



HAL
open science

Faithful model reduction of discrete biological systems

Albin Salazar

► **To cite this version:**

Albin Salazar. Faithful model reduction of discrete biological systems. Computer Science [cs]. ENS Paris, 2023. English. NNT: . tel-04362103v1

HAL Id: tel-04362103

<https://inria.hal.science/tel-04362103v1>

Submitted on 22 Dec 2023 (v1), last revised 22 Apr 2024 (v2)

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License



THÈSE DE DOCTORAT
DE L'UNIVERSITÉ PSL

Préparée à École Normale Supérieure

Faithful model reduction of discrete biological systems

Soutenu par

Albin SALAZAR

Le 14 novembre 2023

École doctorale n°474

**Frontières de l'Innovation
en Recherche et
l'Éducation**

Spécialité

Informatique

Composition du jury :

M. François FAGES INRIA	<i>Rapporteur</i>
M. Jérôme FERET INRIA & ENS-PSL	<i>Directeur de thèse</i>
Mme. Roberta GORI Università di Pisa	<i>Examinatrice</i>
Mme. Anna NIARAKIS Université Toulouse III - Paul Sabatier	<i>Examinatrice</i>
M. Loïc PAULEVÉ CNRS & Université de Bordeaux	<i>Rapporteur</i>
Mme. Tatjana PETROV Universität Konstanz	<i>Examinatrice</i>
M. Marc POUZET ENS-PSL	<i>Examineur</i>
Mme. Elisabeth REMY CNRS & Institut de Mathématiques de Marseille	<i>Examinatrice</i>

Résumé

Les paradigmes de modélisation pour la biologie des systèmes jouent un rôle important dans l'étude de la fonction orchestrée de divers systèmes biologiques. En outre, ils permettent d'étudier un système *in-silico* afin d'obtenir des informations mécanistiques à partir des trajectoires résultant du modèle. Un enjeu majeur pour dériver une représentation idéale pour un processus de système est de fixer le compromis descriptif entre la simplicité et la précision. D'une part, les modèles trop simples ont tendance à ne reproduire que les connaissances *a priori*. D'autre part, les modèles trop descriptifs donnent lieu à des comportements trop difficiles à analyser, voire à calculer. Dans les deux cas, l'acquisition de nouvelles connaissances est entravée. C'est pourquoi il est sans doute important, lors de la sélection des outils de mesurer l'impact de la sélection des modèles sur la capture des phénomènes biologiques, en particulier ceux qui peuvent être vérifiés.

Dans ce manuscrit, nous présentons un cadre formel pour dériver automatiquement des modèles discrets de systèmes biologiques à partir de réseaux de réactions stochastiques. Pour ce faire, nous utilisons des techniques offertes par *Abstract* d'interprétation pour évaluer les comportements résultant des modèles logiques, un outil de modélisation populaire en biologie des systèmes. Malgré le succès des modèles logiques pour résumer les observations expérimentales et prédire les propriétés locales du système, leurs hypothèses de modélisation sous-jacentes restent souvent implicites. Au lieu de cela, les modèles à gros grains que nous obtenons traitent de tous les comportements de la sémantique stochastique des réseaux de réaction initiaux, qui est explicitement définie. Plus précisément, l'espace d'état du réactionnel est divisé en régions abstraites et les transitions non déterministes entre les régions abstraites sont dérivées de manière conservatrice. En outre, nous récupérons les probabilités de transitions du réseau réactionnel de référence, de sorte que les limites du réseau réactionnel non déterministe sont calculées de manière conservatrice. Il est important de souligner que nous pouvons utiliser ce cadre pour évaluer, par le biais du modèle formellement dérivé, les comportements du modèle logique qui l'accompagne. En d'autres termes, le travail établi dans cette thèse ouvre une voie pour évaluer les modèles qui sont naturellement discrets, tout en ouvrant la voie à l'établissement de techniques de réduction de modèles plus efficaces pour les systèmes de réaction.

Mots clés : Interprétation abstraite, modélisation logique, méthodes formelles, réseaux de réaction stochastiques

Abstract

Modeling paradigms for Systems Biology plays an important role in investigating the orchestrated function of various biological systems. Additionally, they permit one to study a system *in-silico* in order to gain mechanistic insights from the trajectories resulting from the model. A common battle to derive an ideal representation for a system process is the descriptive trade-off between the simplicity and accuracy. On one hand, too simple models are prone to reproduce only a priori knowledge. On the other hand, too descriptive models result in behaviors too difficult to parse, or even yet compute. In both cases, gaining new insights is hampered. Thus, it is arguably important in generating tools to measure the impact model selection has in capturing biological phenomena, especially those which can be verified.

In this manuscript, we report a formal framework to automatically derive discrete models of biological systems from stochastic reaction networks. To do so, we utilize techniques offered by Abstract Interpretation (AI) to assess the behaviors resulting from logical models, a popular Systems Biology modeling tool. Albeit the success of logical models in recapitulating experimental observations and predicting local system properties, their underlying modeling assumptions are often kept implicit. Instead, the coarse-grain models that we obtain deal with all the behaviors of the stochastic semantics of the initial reaction networks, which is explicitly defined. More precisely, the state space of the reaction network is split into abstract regions and non-deterministic transitions between abstract regions are derived conservatively. Also, we recover the probabilities of transitions from the reference reaction network, so that bounds to the probability of unlikely behaviors can be computed. Importantly, we emphasize how one can use this framework to assess, via the formally derived model, the behaviors of the accompanying logical model of each reaction system. Namely, the work established in this thesis bares an avenue to assess those models which are naturally discrete, while also paving a path towards establishing more efficient model reduction techniques for stochastic, combinatorial systems.

Keywords : Abstract interpretation, logical modelling, formal methods, stochastic reaction networks

Contents

Résumé	i
Abstract	ii
Table of Contents	ii
1 Introduction	1
1.1 The concentration- and time-scale conundrum in Biology	2
1.2 Quantitive and qualitative models in biology	4
1.2.1 Quantitative models	4
1.2.2 Qualitative models	6
1.3 Discrete models by means of Abstract Interpretation	8
1.4 Thesis outline	9
2 Formal Reaction Networks	12
2.1 Formal stochastic reaction networks	15
2.2 Case Study I: The Birth and Death Model	19
2.2.1 Reaction network	20
2.2.2 Logical model	20
2.2.3 Formal derivation of a coarse-grained model	21
2.3 Case Study II: Model of Resource Competition	25
2.3.1 Reaction network	25
2.3.2 A logical model of competition for resources	25
2.4 A formally derived model of competition for resources	26
2.5 Conclusion	30
3 Mathematical basis for building a coarse-graining framework	31
3.1 Sets and set properties	33
3.2 Distances and projections between sets of elements	34
3.3 Posets and poset properties	40
3.4 Conclusion	46
4 An Abstract Domain with Non-overlapping Intervals	48
4.1 The classical interval lattice domain	50
4.2 Abstract states	54
4.3 Abstract transitions	60
4.4 Abstract traces	67
4.5 Collecting the behaviors of a transition system	74

4.5.1	Concrete collecting semantics	74
4.5.2	Non-overlapping abstract collecting semantics	75
4.6	Non-deterministic, effortless abstract transition systems	77
4.6.1	Revisiting the BD model I	77
4.6.2	Revisiting the Resource Competition System I	78
4.7	Conclusion	79
5	An Abstract Domain with Overlapping Intervals	81
5.1	Characterizing overlapping intervals	85
5.2	Abstractions of quantities of chemical components using reference intervals	92
5.3	Chemical state abstraction using reference macro-states	103
5.4	Macro-transitions between regions of chemical states	109
5.5	Characterizing behaviors for a macro-transition system	114
5.6	Approximation of the collecting semantics	120
5.7	Non-deterministic macro-transition systems with minimal effort	122
5.7.1	Revisiting the BD model II	122
5.7.2	Revisiting the Resource Competition System II	123
5.8	Conclusion	125
6	Chemical Goals: Propagating Probabilities to the Abstract Semantics	127
6.1	Relaying probabilities to macro-transitions	129
6.2	Case Study	134
6.2.1	Biased one-dimensional random walk with non-constant coefficients	134
6.2.2	BD model revisited	138
6.3	Conclusion	140
7	Conclusion	141
	Bibliography	144

Chapter 1

Introduction

The field of Systems Biology is driven by the development of tools to investigate emergent behaviors in populations of biological molecules from single entity interactions. Among these tools, mathematical models of systems of interactions have been critical in identifying hidden mechanisms. Additionally, these tools have been repurposed to conduct perturbation studies, identify potentially novel drivers of disease phenotypes, and in generating new hypotheses. Consequently, a major interest underlies the ability for modeling tools to recapitulate biological phenomena and its repurposing for predictive studies.

In retrospect, developing modeling tools is a continuous field of investigation as it provides means to gain understanding of biological systems through *in silico* studies. A common battle to derive an ideal representation for a system process is the descriptive trade-off between the simplicity and accuracy. On one hand, too simple models are prone to reproduce only *a priori* knowledge. On the other hand, too descriptive models result in behaviors too difficult to parse, or even yet compute. In both cases, gaining new insights is hampered. Thus, it is arguably important in generating tools to measure the impact model selection has in capturing biological phenomena, especially those which can be verified. For example, an overview of various formal modeling frameworks for Systems Biology have been reviewed in [4].

1.1 The concentration- and time-scale conundrum in Biology

It has become a well-established practice that the selection of a modeling framework must be well-suited to capture a biological phenomenon of interest. This model must thus contain the necessary ingredients to investigate a dynamical property of a biological system within a given, contextualized environment. Albeit the process may appear straightforward, the inevitable diversity across biological organisms is vast and the information that can be formalized into a mathematical model and analyzed is computationally limited.

One fundamental aspect that is addressed in this thesis is how the notion of concentration- and time-scale separation that exists between single cells and across organisms (see Fig. 1.1) is treated in logical models, a popular discrete modeling framework used in Systems Biology. Indeed, biological systems exhibit processes at varying size and kinetic scales; those of which involve interacting systems of metabolites, proteins, genes, organelles, cells, and so on. As a result, an overarching goal in Systems Biology is to encapsulate information at each scale and observe communication strategies between scales in order to see how a biological phenotype, such as the onset of an aggressive tumor from healthy cells, could have occurred by following some sequential behavioral trajectory. In theory, we would like to capture this process in a model and be able to develop the predictive capacity to generate new insights on the interplay between the components in the model. In practice, these insights could have to further develop experimental hypotheses or, in many cases, therapeutic strategies to prevent the onset of a disease.

Thus, the goal of modelers is to construct models that faithfully (or to the best of their ability) recapitulate a candidate biological behavior. However, constructing such models is hampered not only by the amount of experimental knowledge available about particular biochemical processes, but also by the sensitivity (or, limitation) offered by the underlying structure of a modeling

property	<i>E. coli</i>	budding yeast	mammalian (HeLa line)
cell volume	0.3–3 μm^3	30–100 μm^3	1000–10,000 μm^3
proteins per μm^3 cell volume	2–4 $\times 10^6$		
mRNA per cell	10^3 – 10^4	10^4 – 10^5	10^5 – 10^6
proteins per cell	$\sim 10^6$	$\sim 10^8$	$\sim 10^{10}$
mean diameter of protein	4–5 nm		
genome size	4.6 Mbp	12 Mbp	3.2 Gbp
number protein coding genes	4300	6600	21,000
regulator binding site length	10–20 bp	5–10 bp	
promoter length	~ 100 bp	~ 1000 bp	$\sim 10^4$ – 10^5 bp
gene length	~ 1000 bp	~ 1000 bp	$\sim 10^4$ – 10^6 bp (with introns)
concentration of one protein per cell	~ 1 nM	~ 10 pM	~ 0.1 – 1 pM
diffusion time of protein across cell ($D \approx 10 \mu\text{m}^2/\text{s}$)	~ 0.01 s	~ 0.2 s	~ 1 – 10 s
diffusion time of small molecule across cell ($D \approx 100 \mu\text{m}^2/\text{s}$)	~ 0.001 s	~ 0.03 s	~ 0.1 – 1 s
time to transcribe a gene	<1 min (80 nts/s)	~ 1 min	~ 30 min (incl. mRNA processing)
time to translate a protein	<1 min (20 aa/s)	~ 1 min	~ 30 min (incl. mRNA export)
typical mRNA lifetime	3 min	30 min	10 h
typical protein lifetime	1 h	0.3–3 h	10–100 h
minimal doubling time	20 min	1 h	20 h
ribosomes/cell	$\sim 10^4$	$\sim 10^5$	$\sim 10^6$
transitions between protein states (active/inactive)	1–100 μs		
time scale for equilibrium binding of small molecule to protein (diffusion limited)	1–1000 ms (1 μM –1 nM affinity)		
time scale of transcription factor binding to DNA site	~ 1 s		
mutation rate	10^{-8} – 10^{-10} /bp/replication		

Figure 1.1: Adapted from [2, 25]. A table of composed of a collection of biological properties at varying physical scales and across three organisms that a modeler must consider when contextualizing a mathematical model. A subset of parameters are conserved across each biological organism (e.g., transition time between an active or inactive protein configuration), while other parameters may be unique to each organism (e.g., the amount of time it takes to translate a protein).

paradigm. Consequently, one must deal with the delicate trade-off between knowledge scarcity and the structural capability of a predictive tool.

Cumulatively, we will delve into the treatment of concentration- and time-scale principles in a common class of modeling styles: discrete models. For us, the goal is to highlight structural limitations underlying a particular type of discrete model in regards to the aforementioned principle, and to offer some avenues in formally assessing the behaviors that may be generated. Namely, we would like to emphasize the generation of a class of discrete models that rely on information obtained from stochastic reaction networks, those of which are built on solid chemical kinetic principles (see [23, Chps. 26 & 27]).

Beforehand, we will overview a few selected candidate modeling frameworks that are often used in Systems Biology. The goal is to discuss these techniques and in which biological context they appear to be most suitable, and then highlight some limitations they experience in practice. This is done to offer the reader of the various flavors of models that exist, and the challenges a modeler must face in order to recapitulate a biological system of interest.

1.2 Quantitive and qualitative models in biology

There are two main classes of modeling paradigms used to investigate biological systems: quantitative and qualitative models. In each class, we will focus on a selected subset of models that one may develop, their application, and limitation.

1.2.1 Quantitative models

In the quantitative setting, models often comprise the use of either differential systems, such as ordinary differential equations (ODEs), or stochastic (probabilistic) transition systems, such as continuous-time Markov chains (CTMCs). These two modeling styles offer two flavors for modeling the dynamics of biological systems.

Although in this thesis work we do not dedicate much work to ODEs, it does merit discussion. In the ODE framework, differential systems are composed of variables expressing chemical concentrations and kinetic parameters detailing reaction rates, those of which constrain the time-evolution dynamics arising from their numeric simulations. In retrospect, these expressions are generally derived from a reaction network diagram depicting how each chemical species in a system may be transformed or produced via their involvement in a reaction. For example, in [29], the authors developed an ODE model describing the hepcidin-regulated distribution of iron, a critical nutrient in several physiological processes, by several biochemical entities in the liver. Thus, this model describes each of the candidates that are involved in regulating the process such as transporting iron into a cell, allocating iron to an appropriate storage unit, or changing the atomic configuration of the iron element itself. A second study [24] focuses on describing populations of antibiotic-resistant bacteria. Here, the authors developed a generic ODE model detailing the resistance pathway that can be instantiated to different forms antibiotic treatments [3]. Thus, the model can be used to characterize bacterial cells obtained in a clinical setting, and predict whether the effectiveness of an antibiotic treatment is realizable. In each case, the ODEs

offer a way to simulate the dynamics of the underlying biological process which can then be observed to study mechanistic interactions between the populations biochemical entities. Yet, in both works as well, the ODE models are heavily dependent on experimental data in order to calibrate the model dynamics to experimental observations; thus making it difficult to scale in the number of variables which can be included in the model. Thus, one must be careful to consider only chemical variables that they deem important to the biological process. Additionally, these systems are often difficult to solve analytically due to non-linear effects arising from interactions between chemical variables, thus modelers must rely on numerical integration strategies to reconstruct the chemical dynamics captured in the ODE expressions. With respect to time-scale separation across the chemical processes, analysis of biochemical processes at different rates can become pesky since numerical integration strategies can lead to unreliable solutions due to step-wise barriers across such processes. This is known as the stiffness problem. Nevertheless, the ODE framework plays a pivotal role in detailing mechanistic interactions between populations of interaction species. The framework that we discuss next focuses on a mechanistic modeling approach that is sensitive to variations that naturally occur between each individual, discrete molecule in a population; thus making it more suitable to investigate biological interactions with low molecule count.

In retrospect, stochastic transition systems are another class of quantitative models which capture physical interactions between each chemical species. In this context, the expressions of a system of interacting agents can be derived from reaction networks using the classical stochastic mass-action kinetics principle (see [23, Ch. 26]). In observing the stochastic dynamics, one models a system state as a vector that composes in each of its indices a discrete value of a chemical species from a reaction network, and transitions between states are driven by the reaction rules. Each reaction inherits a natural kinetic value which is often experimentally deduced and reflects, for each reaction, a statistical quantity regarding the likelihood for reactants to successfully collide to induce a reaction. One famous example of a stochastic model is the one of a genetic circuit observed in the Lambda Switch system observed in viruses [27, 28]. Here, a distinction between two opportunistic pathways of viral integration strategies in bacterial cells are observed in a stochastic model, as opposed to using ODEs. This is mainly because there is competition observed between specific proteins that drive their respective pathways, and which is captured given their abundances and variation in the genetic circuit environment. This phenomenon is observed by fitting experimental, kinetic data to a CTMC model, and then computing probability distributions between state configurations to determine which reaction in the network is likely to occur as compared to another. Interestingly, the Lambda Switch model was reproduced [5] using the KappaLanguage platform [12], thus making it amenable to static analysis techniques such as Abstract Interpretation (AI) [10]. In this line, we would also like to mention a stochastic model of fragments of protein complexes involved in signalling pathways [15]. In this work, the authors developed an AI-based technique to assess the involvement of subsets of clusters of proteins involved in a CTMC model Epidermal growth factor pathway [6], an important pathway in cell homeostasis. By doing so, it was possible to perform model reduction techniques that is capable of recapitulating the equal biological behavior making it more computationally feasible.

Actually, the behavior of a stochastic system can be summarized by its chemical master equation (CME) [18], an instantiation of a CTMC. Indeed, the solution of the CME is an abstraction of the underlying stochastic transition system. That is, in the CME one cannot express the correlation between the probability of being in a pair of states at a pair of moments, while this is possible in the stochastic transition system. However, a limitation to the CME is the fact the one must enumerate the potential states of a system, resulting in a combinatorial explosion in the number of species which can be generated. Additionally, stochastic modeling requires to fit kinetic parameters and to compute distributions over the collection of reactions in order to recapitulate experimental observations, which can also be tricky if one does not have this kinetic information or insights regarding the initial configuration of the system. Thus, one must maneuver carefully an instantiation of a stochastic model for a biological system of interest. In this thesis, we report a sound-over approximation of the stochastic semantics using AI, and use this information to recover probabilities into the type of abstract transition systems we will derive, those of which lose information during the abstraction process. By relying on the stochastic semantics, this information can be integrated into our framework in light of new, quantitative information.

1.2.2 Qualitative models

This leads us to another modeling paradigm we would like to discuss: qualitative models.

Qualitative models differ from quantitative models in that one can may not necessarily require kinetic information about each specific processes in a reaction network. Rather, one can focus on local system properties that may emerge from the changes in chemical quantities arising from systems of interactions. The two qualitative frameworks we will discuss are Petri nets (PNs) and logical models.

PNs are a common modeling tool used to investigate concurrent processes [31, 30], and their applications have found their way into biochemical networks [32] given the structural similarity between PNs and reaction networks. In this framework, one realizes that the syntax of reaction networks are similar to that of Petri nets. As such, applications of PNs have found their way into metabolic modeling [7]. For example, the authors in [37] were able to take advantage of this topological feature to identify structural cycles for a metabolic network model involved in nucleotide (the fundamental structural back bone keeping intact DNA) metabolism. Yet, PNs are also amenable to dynamic analyses. A dynamic analysis of a PN model is inherently similar to a CTMC one, with the exception that the probability values associated to each transition between states have been abstracted. Thus, PNs retain no information regarding time. Thus, formalizing the reaction rules from a reaction network into a PN model and analyzing its dynamics permits one to capture production and consumption events of discrete chemical quantities involved in the selected reaction network. Importantly, the infusion of chemical components through the PN transition system is independent of kinetic parameters. Thus, sequences of transitions carry information about the flux of chemical quantities over some implicit time period (e.g., the order of indices of each transition in a sequence). As a result, iterating over all the possible sequences that can be obtained (which can be infinite in an unlimited resource pool) allows one to keep

track of the token copy numbers dispersed along each sequence. For example, in [21] the authors used PNs to gain insights and validation on a metabolic sub-network of enzymatic components involved in converting sucrose, a sugar, into starch in potatoes. Namely, invariant analysis of this transition system of this PN allowed the authors to determine which pertinent compounds are associated to a subset of reactions of the corresponding metabolic network.

PNs have become a popular starting point for modeling biological systems, given that it can be amenable to extensions in order increase its sensitivities to other biological properties. Additionally, given its formal specification one can also use model checking tools to answer questions of interest, such as reachability regarding certain state properties.

The second qualitative model we will discuss about in-depth are logical models. Logical models are another popular class of discretized models. Recent development have made it an ideal modeling tool to perform perturbation studies for a myriad of biochemical interactions. For example, one interesting model is an iron metabolism system in breast cancer cells [9]. This model captures the regulation of activity levels for different proteins involved in a signaling pathway controlled by *pools* of iron, and how changes in the regulation of these components could alter the behaviors of a healthy breast cell may lead to a cancerous breast cell.

The process of constructing logical models follows three general guidelines [8]:

1. building a regulatory graph;
2. deriving the logical expression for each component in the graph;
3. selecting an update scheme to generate the transition system.

The guidelines can be broken down accordingly. In Step 1, the set chemical species involved in a selected biological process is gathered either from a literature survey or from experimental observations, afterwhich their activity levels are encoded into nodes. Afterwhich, edges connecting nodes correspond evidence-based regulatory interactions: these are the interactions between chemical species and how their respective abundances can affect the activity of other chemical species for whom to they may be connected. In Step 2, the regulatory graph is used to derive logical functions that express how each chemical species are updated (regulated) by their inputs (regulators). Note that the connectivity of the graph does not necessarily lead to the library of functions derived. To continue, these hand-written logical functions are composed of logical operators and a threshold is given in order to capture the activity level across each species variables in the logical expressions. Finally, in Step 3 the logical functions are implemented in a scheduling algorithm to generate the state transition system under a synchronous (all states are updated at the same time) or asynchronous (one state is updated at each iteration) updating policy. This state transition system is structurally similar to Petri nets; indeed, logical models fit into a specific class of PNs called safe Petri-nets. Yet, in logical models the chemical quantities are not discrete quantities but rather an implicit interval of values denoting a certain level of activity.

A key feature of logical models is in obtaining knowledge about a process while only partial information is available about some particular interactions and their kinetics. Albeit their

success in recapitulating experimental observations and predicting local system properties, their underlying modeling assumptions are often kept implicit. There is indeed a gap between hand-written logical models and the models that can be formally derived from a more concrete level of representation (such as a stochastic reaction network, as done for the previous modeling paradigms).

One popular approach to design logical models has been proposed by René Thomas [19, 36]. With this method, each dimension is associated with a unique attractor state (or focal point), which may depend on the current state of the system. Then the transitions of the system are obtained by assuming that on each dimension, the system may get closer to its focal point. In reality, in a more comprehensive description of these systems permits, reverse transitions may also occur (but at low probabilities), but in practice they are neglected. Such simplifications of the model is usually justified by some time-scale separation principles [13] (that are mainly asymptotic reasonings providing convergence results when scales are infinitely separated). In this work, the authors develop a model of a cell division process in mammalian cells and, in order to facilitate the analysis of an asynchronous update scheme on the logical expressions derived, proposes to distinguish between fast or slow process classes of reactions based on literature evidence by ranking the reactions in the order in which they may occur. Two critical observations emerge about this modeling process. Firstly, it is unclear to which extent these simplifications actually impact the behaviors of the systems they try to represent. Secondly, considering all reverse transitions, without any information about their potential likelihood, would lead to inaccurate models with many fictitious non-deterministic behaviors.

1.3 Discrete models by means of Abstract Interpretation

In order to increase confidence in the modeling process, we would like to derive formally discrete models from more precise representations (such as reaction networks). For this purpose we use the abstract interpretation framework to coarse-grain reaction networks into discrete models of abstract regions of states while preserving formal relationships between the respective behaviors of both models. Yet we have to face several issues. Firstly, several behaviors that are usually neglected in the logical models may occur with a low probability in the initial reaction network. Thus a non-deterministic abstraction would be unhelpful, because non-deterministic models provide no means to distinguish rare behaviors from more the common ones. Instead we propose to propagate the probabilities of transitions from the reference reaction network to the coarse-grained model, so that bounds to the probability of unlikely behaviors can be indeed computed. For this purpose, we equip each transition in the abstract model with an interval for their probabilities which is formally derived from the underlying reaction network. Secondly, even with probabilities, naive abstractions lead to very imprecise models. This means that we have to adapt our abstraction in order to highlight the main behaviors of interest. Consequently, we obtain a Discrete-time Markov chain (DTMC) whereby the exclusion of probabilities would structurally mimic logical models, yet providing an accessible tool to quantify differences between logical models and the discrete models obtained from our formalizations. One may be tempted to use

Continuous-time Markov chains (CTMC) rather than DTMCs. Yet, the additional information provided by the continuous setting is not relevant in our context. On one hand, we want to obtain models comparable to logical models where the notion of time has already been abstracted away. On the other hand, the exact moment when each event occurs does not affect the computation of the probabilities of transitions between abstract regions in our coarse-grained models. What matters is only the relative order between these transitions, not the exact moment when they have happened. Yet, abstracting away the notion of time of a CTMC while only keeping the relative order between events induces a DTMC, which justifies our choice thoroughly. Lastly, using DTMCs instead of CTMCs deeply simplifies the underlying mathematics. For instance, in the continuous setting, probability density functions are required to define when events are likely to occur and a topology is necessary to define the probability of which set of model executions can be computed [15]. In contrast, in the discrete setting, only discrete probabilities are necessary and the probability of each execution with a finite amount of steps can be computed.

Ideally, the interval of probabilities computed for transitions between abstract states that correspond to less likely behaviors should be very low. To achieve this goal, it is important to refine the abstraction process and to distinguish the abstract regions of states according to which concrete transitions have been taken to enter them. Furthermore, it is also important that every transition between abstract regions corresponds to sequences composed of at least a few concrete transitions. Thus, we relate carefully the abstract behaviors to the concrete ones. That is, for a concrete behavior to enter a new abstract region, a quantity in each behavior must surpass a threshold given by boundary values. These boundary values correspond to the lower and upper bound of each abstract interval. Consequently, in the formal discretization process an interval is composed of a pair of boundary values that enclose a concrete value. Thus, for each concrete value, an abstract interval computation is sensitive to whether a concrete value has exited a visited interval. Additionally, since concrete values are driven by their stochastic dynamics then we want to ensure exited intervals are not easily re-entered. This motivates the use of overlapping intervals to coarse-grain models such that bordering intervals share a sub-region corresponding to common concrete values. In this style of abstraction, we consider that a value changes to another interval only when it actually leaves its current interval (hence leaving the overlapping region between its previous and current intervals). This way, when entering a new region of states, going in the reverse direction requires crossing through the overlap between two intervals, which is likely to have low probability when it is against the main trend of the dynamics of the system. The so-obtained abstraction ignores small fluctuations while strengthening the sequences of transitions that follow the main trend of the system dynamics.

1.4 Thesis outline

In this Chapter, we introduce the concentration- and time-scale problem in Biology. Afterwhich, we discuss various modeling paradigms used in Systems Biology and provide some study case examples of how each modeling framework may be used. Then we shift our focus to logical models, the main modeling paradigm we challenge in the thesis. We show how these models are

derived, instantiated and how some of the incorporated modeling assumptions can be misleading. Thus, our goal is to build a formal framework not only to derive a sound-approximation of the stochastic semantics but to offer an AI-based tool to assess the underlying mathematical structure of logical models.

In Chapter 2, we formally introduce the stochastic semantics for reaction networks. We often refer to the stochastic semantics as the concrete semantics, since it characterizes each of the mathematical objects we require to interpret the collection of behaviors that can be instantiated from a reaction network. Additionally, we compare our stochastic semantics to a Boolean network semantics in order to provide an intuitive example on how our formal framework can be used to assess the behaviors from a logical-type model. We make an example on a well-studied system, the birth and death-model, and show how one can incorporate probabilities into a formally derived discrete transition system.

In Chapter 3, we introduce our mathematical toolbox, and define the object types, properties, and functions that we will use throughout the text. Mainly, this toolbox will permit us to derive to characterize the abstraction process, generate an abstract semantics for reaction networks, which we refer to as the macro-transition system, and re-introducing probabilities to a macro-transition system.

The goal of Chapter 4 is to derive a concrete and an abstract transition system using the classical interval lattice domain in order to highlight the importance of dependency abstraction. Namely, we derive each concrete and abstract objects required for its respective transition system, establish for each a Galois connection, and show how the abstraction process can greatly incur loss of precision in our analyses.

In Chapter 5, we generalize the classical interval domain to include intervals which overlap. Using both non-overlapping and overlapping intervals, we derive a macro-transition system composed of the abstract elements which over-approximate the chemical transition system in Chapter 2. Similarly to Chapter 4, we establish a Galois connection between each abstract and chemical element, thus permitting us to traverse between these two domains. Additionally, our macro-transition system is a type of discrete transition system since the abstraction process loses information regarding the stochastic behaviors from a reaction network. Thus, the formally derived models lead to a transition system that is structurally similar to logical models. Thus, at this point our goal becomes to re-introduce probabilities to our macro-transition system in order to carefully capture the underlying behaviors of an instantiated reaction network.

In Chapter 6 we re-introduce probabilities between the macro-transitions derived in Chapter 5. Since the abstraction process loses information about probabilities, we recast abstract notions in the concrete domain and take advantage of monotonicity properties of the probability functions derived in Chapter 2 in order to compute probability values for chemical quantities to reach a goal (e.g., an region of chemical quantities) of interest. We show that for some invertible matrices, this probability can be computed exactly. Indeed, we would like to use this information to compare the likelihood between pairs of macro-transitions in order to know which one of the two is more likely. At this point the macro-transition system is no longer non-deterministic. Quantitative measures on these macro-transitions can provide information regarding the gen-

eral trend for the dynamics for a particular reaction network. We instantiate our probabilistic framework on a second case study for a system of competing reactions.

Finally, in Chapter 7 we provide final remarks on our coarse-graining framework for stochastic reaction networks. We highlight some limitations and future directions regarding scaling the framework on practical examples.

Chapter 2

Formal Reaction Networks

Objectives

In this Chapter, we introduce our formal framework to automatically generate the stochastic behaviors emerging from reaction networks. That is, we define the collection of the mathematical objects we require to capture these behaviors, and which will be fundamental for the design of our coarse-graining framework. Finally, we conclude with two case studies in order to motivate our formal discretization framework.

Contents

2.1	Formal stochastic reaction networks	15
2.2	Case Study I: The Birth and Death Model	19
2.2.1	Reaction network	20
2.2.2	Logical model	20
2.2.3	Formal derivation of a coarse-grained model	21
2.3	Case Study II: Model of Resource Competition	25
2.3.1	Reaction network	25
2.3.2	A logical model of competition for resources	25
2.4	A formally derived model of competition for resources	26
2.5	Conclusion	30

A general structure to capture diverse behaviors emerging from interacting (bio-)molecules are reaction networks. A reaction network models the behavior of a system of interactions. For example:



depicts a reaction network composed of two reactions. The tail of a reaction arrow is preceded by reactants, while products follow after the head of the reaction arrow. In this reaction network, the first pseudo-reaction is a synthesis reaction of molecule A with kinetic constant k_A . A second pseudo-reaction is a consumption reaction of molecule A with kinetic constant $k_{A'}$. This reaction system is commonly known as the birth and death (BD) model because there are two reactions that generate counter behaviors when the dynamics are observed. Additionally, each kinetic constant is constrained experimentally and provides information about the speed at which each reaction can occur. Thus, it is possible that the values of these kinetic parameters and the quantities of chemical species can vary for diverse, larger reaction systems, which leads to the notion of time-scale separation discussed in Ch. 1.

A curious observation of interest usually entails how systems of reactions driven by their interconnected species give rise to biological phenomenon in the form of phenotypes, which is a physical characteristic seen in an individual or collection of cell(s). In a mathematical sense, interactions between species are seen as reactions which lead to behaviors upon the application of at least a reaction, while phenotypes may correspond to long-term behaviors (e.g., intrinsic, stable behavior(s) of a system) reachable after several reactions.

Reactions are commonly seen as the semantics for the behaviors generated from a reaction network. Namely, they allow us to instantiate and update the state of the system and observe their evolution.

The reason we use reaction networks is because they are well-founded on physical chemical principles. Thus, one can select a semantics (e.g., such as, for example, ODEs) to observe how the system evolves with respect to the key chemical drivers involved in a physiological process. However, choosing the appropriate modeling framework is non-trivial, as each paradigm offers their own flavor of representation and mechanistic insights. Thus, choosing the appropriate framework to model a given biological system is important to understand a property of interest, such as the quantity of a drug that may inhibit a target or a qualitative phenotype that may indicate a cell is undergoing cellular division. Nevertheless, one (possibly) has as a reference a reaction network and must modestly select an appropriate semantics to investigate the underlying biological behaviors. Our goal is not to exhaust limitations between different modeling frameworks, but one can suggest to the reader [4] in order to highlight comparison and differences between candidate paradigms.

Given the nature of the model construction process, a common concern is the reliability given to one modeling framework over another. That is, whether the framework captures faithfully the underlying biological process. In the case of this work, one goal is to provide formal tools to assess a popular discrete modeling paradigm: logical models. More specifically, an objective is to build more precise discrete models from abstracting behaviors from stochastic reaction

networks. By building discrete models from a formal basis, we hope to offer structural insights on the logical modeling paradigm and provide a tool to build models with more confidence.

2.1 Formal stochastic reaction networks

In Chapter 1, we discussed how the selection of a modeling language can impact the biological properties one may attempt to recapitulate. Indeed, the model selection process plays a critical role in capturing phenomena of interest, and given the large diversity of biological behaviors that can be observed, the modeler must choose a suitable paradigm. That is, a modeler generally faces two major challenges: i) to identify an appropriate mathematical representation for a selected biological system of interest; and, ii) the design of a strategy to generate the selected system's behaviors, which carry information about a property of interest. The two aforementioned challenges are often driven by direct modeling approaches, which are often hampered by the amount of information that can be modeled or even available. For example, in a stochastic setting, a focus on low-copy number interactions between diverse biomolecules may rapidly lead to an intractable system analysis upon enumeration of all the possible interactions. Yet, it could be the case that a subset of these enumerations are not biologically relevant. In such a scenario it is arguable that these challenges could benefit from formal methods which offers a precise mathematical description of the problem at hand which permits the automatic extraction of properties of interest while maneuvering between irrelevant behaviors. Additionally, one may also gain mathematical guarantees.

In this Section, we will derive a formal representation of stochastic chemical reaction networks. More specifically, we define for reaction networks a sufficient collection of mathematical objects that will permit a user to automatically generate their stochastic behaviors. In order to compute the probabilities of these stochastic behaviors, we instantiate a Discrete-time Markov Chain (DTMC) which are driven by kinetics associated to each reaction in a network and the quantity of each chemical species at a given discrete-time instance.

Formally, a reaction is defined as follows (and adapted from [1]).

Definition 2.1 (Chemical Reaction). *Let $\mathcal{V}_{\mathcal{R}}$ be a set of chemical species. A reaction over the set of species $\mathcal{V}_{\mathcal{R}}$ is defined as a triple $r = (M, V, k)$ such that:*

1. $M : \mathcal{V}_{\mathcal{R}} \rightarrow \mathbb{N}$,
2. $V : \mathcal{V}_{\mathcal{R}} \rightarrow \mathbb{Z}$,
3. and $k : \mathbb{N}^{\mathcal{V}_{\mathcal{R}}} \rightarrow \mathbb{R}_{\geq 0}$.

In a chemical reaction, the first component M is a function that maps a chemical species to its multiplicity which is generally regarded to the amount of reactants required. The second component, V , is a function that associates to each reactant chemical species respective consumption (negatively) and production (positively) values. Finally, the third component k is a function assigning a kinetic term to a reaction and details its frequency. Now, we define a reaction network composed of many reactions.

Definition 2.2 (Chemical Reaction Network). A reaction network \mathcal{R} is defined as a pair $(\mathcal{V}_{\mathcal{R}}, (r_j)_{1 \leq j \leq n})$ such that:

1. $\mathcal{V}_{\mathcal{R}}$ is the set of chemical species;
2. $(r_j)_{1 \leq j \leq n}$ is a set of n reactions over the set $\mathcal{V}_{\mathcal{R}}$ indexed with an integer j between 1 and n .

For each integer j between 1 and n , the reaction r_j is also denoted as $(M_{r_j}, V_{r_j}, k_{r_j})$.

In Def. 2.2, a reaction network is defined as a pair such that the first component is the set of chemical species and in the second component are sequences of reactions that involves each chemical species.

Example 2.1. For example, we apply our definition of a chemical reaction network:



Here, the set of chemical species is $\mathcal{V}_{\mathcal{R}} = \{A, B, C\}$. This reaction network is made of two reactions, with respective multiplicity vectors:

$$\begin{aligned} M_{r_1} &= [A \mapsto 1, B \mapsto 0, C \mapsto 0]; \\ M_{r_2} &= [A \mapsto 2, B \mapsto 0, C \mapsto 0], \end{aligned}$$

and with the respective reaction vectors:

$$\begin{aligned} V_{r_1} &= [A \mapsto -1, B \mapsto 1, C \mapsto 0]; \\ V_{r_2} &= [A \mapsto -2, B \mapsto 0, C \mapsto 1]. \end{aligned}$$

Furthermore, each respective reaction has, respectively, kinetic constants k_B and k_C , which induces the kinetic terms

$$\begin{aligned} k_{r_1}(q) &= [q \mapsto k_B \cdot q(A)]; \\ k_{r_2}(q) &= [q \mapsto \frac{1}{2} \cdot k_C \cdot q(A) \cdot (q(A) - 1)]. \end{aligned}$$

Here, q is a vector containing the value of each chemical component; later we refer to this mathematical object as a chemical state. Additionally, each kinetic term is derived from the law of stochastic mass-action kinetics.

Definition 2.3 (Chemical State). A chemical state is defined as a function $q : \mathcal{V}_{\mathcal{R}} \rightarrow \mathbb{N}$. The set of all the chemical states for a reaction network is denoted as $\mathcal{Q}_{\mathcal{R}}$.

In Def. 2.3, a chemical state is a function q with domain and co-domain as, respectively, the set of chemical variables and the set of values. This function contains in each of its component the copy number associated to each chemical species. In retrospect, the values of a subset of chemical variables in a chemical state can be updated if it happens to be involved in a reaction, which triggers a chemical transition.

Definition 2.4 (Chemical Transition). *A chemical transition is a triple $(q, r, q') \in T_{\mathcal{R}}$ relating two chemical states $q, q' \in \mathcal{Q}_{\mathcal{R}}$ by a reaction r such that for all chemical variables $v \in \mathcal{V}_{\mathcal{R}}$:*

1. $M_r(v) \leq q(v)$;
2. $q'(v) = q(v) + V_r(v)$.

The set of all the chemical transitions for a reaction network is denoted as $T_{\mathcal{R}}$.

A chemical transition captures an application of a reaction such that Cond. 1 ensures that for a given reaction there are for the associated chemical species sufficient amounts available for this reaction to occur, while Cond. 2 applies a reaction to update the respective values of each chemical species in a source chemical state to obtain the new values in the corresponding target chemical state.

In the following definition, we compute probabilities for each chemical transition.

Definition 2.5 (Transition probability). *Let $(q, r, q') \in T_{\mathcal{R}}$ be a chemical transition. The probability for a chemical state $q \in \mathcal{Q}_{\mathcal{R}}$ involved in a chemical transition is defined as:*

$$\lambda_r(q) = \frac{k_r(q)}{\sum_{r' \in \{1, \dots, n\}} k_{r'}(q)}.$$

In Def. 2.5 the probability for a chemical transition is equal to the ratio of the kinetic term of a reaction to the sum of the kinetic term of each reaction in a reaction network. Note that each transition may contain diverse quantities of chemical species in its source chemical state, which is reflected in the probability functions. Namely, we concern ourselves with transitions with non-constant probability rates.

Example 2.2. For example, an instantiation of Def. 2.5 used on the reaction network from Ex. 2.1 results in the following transition probability functions:

$$\lambda_{r_1}(q) = \frac{2 \cdot k_B \cdot q(A)}{q(A) \cdot (2 \cdot k_B + k_C \cdot (q(A) - 1))} \text{ and } \lambda_{r_2}(q) = \frac{k_C \cdot q(A) \cdot (q(A) - 1)}{q(A) \cdot (2 \cdot k_B + k_C \cdot (q(A) - 1))}$$

whenever $q(A) > 0$. Note that when $q(A) = 0$, no reaction is enabled and the system is deadlocked.

Starting from an initial chemical state, it is possible to chain chemical transitions. This leads to the notion of a chemical trace.

Definition 2.6 (Chemical Trace). *A trace of length $k \in \mathbb{N}$ is a couple $(q'_0, ((q_i, r_i, q'_i), \mu_i)_{1 \leq i \leq k}) \in \mathcal{Q}_{\mathcal{R}} \times (T_{\mathcal{R}} \times [0, 1])^k$ that satisfies both conditions:*

1. for every integer i between 0 and $k - 1$, we have $q'_i = q_{i+1}$;
2. for every integer i between 1 and k , we have $\mu_i = \lambda_{r_i}(q_i)$.

Such a trace is usually written as $q_1 \xrightarrow[\mu_1]{r_1} \dots \xrightarrow[\mu_k]{r_k} q'_k$. The set of all the chemical traces for a reaction network is denoted as $\mathcal{T}_{\mathcal{R}}$.

The set of all the chemical traces of a reaction network defines all the potential long-term behaviors of its underlying system. Each chemical transition in each chemical trace is associated to a probability, which is driven by the respective source states.

Example 2.3. Given the initial chemical state $q'_0 = [A \mapsto 6, B \mapsto 0, C \mapsto 0]$ and the kinetic constants $k_B = 20$ and $k_C = 1$, an example of a chemical trace for the reaction network in Ex. 2.1 is given as follows:

$$(6, 0, 0) \xrightarrow[0.8]{r_1} (5, 1, 0) \xrightarrow[0.83]{r_1} (4, 2, 0) \xrightarrow[0.13]{r_2} (2, 2, 1) \xrightarrow[0.95]{r_1} (1, 3, 1) \xrightarrow[1]{r_1} (0, 4, 1),$$

where a chemical state q is denoted as the triple $(q(A), q(B), q(C))$. Each transition along this chemical trace has associated to it a probability for a reaction type. At the end of this trace, resources (the quantity of molecule A) are depleted.

It is possible to compute for each chemical trace a probability. That is, a stochastic transition step in a chemical behavior induces a Discrete-time Markov Chain (DTMC).

Definition 2.7 (Discrete-time Markov Chain). *Let $q \in \mathcal{Q}_{\mathcal{R}}$ be a chemical state. A Discrete-time Markov Chain is defined as a stochastic process $\{S_i\}$ with $i \in \mathbb{N}$ and $S_0 = q'_0$ an initial chemical state such that the following property holds for each $q_i \in \mathcal{Q}_{\mathcal{R}}$:*

$$P(S_k = q'_k \mid S_0 = q'_0, \dots, S_{k-1} = q_k) = P(S_k = q'_k \mid S_{k-1} = q_k).$$

In Def. 2.7, a DTMC models the probability for a chemical transition between two states to occur. Namely, the probability to reach a target state via a chemical transition is dependent only on its source state. Thus, along a chemical trace each transition step is independent of each other, and a reaction type occurs with a probability that is governed by the quantities of chemical species contained in each transition's source chemical state and the reaction's kinetic constant. Thus, the following definition associates a probability to each chemical trace.

Definition 2.8. *Let $(q'_0, ((q_i, r_i, q'_i), \mu_i)_{1 \leq i \leq k})$ be a chemical trace that we denote as τ . The probability $P(\tau \mid q'_0)$ of the chemical trace τ , knowing that the system starts in the initial chemical state q'_0 , is defined as $\prod_{1 \leq i \leq k} \mu_i$.*

Example 2.4. For example, the probability for the trace of Ex. 2.3 is: $P(\tau \mid (6, 0, 0)) = 0.08$ (since $0.80 \cdot 0.83 \cdot 0.13 \cdot 0.95 \cdot 1 = 0.08$).

Consequently, we have defined each of the mathematical objects we require to faithfully represent all the stochastic behaviors of a reaction network. Indeed, our goal is to generate these

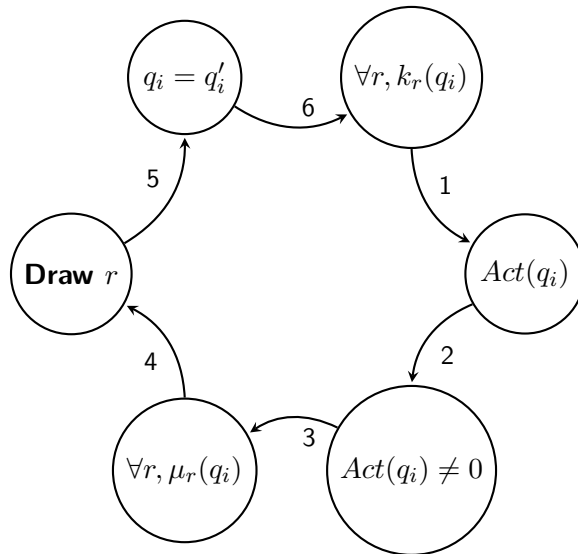


Figure 2.1: The iterative scheme of the Gillespie next reaction method [17] for generating the stochastic behavior of a reaction network. In the tail of step (1), the kinetic term of each reaction in a system is computed at the chemical state q_i . Accordingly, in (2), the activity of the system is computed: $Act(q_i) = \sum_{r' \in \{1, \dots, n\}} k_{r'}(q_i)$. After which, (3) a loop condition verifies whether the activity is null or not and proceeds accordingly. (4) If the activity is non-zero, the probabilities for each reaction at state q_i is computed. In (5), a reaction is chosen according to the underlying distribution. Finally, in (6) the chemical state q_i is replaced with q'_i (see Def. 2.6), and the loop is repeated.

behaviors from a more formal description of reaction networks. For example, in Fig. 2.1 is an iterative scheme, known as Gillespie next reaction method [17], that displays how the stochastic behaviors of a reaction network may be generated. An overall goal is to be able to approximate these chemical behaviors with the interval domain borrowed from the Abstract Interpretation framework. More specifically, we would like to adapt our abstractions to faithfully approximate an underlying stochastic transition system and further pinpoint the system’s dynamics, yet at a lower level of representation. To do so, we use the stochastic transition system to re-introduce probabilities to the resulting abstract transition system. Also, we will show that the replenished abstract transition system can be used to assess the transition system of a corresponding logical model.

2.2 Case Study I: The Birth and Death Model

In this Section and Sect. 2.3, we provide intuition for the type of abstractions we would like to accomplish on stochastic reaction networks. Accordingly, our case studies include the birth and death (BD) model (this Section) and a resource competition system (Sect. 2.3). In each example, we will introduce the system’s reaction network representation, logical model, and a formal model obtained from an instantiation of our framework. Further, we show how one can propagate probabilities from a stochastic (concrete) transition system into the formal model; and, how to use the latter model to assess the behaviors of a corresponding logical model.

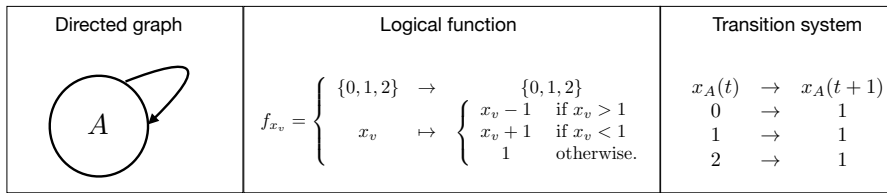


Figure 2.2: A logical BD model. A directed graph (left) displays A as a self-regulator, while the logical function (center) is derived to reflect the expected BD system behaviors. The transition system (right) is the result of applying the logical function to a state $x_A \in \{0, 1, 2\}$, each representing an interval of values of molecule A : low (0), medium (1) and high (2). Note that the notion of time is discrete.

2.2.1 Reaction network

The BD system is characterized by two reactions having opposite behaviors. Both reactions are given as follows:



The reactions represent production and consumption events of molecules of A . The first reaction, r_1 , is a birth event with kinetic constant k_A , while the second, r_2 , is a death event with kinetic constant $k_{A'}$. We denote as q the chemical state of the system. The chemical state of the system maps the components of the system to their copy numbers. In this model, A is the unique component.

By assuming stochastic mass-action kinetics law, we can obtain the propensity k_A for the production event and the propensity $k_{A'} \cdot q(A)$ for the death event. Then, the transition probabilities (see Def. 2.5) $\mu_1(q)$ and $\mu_2(q)$ that the next event is an instance of the reaction r_1 or an instance of the reaction r_2 are defined as follows:

$$\mu_1(q) = \frac{k_A}{k_A + k_{A'} \cdot q(A)} \quad \text{and} \quad \mu_2(q) = \frac{k_{A'} \cdot q(A)}{k_A + k_{A'} \cdot q(A)}.$$

Indeed, a probability for an event type is the ratio between its propensity and the sum of all possible propensities for this system. We can observe that these probabilities are non-constant, as the quantity of A , $q(A)$, may vary.

2.2.2 Logical model

A logical model can be provided regardless of the exact structure of the reactions and the effective values of the kinetic parameters. Following René Thomas's principles [36], what matters is the general trend for the evolution of the copy numbers of each kind of components. In our BD case study, we can indeed distinguish three kinds of states: when the amount of A is such that the state of the system is likely to be stable; when (below this amount) the quantity of A is likely to increase; and when (above this amount) the quantity of A is likely to decrease.

These observations lead to the logical model that is described in Fig. 2.2. Firstly, a directed graph summarizes the potential regulations (or dependencies) between the components. Here

a self arrow on the component A stipulates that the component A auto-regulates itself. We assume that the potential quantities of A are partitioned into three intervals, denoted as 0, 1, and 2. They stand respectively for below the steady state (low / (0)), for around the steady state (medium / (1)), and for above the steady state (high / (2)). Consequently, we derive a logical function that reflects how the system is expected to evolve: below the steady state, the amount of A increases; around the steady state, it remains constant; above the steady state, it decreases. Then, the logical function induces a transition system that captures the integrated process. Note, however, that at this level of abstraction stochastic fluctuations cannot be observed since reversible interval transitions are not permitted. Thus, the BD logical model is capable of capturing only the most expected behaviors.

2.2.3 Formal derivation of a coarse-grained model

Now that a logical model for the BD system has been proposed, we would like to compare its behaviors to those obtained by a formal discretization. We use the same BD reaction network and formally discretize its state space. Then we show that it is possible to restore information on probabilities in this new model.

A common discretization method uses a non-overlapping interval schema. This means that the state space of chemical values are partitioned into intervals that do not share any common values. For example, in the logical BD model interval partitioning are qualitative states with implicit meaning. We can obtain a similar representation in the formally derived model by explicitly choosing a sequence of contiguous intervals (with no intersecting values, see Fig. 2.3). It is worth noting that we expect this new model to cope with many more behaviors than the logical model. Consequently, we obtain a non-deterministic abstraction of the BD transition system. A question then rises as to whether these behaviors are consequence of the imprecision of the formal abstraction, or whether they reveal important behaviors that are missing in the logical models.

To answer this question, we use information about the stochastic behaviors of the reaction network to recover the probabilities to navigate between intervals (see Fig. 2.4). More precisely, when entering an interval, we compute the probability that the process will cross this interval or go back to the previous one. It is worth noting that whether an interval is entered from below or from above matters. So we duplicate each interval accordingly. We expect to observe the convergence of the process towards the interval containing the steady state, which is a stable behavior observed by a system. This is the main qualitative behavior of the reaction network and it should be reflected in the discretized model independently of the choices of the discretization intervals.

Given an interval with lower bound $l \in \mathbb{N}$ and upper bound $u \in \mathbb{N}$, we consider for every chemical state q such that $l \leq q(A) < u$, the probability $P(q)$ such that the quantity of A will reach the upper bound u before reaching the lower bound l , knowing that the system starts in the state q . By reasoning on the potential BD events stemming from the state q and their

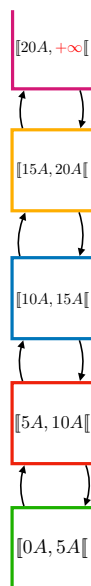


Figure 2.3: A non-deterministic macro-transition system of the BD system with non-overlapping intervals. Each rectangle is a range of values for the molecule A and an edge is a transition from a source to a target interval. Intervals are duplicated to distinguish whether they are entered from below or above.

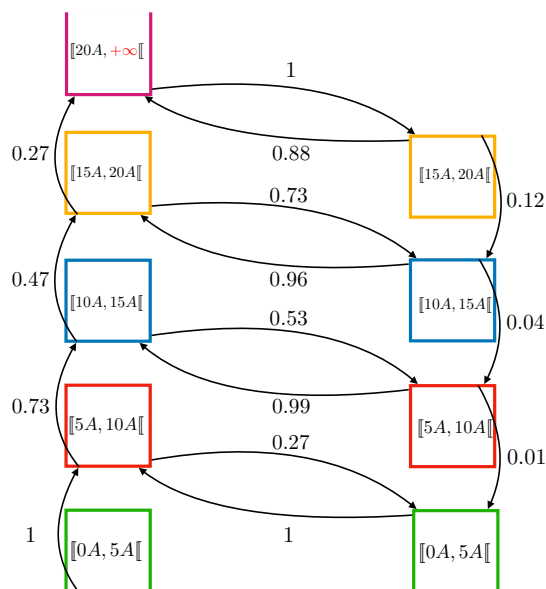


Figure 2.4: The same BD macro-transition system of Fig. 2.3 refined with probabilities. Each rectangle is a range of values for the molecule A and a labeled edge is a transition from a source to a target interval. Intervals are duplicated to distinguish whether they are entered from below or above.

probabilities, we obtain the following relation:

$$P(q) = \begin{cases} 0 & \text{whenever } q(A) = l, \\ 1 & \text{whenever } q(A) = u, \\ \mu_1(q) \cdot P([A \mapsto q(A) + 1]) + (1 - \mu_1(q)) \cdot P([A \mapsto q(A) - 1]) & \text{otherwise.} \end{cases}$$

As boundary conditions, the probabilities $P([A \mapsto l])$ and $P([A \mapsto u])$ are set respectively to 0 and 1. The last case combines the contribution of two processes: the potential increase in the amount of A with probability $\mu_1(q)$ and the potential decrease with probability $1 - \mu_1(q)$.

Finite unfolding of the previous recurrence relation converges to a lower bound on the probability $P(q)$. In Chapter 6 we discuss how one can exploit recurrence relations to bound an exact probability with an interval of probabilities. Yet, in the BD model, a closed form equation can be derived (the proof can be found in Sect. 6.2.2). The probability $P(q)$ is indeed defined by the following equality:

$$P(q) = \frac{Aux(q(A))}{Aux(u)} \quad (2.1)$$

where $Aux(j) = \sum_{l \leq s' < j} \left(\prod_{l < s \leq s'} \left(\frac{k_{A'} \cdot s}{k_A} \right) \right)$, for each $j \in \{u, q(A)\}$. We can use Eq. 2.1 to compute the probability to reach first an upper (resp. lower) bound prior to a lower (resp. upper) bound.

We show in Fig. 2.4 the macro-transition system that we derive this way. Underlying each rectangular region are the dynamics stemming from the BD process with kinetic constants $k_A = 20$ and $k_{A'} = 1$. Thus, the collection of intervals displayed are chosen in order to capture the system's steady state, which is the qualitative behavior of interest. Here, the system steady state is when the quantity of the molecule A is equal to 20, or $q(A) = 20$; and, in this example it is contained in the interval $\llbracket 20, +\infty \llbracket$. Each macro-transition is composed of a source interval, an edge labeled with a probability and a target interval. Note that each probability value is exact thanks to Eq. 2.1. To trigger a macro-transition type, a BD event (birth or death) must push the value $q(A)$ through an interval upper or lower bound value. Thus, it is possible to enter a target interval from below (via an upper bound) or above (via a lower bound). Intervals are duplicated accordingly. The one on the left side of the transition system denotes those entered from below and the one on the right side, those entered from above. Also the exact position of the source (resp. target) of each macro-transition indicates from which border of the interval the macro-transition starts (resp. ends).

We then want to observe whether the trend of the system will proceed upwards or downwards. Since abstraction loses all information about the probabilities of individual reactions, we recover them using Eq. 2.1. As a result, the probability to exit from the upper bound 19 when starting from the value $q(A) = 15$ is equal to 0.27, and its complement 0.73 is the probability to exit from the lower bound 15. Putting these pieces of information together results in the macro-transition $\llbracket 15, 19 \llbracket \xrightarrow{0.73} \llbracket 10, 14 \llbracket$ in Fig. 2.4. Namely, after going up to the interval $\llbracket 15, 19 \llbracket$, the system has a higher tendency to return to a lower interval. Note that this behavior is opposite in direction to the stable interval $\llbracket 20, +\infty \llbracket$. Actually, the general trend of a majority of macro-transitions opposes the direction of the interval containing the steady state point. Mainly this is because it

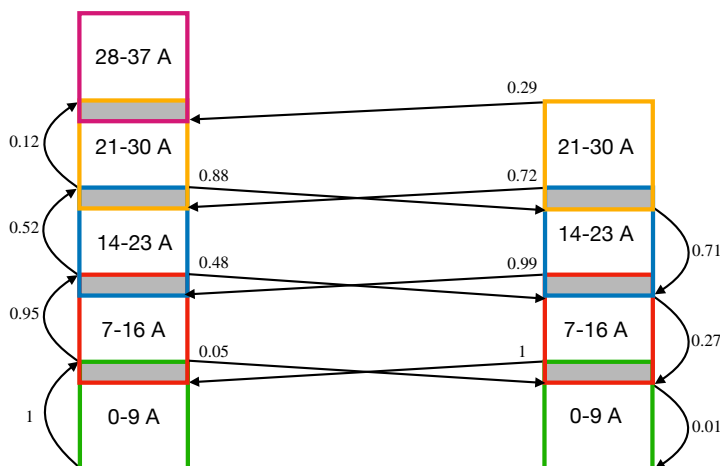


Figure 2.5: An overlapping macro-transition for the BD system. The interpretation is similar to the non-overlapping case in 2.4 with the exception that intervals overlap (denoted by a gray region)

requires only one transition in the initial reaction network to go back to the previous interval, whereas several ones are required to cross an interval entirely. Hence border effects give too much importance to backwards macro-transitions.

To cope with this artifact of the abstraction, we introduce a minimal effort for the system to perform fluctuations between consecutive intervals by using overlapping intervals instead. In Fig. 2.5, we compute a macro-transition system for the same BD system as in Fig. 2.4 but with overlapping intervals (overlaps are indicated in gray). The meaning of macro-transitions has to be defined carefully: we consider a macro-transition between a first interval and a second one, only when the system leaves the first interval (hence crossing the overlapping region). We adjust the position of the source and target of the macro-transitions accordingly in the drawing.

Now, the stable interval is $\llbracket 14, 23 \llbracket$. Starting from the value $q(A) = 10$, the probability to exit from the upper bound 16 of the interval $\llbracket 7, 16 \llbracket$ is equal to 0.95 while the probability to exit from its lower bound 7 is equal to 0.05. Consequently, we obtained the macro-transition $\llbracket 7, 16 \llbracket \xrightarrow{0.95} \llbracket 14, 23 \llbracket$ which shows the tendency to move towards the stable interval. Similar observations can be made about the other macro-transitions leading to the interval $\llbracket 14, 23 \llbracket$.

We notice two major features from our overlapping interval design. Firstly, quantities of molecule A contained in overlapping regions must surpass a buffer region to be able to go back into the previous interval, thus limiting border effects. And, secondly, the general trend of the system reflects a greater likelihood towards the stable interval which was not the case for non-overlapping intervals.

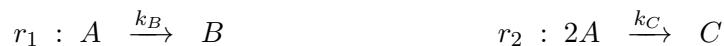
Altogether, we show how our framework is capable of coarse-graining the behaviors emerging from the BD reaction network using a more formal approach. Whereas discretization with non-overlapping intervals was not enough to keep only the likely behaviors as done in the logical models, the use of overlapping intervals introduces a minimal effort for the system to go back after entering an interval. The result is an abstract transition system when unnatural behaviors are assigned low probabilities, hence allowing to quantify the probability of the behaviors that are neglected in the hand-written logical model.

2.3 Case Study II: Model of Resource Competition

Now we shift gears to the second case study: the resource competition system.

2.3.1 Reaction network

The second case study is composed of two reactions:



where two chemical species, B and C are produced and each consume a common resource, A . In reaction r_1 , a quantity of B is produced with a kinetic constant k_B . In reaction r_2 , a quantity of C is produced with kinetic constant k_C . The production of molecule B consumes a quantity of A , while C requires two.

As in the first case study, we assume stochastic mass-action kinetics law to obtain the propensities $k_B \cdot q(A)$ and $\frac{k_C \cdot q(A) \cdot (q(A)-1)}{2}$ for, respectively, the reactions r_1 and r_2 . Consequently, we derive a probability function for each reaction:

$$\mu_1(q) = \frac{2 \cdot k_B \cdot q(A)}{q(A) \cdot (2 \cdot k_B + k_C \cdot (q(A)-1))} \text{ and } \mu_2(q) = \frac{k_C \cdot q(A) \cdot (q(A)-1)}{q(A) \cdot (2 \cdot k_B + k_C \cdot (q(A)-1))}.$$

Contrary to the first case study, this reaction system does not have reversible reactions and the quantities of B and C are strictly increasing up to the point at which all resources become depleted.

2.3.2 A logical model of competition for resources

As in the first case study, we can write by hand a logical model for this second example. The property of interest is the competition between the production of the molecules B and C . Namely, depending on the quantity of the resource A , either B or C is produced more abundantly. When the quantity of A is under a certain value, the quantity of B increases more; and when it is above this value, the quantity of C increases more.

Using this knowledge we obtain the logical model that is described in Fig. 2.6. The directed graph shows different kinds of regulations. Firstly, A regulates B and C because it may be consumed to produce them. Secondly, each component auto-regulates itself since whatever it increases or decreases, its quantity at the next time step depends on the one at the current state. Lastly, A auto-regulates itself negatively (since it is consumed to produce B and C). We can abstract quantities by Boolean values. Namely, the state is a triple of Boolean variables $(x_A, x_B, x_C) \in \{0, 1\}^3$, here 0 stands for low quantity and 1 for high quantity. The logical (Boolean) functions are derived to capture the main feature of the reaction network. A gets consumed, while either B or C is produced according to the qualitative abundance of A .

Several update policies exist to define the operational semantics. Our transition system is derived by assuming the synchronous one, where the value of each chemical species is updated at each time step. It induces the eight transitions that are described in Fig. 2.6. It is worth mentioning that similar results can be achieved in the asynchronous mode by taking into account

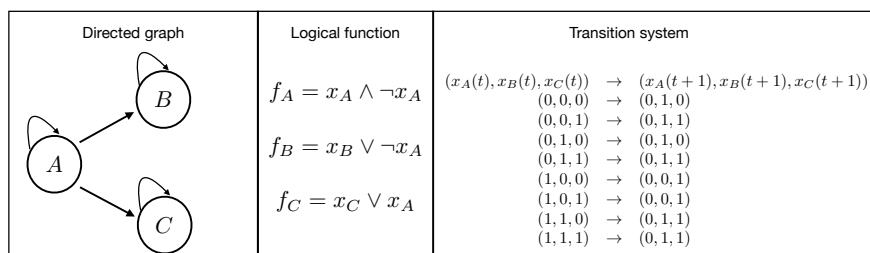


Figure 2.6: A logical model for a system of resource competition. A directed graph (left) displays how molecule B and C have a common regulator, molecule A. Moreover, each kind of molecules auto-regulates it-self. Each logical function (center) is a Boolean rule which reflects the update scheme for a given chemical species. They can take values in the domain 0, 1. The value 0 stands for low, 1 for high. The transition system (right) reflects the system dynamics from the logical functions.

mass preservation of invariants and using priorities [1]. The model indicates that starting with a low amount of the molecule A , the system produces some B , but no C ; whereas with a high amount, the system produces firstly some C , then some B . One may wonder whether more behaviors may occur in the underlying reaction system that have been discarded by the simplification into a logical model.

2.4 A formally derived model of competition for resources

This motivates the formal discretization of the reaction network with overlapping intervals to compare the behaviors of the so-obtained model to the ones of the logical model. Specifically, we wonder whether our framework is capable of highlighting both main behaviors (production of B , or production of C followed by production of B), and provides low upper bounds for the probability of behaving differently.

Firstly, in order to simplify the computation, we would like to eliminate the variable $q(A)$ which stands for the quantity of A in the system. We denote as $q \in \mathcal{Q}_{\mathcal{R}}$ the chemical state, which contains copy numbers of molecules A , B , and C . The system is constrained by the following mass invariant $q(A) = q'_0(A) - (q(B) - q'_0(B)) - 2 \cdot (q(C) - q'_0(C))$, where q'_0 stands for the initial chemical state and is fixed once for all. We can safely replace each occurrence of the variable $q(A)$ with the right hand side of this equality in any expression.

As in the first case study, the domain of values can be sampled into overlapping intervals. This way, chemical states are gathered into rectangular regions (we are left with only two dimensions since we have eliminated the quantity of A), that are called macro-states. Our goal is then to derive some quantitative information about the macro-transitions in the so-obtained discretized model. More specifically, when a chemical state enters a new macro-state, the goal is to compute the probability that the system will cross the corresponding rectangular region and exit along the same axis, or via the alternate axis. Note that when entering a new macro-state, we do not know precisely the chemical state of the system. Thus, any potential position on the entering side must be considered (e.g., the system may be arbitrary close to the corner of the

rectangular region so that exiting the rectangle via the alternate axis may require only one step in the concrete, see Fig. 2.7). In such a scenario, one may have to deal with several more fictitious abstract behaviors that may impact the analysis of the underlying system. Consequently, in order to retain a minimal effort strategy, we do not consider the next consecutive interval in the alternate axis, but the subsequent one (see Fig. 2.8). For instance, when an event drives molecule B into a new abstract interval, we consider as target goals the next consecutive interval for molecule B and the next two consecutive abstract values for molecule C , since we do not know precisely the concrete amount of C . The initial macro-state receives particular treatment: we can safely compute by which rectangular face a chemical state will exit, since the initial state is known perfectly.

The general framework described in Chapter 6 provides for any pair of thresholds for the quantities of molecules B and C and for any chemical state $q \in \mathcal{Q}_{\mathcal{R}}$, the probability that the quantity of the molecule B reaches its threshold before the molecule C , and conversely. We denote by $\llbracket m_B, M_B \rrbracket$ (resp. $\llbracket m_C, M_C \rrbracket$) an interval for the quantity of the molecule B (resp. C). We introduce $P_{g_1}^{\mathcal{G}}(q)$ (resp. $P_{g_2}^{\mathcal{G}}(q)$) as the probability for a chemical state where the quantity of B (resp. C) reaches the threshold M_B (resp. M_C) before that the quantity of C (resp. B) reaches the threshold M_C (resp. M_B) when starting from a state with $q(B)$ and $q(C)$ instances of the molecule B and C . Therefore, the probability $P_{g_1}^{\mathcal{G}}(q)$ satisfies the following relation:

$$P_{g_1}^{\mathcal{G}}(q) = \begin{cases} 1 & \text{whenever } q(B) = M_B, \\ 0 & \text{whenever } q(B) < M_B \text{ and } q(C) = M_C, \\ \mu_1(q) \cdot P_{g_1}^{\mathcal{G}}(q') + \mu_2(q) \cdot P_{g_1}^{\mathcal{G}}(q'') & \text{otherwise.} \end{cases} \quad (2.2)$$

for every $q(B), q(C) \in \mathbb{N}$, such that $m_B \leq q(B) \leq M_B$ and $m_C \leq q(C) \leq M_C$. A similar expression can be obtained for $P_{g_2}^{\mathcal{G}}(q)$ by switching the base cases for the alternate axis. First two cases stand for the boundary conditions (where thresholds are reached) whereas the third case captures an increase in molecule B with probability $\mu_1(q)$ or C with probability $\mu_2(q)$.

In general, Eq. 2.2 can be computed exactly by means of inverting a matrix (or equivalently solving a linear system of equations) or approximated, from below, by using a finite expansion of the sequence of the powers of a sparse matrix. We derive and discuss this computation into detail in Chapter 6. Here, since the quantities of the molecules B and C never decrease, the recurrence relation can be solved exactly (up to rounding errors) in a finite amount of iterations.

In the logical version of the reaction network, the unlikely behaviors when C is produced at low abundance of A and when B is produced at high abundance of A have been discarded. We thus test our framework in capturing a low upper bound on the probability of the corresponding macro-transitions in the formal discretization of the underlying reaction network. As a result, in Fig. 2.8, we computed a macro-transition system for two scenarios: when the copy number of A is low (Fig. 2.8, left) or high (Fig. 2.8, right). As kinetic constant, we took $k_B = 20$ and $k_C = 1$ and parameterize intervals to reflect the system steady state, which is again when $q(A) = 20$. We tune the initial values $q_0(A)$ respectively to 15 and 100. In Fig. 2.8, a rectangular region represents a macro-state and is composed each of a respective range of values for molecules B

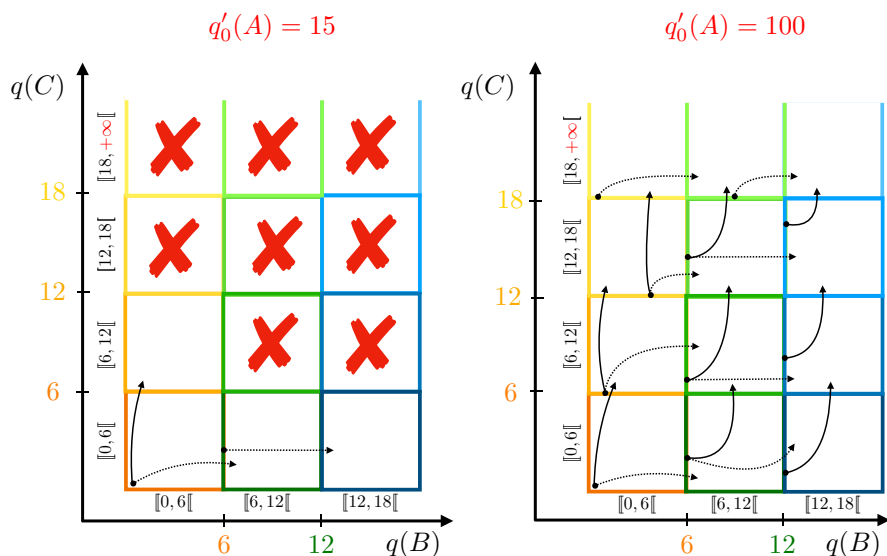


Figure 2.7: A non-deterministic macro-transition system of the resource competition system without a minimal effort system. In the first case (left), the initial amount of A is equal to 15 whereas it is equal to 100 in the second case (right). The states of both systems are related by edges with one source and at least one target.

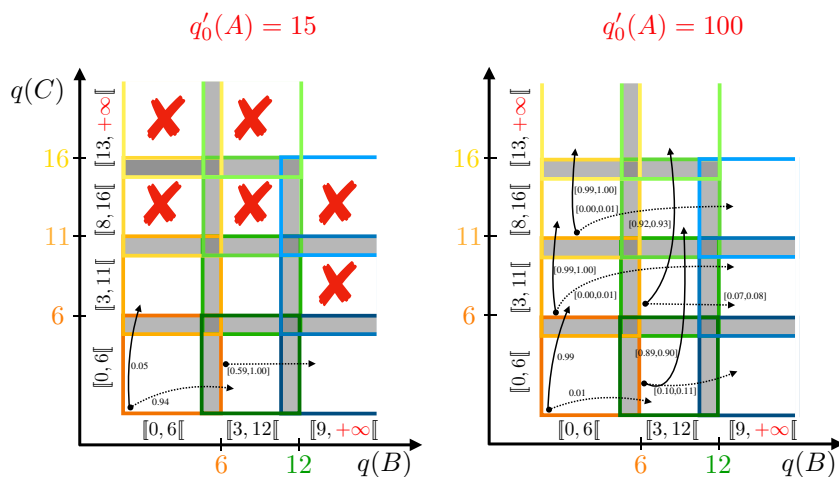


Figure 2.8: The derived coarse-grained transition systems for different initial quantities of the molecule A. In the first case (left), the initial amount of A is equal to 15 whereas it is equal to 100 in the second case (right). The states of both systems are related by edges with one source and two targets, with the following meaning: upon entering a new state (the source), the range of probabilities reflects the probability to reach a target before the other one.

and C . A labeled edge connects a source macro-state to a target macro-state. Each time, a dashed edge, that describes an increase in the quantity of the molecule B , competes with a solid edge, that describes an increase in the quantity of the molecule C .

Let us take an example by considering, in the first scenario, when $q_0(A) = 15$, the sequence of chemical reactions that starting from the initial state (where $q(B) = 0$ and $q(C) = 0$) drives the system out of the region $(\llbracket 0, 5 \rrbracket, \llbracket 0, 5 \rrbracket)$ into the region $(\llbracket 3, 11 \rrbracket, \llbracket 0, 5 \rrbracket)$. To recover the probability for this macro-transition, by Eq. 2.2 we can compute the probabilities to reach each next target macro-state. As such, the probability to exceed, from the initial state, the upper bound $q(B) = 5$ (resp. upper bound $q(C) = 5$) first is equal to 0.94 (resp. 0.05). The remaining 0.01 probability corresponds to the case where the system reaches the state where $q(B) = 5$ and $q(C) = 5$ (that is to say that the molecule A is completely depleted before having left the initial macro-state). This describes precisely the directional tendency of the behavior of the underlying reaction network. Combining these elements results in the macro-transition $(\llbracket 0, 5 \rrbracket, \llbracket 0, 5 \rrbracket) \xrightarrow{0.94} (\llbracket 3, 11 \rrbracket, \llbracket 0, 5 \rrbracket)$ (indicated by a dashed edge on the left in Fig. 2.8). Given a low resource environment, this macro-transition highlights the tendency towards a regime where the molecule B is created abundantly. Additionally, macro-transitions towards creation of the molecule C either occur with low probability or are not possible due to limited resources (indicated by red crosses).

In the second scenario (Fig. 2.8, right) we tune the initial resource pool to $q_0(A) = 100$ and retain the same kinetic conditions. We immediately observe that there are many more macro-transitions than in the first scenario, a consequence to the abundance in the common resource. As an example, we detail the computation for the bounds on the probability of the macro-transition $(\llbracket 0, 5 \rrbracket, \llbracket 3, 10 \rrbracket) \rightarrow (\llbracket 0, 5 \rrbracket, \llbracket 8C, 15C \rrbracket)$, when the region $(\llbracket 0, 5 \rrbracket, \llbracket 3, 10 \rrbracket)$ has been entered from below (i.e. by increasing the abundance of C). Before starting any computation, it is worth noting that when entering the region $(\llbracket 0, 5 \rrbracket, \llbracket 3, 10 \rrbracket)$ from below, $q(B)$ ranges arbitrarily between 0 and 5, while $q(C)$ is equal to 6. Thus in the computation of the probabilities of macro-transitions we have to consider any potential value for $q(B)$ between 0 and 5. Then we compute the minimal probability that the quantity of the molecule C exceeds 10 before the quantity of the molecule B exceeds the quantity 11. By applying Eq. 2.2, we obtain 0.99 it is indeed obtained when entering the region for $(\llbracket 0, 5 \rrbracket, \llbracket 3, 10 \rrbracket)$ with the state $[B \mapsto 5, C \mapsto 6]$, the maximal probability that we obtain is 1.00 (when entering this region with the state $[B \mapsto 0, C \mapsto 6]$). Indeed the value is not exactly 1 but it is conservatively rounded to 1 because of floating point arithmetics. This highlights that the molecule C is abundantly created with very high probability at the begin of the system execution, and then eventually some B is synthesized.

Reflecting on the two scenarios, it becomes clear that one can bound accurately the probabilities on the likelihood for macro-transitions. Our coarse-graining approach has the following benefits. Firstly, our framework provides a means to compute lower and upper bounds on the probabilities of the transitions between macro-states by formally relating the semantics of reaction networks to its abstract counterpart. Hence providing formal confidence in the formally derived discretized models. Secondly, by enforcing a minimal effort for the system to perform any transition between macro-states, we were able to observe the expected dynamics, the same

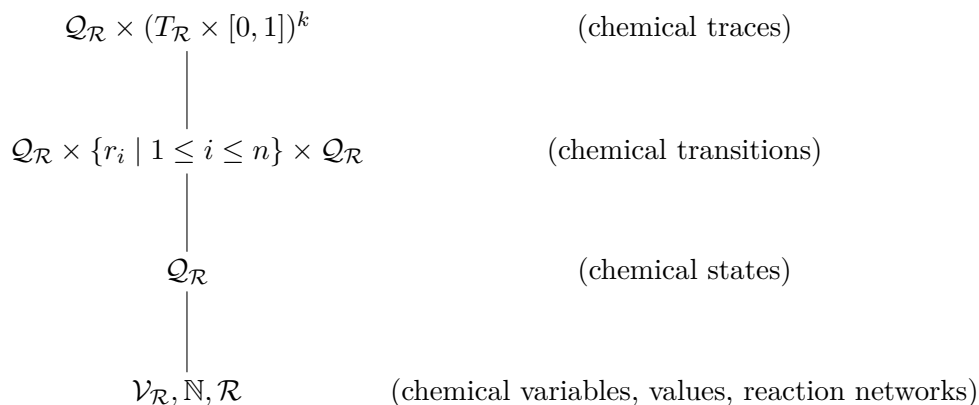


Figure 2.9: The hierarchy of the mathematical objects, each containing chemical information, that we require to characterize the stochastic behaviors of a reaction network.

as in the logical model but without relying on arguments on concentration- and time-scale separation. This has been obtained by twisting the abstraction function instead. This way, the interpretation of the behaviors of the abstract model is less intuitive, but it is still rigorously formally specified. Finally, it is possible to assess the validity of logical models by providing upper bounds to the probabilities of the transitions that has been discarded during the modeling process without any formal justification.

2.5 Conclusion

Accordingly, in this section we derived all the (concrete) mathematical objects we require to describe the behaviors of a stochastic reaction network (Fig. 2.9). Within each object is information regarding the quantity of chemical species and their relations among each other which are drive by the reactions involving one another. Additionally, we compute the likelihood of these reactions to occur, which give rise to the stochastic transition systems we focus on throughout the manuscript.

In Chapter 3 we shift gears and define our mathematical toolbox that will be used to coarse-grain the behaviors of a reaction network. That is, we introduce the set properties, functions, and relations between concrete and abstract domains that will permit us to traverse between varying levels of information of a reaction network and further highlight properties of interest.

Chapter 3

Mathematical basis for building a
coarse-graining framework

Objectives

In this Chapter, introduce our mathematical toolbox that will allow us to relate concrete and abstract sets of information; and, the minimal structures we require to traverse between these set types.

Contents

3.1	Sets and set properties	33
3.2	Distances and projections between sets of elements	34
3.3	Posets and poset properties	40
3.4	Conclusion	46

3.1 Sets and set properties

We begin by introducing several notions regarding sets and the operations to combine them. A set is an unordered collection of non-repeated elements. They are usually denoted as a list of elements, separated by commas, and enclosed by curly brackets. For example, $D = \{2, -4, -1, 5\}$ is an example of a set denoted D that contains four elements. To count the amount of elements in a set, one can compute its cardinality $|D| = 4$. A set that contains zero elements is denoted \emptyset . If an element belongs to a set then we use the symbol \in to denote set membership, otherwise we use \notin . For example, $-1 \in D$ and $4 \notin D$ are true statements. Additionally, one can filter in a set elements that possess an intrinsic property (or hypothesis). For example, the set $\{s \in S \mid H(s)\}$ is composed of the set of elements s from a set S that hold the property $H(s)$. A specific example is $D' = \{d \in D \mid d > 0\}$ which contains the set of elements from D that are positive: $D' = \{2, 5\}$. It is possible to compound properties with a variety of logical mathematical operators: \wedge (AND), \vee (OR), $=$ (equal), to name a few. Further, D' is said to be a subset of D which can be written as $D' \subseteq D$. The powerset is a set containing all the possible subsets of a set D , and is denoted $\wp(D)$. We will also use well known number sets such as the set of real numbers, integers, and natural numbers which are denoted, respectively, \mathbb{R} , \mathbb{Z} , and \mathbb{N} .

We can perform operations over sets of elements. For example, the intersection of two natural number sets O_1 and O'_1 is the set that contain only those elements present in both sets: $O_1 \cap O'_1 = \{o \in \mathbb{N} \mid o \in O_1 \wedge o \in O'_1\}$. the set union $O_1 \cup O'_1 = \{o \in \mathbb{N} \mid o \in O_1 \vee o \in O'_1\}$ is composed of all elements that are present in at least one of the two sets. The complement of a set O_1 for some universe \mathcal{U} is $O_1^c = \{o \in \mathbb{N} \mid o \in \mathcal{U} \wedge o \notin O_1\}$ or $O_1^c = \mathcal{U} \setminus O_1$ by the set difference operator \setminus . A cartesian product of the sets O_1 and O'_1 is a set composed of ordered pairwise components from each set: $O_1 \times O'_1 = \{(o, o') \in \mathbb{N} \times \mathbb{N} \mid o \in O_1 \wedge o' \in O'_1\}$. A more specific example follows.

Example 3.1. Let us take the cartesian product between two integer sets $G = \{-2, -3, 4\}$ and $S = \{0, 5\}$:

$$G \times S = \{(-2, 0), (-2, 5), (-3, 0), (-3, 5), (4, 0), (4, 5)\}.$$

More generally,

$$\underbrace{O \times O \times \dots \times O}_{\ell\text{-times}} = \{(o_1, o_2, \dots, o_\ell) \in \mathbb{N}^\ell \mid o_1 \in O \wedge o_2 \in O \wedge \dots \wedge o_\ell \in O\}$$

can be expressed as O^ℓ and is a set composed of sequences of length $\ell \in \mathbb{N}$. Indeed, an element of O^ℓ is usually written as $(o_\ell)_{\ell \in \mathbb{N}}$ where o_ℓ is the ℓ -th element of the sequence.

We will also use pervasively functions to transform objects. A function $f : X \rightarrow Y$ maps each element x in the domain X to the element $f(x)$ in the co-domain Y . When the domain and co-domain of a function are the same, then the function is referred to as an operator. The

symbol \circ denotes a composition between two functions. For example, a composition between the two functions $g : W \rightarrow Z$ and $h : X \rightarrow W$ is denoted as $g \circ h$. Note that the co-domain of h has to be equal to the domain of g . Certain functions also retain specific properties, and we will derive these on demand.

3.2 Distances and projections between sets of elements

In the following collection of definitions, we define a variety of functions that can either manipulate, assign, or provide a metric among and between sets of elements. For example, the first of a collection of functions we will define pertains to a decomposition of a component value in a sequence onto its coordinate.

Definition 3.1 (Orthogonal projection). *Let \mathcal{V} be a finite set of variables. The orthogonal projection on a component $v \in \mathcal{V}$ is defined as:*

$$\pi_v = \begin{cases} \mathbb{N}^{\mathcal{V}} & \rightarrow \mathbb{N} \\ (\rho_v)_{v \in \mathcal{V}} & \mapsto \rho_v. \end{cases}$$

In Def. 3.1, the domain and co-domain of π_v is, respectively, the set of ordered sequences, each of length $|\mathcal{V}|$, and the set of natural numbers. Thus, an element of $\mathbb{N}^{\mathcal{V}}$ can be decomposed on each of its components by using the respective projections. projection is a decomposition of a $|\mathcal{V}|$ -dimensional sequence of values onto the value in the v -th dimension.

Example 3.2. Let $\{w, z\}$ be the finite set of variables and $x_w, x_z \in \mathbb{N}$ be two natural numbers such that $(x_w, x_z) \in \mathbb{N}^{\{w, z\}}$ is a two-dimensional coordinate. The orthogonal projection of the natural number pair onto the w -th dimension is $\pi_w(x_w, x_z) = x_w$.

We can immediately extend the notion of orthogonal projection over sets of sequences.

Definition 3.2 (Set extension of the i -th coordinate). *Let \mathcal{V} be a finite set of variables. The orthogonal projection of a set of sequences onto a set of values in component $v \in \mathcal{V}$ is defined as:*

$$\pi_v = \begin{cases} \wp(\mathbb{N}^{\mathcal{V}}) & \rightarrow \wp(\mathbb{N}) \\ X & \mapsto \{\pi_v(\rho) \mid \rho \in X\}. \end{cases}$$

In Def. 3.2, the domain and co-domain of the function π_v , respectively, is a powerset of a set of sequences and a set of elements. Thus, π_v inputs a set of sequences and outputs a set of values whose value in the sequence has been orthogonally projected in the dimension $v \in \mathcal{V}$.

The next function we would like to define is a distance metric between elements belonging to a set.

Definition 3.3 (Gaussian distance). *The Gaussian distance between point pairs is:*

$$\text{dist}_{\mathbb{N}^\nu \times \mathbb{N}^\nu} = \begin{cases} \mathbb{N}^\nu \times \mathbb{N}^\nu & \rightarrow \mathbb{R} \\ (x, x') & \mapsto \sqrt{\sum_{v \in \mathcal{V}} (\pi_v(x) - \pi_v(x'))^2}. \end{cases}$$

In Def. 3.3, the domain and co-domain of the function $\text{dist}_{\mathbb{N}^\nu \times \mathbb{N}^\nu}$ is, respectively, a set composed of paired natural number sequences and the set of real numbers. A distance computation between paired sequences is done by taking the squared sum of the difference between each respective component values obtained from their respective orthogonal projections. In Sect. 5 we will use the distance function pervasively to compare the position of overlapping intervals. Indeed this will allow us to determine which interval is nearest when, and if, a sequence of values exit from one into another. Actually, the next definition will help us to make these type of computations, as the objects we consider distances between are products of sets containing each a range of values. These result in sets with are convex.

Definition 3.4 (Convex sets). *Let X is a set of natural numbers. The set X is convex if and only if one of the following assertions is satisfied:*

1. $X = \emptyset$;
2. $\exists x, x' \in \mathbb{N}$ such that $X = \{n \in \mathbb{N} \mid x \leq n \leq x'\}$;
3. or, $\exists x$ such that $X = \{n \in \mathbb{N} \mid x \leq n\}$;

We introduce few notations for non-empty convex sets of natural numbers. The bounded non-empty convex set $\{n \in \mathbb{N} \mid x \leq n \leq x'\}$ is usually written as $\llbracket x, x' \rrbracket$ or, equivalently, $\llbracket x, x' + 1 \rrbracket$, whereas the unbounded convex set $\{n \in \mathbb{N} \mid x \leq n\}$ is written as $\llbracket x, +\infty \rrbracket$.

In the following definition, we define the projection of an element onto a set.

Definition 3.5 (Projection onto a set). *Let $X \subseteq \mathbb{N}$ be a non-empty convex set of elements and $x, y \in \mathbb{N}$ be two natural numbers. An element y is a projection of x onto X if and only if:*

1. $y \in X$;
2. and, $\forall y' \in X, \text{dist}_{\mathbb{N}^\nu \times \mathbb{N}^\nu}(y - x) \leq \text{dist}_{\mathbb{N}^\nu \times \mathbb{N}^\nu}(y - y')$.

In Def. 3.5, the one-dimensional projection of an element onto a set is an element in the set nearest to the one projected. By nearest, we refer to the distance between two elements that retains the smallest value. In the following lemma we show that the one-dimensional projection of an element is unique.

Lemma 3.1. *Let Y be a non-empty convex set. Let $x \in \mathbb{N}$. Then x has a unique projection on Y .*

Proof. Let Y be a non-empty interval. Let $x \in \mathbb{N}$.

We make the proof of Lemma 3.1 by disjunction of cases, depending on whether x is smaller than the elements of Y , an element of Y , or greater than the elements of Y .

1. In the case where Y is of the form $\llbracket m, M \rrbracket$ or $\llbracket m, +\infty \llbracket$, with $x < m$.

Let us prove that m is the projection of x onto the set Y .

- (a) By definition we have $m \in Y$.
 (b) Let x' be an element of $Y \setminus \{m\}$.

We have: $m < x'$.

Thus, $x < m < x'$.

That is to say that $|x - m| < |x - x'|$.

We can conclude that x has a unique projection on the set Y and that this projection is equal to m .

2. In the case where Y is of the form $\llbracket m, M \rrbracket$, with $x > M$.

- (a) By definition we have $M \in Y$.
 (b) Let x' be an element of $Y \setminus \{M\}$.

We have: $x' < M$.

Thus, $x' < M < x$.

That is to say that $|x - M| < |x - x'|$.

We can conclude that x has a unique projection on the set Y and that this projection is equal to M .

3. The third case where $x \in Y$.

We have $x \in Y$ and $|x - x| = 0$.

Moreover, for any other element x' in Y , we have $|x - x'| > 0$.

Thus, x has a unique projection on the set Y and that this projection is equal to x .

□

We denote as $proj_{\mathbb{N}}(x, Y)$ the projection of x onto the non-empty convex set Y .

Accordingly, we define a special kind of convex sets that we refer to as boxes.

Definition 3.6 (Box). *Let \mathcal{V} be a set of variables. A box B in $\llbracket 1, n \rrbracket^{\mathcal{V}}$ can be defined either as the empty set or as the Cartesian product of some non-empty bounded convex sets. That is to say that:*

$$B = \prod_{v \in \mathcal{V}} \llbracket m_v, M_v \rrbracket$$

where $(m_v)_{v \in \mathcal{V}}$ and $(M_v)_{v \in \mathcal{V}}$ are two n -tuples of elements in \mathbb{N}^v such that for every v in \mathcal{V} , the property $m_v \leq M_v$ is satisfied.

The set of all the boxes in $\llbracket 1, n \rrbracket^{\mathcal{V}}$ is denoted $\text{Box}(\llbracket 1, n \rrbracket^{\mathcal{V}})$

Projections can also be performed over non-empty boxes.

Definition 3.7 (Sequence Projection). *Let B be a non-empty box. Let $u, b \in \mathbb{N}^{\mathcal{V}}$. We say that b is a projection of the element u on the set B if and only if:*

1. $b \in B$;
2. and, $\forall b' \in B, \text{dist}_{\mathbb{N}^{\mathcal{V}} \times \mathbb{N}^{\mathcal{V}}}(u - b) \leq \text{dist}_{\mathbb{N}^{\mathcal{V}} \times \mathbb{N}^{\mathcal{V}}}(u - b')$.

Here, Def. 3.7 is similar to 3.5 with the exception that the operations are done over many components. In the following lemma, we show that a box set membership is preserved under orthogonal projection.

Lemma 3.2. *Let \mathcal{V} be a set of variables. Let Y be a box in $\mathcal{B}ox(\llbracket 1, n \rrbracket^{\mathcal{V}})$. Let $u \in \mathbb{N}^{\mathcal{V}}$. Then:*

$$\begin{aligned} u \in Y &\iff \forall v \in \mathcal{V}, \pi_v(u) \in \dot{\pi}_v(Y) \\ u \in Y &\iff \forall v \in \mathcal{V}, \exists b \in Y, \pi_v(u) = \pi_v(b). \end{aligned}$$

Proof. Let Y be a box in $\mathcal{B}ox(\llbracket 1, n \rrbracket^{\mathcal{V}})$.

Let $u \in \mathbb{N}^{\mathcal{V}}$.

1. If Y is equal to the empty set.

- (a) The property $u \in Y$ is false.
- (b) Let us prove that the property $\forall v \in \mathcal{V}, \pi_v(u) \in \dot{\pi}_v(Y)$ is also false.

Let v be an element in \mathcal{V} (which is non-empty).

The set $\dot{\pi}_v(Y)$ is empty (since Y is empty).

Were x an element of $\dot{\pi}_v(Y)$,

there would be an element $y \in Y$ such that: $x = \pi_v(y)$,

which is absurd, since Y is empty.

Thus $\pi_v(u) \notin \dot{\pi}_v(Y)$.

- (c) Let us prove that the property $\forall v \in \mathcal{V}, \exists y \in Y, \pi_v(u) = \pi_v(y)$ is also false.

This is the case because \mathcal{V} is non-empty while Y is.

2. If Y is non-empty.

By definition 3.6, we can introduce two n -tuples $(m_v)_{v \in \mathcal{V}}$ and $(M_v)_{v \in \mathcal{V}}$ of elements in \mathcal{I}^\sharp such that $m_v \leq M_v$ for every $v \in \mathcal{V}$ and:

$$Y = \prod_{v \in \mathcal{V}} \llbracket m_v, M_v \rrbracket.$$

Thus we have: $\llbracket m_v, M_v \rrbracket = \dot{\pi}_v(Y)$ for every $v \in \mathcal{V}$.

Moreover u can be written as $(u_v)_{v \in \mathcal{V}}$ with $u_v \in \mathcal{I}^\sharp$ for every $v \in \mathcal{V}$.

Indeed, $u_v = \pi_v(u)$ for every $v \in \mathcal{V}$.

- (a) $u \in Y \Rightarrow \forall v \in \mathcal{V}, \pi_v(u) \in \dot{\pi}_v(Y)$.

We assume that $u \in Y$.

It follows that: $(u_v)_{v \in \mathcal{V}} \in \prod_{v \in \mathcal{V}} \llbracket m_v, M_v \rrbracket$.

Thus, for every $v \in \mathcal{V}$, $u_v \in \llbracket m_v, M_v \rrbracket$.

Thus, for every $v \in \mathcal{V}$, $\pi_v(u) \in \pi_v(Y)$.

(b) $u \in Y \Leftrightarrow \forall v \in \mathcal{V}, \pi_v(u) \in \pi_v(Y)$.

We assume that $\forall v \in \mathcal{V}, \pi_v(u) \in \pi_v(Y)$.

Let $v \in \mathcal{V}$.

By assumption, we have: $\pi_v(u) \in \pi_v(Y)$.

It must follow that: $u_v \in \llbracket m_v, M_v \rrbracket$.

Thus, $(u_v)_{v \in \mathcal{V}} \in \prod_{v \in \mathcal{V}} \llbracket m_v, M_v \rrbracket$.

That is to say, $u \in Y$.

(c) $u \in Y \Rightarrow \forall v \in \mathcal{V}, \exists y \in Y, \pi_v(u) = \pi_v(y)$

We assume that $u \in Y$.

Let i be an element in \mathcal{V} .

We know that $u \in Y$ and by extension, $\pi_v(u) = \pi_v(u)$.

Thus, there exists an element $y \in Y$ such that $\pi_v(y) = \pi_v(u)$.

(d) $u \in Y \Leftrightarrow \forall v \in \mathcal{V}, \exists y \in Y, \pi_v(u) = \pi_v(y)$

We assume that $\forall v \in \mathcal{V}, \exists y \in Y, \pi_v(u) = \pi_v(y)$.

Let $v \in \mathcal{V}$.

Let, $\exists y \in Y, \pi_v(u) = \pi_v(y)$.

Let $y \in Y$ such that $\pi_v(u) = \pi_v(y)$

Since $Y = \prod_{v \in \mathcal{V}} \llbracket m_v, M_v \rrbracket$, we have: $\pi_v(y) \in \llbracket m_v, M_v \rrbracket$.

So $\pi_v(u) \in \llbracket m_v, M_v \rrbracket$.

That is to say $u_v \in \llbracket m_v, M_v \rrbracket$.

Thus for every $v \in \mathcal{V}$, $u_v \in \llbracket m_v, M_v \rrbracket$.

Since $u = (u_v)_{v \in \mathcal{V}}$, we can conclude that $u \in \prod_{v \in \mathcal{V}} \llbracket m_v, M_v \rrbracket$.

That is to say that $u \in Y$.

□

Further, we show that when an element in $\mathbb{N}^{\mathcal{V}}$ is closer to an element than another one, then it is necessary also true for at least one coordinate.

Lemma 3.3. *Let $u, u', u'' \in \mathbb{N}^{\mathcal{V}}$. Then:*

$\forall u, u', u'' \in \mathbb{N}^{\mathcal{V}}$,

$dist_{\mathbb{N}^{\mathcal{V}} \times \mathbb{N}^{\mathcal{V}}}(u, u') < dist_{\mathbb{N}^{\mathcal{V}} \times \mathbb{N}^{\mathcal{V}}}(u, u'') \Rightarrow \exists v \in \mathcal{V}, |\pi_v(u) - \pi_v(u')| < |\pi_v(u) - \pi_v(u'')|$.

Proof. Let $u, u', u'' \in \mathbb{N}^{\mathcal{V}}$ such that $dist_{\mathbb{N}^{\mathcal{V}} \times \mathbb{N}^{\mathcal{V}}}(u, u') < dist_{\mathbb{N}^{\mathcal{V}} \times \mathbb{N}^{\mathcal{V}}}(u, u'')$.

By definition, we have:

$$\sqrt{\sum_{v \in \mathcal{V}} (\pi_v(q^\sharp) - \pi_v(q^\sharp \iota))^2} < \sqrt{\sum_{v \in \mathcal{V}} (\pi_v(q^\sharp) - \pi_v(q^\sharp \iota \iota))^2}$$

Since both hand sides are non negative, we can conclude that:

$$\sum_{v \in \mathcal{V}} (\pi_v(q^\sharp) - \pi_v(q^\sharp \iota))^2 < \sum_{v \in \mathcal{V}} (\pi_v(q^\sharp) - \pi_v(q^\sharp \iota \iota))^2.$$

Then by compatibility of the sum, it follows that there exists an element $v \in \mathcal{V}$ such that:

$$(\pi_v(u) - \pi_v(u'))^2 < (\pi_v(u) - \pi_v(u''))^2.$$

Let i be an element in \mathcal{V} such that:

$$(\pi_v(u) - \pi_v(u'))^2 < (\pi_v(u) - \pi_v(u''))^2.$$

Then, since $\sqrt{\cdot}$ is monotonic, we have:

$$\sqrt{(\pi_v(u) - \pi_v(u'))^2} < \sqrt{(\pi_v(u) - \pi_v(u''))^2}.$$

That is to say:

$$|\pi_v(u) - \pi_v(u')| < |\pi_v(u) - \pi_v(u'')|.$$

We conclude that there exists $v \in \mathcal{V}$ such that:

$$|\pi_v(u) - \pi_v(u')| < |\pi_v(u) - \pi_v(u'')|.$$

□

By reasoning on Lemmas 3.2 and 3.3 leads to the following proposition showing that the projection of a sequence onto a box is unique.

Proposition 3.1. *Let Y be a non-empty box in $\mathcal{B}ox(\llbracket 1, n \rrbracket^\mathcal{V})$. Let $u \in \mathbb{N}^\mathcal{V}$. Then,*

1. u has a projection on Y ;
2. and, $proj(u, Y) = (proj(\pi_v(u), \dot{\pi}_v(Y)))_{v \in \mathcal{V}}$.

Proof. Let Y be a non-empty box in $\mathcal{B}ox(\llbracket 1, n \rrbracket^\mathcal{V})$.

Let $u \in \mathbb{N}^\mathcal{V}$.

1. Let us prove that $(proj(\pi_v(u), \dot{\pi}_v(Y)))_{v \in \mathcal{V}} \in Y$.

Let $v \in \mathcal{V}$.

By Def. 3.5, we know that $proj(\pi_v(u), \dot{\pi}_v(Y)) \in \dot{\pi}_v(Y)$.

Let us recall that $\dot{\pi}_v(Y) = \llbracket m_v, M_v \rrbracket$.

So $proj(\pi_v(u), \pi_v(Y)) \in \llbracket m_v, M_v \rrbracket$.

Additionally, we know that $Y = \prod_{v \in \mathcal{V}} \llbracket m_v, M_v \rrbracket$.

Thus, $(proj(\pi_v(u), \pi_v(Y)))_{v \in \mathcal{V}} \in Y$.

2. Let us prove that for any element $y \in Y$, we have

$$dist_{\mathbb{N}^{\mathcal{V}} \times \mathbb{N}^{\mathcal{V}}}(u, (proj(\pi_v(u), \pi_v(Y)))_{v \in \mathcal{V}}) \leq dist_{\mathbb{N}^{\mathcal{V}} \times \mathbb{N}^{\mathcal{V}}}(u, y).$$

Let y be an element of Y .

By contradiction, let us assume that:

$$dist_{\mathbb{N}^{\mathcal{V}} \times \mathbb{N}^{\mathcal{V}}}(u, y) < dist_{\mathbb{N}^{\mathcal{V}} \times \mathbb{N}^{\mathcal{V}}}(u, (proj(\pi_v(u), \pi_v(Y)))_{v \in \mathcal{V}}).$$

Recall by Lemma 3.3, $\exists v \in \mathcal{V}, |\pi_v(u) - \pi_v(y)| <$

$|\pi_v(u) - \pi_v((proj(\pi_v(u), \pi_v(Y)))_{v \in \mathcal{V}})|$.

Thus, there must be an $v \in \mathcal{V}$ such that:

$$|\pi_v(u) - \pi_v(y)| < |\pi_v(u) - \pi_v((proj(\pi_v(u), \pi_v(Y)))_{v \in \mathcal{V}})|.$$

Yet, $y \in Y$. It follows, by Def. 3.2 that $\pi_v(y) \in \pi_v(Y)$.

This is absurd by Def. 3.5.

It follows, by Def. 3.5 that u has a projection on Y and that this projection is indeed the element $(proj(\pi_v(u), \pi_v(Y)))_{v \in \mathcal{V}}$. \square

At this point we have carefully defined the necessary computations we require over sets to design our coarse-graining framework. In Sect. 3.3, we will define a particular set that will permit us to perform coarse-graining and highlighting certain features belonging to an ordered set of interest.

3.3 Posets and poset properties

We would like to begin by introducing a minimal structure to compare elements in a set: partially ordered sets.

Definition 3.8 (Partially ordered sets). *A partially ordered set (or, poset) over a set P is a binary relation $\sqsubseteq \in P \times P$ between elements in the set P that is:*

1. reflexive: $\forall p \in P, p \sqsubseteq p$;
2. anti-symmetric: $\forall p, p' \in P, p \sqsubseteq p' \wedge p' \sqsubseteq p \implies p = p'$;
3. and, transitive: $\forall p, p', p'' \in P, p \sqsubseteq p' \wedge p' \sqsubseteq p'' \implies p \sqsubseteq p''$

In such a case, a poset is denoted (P, \sqsubseteq) .

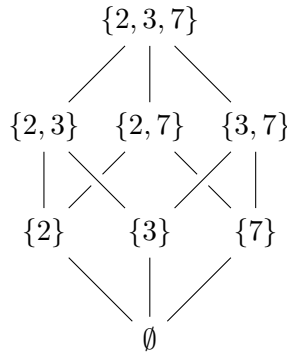


Figure 3.1: The Hasse diagram for the powerset of $A = \{2, 3, 7\}$.

A poset is a set with a binary relation. This relation may have certain properties. We would like a notion of compare pairs of elements in a set, thus the three poset axioms describes one such particular order. In our context, a poset carries information with varying degrees of precision. Namely, in the context of a powerset ordered by inclusion, the cardinality of a poset reflects this degree of precision such that the poset with a smaller number of elements (or, solutions) is deemed more precise.

Example 3.3. In Fig. 3.1 is a Hasse diagram for the powerset of $A = \{2, 3, 7\}$, which is an example of a partial order by set inclusion. Each node of a Hasse diagram is an element belonging to $\wp(A)$, and each edge (read from bottom to top) connects two sets that are ordered by \subseteq . Two elements that are partially ordered in a poset can provide information regarding precision. For example, in the following sequence $\{2\} \subseteq \{2, 3\} \subseteq \{2, 3, 7\}$ of the Hasse diagram the subset $\{2, 3\}$ is more precise than $\{2, 3, 7\}$ and less precise than $\{2\}$. Indeed, a set denotes the potential values for an element. Hence, knowing that an element $x \in \{2, 3\}$ is more precise than knowing that $x \in \{2, 3, 7\}$.

In a poset, it is possible to extract elements which possess a particular property such as upper and lower bounds of a subset and, respectively, the least and greatest one.

Definition 3.9 (Upper bound). *Let $\mathcal{X} \subseteq P$. $u \in P$ is an upper bound if $\forall x' \in \mathcal{X}, x' \subseteq u$. The set of upper bounds of \mathcal{X} is denoted \mathcal{X}^u .*

Definition 3.10 (Lower bound). *Let $\mathcal{X} \subseteq P$. $l \in P$ is a lower bound if $\forall x' \in \mathcal{X}, l \subseteq x'$. The set of lower bounds of \mathcal{X} is denoted \mathcal{X}^l .*

Definition 3.11 (Least upper bound). *Let $\mathcal{X}^u \subseteq P$. An upper bound $u \in \mathcal{X}^u$ is a least upper bound if $\forall u' \in \mathcal{X}^u, u \subseteq u'$.*

In the following proposition, we want to show that if a poset has a least upper bound, then this element is unique.

Proposition 3.2. *Let $\mathcal{X} \subseteq P$. If w and z are least upper bounds of the set \mathcal{X} , then $w = z$.*

Proof. Let w and z be two least upper bounds of the set \mathcal{X} .

By Def. 3.9, $w, z \in \mathcal{X}^{\mathcal{U}}$.

By assumption, w is a least upper bound, thus $w \sqsubseteq z$.

Also, z is a least upper bound, thus $z \sqsubseteq w$.

By the anti-symmetry axiom of posets, $w = z$. □

If it exists, the least upper bound for a set \mathcal{X} is denoted $\sqcup \mathcal{X}$ and is also called the join.

We would like to accomplish the same task and characterize the greatest lower bound of a poset.

Definition 3.12 (Greatest lower bound). *Let $X^{\mathcal{L}} \sqsubseteq P$. A lower bound $l \in X^{\mathcal{L}}$ is a greatest lower bound if $\forall l' \in X^{\mathcal{L}}, l' \sqsubseteq l$.*

Proposition 3.3. *Let $\mathcal{X} \sqsubseteq P$. If w and z are greatest lower bounds of the set \mathcal{X} , then $w = z$.*

Proof. Let w and z be two greatest lower bounds of the set \mathcal{X} .

By Def. 3.10, $w, z \in X^{\mathcal{L}}$.

By assumption, w is a greatest lower bound, thus $z \sqsubseteq w$.

Also, z is a greatest lower bound, thus $w \sqsubseteq z$.

By the anti-symmetry axiom of posets, $w = z$. □

If it exists, the greatest lower bound for a set \mathcal{X} is denoted $\sqcap \mathcal{X}$ and is also called the meet.

Given an arbitrary poset, Defs. 3.9 and 3.10 describe for any subset of this poset, respectively, a set of lower and a set of upper bounds; however, these elements may not necessarily bare set membership to the subset. Among the upper bounds, there may exist a least one (Def. 3.11). Similarly, among the lower bounds, there may be a greatest one (Def. 3.12). The proofs of Prop. 3.2 - 3.3 characterize that the least upper bound and greatest lower bound of a set are, when they exist, unique elements.

Example 3.4. We compute an example of the join and meet for the poset in Fig. 3.1. In this Hasse diagram, the set of upper bounds for the elements $\{2\}$ and $\{7\}$ is $\mathcal{X}^{\mathcal{U}} = \{\{2, 7\}, \{2, 3, 7\}\}$. By Def. 3.11, the join for these two given elements is $\sqcup \mathcal{X}^{\mathcal{U}} = \{2, 7\}$.

Respectively, for the two elements $\{2, 3\}$ and $\{3, 7\}$ the set of lower bounds is $X^{\mathcal{L}} = \{\emptyset, \{3\}\}$. Note that \emptyset is a member of any set. By Def. 3.12, the meet is $\sqcap X^{\mathcal{L}} = \{3\}$.

Example 3.5. Let $A = \{2, 3, 7\}$. We will look at a second example of a poset where a join may not exist.

The set $\wp_{\leq 2}(A) = \{Y \subseteq A \mid |Y| \leq 2\}$ has no join operator. Infeed, a subset of it has a join only if it contains at most one element. In such a case, it is its own lub.

Example 3.6. We consider a third example of a pair of elements with upper bounds, but no join. We consider the set $\mathcal{A} = \mathbb{Z} \cup \{-\infty_1, \infty_2\}$, with the relation R that is defined as xRy if and only if one of the following three properties is satisfied:

1. $x = y$;
2. $x, y \in \mathbb{Z}$ and $x \leq y$;
3. $y \in \mathbb{Z}$ and $x \in \{-\infty_1, -\infty_2\}$;

Intuitively, integers are ordered usually. Moreover, the elements $-\infty_1$ and $-\infty_2$ are two incompatible copies of the minus infinity element. Hence the two elements $-\infty_1$ and $-\infty_2$ are bounded by any number, but they do not have a least upper bound.

Interestingly, the powerset of any arbitrary set (including the one in our example), benefit from the existence of a join and a meet for any subset belonging to the powerset. This type of structural feature forms a special type of poset called a complete lattice.

Definition 3.13 (Complete lattice). *A complete lattice is a poset $(P, \sqsubseteq, \sqcup, \sqcap, \emptyset, P)$ such that each $P' \sqsubseteq P$ has a least upper bound and a greatest lower bound.*

Our efforts are geared towards using complete lattices. For example, in Chapters 4 and 5 we will use complete lattices during the abstraction process to highlight the type of imprecisions that can occur when one abstracts a concrete element into an abstract one also to coarse-grain the stochastic transition system for reaction networks. Actually, a useful complete lattice that we will use pervasively is the powerset domain.

As discussed previously, the powerset domain is a poset made of subsets ordered by the inclusion operator and expresses a useful mathematical structure to interrogate for information. For example, the example in Fig. 3.1 forms a complete lattice: $(\wp(A), \subseteq, \cup, \cap, \emptyset, A)$. The existence of a join and meet for each part of the powerset provides the ability to coarse-grain information at varying scales of (more or less precise) information. For example, in the case of reaction networks, we can check which values of chemical species belong to a region of interest, which can reflect a region where chemical behavior for the system is thermodynamically favorable. Thus, this ability to coarse-grain will be very useful during the process of abstraction in order to pinpoint general dynamical trends in a coarse-grained context rather than in the actual transition system, which can prove to be more cost effective. Thus, one can design a syntax to symbolically represent precise features present in certain sets and further build a semantics to automatically extract this information from a complete lattice.

In Sect. 4, we will introduce a generic concrete and an abstract semantics to demonstrate how it is possible to utilize complete lattices to highlight interesting features for an arbitrary transition system. This is accomplished by building the necessary mathematical objects to describe a concrete and abstract generic transition system and then build relations between sets of respective concrete and abstract elements and for each mathematical object. The goal,

thus, is to parse from the collection of sets of element information which can reveal interesting information about each transition system.

In the following collection of definitions, we will show how one can constructively build relations between concrete and abstract elements. We consider until the rest of the chapter two sets of elements C and D .

Definition 3.14 (Galois connection). *A Galois connection between the powerdomains $(\wp(C), \subseteq)$ and $(\wp(D), \subseteq)$ is a pair of functions $\alpha : \wp(C) \rightarrow \wp(D)$ and $\gamma : \wp(D) \rightarrow \wp(C)$ such that:*

$$\forall X \in \wp(C), Y \in \wp(D), X \subseteq \gamma(Y) \iff \alpha(X) \subseteq Y.$$

A Galois connection is typically represented as $(\wp(C), \subseteq) \xleftrightarrow[\alpha]{\gamma} (\wp(D), \subseteq)$.

In Def. 3.14, a Galois connection provides two representations between sets of elements belonging to two different domains: a concrete and an abstract one. Thus, a Galois connection provides a method to tunnel between sets of concrete and abstract elements. More specifically, the left-hand side of the equivalence in Def. 3.14 compares sets of concrete elements. Here, a concrete set of elements X is compared to a concretized set of abstract elements $\gamma(Y)$ with a set inclusion operator. Two interpretations arise from this comparison: (i) the concrete set X is more precise than the concrete set $\gamma(Y)$, and (ii) Y is an abstraction of the set X . The second interpretation is often referred to as soundness, namely that each concrete element in a concrete set is accounted for in the concrete representation of a corresponding abstract set (e.g., $\forall x \in X \implies x \in \gamma(Y)$). The right-hand side of the equivalence in Def. 3.14 compares sets of abstract elements. The interpretation is similar to the concrete case. That is, an abstract representation of a concrete set X , $\alpha(X)$, is more precise than a comparable, sound abstract set Y . Thus in a Galois connection, $\alpha(X)$ is an abstraction of the set X and among all the abstraction of X it is the most precise.

It is often useful to build for each element described in a semantics a Galois connection. This way, one ensures any computations done in one domain can be preserved in the other. For practical purposes, it may be useful to perform some computations in the abstract domain, which can be further refined on demand by reasoning in the concrete domain.

As an example of how to establish a Galois connection, we will derive a relation between sets of concrete and abstract elements. The first step is to derive an abstraction function that will traverse from a concrete domain to an abstract domain.

Definition 3.15 (Generic set abstract function). *The abstraction function maps a concrete element to an abstract element as:*

$$\beta : \begin{cases} \mathcal{C} & \rightarrow & D \\ c & \mapsto & a \end{cases}$$

In Def. 3.15, the domain and co-domain of β is, respectively, a set of concrete elements \mathcal{C} and a set of abstract elements D . Thus, the abstraction function is responsible for mapping a concrete element $c \in \mathcal{C}$ to a sound abstract element $\beta(c) \in D$.

Now, we lift this notion to sets of concrete elements to obtain sets of abstract elements.

Definition 3.16 (Set abstraction). *The abstraction of a set of concrete elements to a set of abstract elements is:*

$$\alpha = \begin{cases} \wp(C) & \rightarrow & \wp(D) \\ X & \mapsto & \{\beta(c) \in D \mid c \in X\} \end{cases} \quad (3.1)$$

In Def. 3.16, the domain and co-domain of α is, respectively, powersets of a set concrete and abstract elements. The function α will therefore take as an input a set of concrete elements and lift β onto each concrete element in the input set. The consequence is a set of abstract elements each over-approximating of its concrete counterpart. This is accomplished for all concrete sets, which leads soundness of all possible concrete elements.

Respectively, we can proceed in the other direction in order to traverse from sets of abstract elements to sets of concrete ones.

Definition 3.17 (Set concretization). *The concretization of a set of abstract elements to a set of concrete elements is:*

$$\gamma = \begin{cases} \wp(D) & \rightarrow & \wp(C) \\ Y & \mapsto & \{c \in C \mid \beta(c) \in Y\} \end{cases} \quad (3.2)$$

In Def. 3.17, the domain and co-domain of γ are reverse to Def. 3.16 and is, respectively, powersets of a set of abstract and concrete elements. Accordingly, this concretization function will input an abstract set and return a concrete set whose elements have the property that its abstraction belongs to the input set. Note that since each abstract element is an over-approximation of a concrete element, then the concrete set may contain several more solutions than a native concrete set.

Example 3.7. We consider the abstract domain D is the set of intervals $\{\llbracket 0, 4\llbracket, \llbracket 5, 9\llbracket, \llbracket 9, +\infty\llbracket\}$ and the abstraction function $\beta : \mathbb{N} \rightarrow D$ that maps each natural number $x \in \mathbb{N}$ to the unique interval I in D such that $x \in I$.

For example, abstracting the value 5 gives $\beta(5) = \llbracket 5, 9\llbracket$. We say the interval $\llbracket 5, 9\llbracket$ is a sound abstraction of the value 5.

We now consider the abstraction function α and the concretization function γ that are induced by the function β , as in Defs. 3.16 - 3.17. The abstraction of the set of natural numbers $\{3, 8, 9\}$ is $\alpha(\{3, 8, 9\}) = \{\llbracket 0, 4\llbracket, \llbracket 5, 9\llbracket\}$. Concretizing this abstract set leads to $\gamma(\{\llbracket 0, 4\llbracket, \llbracket 5, 9\llbracket\}) = \llbracket 0, 9\llbracket$. Indeed, $[\gamma \circ \alpha](\{3, 8, 9\})$ refers to the abstraction process of the concrete set $\{3, 8, 9\}$ and incurs a loss of precision. That is, $\{3, 8, 9\} \subseteq \{x \in \mathbb{N} \mid 0 \leq x < 9\}$.

Loss of precision incurred by an abstract analysis is discussed extensively in Chapter 4.

In the following proposition, we will show that α in Def. 3.16 and γ in Def. 3.17 form a Galois connection with respect to the abstraction function β in Def. 3.15.

Proposition 3.4. *Let C be the set of concrete elements and D be the set of abstract elements. Then the function pair α and γ form a Galois connection.*

Proof. By the definition of a Galois connection, we have:

$$\forall Y \in \wp(D), X \in \wp(C), X \subseteq \gamma(Y) \iff \alpha(X) \subseteq Y.$$

(\Leftarrow) Assume that $\alpha(X) \subseteq Y$.

Take $c \in X$ to be a concrete element.

By assumption, we know that $\alpha(X) \subseteq Y$ and $\beta(c) \in \alpha(X)$.

So, $\beta(c) \in Y$.

Thus, $c \in \{c \in C \mid \beta(c) \in Y\}$

So, $c \in \gamma(Y)$.

(\Rightarrow) Assume that $X \subseteq \gamma(Y)$.

Take $a \in \alpha(X)$ to be an abstract element.

There exists a concrete element $c \in X$ such that $a = \beta(c)$.

There exists a concrete element $c \in \gamma(Y)$ such that $a = \beta(c)$.

Let $c \in \gamma(Y)$ be a concrete element with the property $a = \beta(c)$.

We have $\beta(c) \in Y$.

So, $a \in Y$.

□

Recollecting this information, we show how one can build a Galois connection between sets of concrete elements and sets of abstract elements, although not all Galois connection can be built this way. In Sect. 4 and Sect. 5 we build Galois connection between various sets of concrete and abstract elements. The overall goal is to ensure that the abstractions we incur on each concrete element is sound. Additionally, we build a hierarchy of concrete and abstract semantics by building on each abstraction of a concrete element such sound-approximations from the simplest concrete element.

3.4 Conclusion

In this Chapter we devoted attention to introducing the mathematical toolbox used throughout the duration of the manuscript. In particular, we place an emphasis on the use of posets in order to provide a structure that would permit us to extract pertinent information about a system of interest. In addition, we introduced an important relation between two domains: the concrete and the abstract domain. This can be accomplished by establishing a Galois connection between two domains.

We will see in Chapters 4 and 5 that we use extensively the structures introduced in this Chapter in order to analyze generic transition systems and chemical transition systems. To do so, we derive the necessary concrete and abstract objects that allows one to design an abstraction system on a concrete system. Further, by establishing correspondances between the two aforementioned system it will become clear that the former system can be used to address similar questions at a lower level of representation than in the latter one.

Chapter 4

An Abstract Domain with Non-overlapping Intervals

Objectives

In this Chapter, we detail the classical interval framework of Abstract Interpretation which uses non-overlapping intervals. A major focus is given to the design of a generic abstract transition system, which is derived from a concrete transition system. In particular, a special emphasis is given to the analysis of the obtained non-deterministic abstractions resulting from the abstraction process. After, we conclude with two case studies using this classical framework.

Contents

4.1	The classical interval lattice domain	50
4.2	Abstract states	54
4.3	Abstract transitions	60
4.4	Abstract traces	67
4.5	Collecting the behaviors of a transition system	74
4.5.1	Concrete collecting semantics	74
4.5.2	Non-overlapping abstract collecting semantics	75
4.6	Non-deterministic, effortless abstract transition systems	77
4.6.1	Revisiting the BD model I	77
4.6.2	Revisiting the Resource Competition System I	78
4.7	Conclusion	79

As we have previously mentioned, AI is a technique used in the approximation of mathematical structures by varying levels of granularity. The AI framework owns a collection of abstract domains which aim at abstracting object types with several approximation strategies. For example, an abstract domain that we will focus on pervasively is the interval lattice domain [10] to over-approximate numerical values with an interval representing a range of values. For an in-depth understanding of the tools of AI, the reader is referred to [33] and [11]. For now, a focus is given to building abstract notions for a specific case of an arbitrary transition system abstracted by means of non-overlapping intervals.

Generally, an abstract transition system is generated from a specified abstract semantics. An abstract semantics is composed of a collection of mathematical, abstract objects that permit one to automatically abstract a corresponding concrete semantics. This abstraction represents which is the concrete counterpart to the abstract semantics. Yet, a difficult task remains in the design of an appropriate approximation scheme to highlight certain behaviors of interest. Additionally, the abstract transition system can introduce unwanted fictitious behaviors due to a loss of precision. We will also take the opportunity to discuss how this imprecision arises and how it can be dealt with. Nevertheless, accomplishing this difficult task can lead to several advantages such as a reduction in behavior complexity and a versatile, modular abstract framework which can be refined on command to highlight properties of interest.

4.1 The classical interval lattice domain

The first mathematical object belonging to the abstract semantics in this Chapter are abstract intervals, and the following definition characterizes them.

Definition 4.1 (Abstract intervals). *We consider a family $(\llbracket \underline{q}_p^\#, \bar{q}_p^\# \rrbracket)_{1 \leq p \leq n}$ of n couples of values in $\mathbb{N} \cup \{+\infty\}$ where n is a natural number in \mathbb{N} such that all the following properties are satisfied:*

1. *for every natural number p between 1 and n , $\underline{q}_p^\# < \bar{q}_p^\#$*
2. *for every natural number p between 2 and n , $\bar{q}_{p-1}^\# = \underline{q}_p^\#$;*
3. *$\underline{q}_1^\# = 0$ and $\bar{q}_n^\# = +\infty$.*

We denote this set of non-overlapping intervals $\{\llbracket \underline{q}_p^\#, \bar{q}_p^\# \rrbracket \mid 1 \leq p \leq n\}$ by $\mathcal{D}^\#$.

In Def. 4.1, an abstract interval is an interval set denoted by a lower bound and an upper bound. Each interval represents a range of values which includes the lower bound and excludes the upper bound. In Cond. 1, we ensure intervals are properly formed. In Cond. 2, for two consecutive intervals the upper bound of the first is equal to the lower bound of the second. This is done to ensure that each interval representation do not share any common values, or that intervals do not overlap. In Cond. 3, the lower bound of the first interval $\llbracket \underline{q}_1^\#, \bar{q}_1^\# \rrbracket$ is set to zero ($\underline{q}_1^\# = 0$), while the last interval $\llbracket \underline{q}_n^\#, \bar{q}_n^\# \rrbracket$ has no upper bound ($\bar{q}_n^\# = +\infty$). This is done to ensure that the set of intervals covers each value in \mathbb{N} .

To begin the first of many type abstractions, we would like to compute abstract intervals from concrete values, which is the reason why we derive the non-overlapping value abstraction function using the interval domain in Def. 4.1.

Definition 4.2 (Non-overlapping value abstraction). *The value abstraction function $\beta^{\mathbb{N}} \in \mathbb{N} \rightarrow \mathcal{D}^{\sharp}$ maps a natural number $x \in \mathbb{N}$ to the unique interval $\llbracket \underline{q}_p^{\sharp}, \bar{q}_p^{\sharp} \rrbracket \in \mathcal{D}^{\sharp}$ such that $\underline{q}_p^{\sharp} \leq x < \bar{q}_p^{\sharp}$.*

In Def. 4.2, the domain and co-domain of the function $\beta^{\mathbb{N}}$ are, respectively, the set of natural numbers and the non-overlapping interval domain. We often refer to a natural number as a concrete value. Thus, the value abstraction function will map a concrete value $x \in \mathbb{N}$ to the unique abstract interval $\beta^{\mathbb{N}}(x) \in \mathcal{D}^{\sharp}$ that contains this concrete value. This interval exists since intervals covers all the values and is unique since they do not overlap. We will show how to use the value abstraction function in the following example.

Example 4.1. Let $\mathcal{D}^{\sharp} = \{\llbracket 0, 6 \rrbracket, \llbracket 6, +\infty \rrbracket\}$ be the set of intervals. We would like to use the value abstraction function in Def. 4.2 to abstract the concrete values 2, 3, 7 into intervals in our domain. By applying $\beta^{\mathbb{N}}$ onto each concrete value, we obtain the intervals $\llbracket 0, 6 \rrbracket$, $\llbracket 0, 6 \rrbracket$, and $\llbracket 6, +\infty \rrbracket$.

Consequently, abstracting a concrete value results in an interval that over-approximates this concrete value. As discussed in Sect. 3.3, we commonly refer to the abstract image of a concrete value as a sound approximation since by construct of Def. 4.2 an interval encloses a concrete value.

Now, we will focus on establishing a Galois connection (see Def. 3.14) between the concrete value and abstract interval domains. That is, we want to ensure that each interval (resp. concrete) set has a corresponding concrete (resp. abstract) representation.

The purpose of the following definition is to derive the abstraction function of the value Galois connection, which is responsible for generating sets of intervals from sets of concrete values.

Definition 4.3 (Concrete set abstraction). *The concrete set abstraction function maps each set of concrete values to a set of abstract intervals:*

$$\alpha^{\mathbb{N}} = \begin{cases} \wp(\mathbb{N}) & \rightarrow & \wp(\mathcal{D}^{\sharp}) \\ X & \mapsto & \left\{ \beta^{\mathbb{N}}(x) \in \mathcal{D}^{\sharp} \mid x \in X \right\}. \end{cases}$$

In Def. 4.3, the domain and co-domain of the function $\alpha^{\mathbb{N}}$ are, respectively, the powerset of the set of concrete values and the powerset of the set of intervals. Thus, the concrete set abstraction function will input a set of concrete values X and output the set of intervals $\alpha^{\mathbb{N}}(X)$ by lifting $\beta^{\mathbb{N}}$ onto each concrete value in X . Consequently, the output set is composed of intervals that are sound approximations of each concrete value from the input set. Note, also, that by Def. 3.13, the domain and co-domain are complete lattices. This is particularly useful because it permits us to extract sets with pertinent information such as the join or the meet of

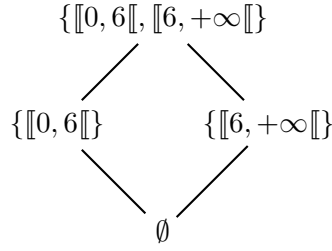


Figure 4.1: The Hasse diagram composed of the sets of intervals obtained by abstracting each set of concrete values in Fig. 3.1 by $\alpha^{\mathbb{N}}$ (see Def. 4.3).

elements belonging to each powerset, which may compose a set of elements carrying a property of interest. This structure will become important in Sect. 4.5 to generate automatically all the abstract (resp. concrete) behaviors for an abstract (resp. a concrete) transition system.

The problem of abstracting a set of concrete values can be specified into collecting the set of abstract intervals such that each value in a concrete set has the property that it is bounded by an interval in the corresponding abstract set. This motivates the following explicit form of the concrete set abstraction function.

Proposition 4.1. *Let $X \in \wp(\mathbb{N})$ be a set of concrete values. Then:*

$$\alpha^{\mathbb{N}}(X) = \left\{ \llbracket \underline{q}_p^{\sharp}, \bar{q}_p^{\sharp} \llbracket \in \mathcal{D}^{\sharp} \mid \exists x \in X : x \in \llbracket \underline{q}_p^{\sharp}, \bar{q}_p^{\sharp} \llbracket \right\}.$$

Proof. Let $X \in \wp(\mathbb{N})$ be a set of natural numbers.

$$\begin{aligned}
 \alpha^{\mathbb{N}}(X) &= \left\{ \beta^{\mathbb{N}}(x) \in \mathcal{D}^{\sharp} \mid x \in X \right\} \\
 &= \left\{ \llbracket \underline{q}_p^{\sharp}, \bar{q}_p^{\sharp} \llbracket \in \mathcal{D}^{\sharp} \mid \exists x \in X : \beta^{\mathbb{N}}(x) = \llbracket \underline{q}_p^{\sharp}, \bar{q}_p^{\sharp} \llbracket \right\} \\
 &= \left\{ \llbracket \underline{q}_p^{\sharp}, \bar{q}_p^{\sharp} \llbracket \in \mathcal{D}^{\sharp} \mid \exists x \in X : \underline{q}_p^{\sharp} \leq x < \bar{q}_p^{\sharp} \right\} \\
 &= \left\{ \llbracket \underline{q}_p^{\sharp}, \bar{q}_p^{\sharp} \llbracket \in \mathcal{D}^{\sharp} \mid \exists x \in X : x \in \llbracket \underline{q}_p^{\sharp}, \bar{q}_p^{\sharp} \llbracket \right\}.
 \end{aligned}$$

□

The expression in Prop. 4.1 provides a more descriptive process as to how the abstraction of concrete values by $\beta^{\mathbb{N}}$ in a concrete set results in an abstract set composed of those abstract intervals each of which bound the concrete values in a prescribed set.

Example 4.2. Let $\mathcal{D}^{\sharp} = \{\llbracket 0, 6\llbracket, \llbracket 6, +\infty\llbracket\}$ be the set of intervals. Let $\mathcal{A} = \{2, 3, 7\}$ be the set of concrete values. We use the value set abstraction function $\alpha^{\mathbb{N}}$ with interval parameters \mathcal{D}^{\sharp} on each concrete set in the powerset $\wp(\mathcal{A})$ (see Fig. 3.1) to obtain the interval sets in the powerset $\wp(\mathcal{A}^{\sharp})$ (see Fig. 4.1).

Similarly to Ex. 3.4, we can compute the join and meet for any element of the powerset $\wp(\mathcal{A}^{\sharp})$. For example, the set of upper bounds for the interval sets $\{\llbracket 0, 6\llbracket\}$ and $\{\llbracket 6, +\infty\llbracket\}$ is

$\mathcal{X}^{\sharp} = \{\{[0, 6], [6, +\infty]\}\}$. Respectively, the join for the two aforementioned interval sets is $\sqcup \mathcal{X}^{\sharp} = \{[0, 6], [6, +\infty]\}$.

Accordingly, the set of lower bounds for the interval sets $\{[0, 6]\}$ and $\{[6, +\infty]\}$ is $\mathcal{X}^{\mathcal{L}\sharp} = \{\emptyset\}$. Respectively, the meet is $\sqcap \mathcal{X}^{\mathcal{L}\sharp} = \emptyset$.

In a similar manner to Def. 4.3, we can derive the concretization function to map sets of intervals and to sets of concrete values. This is the second function required for our value Galois connection.

Definition 4.4 (Abstract set concretization function). *The abstract set concretization function maps each set of intervals to a set of concrete values:*

$$\gamma^{\mathcal{D}\sharp} = \begin{cases} \wp(\mathcal{D}^{\sharp}) & \rightarrow & \wp(\mathbb{N}) \\ Y & \mapsto & \{x \in \mathbb{N} \mid \beta^{\mathbb{N}}(x) \in Y\}. \end{cases}$$

In Def. 4.4, the domain and co-domain of the function $\gamma^{\mathcal{D}\sharp}$ are, respectively, the powerset of the set of intervals and the powerset of the set of concrete values. Here, the domain and co-domain is reversed when compared to the concrete set abstraction function (see Def. 4.3). Thus, the abstract set concretization function will input a set of intervals Y and output the set of concrete values $\gamma^{\mathcal{D}\sharp}(Y)$ composed of those concrete values with the property that its abstraction belongs to the input set. A more mechanical description of this concretization function is detailed in the following proposition: since each interval represents a range of concrete values, then we collect by set union each range of values represented by the abstract intervals in the candidate set to be concretized.

Proposition 4.2. *Let $Y \in \wp(\mathcal{D}^{\sharp})$ be a set of intervals. Then:*

$$\gamma^{\mathcal{D}\sharp}(Y) = \bigcup_{\llbracket \underline{q}_p^{\sharp}, \bar{q}_p^{\sharp} \rrbracket \in Y} \llbracket \underline{q}_p^{\sharp}, \bar{q}_p^{\sharp} \rrbracket.$$

Proof. Let $Y \in \wp(\mathcal{D}^{\sharp})$ be a set of intervals.

$$\begin{aligned} \gamma^{\mathcal{D}\sharp}(Y) &= \{x \in \mathbb{N} \mid \beta^{\mathbb{N}}(x) \in Y\} \\ &= \{x \in \mathbb{N} \mid \exists \llbracket \underline{q}_p^{\sharp}, \bar{q}_p^{\sharp} \rrbracket \in Y : \beta^{\mathbb{N}}(x) = \llbracket \underline{q}_p^{\sharp}, \bar{q}_p^{\sharp} \rrbracket\} \\ &= \{x \in \mathbb{N} \mid \exists \llbracket \underline{q}_p^{\sharp}, \bar{q}_p^{\sharp} \rrbracket \in Y : \underline{q}_p^{\sharp} \leq x < \bar{q}_p^{\sharp}\} \\ &= \{x \in \mathbb{N} \mid \exists \llbracket \underline{q}_p^{\sharp}, \bar{q}_p^{\sharp} \rrbracket \in Y : x \in \llbracket \underline{q}_p^{\sharp}, \bar{q}_p^{\sharp} \rrbracket\} \\ &= \bigcup_{\llbracket \underline{q}_p^{\sharp}, \bar{q}_p^{\sharp} \rrbracket \in Y} \{x \in \mathbb{N} \mid x \in \llbracket \underline{q}_p^{\sharp}, \bar{q}_p^{\sharp} \rrbracket\} \\ &= \bigcup_{\llbracket \underline{q}_p^{\sharp}, \bar{q}_p^{\sharp} \rrbracket \in Y} \llbracket \underline{q}_p^{\sharp}, \bar{q}_p^{\sharp} \rrbracket. \end{aligned}$$

□

Example 4.3. Let $\mathcal{D}^\sharp = \{\llbracket 0, 6 \rrbracket, \llbracket 6, +\infty \rrbracket\}$ be the set of intervals. Let $\mathcal{A}^\sharp = \{\llbracket 0, 6 \rrbracket, \llbracket 6, +\infty \rrbracket\}$ (see Fig. 4.1). We would like to concretize the set \mathcal{A}^\sharp by $\gamma^{\mathcal{D}^\sharp}$ (see Def. 4.4).

Thus, we obtain $\gamma^{\mathcal{D}^\sharp}(\mathcal{A}^\sharp) = \{x \in \mathbb{N} \mid 0 \leq x < +\infty\}$.

Note that the abstraction process $\gamma^{\mathcal{D}^\sharp} \circ \alpha^{\mathbb{N}}$ incurs a loss of precision. For example, consider the original set of concrete values $\mathcal{X} = \{3, 7\}$ (see Fig. 3.1) that underwent the abstraction process using the interval parameters \mathcal{D}^\sharp . Therefore, $\mathcal{X} \subseteq [\gamma^{\mathcal{D}^\sharp} \circ \alpha^{\mathbb{N}}](\mathcal{X})$.

Very often, abstracting concrete values incurs a loss in precision because the abstraction process can introduce fictitious concrete values. This is because the first step in the abstraction process computes a set of intervals from an initial set of concrete values. Since each interval is an over-approximation of a concrete value, then more values are considered for the sake of soundness. Consequently, the second step of the abstraction process is to concretize the interval obtained from the first step. Therefore, concretizing this interval set results in the set union of the the concrete values captured by each interval from step one. We saw this was the case in Ex. 4.3.

Additionally, the abstraction process is further impacted as to whether or not we have at our disposal some concrete or abstract dependencies. For each of the abstract elements derived in Sects. 4.2 - 4.4, we will show how each respective abstract element incurs loss in precision by comparing relational and non-relational abstractions.

4.2 Abstract states

In Sect. 4.1, we focused on deriving our first abstract function that maps values to intervals. In this section, we derive the second abstract function that maps concrete states to abstract states, thus building on the previous abstract element. In order to compute abstract states, we must first derive its concrete counterpart. A concrete state is a function that associates a variable to a quantity.

Definition 4.5 (Concrete state). *Let \mathcal{V} be a set of variables. A concrete state is a function $q : \mathcal{V} \rightarrow \mathbb{N}$ that maps each variable $v \in \mathcal{V}$ onto the concrete value $q(v) \in \mathbb{N}$. The set of concrete states is denoted as \mathcal{Q} .*

The domain and co-domain of the function q are, respectively, the set of variables and the set of concrete values. Accordingly, a concrete state $q \in \mathcal{Q}$ carries information regarding the quantity of each variable in a given set of variables. For example, for a given concrete state $q \in \mathcal{Q}$, one refers to $q(v)$ as the concrete value of the variable $v \in \mathcal{V}$. A similar mathematical structure can be derived for abstract states, with the difference being that a variable is associated to an interval rather than a concrete value.

Definition 4.6 (Abstract state). *An abstract state is a function $q^\sharp : \mathcal{V} \rightarrow \mathcal{D}^\sharp$ and that maps each variable $v \in \mathcal{V}$ to the abstract interval $q^\sharp(v) \in \mathcal{D}^\sharp$. The set of abstract states is denoted as \mathcal{Q}^\sharp .*

The domain and co-domain of the function q^\sharp are, respectively, the set of variables and the set of intervals. Here, an abstract state holds the potential interval values for a given set of variables. Thus, for a given abstract state $q^\sharp \in \mathcal{Q}^\sharp$, $q^\sharp(v)$ is the abstract interval value for the variable $v \in \mathcal{V}$. In order to obtain abstract states, one would have to abstract concrete ones. This will be tasked to the following state abstraction function.

Definition 4.7 (State abstraction). *The state abstraction function maps each concrete state to an abstract state such that:*

$$\beta^{\mathcal{Q}} : \begin{cases} \mathcal{Q} & \rightarrow & \mathcal{Q}^\sharp \\ q & \mapsto & [v \in \mathcal{V} \mapsto \beta^{\mathcal{N}}(q(v)) \in \mathcal{D}^\sharp]. \end{cases}$$

The domain and co-domain of the function $\beta^{\mathcal{Q}}$ are, respectively, the set of concrete states and the set of abstract states. Thus, the state abstraction function maps each concrete state to an abstract state. This is accomplished by lifting the value abstraction function from Def. 4.2 component-wise onto each component quantity in a candidate concrete state to be abstracted. This results in the abstract state such that each of its component is an interval value that over-approximates the respective concrete value from the corresponding concrete state.

Example 4.4. Let $\mathcal{D}^\sharp = \{\llbracket 0, 6\llbracket, \llbracket 6, 12\llbracket, \llbracket 12, +\infty\llbracket\}$ be the set of intervals. Let $\mathcal{V} = \{x, y, w\}$ be the set of variables and $q = [x \mapsto 2, y \mapsto 3, w \mapsto 7] \in \mathcal{Q}$ be the concrete state such that the variables x, y , and w are mapped to the respective concrete values 2, 3, and 7. We would like to abstract the concrete state q using the state abstraction function (see Def. 4.7).

By applying the state abstraction function onto q one obtains the abstract state $q^\sharp = \beta^{\mathcal{Q}}(q) = [x \mapsto \llbracket 0, 6\llbracket, y \mapsto \llbracket 0, 6\llbracket, w \mapsto \llbracket 6, 12\llbracket$ such that the variables x, y , and w are mapped respectively to the intervals $\llbracket 0, 6\llbracket, \llbracket 0, 6\llbracket$, and $\llbracket 6, 12\llbracket$. As a result, the concrete value of each component in the given concrete state is abstracted into the interval that contains it.

Note, also, that each interval in the abstract state q^\sharp is a sound-approximation of its concrete counterpart. Thus, our abstractions propagates the soundness property from the abstract interval object to abstract states.

Similarly to what was done in Sect. 4.1, we can take advantage of our soundness property in order to establish a Galois connection between the concrete and abstract domains. Thus, we derive now the abstract function that maps sets of concrete states to sets of abstract states.

Definition 4.8 (Abstraction of sets of concrete states). *The state set abstraction function maps*

a set of concrete states to the set of abstract states:

$$\alpha^{\mathcal{Q}} = \begin{cases} \wp(\mathcal{Q}) & \rightarrow & \wp(\mathcal{Q}^{\#}) \\ X & \mapsto & \{ \beta^{\mathcal{Q}}(q) \in \mathcal{Q}^{\#} \mid q \in X \} \end{cases}$$

In Def. 4.8, the domain and co-domain of the function $\alpha^{\mathcal{Q}}$ are, respectively, the powerset of the set of concrete states and the powerset of the set of abstract states. Here, the state set abstraction function inputs a set of concrete states and output the set of abstract states with the property that each concrete state when abstracted by $\beta^{\mathcal{Q}}$ (see Def. 4.7) belongs to the input set.

Accordingly, we would like to derive a more specific version of Def. 4.8 with the goal of assessing how abstracting a set of concrete values can impact the accuracy of the abstraction.

Therefore, in the following proposition one can unfold the abstraction of a given set of concrete states by considering for each concrete the component-wise abstraction of a variable value along each coordinate.

Proposition 4.3. *Let $X \in \wp(\mathcal{Q})$ be a set of concrete states. Then:*

$$\alpha^{\mathcal{Q}}(X) = \left\{ q^{\#} \in \mathcal{Q}^{\#} \mid \exists q \in X, \forall v \in \mathcal{V} : q^{\#}(v) \in \alpha^{\mathbb{N}}(\{q(v)\}) \right\}.$$

Proof. Let $X \in \wp(\mathcal{Q})$ be a set of concrete states.

$$\begin{aligned} \alpha^{\mathcal{Q}}(X) &= \left\{ \beta^{\mathcal{Q}}(q) \in \mathcal{Q}^{\#} \mid q \in X \right\} \\ &= \left\{ q^{\#} \in \mathcal{Q}^{\#} \mid \exists q \in X : \beta^{\mathcal{Q}}(q) = q^{\#} \right\} \\ &= \left\{ q^{\#} \in \mathcal{Q}^{\#} \mid \exists q \in X, \forall v \in \mathcal{V} : \beta^{\mathbb{N}}(q(v)) = q^{\#}(v) \right\} \\ &= \left\{ q^{\#} \in \mathcal{Q}^{\#} \mid \exists q \in X, \forall v \in \mathcal{V} : \underline{q}^{\#}(v) \leq q(v) < \bar{q}^{\#}(v) \right\} \\ &= \left\{ q^{\#} \in \mathcal{Q}^{\#} \mid \exists q \in X, \forall v \in \mathcal{V} : q^{\#}(v) \in \alpha^{\mathbb{N}}(\{q(v)\}) \right\}. \end{aligned}$$

□

In the following, we denote as $\alpha^{\mathcal{Q},1}$ the function mapping every set of concrete states $X \subseteq \mathcal{Q}$ into the set of abstract states $\left\{ q^{\#} \in \mathcal{Q}^{\#} \mid \exists q \in X, \forall v \in \mathcal{V} : q^{\#}(v) \in \alpha^{\mathbb{N}}(\{q(v)\}) \right\}$. Note that, by Prop. 4.3, the functions $\alpha^{\mathcal{Q}}$ and $\alpha^{\mathcal{Q},1}$ are the same.

Proposition 4.4. *Let $X \in \wp(\mathcal{Q})$ be a set of concrete states. Then:*

$$\alpha^{\mathcal{Q}}(X) \subseteq \left\{ q^{\#} \in \mathcal{Q}^{\#} \mid \forall v \in \mathcal{V} : q^{\#}(v) \in \alpha^{\mathbb{N}}(\{q(v) \mid q \in X\}) \right\}.$$

Proof.

$$\begin{aligned} \alpha^{\mathcal{Q}}(X) &= \left\{ q^{\#} \in \mathcal{Q}^{\#} \mid \exists q \in X, \forall v \in \mathcal{V} : \underline{q}^{\#}(v) \leq q(v) < \bar{q}^{\#}(v) \right\} \\ &\subseteq \left\{ q^{\#} \in \mathcal{Q}^{\#} \mid \forall v \in \mathcal{V} : q^{\#}(v) \in \alpha^{\mathbb{N}}(\{q(v) \mid q \in X\}) \right\}. \end{aligned}$$

□

In the following, we denote as $\alpha^{\mathcal{Q},2}$ the function mapping every set of concrete states $X \subseteq \mathcal{Q}$ into the set of abstract states $\{q^\sharp \in \mathcal{Q}^\sharp \mid \forall v \in \mathcal{V} : q^\sharp(v) \in \alpha^{\mathbb{N}}(\{q(v) \mid q \in X\})\}$.

In Prop. 4.3, we derived the relational version of the abstraction function from Def. 4.8 that we can use to obtain the best abstraction of a set of concrete states. In this scenario, the expression $\alpha^{\mathcal{Q},1}$ abstracts each component value in a concrete state (belonging to the input set) provided relational information about the value of the other components in the state. On the contrary, the non-relational version of this abstraction function $\alpha^{\mathcal{Q},2}$, given by Prop. 4.4, leads to each component in a concrete state being abstracted independently of one another.

Example 4.5. Let $\mathcal{D}^\sharp = \{\llbracket 0, 6\llbracket, \llbracket 6, 12\llbracket, \llbracket 12, +\infty\llbracket\}$ be the set of intervals and $\mathcal{V} = \{v_1, v_2\}$ be the set of variables. Let $\mathcal{A} = \{q_1, q_2\}$ be the set of concrete states such that $q_1 = [v_1 \mapsto 2, v_2 \mapsto 3]$ and $q_2 = [v_1 \mapsto 8, v_2 \mapsto 7]$. We would like to compare a relational (by Prop. 4.3) and non-relational (by Prop. 4.4) abstraction of the set \mathcal{A} .

Using the relational abstraction expression, we obtain

$$\mathcal{A}_1^\sharp = \alpha^{\mathcal{Q},1}(\mathcal{A}) = \{(\llbracket 0, 6\llbracket, \llbracket 0, 6\llbracket), (\llbracket 6, 12\llbracket, \llbracket 6, 12\llbracket)\}$$

such that each component in each concrete state are abstracted together (see Fig. 4.2, solid crosses).

Otherwise, using the non-relational abstraction expression results in

$$\mathcal{A}_2^\sharp = \alpha^{\mathcal{Q},2}(\mathcal{A}) = \{\{\llbracket 0, 6\llbracket, \llbracket 6, 12\llbracket\} \times \{\llbracket 0, 6\llbracket, \llbracket 6, 12\llbracket\}\}$$

such that each set in \mathcal{A}_2^\sharp contains an interval value for each component. This leads to four potential abstract states: $\{(\llbracket 0, 6\llbracket, \llbracket 0, 6\llbracket), (\llbracket 0, 6\llbracket, \llbracket 6, 12\llbracket), (\llbracket 6, 12\llbracket, \llbracket 0, 6\llbracket), (\llbracket 6, 12\llbracket, \llbracket 6, 12\llbracket)\}$ (see Fig. 4.2, solid and dashed crosses).

Thus, the non-relational set of intervals \mathcal{A}_2^\sharp generates additional abstract states in comparison to the relational set of intervals \mathcal{A}_1^\sharp since we lose dependency from each component in each concrete state in \mathcal{A} , and is the reason why we choose relational abstractions for the state abstraction process. Indeed, the relational set of intervals is more precise than the non-relational set, e.g., $\mathcal{A}_1^\sharp \subseteq \mathcal{A}_2^\sharp$.

Definition 4.9 (Concretization of sets of abstract states). *The abstract state set concretization function maps a set of abstract states to the set of concrete states:*

$$\gamma^{\mathcal{Q}^\sharp} = \begin{cases} \wp(\mathcal{Q}^\sharp) & \rightarrow & \wp(\mathcal{Q}) \\ Y & \mapsto & \{q \in \mathcal{Q} \mid \beta^{\mathcal{Q}}(q) \in Y\} \end{cases}$$

In Def. 4.9, the domain and co-domain of the function $\gamma^{\mathcal{Q}^\sharp}$ are, respectively, the powerset of the set of abstract states and the powerset of the set of concrete states. The state concretization function inputs a set of abstract states and outputs the collection of concrete states with the property their abstraction belongs to the input set. Indeed, this function functions similarly to

the one derived for values (see Def. 4.4), yet now we consider this concretization procedure over abstract states containing several possible interval values for each variable. We can further decompose this concretization process similar to what we did for the state set abstraction function (see Prop. 4.3). Similarly, in the following proposition, the concretization of a set of abstract states is decomposed into the problem of taking the union of sets of concrete states such that each variable value belong to the delimited region conferred by an abstract state.

Proposition 4.5. *Let $Y \in \wp(\mathcal{Q}^\sharp)$ be a set of abstract states. Then:*

$$\gamma^{\mathcal{Q}^\sharp}(Y) = \bigcup_{q^\sharp \in Y} \left(\prod_{v \in \mathcal{V}} \llbracket \underline{q}^\sharp_v, \bar{q}^\sharp_v \rrbracket \right).$$

Proof. Let $Y \in \wp(\mathcal{Q}^\sharp)$ be a set of abstract states.

$$\begin{aligned} \gamma^{\mathcal{Q}^\sharp}(Y) &= \{q \in \mathcal{Q} \mid \beta^{\mathcal{Q}}(q) \in Y\} \\ &= \{q \in \mathcal{Q} \mid \exists q^\sharp \in Y : \beta^{\mathcal{Q}}(q) = q^\sharp\} \\ &= \{q \in \mathcal{Q} \mid \exists q^\sharp \in Y, \forall v \in \mathcal{V} : \beta^{\mathbb{N}}(q(v)) = q^\sharp(v)\} \\ &= \bigcup_{q^\sharp \in Y} \{q \in \mathcal{Q} \mid \forall v \in \mathcal{V} : \beta^{\mathbb{N}}(q(v)) = q^\sharp(v)\} \\ &= \bigcup_{q^\sharp \in Y} \{q \in \mathcal{Q} \mid \forall v \in \mathcal{V} : \underline{q}^\sharp(v) \leq q(v) < \bar{q}^\sharp(v)\} \\ &= \bigcup_{q^\sharp \in Y} \left(\prod_{v \in \mathcal{V}} \llbracket \underline{q}^\sharp(v), \bar{q}^\sharp(v) \rrbracket \right). \end{aligned}$$

□

In the following, we denote as $\gamma^{\mathcal{Q}^\sharp,1}$ the function mapping every set of abstract states $Y \subseteq \mathcal{Q}^\sharp$ into the set of concrete states $\bigcup_{q^\sharp \in Y} \left(\prod_{v \in \mathcal{V}} \llbracket \underline{q}^\sharp(v), \bar{q}^\sharp(v) \rrbracket \right)$. Note that, by Prop. 4.5, the functions $\gamma^{\mathcal{Q}^\sharp}$ and $\gamma^{\mathcal{Q}^\sharp,1}$ are the same.

Proposition 4.6. *Let $Y \in \wp(\mathcal{Q}^\sharp)$ be a set of abstract states. Then:*

$$\gamma^{\mathcal{Q}^\sharp}(Y) \subseteq \bigcup \left\{ q \in \mathcal{Q} \mid \forall v \in \mathcal{V} : q(v) \in \gamma^{\mathcal{D}^\sharp} \left(\{q^\sharp(v) \mid q^\sharp \in Y\} \right) \right\}.$$

Proof. Let $Y \in \wp(\mathcal{Q}^\sharp)$ be a set of abstract states.

$$\begin{aligned} \gamma^{\mathcal{Q}^\sharp}(Y) &= \bigcup_{q^\sharp \in Y} \{q \in \mathcal{Q} \mid \forall v \in \mathcal{V} : \beta^{\mathbb{N}}(q(v)) = q^\sharp(v)\} \\ &= \bigcup_{q^\sharp \in Y} \{q \in \mathcal{Q} \mid \forall v \in \mathcal{V} : q(v) \in \gamma^{\mathcal{D}^\sharp}(\{q^\sharp(v)\})\} \\ &\subseteq \bigcup \left\{ q \in \mathcal{Q} \mid \forall v \in \mathcal{V} : q(v) \in \gamma^{\mathcal{D}^\sharp} \left(\{q^\sharp(v) \mid q^\sharp \in Y\} \right) \right\}. \end{aligned}$$

□

In the following, we denote as $\gamma^{\mathcal{Q}^\#,2}$ the function mapping every set of abstract states $Y \subseteq \mathcal{Q}^\#$ into the set of concrete states $\bigcup \left\{ q \in \mathcal{Q} \mid \forall v \in \mathcal{V} : q(v) \in \gamma^{\mathcal{D}^\#} \left(\left\{ q^\#(v) \mid q^\# \in Y \right\} \right) \right\}$.

Similarly, in Prop. 4.5 is the relational version of the concretization expression from Def. 4.9. In this case, each abstract state in a set concretized by $\gamma^{\mathcal{Q}^\#,1}$ is guided by the interval values of each variable in an abstract state. Otherwise, the non-relational version of this concretization function by $\gamma^{\mathcal{Q}^\#,2}$ from Prop. 4.6 leads to one each interval value per component of each abstract state independently. Note that $\wp(\mathcal{Q}) \xleftrightarrow[\alpha_{\mathcal{Q},2}]{\gamma^{\mathcal{Q}^\#,2}} \wp(\mathcal{Q}^\#)$ forms another Galois connection between the states $\wp(\mathcal{Q})$ and $\wp(\mathcal{Q}^\#)$. It is indeed the non-relational counterpart of the Galois connection $\wp(\mathcal{Q}) \xleftrightarrow[\alpha_{\mathcal{Q},1}]{\gamma^{\mathcal{Q}^\#,1}} \wp(\mathcal{Q}^\#)$.

Example 4.6. Let $\mathcal{D}^\# = \{ \llbracket 0, 6 \llbracket, \llbracket 6, 12 \llbracket, \llbracket 12, +\infty \llbracket \}$ be the set of intervals and $\mathcal{V} = \{v_1, v_2\}$ be the set of variables. Let $\mathcal{A} = \{q_1, q_2\}$ be the set of concrete states such that $q_1 = [v_1 \mapsto 2, v_2 \mapsto 3]$ and $q_2 = [v_1 \mapsto 8, v_2 \mapsto 7]$.

We would like to apply the relational concretization $\gamma^{\mathcal{Q}^\#,1}$ to the relational and to the non-relational sets of states obtained in the result of Ex. 4.5:

$$\mathcal{A}_1^\# = \{q_1^\#, q_2^\#\}$$

such that $q_1^\# = [v_1 \mapsto \llbracket 0, 6 \llbracket, v_2 \mapsto \llbracket 0, 6 \llbracket]$ and $q_2^\# = [v_1 \mapsto \llbracket 6, 12 \llbracket, v_2 \mapsto \llbracket 6, 12 \llbracket]$, and:

$$\mathcal{A}_2^\# = \{q_1^\#, q_2^\#, q_3^\#, q_4^\#\}$$

such that additionally $q_3^\# = [v_1 \mapsto \llbracket 0, 6 \llbracket, v_2 \mapsto \llbracket 6, 12 \llbracket]$ and $q_4^\# = [v_1 \mapsto \llbracket 6, 12 \llbracket, v_2 \mapsto \llbracket 6, 12 \llbracket]$.

Using the relational concretization one obtains

$$\gamma^{\mathcal{Q}^\#,1}(\mathcal{A}_1^\#) = (\llbracket 0, 6 \llbracket \times \llbracket 0, 6 \llbracket) \cup (\llbracket 6, 12 \llbracket \times \llbracket 6, 12 \llbracket).$$

By comparison, the non-relational concretization of the set of abstract states $\mathcal{A}_2^\#$ is:

$$\gamma^{\mathcal{Q}^\#,2}(\mathcal{A}_2^\#) = \llbracket 0, 6 \llbracket \times \llbracket 6, 12 \llbracket.$$

Thus, the concretization of each abstract state leads to the regions of concrete states delimited by each interval value.

Up to this point, we have defined the notion of abstract states, vectors which contain in its components the interval values of variables. After which, we showed it possible to obtain abstract states from concrete states using the state abstraction function (see Def. 4.7). Further, one can propagate this abstract function onto those concrete states belonging to a set, which was done for the concrete state set abstraction function of Def. 4.8. This generates the sets of possible abstract states that can be obtained from concrete ones. In contrast, using the abstract

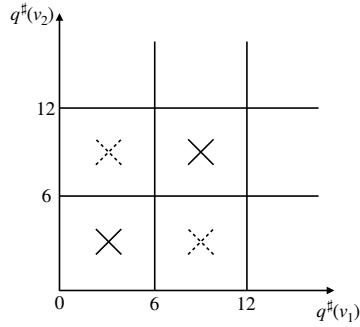


Figure 4.2: The relational versus non-relational state abstraction of the set of concrete states $\mathcal{A} = \{q_1, q_2\}$ such that a concrete state is denoted by a couple $(q(v_1), q(v_2))$ containing the value of the variables v_1 and v_2 . Here, $q_1 = (2, 3)$ and $q_2 = (8, 7)$. Each box represents an abstract state. Using the set of intervals $\mathcal{Q}^\# = \{\llbracket 0, 6\llbracket, \llbracket 6, 12\llbracket, \llbracket 12, +\infty\llbracket\}$, the boxes with solid crosses correspond to the set of abstract states obtained from by a relational abstraction of \mathcal{A} by $\alpha^{\mathcal{Q},1}$. Respectively, the solid and dashed crosses correspond to the set of abstract states obtained by the non-relational abstraction of this same concrete set by $\alpha^{\mathcal{Q},2}$. See Ex. 4.5 for more details regarding this example.

state concretization function allows one to proceed in the reverse direction. That is, to obtain sets of concrete states from abstract ones. Consequently, we were able to establish a Galois connection between the concrete and abstract states domain. Finally, it has become clear how the abstraction process can impact the accuracy of an analysis. For example, we noted that the state abstraction process $\gamma^{\mathcal{Q}^\#} \circ \alpha^{\mathcal{Q}}$ outputs an imprecise set of states where larger regions of values for each variable considered.

Now, we will shift our efforts to deriving in Sect. 4.3 the next abstract object: abstract transitions. This will us to build relations between abstract states.

4.3 Abstract transitions

The third abstract object of this Chapter that we derive are abstract transitions. Abstract transitions are obtained from concrete ones, and a concrete transition is a relation between two concrete states.

We assume that transitions are labelled with a natural number in $\llbracket 1, \sigma\rrbracket$, where $\sigma \in \mathbb{N}$ is the number of kinds of transitions labels.

Definition 4.10 (Concrete transitions). *A concrete transition is an element $(q, l, q') \in \mathcal{Q} \times \llbracket 1, \sigma\rrbracket \times \mathcal{Q}$ such that $l \in \llbracket 1, \sigma\rrbracket$ is a label between two concrete states $q, q' \in \mathcal{Q}$.*

We denote the set of all concrete transitions T .

For Def. 4.10, a concrete transition can also be written as $q \xrightarrow{l} q'$ and indicates how a source (or, predecessor) state q can become a target (or, successor) state q' when an event with label l is applied to q . Thus, the source and target state of a concrete transition are related by an event, which can potentially change the value of a subset of variable values in the source state of this concrete transition.

Example 4.7. Let $\mathcal{V} = \{v_1, v_2\}$ be the set of variables. Let $q, q' \in \mathcal{Q}$ be two concrete states. Consider the following concrete transition event relating the two concrete states q and q' :

$$[v_1 \mapsto q(v_1), v_2 \mapsto q(v_2)] \xrightarrow{1} [v_1 \mapsto q(v_1) + 2, v_2 \mapsto q(v_2) + 3].$$

In this generic concrete transition the event labeled 1 is applied to the source state q to obtain the target state q' , which we can also write as $q \xrightarrow{1} q'$.

In a specific example, consider the initialization of the source state to be $q = [v_1 \mapsto 2, v_2 \mapsto 3]$. By applying the event 1, we obtain:

$$[v_1 \mapsto 2, v_2 \mapsto 3] \xrightarrow{1} [v_1 \mapsto 4, v_2 \mapsto 6].$$

Here, the variable v_1 (resp. v_2) has been increased by two (resp. three) from the source to the target state in the concrete transition produced.

An abstract transition has a similar mathematical structure as a concrete one with the difference being that it expresses a relation between two abstract states.

Definition 4.11 (Abstract transitions). *An abstract transition is an element $(q^\#, l, q^{\#\prime}) \in \mathcal{Q}^\# \times \llbracket 1, \sigma \rrbracket \times \mathcal{Q}^\#$ such that an event $l \in \llbracket 1, \sigma \rrbracket$ is a label between two abstract states $q^\#, q^{\#\prime} \in \mathcal{Q}^\#$.*

The set of all abstract transitions is denoted as $T^\#$.

Similarly to a concrete transition, an abstract transition in Def. 4.11 can be written as $q^\# \xrightarrow{l} q^{\#\prime}$. Namely, an abstract transition is composed of source and target abstract states, which are related by an event label l . In the abstract context, an application of an event l to a source abstract state may change the interval value of a subset of variables and is reflected in the target abstract state. An interpretation of abstract transitions can be the capacity for a variable to traverse between contiguous regions of values. Now the goal is to obtain abstract transitions from concrete ones, which is tasked to the transition abstraction function.

Definition 4.12 (Transition abstraction function). *The concrete transition abstraction function β^T maps each concrete transition to the abstract transition:*

$$\beta^T : \begin{cases} T & \rightarrow & T^\# \\ (q^\#, l, q^{\#\prime}) & \mapsto & (\beta^{\mathcal{Q}}(q), l, \beta^{\mathcal{Q}}(q^{\#\prime})). \end{cases}$$

In Def. 4.12, the domain and co-domain of the function β^T is the set of concrete transitions and the set of abstract transitions. The transition abstraction function inputs a concrete transition and applies to its source and target state the state abstraction function (see Def. 4.7). Consequently, the output is the abstract transition composed of the abstract concrete states belonging to the input concrete transition. Note that in Def. 4.12, the label of the abstracted concrete transition is preserved. The reason for this is because the label will be important when

it is time to recover concrete transitions via concretization of a collection of abstract transitions (see Def. 4.14)

Example 4.8. We consider the abstraction of a pair of concrete transitions. Let $\mathcal{D}^\# = \{\llbracket 0, 6\llbracket, \llbracket 6, 12\llbracket, \llbracket 12, +\infty\llbracket\}$ be the set intervals and $\mathcal{V} = \{v_1, v_2\}$ be the set of variables. We would like to abstract the following concrete transitions:

$$\begin{aligned} [v_1 \mapsto 2, v_2 \mapsto 3] &\xrightarrow{1} [v_1 \mapsto 4, v_2 \mapsto 6]; \\ [v_1 \mapsto 4, v_2 \mapsto 6] &\xrightarrow{1} [v_1 \mapsto 6, v_2 \mapsto 9]. \end{aligned}$$

Using the transition abstraction function in Def. 4.12 together with the interval parameters $\mathcal{Q}^\#$, one obtains:

$$\begin{aligned} [v_1 \mapsto \llbracket 0, 6\llbracket, v_2 \mapsto \llbracket 0, 6\llbracket] &\xrightarrow{1} [v_1 \mapsto \llbracket 0, 6\llbracket, v_2 \mapsto \llbracket 6, 12\llbracket]; \\ [v_1 \mapsto \llbracket 0, 6\llbracket, v_2 \mapsto \llbracket 6, 12\llbracket] &\xrightarrow{1} [v_1 \mapsto \llbracket 6, 12\llbracket, v_2 \mapsto \llbracket 6, 12\llbracket]. \end{aligned}$$

Thus, each related source and target state belonging to the candidate concrete transitions are abstracted, while the event label is preserved.

Accordingly, we can further construct a Galois connection between sets of concrete transitions and sets of abstract transitions by deriving the corresponding abstraction and concretization functions.

Definition 4.13 (Abstraction of sets of concrete transitions). *The transition set abstraction function maps a set of concrete transitions to the set of abstract transitions:*

$$\alpha^T = \begin{cases} \wp(T) & \rightarrow \wp(T^\#) \\ X & \mapsto \left\{ \beta^T(q, l, q') \mid (q, l, q') \in X \right\}. \end{cases}$$

In Def. 4.13, the domain and co-domain of the function α^T are, respectively, the powerset of the set of concrete transitions and the powerset of the set of abstract transitions. The concrete transition set abstraction function inputs a set of concrete transitions and abstracts each transition in this set with the transition abstraction function (see Def. 4.12). Consequently, the output are the set of abstract transitions that can be obtained from the input set. In order to obtain a more explicit version of Def. 4.13, we can use our results obtained in Sect. 4.2 regarding the concrete state set abstraction function (see Prop. 4.3). Accordingly, we abstract sets of concrete transitions using a relational approach that preserves the relation between source and target states of each concrete transition.

Proposition 4.7. *Let $X \in \wp(T)$ be a set of concrete transitions. Then:*

$$\alpha^T(X) = \left\{ (q^\#, l, q^{\#'}) \in T^\# \mid \exists q, q' \in \mathcal{Q} : (q, l, q') \in X \wedge q^\# \in \alpha^{\mathcal{Q}}(\{q\}) \wedge q^{\#'} \in \alpha^{\mathcal{Q}}(\{q'\}) \right\}.$$

Proof. Let $X \in \wp(T)$ be a set of concrete transitions.

$$\begin{aligned}
\alpha^T(X) &= \left\{ \beta^T(q, l, q') \mid (q, l, q') \in X \right\} \\
&= \left\{ (q^\#, l, q^{\#\prime}) \in T^\# \mid \exists (q, l, q') \in X : \beta^T(q, l, q') = (q^\#, l, q^{\#\prime}) \right\} \\
&= \left\{ (q^\#, l, q^{\#\prime}) \in T^\# \mid \exists q, q' \in \mathcal{Q} : (q, l, q') \in X \wedge \beta^{\mathcal{Q}}(q) = q^\# \wedge \beta^{\mathcal{Q}}(q') = q^{\#\prime} \right\} \\
&= \left\{ (q^\#, l, q^{\#\prime}) \in T^\# \mid \exists q, q' \in \mathcal{Q} : (q, l, q') \in X \wedge q^\# \in \alpha^{\mathcal{Q}}(\{q\}) \wedge q^{\#\prime} \in \alpha^{\mathcal{Q}}(\{q'\}) \right\}.
\end{aligned}$$

□

In the following, we denote $\alpha^{T,1}$ the function mapping every set of concrete transitions $X \subseteq T$ into the set of abstract transitions:

$$\left\{ (q^\#, l, q^{\#\prime}) \in T^\# \mid \exists q, q' \in \mathcal{Q} : (q, l, q') \in X \wedge q^\# \in \alpha^{\mathcal{Q}}(\{q\}) \wedge q^{\#\prime} \in \alpha^{\mathcal{Q}}(\{q'\}) \right\}.$$

Note that, by Prop. 4.7, the functions α^T and $\alpha^{T,1}$ are equal.

Proposition 4.8. *Let $X \in \wp(T)$ be a set of concrete transitions. Then:*

$$\alpha^T(X) \subseteq \left\{ (q^\#, l, q^{\#\prime}) \in T^\# \mid q^\# \in \alpha^{\mathcal{Q}}(\{q \mid (q, l, q') \in X\}) \wedge q^{\#\prime} \in \alpha^{\mathcal{Q}}(\{q' \mid (q, l, q') \in X\}) \right\}.$$

Proof. Let $X \in \wp(T)$ be a set of concrete transitions.

$$\begin{aligned}
\alpha^T(X) &= \left\{ (q^\#, l, q^{\#\prime}) \in T^\# \mid \exists q, q' \in \mathcal{Q} : (q, l, q') \in X \wedge \beta^{\mathcal{Q}}(q) = q^\# \wedge \beta^{\mathcal{Q}}(q') = q^{\#\prime} \right\} \\
&\subseteq \left\{ (q^\#, l, q^{\#\prime}) \in T^\# \mid q^\# \in \alpha^{\mathcal{Q}}(\{q \mid (q, l, q') \in X\}) \wedge q^{\#\prime} \in \alpha^{\mathcal{Q}}(\{q' \mid (q, l, q') \in X\}) \right\}.
\end{aligned}$$

□

In the following, we denote $\alpha^{T,2}$ the function mapping every set of concrete transitions $X \subseteq T$ into the set of abstract transitions:

$$\left\{ (q^\#, l, q^{\#\prime}) \in T^\# \mid q^\# \in \alpha^{\mathcal{Q}}(\{q \mid (q, l, q') \in X\}) \wedge q^{\#\prime} \in \alpha^{\mathcal{Q}}(\{q' \mid (q, l, q') \in X\}) \right\}.$$

In Prop. 4.7, we derive the relational version of the abstraction function from Def. 4.13 that we can use to obtain the best abstraction of a set of concrete transitions. Here, the expression $\alpha^{T,1}$ abstracts each concrete transition from an input set by preserving the relation between each respective source and target states. In contrast, the expression $\alpha^{T,2}$, given by Prop. 4.8, loses the relation between each source and target state of each concrete transition belonging to the input set. Thus, one must consider several possible combinations between each source and target states that arise when these components are abstracted independently.

Example 4.9.

Let $\mathcal{D}^\sharp = \{\llbracket 0, 6\llbracket, \llbracket 6, 12\llbracket, \llbracket 12, +\infty\llbracket$ be the set of intervals. Let $\mathcal{V} = \{v_1, v_2\}$ be the set of variables.

We would like to abstract the following set of concrete transitions (see Ex. 4.8):

$$\mathcal{A} = \left\{ \begin{array}{l} [v_1 \mapsto 2, v_2 \mapsto 3] \xrightarrow{1} [v_1 \mapsto 4, v_2 \mapsto 6]; \\ [v_1 \mapsto 4, v_2 \mapsto 6] \xrightarrow{1} [v_1 \mapsto 6, v_2 \mapsto 9] \end{array} \right\}$$

which we can also denote as $\mathcal{A} = \{(2, 3) \xrightarrow{1} (4, 6), (4, 6) \xrightarrow{1} (6, 9)\}$ such that each concrete state, for example q , is expressed as a tuple $(q(v_1), q(v_2))$.

The relational abstraction by $\alpha^{T,1}$ results in the following set of abstract transitions:

$$\mathcal{A}_1^\sharp = \left\{ \begin{array}{l} (\llbracket 0, 6\llbracket, \llbracket 0, 6\llbracket \xrightarrow{1} (\llbracket 0, 6\llbracket, \llbracket 6, 12\llbracket, \\ (\llbracket 0, 6\llbracket, \llbracket 6, 12\llbracket \xrightarrow{1} (\llbracket 6, 12\llbracket, \llbracket 6, 12\llbracket \end{array} \right\}.$$

such that each abstract state, for example q^\sharp , is expressed by a tuple $(q^\sharp(v_1), q^\sharp(v_2))$ (see Fig. 4.3, solid arrows). Indeed, the set \mathcal{A}_1^\sharp is composed of the same abstract transitions obtained in Ex. 4.8.

In contrast, the non-relational abstraction using $\alpha^{T,2}$ results in:

$$\mathcal{A}_2^\sharp = \left\{ \begin{array}{l} (\llbracket 0, 6\llbracket, \llbracket 0, 6\llbracket \xrightarrow{1} (\llbracket 0, 6\llbracket, \llbracket 6, 12\llbracket, \\ (\llbracket 0, 6\llbracket, \llbracket 6, 12\llbracket \xrightarrow{1} (\llbracket 0, 6\llbracket, \llbracket 6, 12\llbracket, \\ (\llbracket 0, 6\llbracket, \llbracket 6, 12\llbracket \xrightarrow{1} (\llbracket 6, 12\llbracket, \llbracket 6, 12\llbracket, \\ (\llbracket 0, 6\llbracket, \llbracket 6, 12\llbracket \xrightarrow{1} (\llbracket 6, 12\llbracket, \llbracket 6, 12\llbracket. \end{array} \right\}$$

In this scenario, in addition to the actual abstract transitions in the set \mathcal{A}_1^\sharp , two fictitious abstract transitions in addition are taken into account in the set \mathcal{A}_2^\sharp (see Fig. 4.3, solid and dashed arrows). This is because the non-relational abstraction loses the dependency between the source and target states of each concrete transition that is abstracted in the initial concrete set. Note also that we use the relational abstraction over these concrete states. An even coarser abstraction could have been obtained by also considering the non-relational state abstraction.

Thus, the non-relational analysis takes into account additional abstract transitions and is more imprecise than its relational counterpart, e.g., $\mathcal{A}_1^\sharp \subset \mathcal{A}_2^\sharp$.

Accordingly, we derive the corresponding concretization function that will permit one to compute sets of concrete transitions from sets of abstract transitions.

Definition 4.14 (Concretization of sets of abstract transitions). *The abstract transition set concretization function maps a set of abstract transitions to the set of concrete transitions such that:*

$$\gamma^{T^\sharp} = \left\{ \begin{array}{ll} \wp(T^\sharp) & \rightarrow \wp(T) \\ Y & \mapsto \{(q, l, q') \in T \mid \beta^T(q, l, q') \in Y\}. \end{array} \right.$$

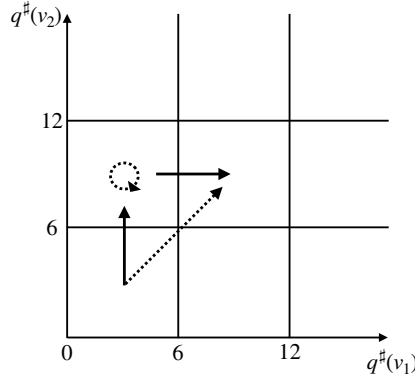


Figure 4.3: The relational versus non-relational abstraction of the set concrete transitions $\mathcal{A} = \{(2, 3) \xrightarrow{1} (4, 6), (4, 6) \xrightarrow{1} (6, 9)\}$ with the interval domain $\mathcal{D}^\# = \{\llbracket 0, 6\llbracket, \llbracket 6, 12\llbracket, \llbracket 12, +\infty\llbracket\}$. Each box is an abstract state. An abstract transition proceeds from a source abstract state (an arrow tail) to a target abstract state (arrow head). The relational abstraction of the set \mathcal{A} by $\alpha^{T,1}$ results in the set of abstract transitions composed of the solid abstract transitions. However, the non-relational of the same concrete set by $\alpha^{T,2}$ outputs the set of abstract transitions composed of both the solid and dashed abstract transitions. For more details see Ex. 4.9.

In Def. 4.14, the domain and co-domain of the function $\gamma^{T^\#}$ are, respectively, the powerset of the set of abstract transitions and the powerset of the set of concrete transitions. The abstract transition set concretization function inputs a set of abstract transitions and outputs the set of concrete transitions with the property that the abstraction of each concrete transition belongs to the input set. A specific version of this function is similar in structure to the explicit concrete transition set abstraction function (see Prop. 4.7). Namely, given a set of abstract transitions, the relational abstract transition set concretization function retains the relation between each source and target abstract states of each abstract transition.

Proposition 4.9. *Let $Y \in \wp(T^\#)$ be a set of abstract transitions. Then:*

$$\gamma^{T^\#}(Y) = \bigcup_{q^\#, q^{\#'} \in \mathcal{Q}^\#} \left\{ (q, l, q') \in T \mid (q^\#, l, q^{\#'}) \in Y \wedge q \in \gamma^{\mathcal{Q}^\#}(\{q^\#\}) \wedge q' \in \gamma^{\mathcal{Q}^\#}(\{q^{\#'}\}) \right\}.$$

Proof. Let $Y \in \wp(T^\#)$ be a set of abstract transitions.

$$\begin{aligned} \gamma^{T^\#}(Y) &= \{(q, l, q') \in T \mid \beta^T(q, l, q') \in Y\} \\ &= \{(q, l, q') \in T \mid \exists (q^\#, l, q^{\#'}) \in Y : \beta^T(q, l, q') = (q^\#, l, q^{\#'})\} \\ &= \{(q, l, q') \in T \mid \exists q^\#, q^{\#'} \in \mathcal{Q}^\# : (q^\#, l, q^{\#'}) \in Y \wedge \beta^{\mathcal{Q}^\#}(q) = q^\# \wedge \beta^{\mathcal{Q}^\#}(q') = q^{\#'}\} \\ &= \{(q, l, q') \in T \mid \exists q^\#, q^{\#'} \in \mathcal{Q}^\# : (q^\#, l, q^{\#'}) \in Y \wedge q \in \gamma^{\mathcal{Q}^\#}(\{q^\#\}) \wedge q' \in \gamma^{\mathcal{Q}^\#}(\{q^{\#'}\})\} \\ &= \bigcup_{q^\#, q^{\#'} \in \mathcal{Q}^\#} \left\{ (q, l, q') \in T \mid (q^\#, l, q^{\#'}) \in Y \wedge q \in \gamma^{\mathcal{Q}^\#}(\{q^\#\}) \wedge q' \in \gamma^{\mathcal{Q}^\#}(\{q^{\#'}\}) \right\}. \end{aligned}$$

□

In the following, we denote $\gamma^{T^\sharp,1}$ the function mapping every set of abstract transitions $Y \subseteq T^\sharp$ into the set of concrete transitions

$$\gamma^{T^\sharp}(Y) = \bigcup_{q^\sharp, q^{\sharp'} \in \mathcal{Q}^\sharp} \left\{ (q, l, q') \in T \mid (q^\sharp, l, q^{\sharp'}) \in Y \wedge q \in \gamma^{\mathcal{Q}^\sharp}(\{q^\sharp\}) \wedge q' \in \gamma^{\mathcal{Q}^\sharp}(\{q^{\sharp'}\}) \right\}.$$

Note that, by Prop. 4.9, the functions γ^{T^\sharp} and $\gamma^{T^\sharp,1}$ are the same.

Proposition 4.10. *Let $Y \in \wp(T^\sharp)$ be a set of abstract transitions. Then:*

$$\gamma^{T^\sharp}(Y) \subseteq \bigcup \left\{ (q, l, q') \in T \mid q \in \gamma^{\mathcal{Q}^\sharp}(\{q^\sharp \mid (q^\sharp, l, q^{\sharp'}) \in Y\}) \wedge q' \in \gamma^{\mathcal{Q}^\sharp}(\{q^{\sharp'} \mid (q^\sharp, l, q^{\sharp'}) \in Y\}) \right\}.$$

Proof. Let $Y \in \wp(T^\sharp)$ be a set of abstract transitions.

$$\begin{aligned} \gamma^{T^\sharp}(Y) &= \left\{ (q, l, q') \in T \mid \exists q^\sharp, q^{\sharp'} \in \mathcal{Q}^\sharp : (q^\sharp, l, q^{\sharp'}) \in Y \wedge q \in \gamma^{\mathcal{Q}^\sharp}(\{q^\sharp\}) \wedge q' \in \gamma^{\mathcal{Q}^\sharp}(\{q^{\sharp'}\}) \right\} \\ &\subseteq \bigcup \left\{ (q, l, q') \in T \mid q \in \gamma^{\mathcal{Q}^\sharp}(\{q^\sharp \mid (q^\sharp, l, q^{\sharp'}) \in Y\}) \wedge q' \in \gamma^{\mathcal{Q}^\sharp}(\{q^{\sharp'} \mid (q^\sharp, l, q^{\sharp'}) \in Y\}) \right\}. \end{aligned}$$

□

In the following, we denote as $\gamma^{T^\sharp,2}$ the function mapping every set of abstract transitions $Y \subseteq T^\sharp$ into the set of concrete transitions:

$$\bigcup \left\{ (q, l, q') \in T \mid q \in \gamma^{\mathcal{Q}^\sharp}(\{q^\sharp \mid (q^\sharp, l, q^{\sharp'}) \in Y\}) \wedge q' \in \gamma^{\mathcal{Q}^\sharp}(\{q^{\sharp'} \mid (q^\sharp, l, q^{\sharp'}) \in Y\}) \right\}.$$

Similarly, in Prop. 4.9 is the relational version of the concretization expression from Def. 4.14. In this case, each abstract transition in a set concretized by $\gamma^{T^\sharp,1}$ is guided by the relation between the source and target states of each abstract transition. Otherwise, the non-relational version of this concretization function by $\gamma^{T^\sharp,2}$ from Prop. 4.6 loses this relation and one must concretize each source and target states independently to one another.

Note that $\wp(T) \xrightleftharpoons[\alpha^{T,2}]{\gamma^{T^\sharp,2}} \wp(T^\sharp)$ forms another Galois connection between the transitions $\wp(T)$ and $\wp(T^\sharp)$. It is indeed the non-relational counterpart of the Galois connection $\wp(T) \xrightleftharpoons[\alpha^{T,1}]{\gamma^{T^\sharp,1}} \wp(T^\sharp)$.

We omit an example because the abstraction process $\gamma^{T^\sharp,1} \circ \alpha^{T,1}$ and $\gamma^{T^\sharp,2} \circ \alpha^{T,2}$ for a set of concrete transitions as the output can rapidly become exhaustive. Yet, it is important to note that each abstraction process builds on the relational state abstraction process. As a result, we are able to track the degree of imprecision that may be propagated to the abstraction of a concrete element that is slightly more complex.

In this Section, we introduced our third abstract object, abstract transitions; and, derived the transition abstraction function (see Def. 4.12) to obtain abstract transitions from concrete transitions. After which, we established a Galois connection between the abstract transition and concrete transition domains. We use this correspondance between the two domains to examine how the abstraction process for a set of concrete transitions can introduce fictitious abstract transitions in our analysis. We will see in Chapter 5 alternative ways that an abstract analysis

of a concrete system can introduce different types fictitious behaviors. Accordingly, we move on to the final abstract element of this Chapter: abstract traces.

4.4 Abstract traces

The final abstract element that we consider are the abstract traces, those of which can be obtained by concrete traces. Informally, a concrete trace is a sequence of transitions that contain trajectorial information regarding a combinatorial number of transition events that may occur this sequence. Formally, a concrete trace is the following.

Definition 4.15 (Concrete traces). *Let $k \in \mathbb{N}$. A concrete trace of size k is defined as an element $(q'_0, (q_i, l_i, q'_i)_{1 \leq i \leq k}) \in \mathcal{Q} \times T^k$ such that $q'_i = q_{i+1}$ for any integer i such that $0 \leq i < k$. The set of all concrete traces is denoted as \mathcal{T} .*

In Def. 4.15, a concrete trace is defined as a couple that contains in the first component an initial state $q'_0 \in \mathcal{Q}$ and in the second component a sequence of concrete transitions $((q_i, l_i, q'_i)_{1 \leq i \leq k}) \in T^k$. The former component contains information about the initial amount of each variable available, while the latter component correspond to those transitions which can be obtained from the initial state and by the respective transitions. Each trace has the property that for two consecutive transitions the target state of the first transition is equal the source state of the second. For two consecutive transitions t_{i-1}, t_i , we often represent this aforementioned property as $pre(t_i) = post(t_{i-1})$ denoting the equality between the source (left) and target (right) states of each respective transition in a concrete trace. Further, for a concrete trace τ , we denote the first state in a sequence as $first(\tau)$ and it is equal to the initial state of a trace ($first(\tau) = q'_0$). The final state q'_k of a concrete trace is denoted $final(\tau)$. A trace can also be elongated from its terminal state by a concrete transition $(q_{k+1}, l_{k+1}, q'_{k+1}) \in T$ using the junction operator \frown : $\tau \frown (q_{k+1}, l_{k+1}, q'_{k+1})$ provided that $final(\tau) = q_{k+1}$.

Example 4.10. Let $\mathcal{V} = \{v_1, v_2\}$ be the set of variables. Let $q'_0 = (2, 0)$ be the initial concrete state denoting the tuple $(q'_0(v_1), q'_0(v_2))$. Consider the following concrete transition events:

$$\begin{aligned} [v_1 \mapsto q(v_1), v_2 \mapsto q(v_2)] &\xrightarrow{1} [v_1 \mapsto q(v_1) + 2, v_2 \mapsto q(v_2) + 3] \\ [v_1 \mapsto q(v_1), v_2 \mapsto q(v_2)] &\xrightarrow{2} [v_1 \mapsto q(v_1) + 1, v_2 \mapsto q(v_2) - 1]. \end{aligned}$$

We would like to examine a concrete trace that can be obtained from the initial state q'_0 via event 1 or 2. Thus, one sampled trace that can be obtained is:

$$\tau_1 = (2, 0) \xrightarrow{1} (4, 3) \xrightarrow{1} (6, 6) \xrightarrow{1} (8, 9).$$

Alternatively, a second possible trace is:

$$\tau_2 = (2, 0) \xrightarrow{1} (4, 3) \xrightarrow{2} (5, 2) \xrightarrow{1} (7, 5) \xrightarrow{2} (8, 4).$$

Now, we define an abstract trace which inherits a similar structure to concrete traces.

Definition 4.16 (Abstract traces). *Let $k \in \mathbb{N}$. An abstract trace of size k is an element $(q_0^\#, (q_i^\#, l_i, q_i^{\#'})_{1 \leq i \leq k}) \in \mathcal{Q}^\# \times T^{\#k}$ such that $q_i^\# = q_{i-1}^{\#'}$.*

The set of all abstract traces is denoted as $\mathcal{T}^\#$.

With the same due respect, an abstract trace in Def. 4.16 is a couple that contains in the first component an initial abstract state $q_0^\# \in \mathcal{Q}^\#$ and in the second component a sequence of abstract transitions $(q_i^\#, l_i, q_i^{\#'})_{1 \leq i \leq k} \in T^{\#k}$. The former contains information about the initial interval value of each variable, while the latter component are the abstract transitions that can be obtained from the initial abstract state. Each abstract trace has the property that for two consecutive abstract transitions the target abstract state of the first transition is equal the source abstract state of the second. For two consecutive transitions $t_{i-1}^\#, t_i^\#$, we often represent this aforementioned property as $pre(t_i^\#) = post(t_{i-1}^\#)$ denoting the equality between the source (left) and target (right) abstract states of each respective transition in an abstract trace. We can extract certain information from abstract traces and perform operations similar to concrete traces. Namely, given an abstract trace $\tau^\#$, the first abstract state is $first(\tau^\#) = q_0^\#$, while the terminal one is $final(\tau^\#) = q_k^{\#'}$. Finally, an abstract trace can also be elongated by an abstract transition $(q_{k+1}^\#, l_{k+1}, q_{k+1}^{\#'}) \in T^\#$: $\tau^\# \frown (q_{k+1}^\#, l_{k+1}, q_{k+1}^{\#'})$ provided that $final(\tau^\#) = q_{k+1}^\#$.

Abstract traces are computed from concrete ones. Indeed, an abstract trace is an over-approximation of a concrete trace. In order to obtain such abstract traces, we abstract in each concrete trace its initial concrete state with the abstract state function in Def. 4.7 and each of its concrete transitions with the abstract transition function derived in Def. 4.12. This operation is encapsulated in the following definition.

Definition 4.17 (Trace abstraction function). *The trace abstraction function maps a concrete trace to the abstract trace:*

$$\beta^{\mathcal{T}} : \left\{ \begin{array}{ccc} \mathcal{T} & \rightarrow & \mathcal{T}^\# \\ (q'_0, (q_i, l_i, q'_i)_{1 \leq i \leq k}) & \mapsto & \left(\beta^{\mathcal{Q}}(q'_0), \left(\beta^{\mathcal{T}}(q_i, l_i, q'_i) \right)_{1 \leq i \leq k} \right). \end{array} \right.$$

In Def. 4.17, the domain and co-domain of the function $\beta^{\mathcal{T}}$ are, respectively, the set of concrete traces and the set of abstract traces. Thus, the trace abstraction function inputs a concrete trace and outputs the abstract trace obtained by lifting point-wise the appropriate abstract functions to each concrete element in the input concrete trace.

Example 4.11.

Let $\mathcal{D}^\# = \{[0, 6], [6, 12], [12, +\infty]\}$ be the set of intervals. We would like to abstract the concrete traces obtained in Ex. 4.10:

$$\begin{aligned} \tau_1 &= (2, 0) \xrightarrow{1} (4, 3) \xrightarrow{1} (6, 6) \xrightarrow{1} (8, 9); \\ \tau_2 &= (2, 0) \xrightarrow{1} (4, 3) \xrightarrow{2} (5, 2) \xrightarrow{1} (7, 5) \xrightarrow{2} (8, 4). \end{aligned}$$

Using the trace abstraction function in Def. 4.17 with interval parameters \mathcal{D}^\sharp , we obtain for the first trace abstraction:

$$\beta^{\mathcal{T}}(\tau_1) = (\llbracket 0, 6 \rrbracket, \llbracket 0, 6 \rrbracket) \xrightarrow{1} (\llbracket 0, 6 \rrbracket, \llbracket 0, 6 \rrbracket) \xrightarrow{1} (\llbracket 6, 12 \rrbracket, \llbracket 6, 12 \rrbracket) \xrightarrow{1} (\llbracket 6, 12 \rrbracket, \llbracket 6, 12 \rrbracket)$$

which we denote by τ_1^\sharp , and for the second:

$$\beta^{\mathcal{T}}(\tau_2) = (\llbracket 0, 6 \rrbracket, \llbracket 0, 6 \rrbracket) \xrightarrow{1} (\llbracket 0, 6 \rrbracket, \llbracket 0, 6 \rrbracket) \xrightarrow{2} (\llbracket 0, 6 \rrbracket, \llbracket 0, 6 \rrbracket) \xrightarrow{1} (\llbracket 6, 12 \rrbracket, \llbracket 0, 6 \rrbracket) \xrightarrow{2} (\llbracket 6, 12 \rrbracket, \llbracket 0, 6 \rrbracket),$$

which is denoted as τ_2^\sharp .

Note that in the abstract trace τ_1^\sharp each variable eventually exit their initial interval, whereas in τ_2^\sharp only the variable v_1 experiences an interval change.

We can lift our notion of abstracting concrete traces to sets which contain them. This is the first step in establishing a Galois connection between the concrete and abstract trace domains. Thus, the abstraction of sets of concrete traces is defined in the following definition.

Definition 4.18 (Abstraction of sets of concrete traces). *The trace set abstraction function maps a set of concrete traces to the set of abstract traces:*

$$\alpha^{\mathcal{T}} = \begin{cases} \wp(\mathcal{T}) & \rightarrow & \wp(\mathcal{T}^\sharp) \\ X & \mapsto & \{ \beta^{\mathcal{T}}(\tau) \in \mathcal{T}^\sharp \mid \tau \in X \}. \end{cases}$$

In Def. 4.18, the domain and co-domain of the function $\alpha^{\mathcal{T}}$ are, respectively, the powerset of the set of concrete traces and the powerset of the set of abstract traces. The concrete trace set abstraction function inputs a set of concrete traces and outputs the set of abstract traces that can be obtained by abstracting each concrete trace in the input set by the trace abstraction function (see Def. 4.17).

A more specific version of Def. 4.18 can be achieved accordingly. Given a set of concrete traces, the trace set abstraction function pieces together the initial abstract states that can be obtained from abstracting each initial concrete state (e.g., the trace's first component); and, the abstract transitions that can be obtained from abstracting the concrete transitions reachable from the initial concrete state (e.g., the trace's second component). Indeed, those abstract traces obtained must also satisfy the structural property of abstract traces (see Def. 4.16). This line of reasoning is specified in the following proposition.

Proposition 4.11. *Let $X \in \wp(\mathcal{T})$ be a set of concrete traces. Then:*

$$\alpha^{\mathcal{T}}(X) = \left\{ (q_0^\sharp, (t_i^\sharp)_{1 \leq i \leq k}) \in \mathcal{T}^\sharp \mid (q_0', (t_i)_{1 \leq i \leq k}) \in X : q_0^\sharp \in \alpha^{\mathcal{Q}}(\{q_0'\}) \wedge \forall i \in \llbracket 1, k \rrbracket, t_i^\sharp \in \alpha^{\mathcal{T}}(\{t_i\}) \right\}.$$

Proof. Let $X \in \wp(\mathcal{T})$ be a set of concrete traces.

$$\begin{aligned} \alpha^{\mathcal{T}}(X) &= \{ \beta^{\mathcal{T}}(q'_0, (t_i)_{1 \leq i \leq k}) \in \mathcal{T}^{\sharp} \mid (q'_0, (t_i)_{1 \leq i \leq k}) \in X \} \\ &= \left\{ (q_0^{\sharp'}, (t_i^{\sharp})_{1 \leq i \leq k}) \in \mathcal{T}^{\sharp} \mid \exists (q'_0, (t_i)_{1 \leq i \leq k}) \in X : \beta^{\mathcal{T}}(q'_0, (t_i)_{1 \leq i \leq k}) = (q_0^{\sharp'}, (t_i^{\sharp})_{1 \leq i \leq k}) \right\} \\ &= \left\{ (q_0^{\sharp'}, (t_i^{\sharp})_{1 \leq i \leq k}) \in \mathcal{T}^{\sharp} \mid (q'_0, (t_i)_{1 \leq i \leq k}) \in X : q_0^{\sharp'} \in \alpha^{\mathcal{Q}}(\{q'_0\}) \wedge \forall i \in \llbracket 1, k \rrbracket, t_i^{\sharp} \in \alpha^{\mathcal{T}}(\{t_i\}) \right\}. \end{aligned}$$

□

In the following, we denote as $\alpha^{\mathcal{T},1}$ the function mapping every set of concrete traces $X \subseteq \mathcal{T}$ into the set of abstract traces

$$\left\{ (q_0^{\sharp'}, (t_i^{\sharp})_{1 \leq i \leq k}) \in \mathcal{T}^{\sharp} \mid (q'_0, (t_i)_{1 \leq i \leq k}) \in X : q_0^{\sharp'} \in \alpha^{\mathcal{Q}}(\{q'_0\}) \wedge \forall i \in \llbracket 1, k \rrbracket, t_i^{\sharp} \in \alpha^{\mathcal{T}}(\{t_i\}) \right\}.$$

Note that, by Prop. 4.11, the functions $\alpha^{\mathcal{T}}$ and $\alpha^{\mathcal{T},1}$ are the same.

Proposition 4.12. *Let $X \in \wp(\mathcal{T})$ be a set of concrete traces. Then:*

$$\alpha^{\mathcal{T}}(X) \subseteq \left\{ (q_0^{\sharp'}, (t_i^{\sharp})_{1 \leq i \leq k}) \in \mathcal{T}^{\sharp} \mid \begin{aligned} &q_0^{\sharp'} \in \alpha^{\mathcal{Q}}(\{q'_0 \mid (q'_0, (t_i)_{1 \leq i \leq k}) \in X\}) \\ &\wedge \forall i \in \llbracket 1, k \rrbracket, t_i^{\sharp} \in \alpha^{\mathcal{T}}(\{t_i \mid (q'_0, (t_i)_{1 \leq i \leq k}) \in X\}) \wedge \text{pre}(t_i^{\sharp}) = \text{post}(t_{i-1}^{\sharp}) \end{aligned} \right\}.$$

Proof. Let $X \in \wp(\mathcal{T})$ be a set of concrete traces.

$$\begin{aligned} \alpha^{\mathcal{T}}(X) &= \left\{ (q_0^{\sharp'}, (t_i^{\sharp})_{1 \leq i \leq k}) \in \mathcal{T}^{\sharp} \mid \exists (q'_0, (t_i)_{1 \leq i \leq k}) \in X : \beta^{\mathcal{T}}(q'_0, (t_i)_{1 \leq i \leq k}) = (q_0^{\sharp'}, (t_i^{\sharp})_{1 \leq i \leq k}) \right\} \\ &= \left\{ (q_0^{\sharp'}, (t_i^{\sharp})_{1 \leq i \leq k}) \in \mathcal{T}^{\sharp} \mid (q'_0, (t_i)_{1 \leq i \leq k}) \in X : \beta^{\mathcal{Q}}(q'_0) = q_0^{\sharp'} \wedge \forall i \in \llbracket 1, k \rrbracket : \beta^{\mathcal{T}}(t_i) = t_i^{\sharp} \right\} \\ &= \left\{ (q_0^{\sharp'}, (t_i^{\sharp})_{1 \leq i \leq k}) \in \mathcal{T}^{\sharp} \mid (q'_0, (t_i)_{1 \leq i \leq k}) \in X : \beta^{\mathcal{Q}}(q'_0) = q_0^{\sharp'} \wedge \forall i \in \llbracket 1, k \rrbracket : \right. \\ &\quad \left. \beta^{\mathcal{T}}(t_i) = t_i^{\sharp} \wedge \text{pre}(t_i^{\sharp}) = \text{post}(t_{i-1}^{\sharp}) \right\} \\ &\subseteq \left\{ (q_0^{\sharp'}, (t_i^{\sharp})_{1 \leq i \leq k}) \in \mathcal{T}^{\sharp} \mid \begin{aligned} &q_0^{\sharp'} \in \alpha^{\mathcal{Q}}(\{q'_0 \mid (q'_0, (t_i)_{1 \leq i \leq k}) \in X\}) \\ &\wedge \forall i \in \llbracket 1, k \rrbracket, t_i^{\sharp} \in \alpha^{\mathcal{T}}(\{t_i \mid (q'_0, (t_i)_{1 \leq i \leq k}) \in X\}) \wedge \text{pre}(t_i^{\sharp}) = \text{post}(t_{i-1}^{\sharp}) \end{aligned} \right\}. \end{aligned}$$

□

In the same spirit, we denote as $\alpha^{\mathcal{T},2}$ the function mapping every set of concrete traces $X \subseteq \mathcal{T}$ into the set of abstract traces

$$\left\{ (q_0^{\sharp'}, (t_i^{\sharp})_{1 \leq i \leq k}) \in \mathcal{T}^{\sharp} \mid \begin{aligned} &q_0^{\sharp'} \in \alpha^{\mathcal{Q}}(\{q'_0 \mid (q'_0, (t_i)_{1 \leq i \leq k}) \in X\}) \\ &\wedge \forall i \in \llbracket 1, k \rrbracket, t_i^{\sharp} \in \alpha^{\mathcal{T}}(\{t_i \mid (q'_0, (t_i)_{1 \leq i \leq k}) \in X\}) \wedge \text{pre}(t_i^{\sharp}) = \text{post}(t_{i-1}^{\sharp}) \end{aligned} \right\}.$$

In Prop. 4.11, we derive the relational version of the abstraction function from Def. 4.18 that we can use to obtain the best abstraction of a set of concrete traces. In this scenario, the expression $\alpha^{\mathcal{T},1}$ abstracts each component in a concrete trace (see Def. 4.15) by preserving the relation between a concrete trace's initial state and those states reachable by continuing the initial state by at least some transitions. In contrast, in the non-relational version of this

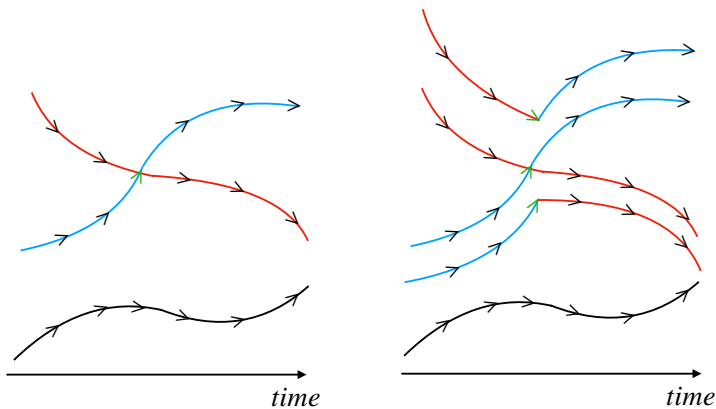


Figure 4.4: Abstraction of a set composed of three concrete traces. Each scenario represents a set of abstract traces obtained by either $\alpha^{\mathcal{T},1}$, or $\alpha^{\mathcal{T},2}$. Note that time denotes the index of an abstract transition along each abstract trace. Each arrow head denotes a target (resp. source) abstract state belonging to two consecutive abstract transitions along each trace (see Def. 4.16). The relational abstraction by $\alpha^{\mathcal{T},1}$ of the arbitrary concrete set leads to the set of abstract traces composed of three abstract traces (left). The non-relational abstraction by $\alpha^{\mathcal{T},2}$ results in the set of abstract traces with five abstract traces (right). For more details, see the explanation in the text.

abstraction function $\alpha^{\mathcal{T},2}$, given by Prop. 4.12, leads to each component in a concrete trace being abstracted independently.

An example of each abstraction expression on a set composed of three concrete traces is given in Fig. 4.4. On the left, we obtain the best abstraction of the aforementioned set. That is, a set composed of three abstract traces. On the right, a situation arises when the abstraction must deal with intersecting concrete traces (e.g., concrete states that are shared across concrete traces) and as a result two additional abstract traces are obtained due to a crossing between intersecting traces (and occurs at the green arrow heads), and which arises from the fact that we preserve structural integrity. Thus, the additional two fictitious traces are a hybrid of the actual abstract traces.

The second part of establishing a Galois connection between abstract and concrete traces is the respective concretization function. The concretization function for abstract traces will output for each abstract trace the corresponding concrete traces. A generic version of the trace set concretization function follows.

Definition 4.19 (Concretization of sets of abstract traces). *The abstract trace set concretization function maps a set of abstract traces to the set of concrete traces:*

$$\gamma^{\mathcal{T}^\#} : \begin{cases} \wp(\mathcal{T}^\#) & \rightarrow & \wp(\mathcal{T}) \\ Y & \mapsto & \{\tau \in \mathcal{T} \mid \beta^{\mathcal{T}}(\tau) \in Y\}. \end{cases}$$

In Def. 4.19, the domain and co-domain of the function $\gamma^{\mathcal{T}^\#}$ are, respectively, the powerset of the set of abstract traces and the powerset of the set of concrete traces. The abstract trace set concretization function takes as an input a set of abstract traces and outputs the set of concrete

traces with the property that abstracting each concrete trace by the trace abstraction function (see Def. 4.17) is an element of the input set. The relational version of Def. 4.19 is given in the following proposition and follows a similar procedure to the relational trace set abstraction function of Prop. 4.18.

Proposition 4.13. *Let $Y \in \wp(\mathcal{T}^\#)$ be a set of abstract traces. Then:*

$$\begin{aligned} \gamma^{\mathcal{T}^\#}(Y) = & \bigcup_{(q_0^\#, (t_i^\#)_{1 \leq i \leq k}) \in Y} \left\{ (q'_0, (t_i)_{1 \leq i \leq k}) \in \mathcal{T} \mid q'_0 \in \gamma^{\mathcal{Q}^\#} \left(\{q_0^\#\} \right) \right. \\ & \left. \wedge \forall i \in \llbracket 1, k \rrbracket, t_i \in \gamma^{\mathcal{T}^\#} \left(\{t_i^\#\} \right) \wedge \text{pre}(t_i) = \text{post}(t_{i-1}) \right\}. \end{aligned}$$

Proof. Let $Y \in \wp(\mathcal{T}^\#)$ be a set of abstract traces.

$$\begin{aligned} \gamma^{\mathcal{T}^\#}(Y) &= \{ (q'_0, (t_i)_{1 \leq i \leq k}) \in \mathcal{T} \mid \beta^{\mathcal{T}}(q'_0, (t_i)_{1 \leq i \leq k}) \in Y \} \\ &= \left\{ (q'_0, (t_i)_{1 \leq i \leq k}) \in \mathcal{T} \mid \exists (q_0^\#, (t_i^\#)_{1 \leq i \leq k}) \in Y : \beta^{\mathcal{T}}(q'_0, (t_i)_{1 \leq i \leq k}) = (q_0^\#, (t_i^\#)_{1 \leq i \leq k}) \right\} \\ &= \left\{ (q'_0, (t_i)_{1 \leq i \leq k}) \in \mathcal{T} \mid \exists (q_0^\#, (t_i^\#)_{1 \leq i \leq k}) \in Y : \beta^{\mathcal{Q}}(q'_0) = q_0^\# \right. \\ & \quad \left. \wedge \forall i \in \llbracket 1, k \rrbracket, \beta^{\mathcal{T}}(t_i) = t_i^\# \wedge \text{pre}(t_i) = \text{post}(t_{i-1}) \right\} \\ &= \left\{ (q'_0, (t_i)_{1 \leq i \leq k}) \in \mathcal{T} \mid \exists (q_0^\#, (t_i^\#)_{1 \leq i \leq k}) \in Y : q'_0 \in \gamma^{\mathcal{Q}^\#} \left(\{q_0^\#\} \right) \right. \\ & \quad \left. \wedge \forall i \in \llbracket 1, k \rrbracket, t_i \in \gamma^{\mathcal{T}^\#} \left(\{t_i^\#\} \right) \wedge \text{pre}(t_i) = \text{post}(t_{i-1}) \right\} \\ &= \bigcup_{(q_0^\#, (t_i^\#)_{1 \leq i \leq k}) \in Y} \left\{ (q'_0, (t_i)_{1 \leq i \leq k}) \in \mathcal{T} \mid q'_0 \in \gamma^{\mathcal{Q}^\#} \left(\{q_0^\#\} \right) \right. \\ & \quad \left. \wedge \forall i \in \llbracket 1, k \rrbracket, t_i \in \gamma^{\mathcal{T}^\#} \left(\{t_i^\#\} \right) \wedge \text{pre}(t_i) = \text{post}(t_{i-1}) \right\}. \end{aligned}$$

□

In the following, we denote $\gamma^{\mathcal{T}^\#,1}$ the function mapping every set of abstract traces $Y \subseteq \mathcal{T}^\#$ into the set of concrete traces:

$$\begin{aligned} \bigcup_{(q_0^\#, (t_i^\#)_{1 \leq i \leq k}) \in Y} \left\{ (q'_0, (t_i)_{1 \leq i \leq k}) \in \mathcal{T} \mid q'_0 \in \gamma^{\mathcal{Q}^\#} \left(\{q_0^\#\} \right) \right. \\ \left. \wedge \forall i \in \llbracket 1, k \rrbracket, t_i \in \gamma^{\mathcal{T}^\#} \left(\{t_i^\#\} \right) \wedge \text{pre}(t_i) = \text{post}(t_{i-1}) \right\}. \end{aligned}$$

Note that, by Prop. 4.13, the functions $\gamma^{\mathcal{T}^\#}$ and $\gamma^{\mathcal{T}^\#,1}$ are the same.

Proposition 4.14. *Let $Y \in \wp(\mathcal{T}^\#)$ be a set of abstract traces. Then:*

$$\begin{aligned} \gamma^{\mathcal{T}^\#}(Y) \subseteq & \left\{ (q'_0, (t_i)_{1 \leq i \leq k}) \in \mathcal{T} \mid q'_0 \in \gamma^{\mathcal{Q}^\#} \left(\left\{ q_0^\# \mid (q_0^\#, (t_i^\#)_{1 \leq i \leq k}) \in Y \right\} \right) \wedge \forall i \in \llbracket 1, k \rrbracket, \right. \\ & \left. t_i \in \gamma^{\mathcal{T}^\#} \left(\left\{ t_i^\# \mid (q_0^\#, (t_i^\#)_{1 \leq i \leq k}) \in Y \right\} \right) \wedge \text{pre}(t_i) = \text{post}(t_{i-1}) \right\}. \end{aligned}$$

Proof. Let $Y \in \wp(\mathcal{T}^\sharp)$ be a set of abstract traces.

$$\begin{aligned} \gamma^{\mathcal{T}^\sharp}(Y) &= \left\{ (q'_0, (t_i)_{1 \leq i \leq k}) \in \mathcal{T} \mid \exists (q_0^{\sharp'}, (t_i^\sharp)_{1 \leq i \leq k}) \in Y : q'_0 \in \gamma^{\mathcal{Q}^\sharp}(\{q_0^{\sharp'}\}) \right. \\ &\quad \left. \wedge \forall i \in \llbracket 1, k \rrbracket, t_i \in \gamma^{\mathcal{T}^\sharp}(\{t_i^\sharp\}) \wedge \text{pre}(t_i) = \text{post}(t_{i-1}) \right\} \\ &\subseteq \left\{ (q'_0, (t_i)_{1 \leq i \leq k}) \in \mathcal{T} \mid q'_0 \in \gamma^{\mathcal{Q}^\sharp}(\{q_0^{\sharp'} \mid (q_0^{\sharp'}, (t_i^\sharp)_{1 \leq i \leq k}) \in Y\}) \wedge \forall i \in \llbracket 1, k \rrbracket, \right. \\ &\quad \left. t_i \in \gamma^{\mathcal{T}^\sharp}(\{t_i^\sharp \mid (q_0^{\sharp'}, (t_i^\sharp)_{1 \leq i \leq k}) \in Y\}) \wedge \text{pre}(t_i) = \text{post}(t_{i-1}) \right\}. \end{aligned}$$

□

Accordingly, we denote $\gamma^{\mathcal{T}^\sharp, 2}$ the function mapping every set of abstract traces $Y \subseteq \mathcal{T}^\sharp$ into the set of concrete traces

$$\left\{ (q'_0, (t_i)_{1 \leq i \leq k}) \in \mathcal{T} \mid q'_0 \in \gamma^{\mathcal{Q}^\sharp}(\{q_0^{\sharp'} \mid (q_0^{\sharp'}, (t_i^\sharp)_{1 \leq i \leq k}) \in Y\}) \wedge \forall i \in \llbracket 1, k \rrbracket, \right. \\ \left. t_i \in \gamma^{\mathcal{T}^\sharp}(\{t_i^\sharp \mid (q_0^{\sharp'}, (t_i^\sharp)_{1 \leq i \leq k}) \in Y\}) \wedge \text{pre}(t_i) = \text{post}(t_{i-1}) \right\}.$$

In Prop. 4.13 is the relational version of the concretization expression from Def. 4.19. In this case, each abstract trace in a set concretized by $\gamma^{\mathcal{T}^\sharp, 1}$ is guided by preserving the relation between each component of an abstract trace and preserving structural integrity. More specifically, the relational concretization of a set of abstract traces results in the union of sets of concrete traces such that each trace in the collective set is pieced together by an initial concrete state from a region described by an initial abstract state (e.g., the first component of an abstract trace); and, by the sequences of concrete transitions delimited by abstract transitions between regions (e.g., the second component of a trace). Otherwise, the non-relational version of this concretization function by $\gamma^{\mathcal{T}^\sharp, 2}$ from Prop. 4.14 loses this relation and one must concretize each abstract component independently.

Note that $\wp(\mathcal{T}) \xleftrightarrow[\alpha^{\mathcal{T}, 2}]{\gamma^{\mathcal{T}^\sharp, 2}} \wp(\mathcal{T}^\sharp)$ forms another Galois connection between the traces $\wp(\mathcal{T})$ and $\wp(\mathcal{T}^\sharp)$. It is indeed the non-relational counterpart of the Galois connection $\wp(\mathcal{T}) \xleftrightarrow[\alpha^{\mathcal{T}, 1}]{\gamma^{\mathcal{T}^\sharp, 1}} \wp(\mathcal{T}^\sharp)$.

Consequently, one may consider two potential abstraction processes for a given set of concrete traces. In each function, note that we use the relational abstraction process for transitions as derived in Sect. 4.3.

This concludes the construction of the final abstract object considered in this Chapter. In this Section, we introduced concrete and abstract traces, and showed how one can obtain the latter from the former. Further, we established a Galois connection between the set of concrete and abstract traces, and detailed how either a relational or non-relational analysis can impact the accuracy of the abstraction process. In the upcoming section, we derive a collection of operators that we will use to automatically generate the complete set of concrete behaviors, along with its respective abstract counterpart.

4.5 Collecting the behaviors of a transition system

In Sects. 4.1 - 4.4, we formally derived all the necessary abstract elements we require to abstract a concrete transition system using the classical interval lattice domain. These results are steered by a collection of abstraction functions that are used to obtain abstract elements from concrete ones by ensuring that each abstract element is an over-approximation to its concrete counterpart. Further, we have encountered different cases when the abstraction process can trigger varying levels of accuracy which is generally associated to the amount of concrete information available.

In this Section we shift gears and focus our efforts to deriving a collection of operators to collect the behaviors of a concrete and an abstract transition system. As we will see soon, operators have several useful mathematical properties that can permit the user to infer properties of interest, such as least fixed point computations (see Def. 3.11). In total, two operators will be derived. One operator for defining the set of concrete behaviors, and the second one for defining the set of abstract behaviors.

4.5.1 Concrete collecting semantics

Firstly we introduce $\mathcal{Q}_0 \subseteq \mathcal{Q}$ a set of potential initial states and $T' \subseteq T$ a set of potential transitions. We are interested in computing, the collecting semantics $\mathcal{T}_{\mathcal{Q}_0, T'}$, which is the set of traces starting from a state in \mathcal{Q}_0 and using transitions in T' only. We denote as $\mathcal{T}_{\mathcal{Q}_0}$ the set of traces $\{(q, ()) \mid q \in \mathcal{Q}_0\}$, that is the set of the traces made of a single state and no transition. Then we introduce the first, concrete elongation operator.

Definition 4.20 (Concrete trace elongation). *The concrete trace elongation operator is defined as:*

$$\mathbb{F} : \begin{cases} \wp(\mathcal{T}) & \rightarrow \wp(\mathcal{T}) \\ X & \mapsto X \cup \{\tau \frown (q, l, q') \in \mathcal{T} \mid \tau \in X \wedge (q, l, q') \in T' \wedge q = \text{final}(\tau)\}. \end{cases}$$

In Def. 4.20, the domain and co-domain of the operator \mathbb{F} is the powerset of the set of concrete traces. The concrete trace elongation operator inputs a set of concrete traces and outputs the set of elongated concrete traces such that each trace in the input set has been extended on its terminal end by a concrete transition in T' . This can be interpreted as an iterative process, and could potentially proceed without termination. Additionally, each trace could be extended by several transitions, or not. Consequently, after each iteration, we take the set union with the input set to filter repeated traces and collect the new reachable traces. When it is not possible to elongate any concrete trace in a set any longer, then we can say that the set of all reachable traces has been reached. Indeed, it is usually expensive to compute the set of all reachable concrete traces, and one may prefer to observe only a subset of such traces. For now, we will focus on characterizing two expressions that will result in the set of all concrete traces. In Chapter 5 we will discuss how one can highlight useful information regarding the collection of chemical behaviors of a reaction network.

We would like to detail two mathematical results for generating the set of all concrete traces from Def. 4.20. The first of these is known as Tarski's fixpoint theorem [35] and the second is Kleene's fixpoint iteration [20].

The first strategy by Tarski is relatively appealing because it is a mathematical characterization on the computation of the set of concrete traces and relies on two main ingredients on the concrete trace elongation operator: an operator which is monotonic and which operates over a complete lattice. A careful examination of Def. 4.20 reveals that the function is monotonic, since the bigger the set in argument is, the bigger the set in result is. Secondly, \mathbb{F} is an operator with a powerset domain and co-domain, therefore making it a complete lattice by Def. 3.13. As a result, we can use Tarski's result to compute the meet (see Def. 3.12) of the set of the traces that are invariant by \mathbb{F} and that contains the initial traces, that of which is equal to the set of concrete traces (see Def. 4.15):

$$\mathcal{T}_{\mathcal{Q}_0, T'} = \bigcap \{X \in \wp(\mathcal{T}) \mid \mathbb{F}(X) \subseteq X \wedge \mathcal{T}_{\mathcal{Q}_0} \subseteq X\}. \quad (4.1)$$

Often, the result in Eq. 4.1 is referred to as the concrete collecting semantics. The concrete collecting semantics of a concrete transition system describes all the behaviors that can be obtained. Since each behavior in $\mathcal{T}_{\mathcal{Q}_0, T'}$ is a concrete trace, then it is possible to design formal procedure to extract properties shared among a subset of behaviors that may highlight a feature of interest.

The second, alternative, strategy by Kleene to compute the concrete collecting semantics is of an algorithmic nature and corresponds to iterating the operator \mathbb{F} until the reachable set of concrete traces is achieved. In practice, it consists in passing to the limit of the iterates. In this context, the set of traces can be expressed as:

$$\mathcal{T}_{\mathcal{Q}_0, T'} = \bigcup_{n \in \mathbb{N}} \{\mathbb{F}^n(\mathcal{T}_{\mathcal{Q}_0})\}. \quad (4.2)$$

In Eq. 4.2, one computes at each iteration n the set of reachable traces and filters after each round to collect subsequent iterations. Whether one uses Eq. 4.1 or 4.2 to compute the reachable set of concrete traces by Def. 4.20, these results naturally describe the same set of objects: the set of all concrete traces for a concrete transition system. However, the procedure in Eq. 4.2 is more costly since it requires the user to build an iterative algorithm to generate the set $\mathcal{T}_{\mathcal{Q}_0, T'}$ and to pass to the limit, whereas Eq. 4.1 is a mathematical characterization.

In Sect. 4.5.2, we will follow a similar procedure to derive the abstract collecting semantics responsible in generating the collection of abstract behaviors from a abstract transition system.

4.5.2 Non-overlapping abstract collecting semantics

The second operator of this Section corresponds to the abstract elongation operator.

Definition 4.21 (Abstract trace elongation). *The abstract trace elongation operator is defined as:*

$$\mathbb{F}^\sharp = \begin{cases} \wp(\mathcal{T}^\sharp) & \rightarrow & \wp(\mathcal{T}^\sharp) \\ Y & \mapsto & \alpha^{\mathcal{T}}(\mathbb{F}(\gamma^{\mathcal{T}^\sharp}(Y))). \end{cases}$$

In Def. 4.21, the domain and co-domain of the operator \mathbb{F}^\sharp is the powerset of the set of abstract traces. The abstract trace elongation operator inputs a set of abstract traces and

outputs the set of elongated abstract traces obtained by (potentially) extending each abstract trace of the input set along its terminal by a reachable abstract transition. It does so accordingly. First, a set of abstract traces is concretized by the abstract trace set concretization function (see Def. 4.19), and second each concrete trace is elongated by the concrete trace elongation operator (see Def. 4.20). After which, the set containing the elongated traces is fed into the concrete trace set abstraction function (see Def. 4.18). Consequently, the abstract trace elongation operator is naturally an over-approximation of the set of elongated traces that one can achieve from Def. 4.20. A more abstract version of Def. 4.21 is enclosed in the following proposition.

Proposition 4.15. *Let $Y \in \wp(\mathcal{T}^\sharp)$ be a set of abstract traces. Then:*

$$\mathbb{F}^\sharp(Y) \subseteq Y \cup \left\{ \tau^\sharp \frown (q^\sharp, l, q^{\sharp'}) \mid \tau^\sharp \in Y \wedge (q^\sharp, l, q^{\sharp'}) \in \alpha^T(\{T'\}) \wedge \text{final}(\tau^\sharp) = q^\sharp \right\}.$$

Proof. Let $Y \in \wp(\mathcal{T}^\sharp)$.

$$\begin{aligned} \mathbb{F}^\sharp(Y) &= \alpha^T(\mathbb{F}(\gamma^{\mathcal{T}^\sharp}(Y))) \\ &= \alpha^T \left(\gamma^{\mathcal{T}^\sharp}(Y) \cup \left\{ \tau \frown (q, l, q') \in \mathcal{T} \mid \tau \in \gamma^{\mathcal{T}^\sharp}(Y) \wedge (q, l, q') \in T' \wedge \text{final}(\tau) = q \right\} \right) \\ &= \alpha^T \left(\gamma^{\mathcal{T}^\sharp}(Y) \right) \cup \alpha^T \left(\left\{ \tau \frown (q, l, q') \in \mathcal{T} \mid \tau \in \gamma^{\mathcal{T}^\sharp}(Y) \wedge (q, l, q') \in T' \wedge \text{final}(\tau) = q \right\} \right) \\ &= \alpha^T \left(\gamma^{\mathcal{T}^\sharp}(Y) \right) \cup \left\{ \beta^T(\tau \frown (q, l, q')) \in \mathcal{T}^\sharp \mid \tau \in \gamma^{\mathcal{T}^\sharp}(Y) \wedge (q, l, q') \in T' \wedge \text{final}(\tau) = q \right\} \\ &= \alpha^T \left(\gamma^{\mathcal{T}^\sharp}(Y) \right) \cup \left\{ \beta^T(\tau) \frown \beta^T(q, l, q') \in \mathcal{T}^\sharp \mid \tau \in \gamma^{\mathcal{T}^\sharp}(Y) \wedge (q, l, q') \in T' \wedge \text{final}(\tau) = q \right\} \\ &= \alpha^T \left(\gamma^{\mathcal{T}^\sharp}(Y) \right) \cup \left\{ \tau^\sharp \frown (q^\sharp, l, q^{\sharp'}) \in \mathcal{T}^\sharp \mid \exists \tau \in \gamma^{\mathcal{T}^\sharp}(Y), \exists (q, l, q') \in T' : \right. \\ &\quad \left. \beta^T(\tau) = \tau^\sharp \wedge \beta^T(q, l, q') = (q^\sharp, l, q^{\sharp'}) \wedge \text{final}(\tau^\sharp) = q^\sharp \right\} \\ &= \alpha^T \left(\gamma^{\mathcal{T}^\sharp}(Y) \right) \cup \left\{ \tau^\sharp \frown (q^\sharp, l, q^{\sharp'}) \in \mathcal{T}^\sharp \mid \exists \tau \in \gamma^{\mathcal{T}^\sharp}(Y), \exists (q, l, q') \in T' : \right. \\ &\quad \left. \tau^\sharp \in \alpha^T(\{\tau\}) \wedge (q^\sharp, l, q^{\sharp'}) \in \alpha^T(\{(q, l, q')\}) \wedge \text{final}(\tau^\sharp) = q^\sharp \right\}. \\ &= \alpha^T \left(\gamma^{\mathcal{T}^\sharp}(Y) \right) \cup \left\{ \tau^\sharp \frown (q^\sharp, l, q^{\sharp'}) \in \mathcal{T}^\sharp \mid \tau^\sharp \in \alpha^T \left(\gamma^{\mathcal{T}^\sharp}(Y) \right) \wedge (q^\sharp, l, q^{\sharp'}) \in \alpha^T(\{T'\}) \wedge \text{final}(\tau^\sharp) = q^\sharp \right\} \\ &\subseteq Y \cup \left\{ \tau^\sharp \frown (q^\sharp, l, q^{\sharp'}) \in \mathcal{T}^\sharp \mid \tau^\sharp \in Y \wedge (q^\sharp, l, q^{\sharp'}) \in \alpha^T(\{T'\}) \wedge \text{final}(\tau^\sharp) = q^\sharp \right\}. \end{aligned}$$

□

In the following, we denote $\mathbb{F}^{\sharp,1}$ the function mapping every set of abstract traces $Y \subseteq \mathcal{T}^\sharp$ into the set of abstract traces

$$Y \cup \left\{ \tau^\sharp \frown (q^\sharp, l, q^{\sharp'}) \in \mathcal{T}^\sharp \mid \tau^\sharp \in Y \wedge (q^\sharp, l, q^{\sharp'}) \in \alpha^T(\{T'\}) \wedge \text{final}(\tau^\sharp) = q^\sharp \right\}.$$

Note that, by Prop. 4.15, the function $\mathbb{F}^{\sharp,1}$ is an over-approximation of the function \mathbb{F}^\sharp .

In a similar manner to the concrete trace elongation operator, we can derive fixpoint results for the abstract trace elongation operator since it inherits the appropriate functional properties such as monotonicity, operating over powersets and continuity. Accordingly, Tarski's results leads to:

$$\mathcal{T}_{\mathcal{Q}_0, T'}^\sharp = \bigcap \left\{ Y \in \wp(\mathcal{T}^\sharp) \mid \mathbb{F}^{\sharp,1}(Y) \subseteq Y \wedge \alpha^T(\mathcal{T}_{\mathcal{Q}_0}) \subseteq Y \right\}. \quad (4.3)$$

In turn, Kleene’s iterative scheme for sets of abstract traces is:

$$\mathcal{T}_{\mathcal{Q}_0, T'}^\# = \bigcup_{n \in \mathbb{N}} \left\{ \left[\mathbb{F}^{\#, 1} \right]^n (\alpha^{\mathcal{T}}(\mathcal{T}_{\mathcal{Q}_0})) \right\}. \quad (4.4)$$

Thus, Eqs. 4.3 - 4.4 equips one with two constructs that can be used to compute the set of abstract behaviors $\mathcal{T}_{\mathcal{Q}_0, T'}^\#$ for an abstract transition system. We refer to such expressions as the abstract collecting semantics. The abstract collecting semantics is an over-approximation of the concrete collecting semantics, that is to say that: $\mathcal{T}_{\mathcal{R}} \subseteq \bar{\gamma}(\mathcal{T}_{\mathcal{R}}^\#)$.

Thus, one can, for example, tune the degree of granularity offered by the intervals characterized in Def. 4.1. This is because the intervals, corresponding to a region of concrete values, are propagated throughout our framework since each of the abstract objects we derived in Sects. 4.2 - 4.4 are built on top of the interval domain. Thus, albeit affecting the accuracy of an analysis by abstraction, one can use interval coarse-graining to pinpoint general trends that may be observed in the concrete collecting semantics, while achieving this at a lower level of representation.

Up to now, we have derived the concrete and abstract collecting semantics that are used to generate, respectively, the collection of all concrete and abstract behaviors. Importantly, we use two celebrated results to do so, namely Tarski’s and Kleene’s fixpoint characterization schemes. Naturally the abstract collecting semantics is an over-approximation of the concrete collecting semantics, and the former incorporates the trace Galois connection we derived in Sect. 4.4 in order to achieve this.

In the following Section, we will highlight a non-deterministic abstraction of two concrete transition systems from the case studies given in Chapter 2. The goal is to show how the abstraction process can be implemented on two stochastic system using the classical interval domain.

4.6 Non-deterministic, effortless abstract transition systems

In this Section, we examine the non-deterministic abstraction of the BD and resource competition reaction systems investigated in Chapter 2 with the absence of a minimal effort system. A non-deterministic abstraction corresponds to an induced abstract transition system that forgets about the underlying trend of its respective concrete transition system. Our goal is to observe how such non-deterministic abstractions can induce fictitious behaviors arising from our classical abstraction procedures derived in this Chapter.

4.6.1 Revisiting the BD model I

In Fig. 4.5 we display the non-deterministic version of Fig. 2.4. That is, we lift the classical framework derived in this Chapter in order to generate an abstract transition system using the classical interval domain: those which compose intervals that do not overlap. Accordingly, in our diagram each rectangular box represents an interval corresponding to the respective ranges of values for the molecule A . Underlying each rectangle are the concrete behaviors that can emerge from the BD model, and under the same set of assumptions made in Sect. 2.2. We

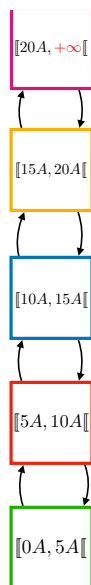


Figure 4.5: The non-deterministic version of Fig. 2.4 of a BD macro-transition system using non-overlapping interval parameters.

note that the abstract transition system in Fig. 4.5 is induced by sampling various concrete behaviors from this reaction network, and performing a relational abstraction of this set of chemical behaviors with the concrete trace set abstraction function (see Def. 4.18) with interval parameters $\mathcal{D}^\sharp = \{[0, 5[, [5, 10[, [10, 15[, [15, 20[, [20, +\infty[\}$. We notice that at this level of abstraction, since intervals do not overlap, the coarse-graining may consider oscillations near the interval borders. This, in retrospect, may add fictitious behaviors into our analysis of this abstract transition system. Additionally, we no longer retain probabilities from the concrete transition system, thus have no information regarding the general trend of the underlying BD dynamics.

4.6.2 Revisiting the Resource Competition System I

In Fig. 4.6 is the non-deterministic version of Fig. 2.8. Each box depicts an abstract state and arises from the product of the respective interval set along each coordinate. Underlying each abstract state are the concrete behaviors that can emerge from the resource competition model, and under the same set of assumptions made in Sect. 2.3. Respectively, this abstract transition system is induced by sampling the concrete behaviors of the underlying reaction network, and abstracting by relational means the set containing these behaviors by the concrete trace set abstraction function (see Def. 4.18) using the interval parameters $\mathcal{D}^\sharp = \{[0, 6[, [6, 12[, [12, 18[, [18, +\infty[\}$. At this level of abstraction, we observe additional abstract transitions between abstract states in the non-deterministic abstraction transition system in Fig. 4.6. This leads to more potential fictitious behaviors. Additionally, since we have no information regarding probabilities then one further cannot deduce the distinction between the competing reaction within their stable, respective regions as observed in Fig. 2.8.

Consequently, using our classical framework derived in this Chapter leads us to consider that

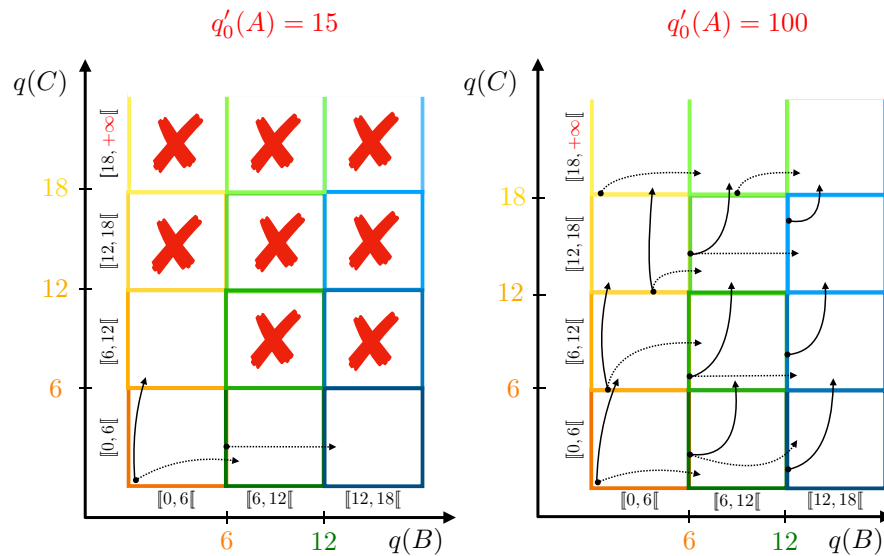


Figure 4.6: The non-deterministic version of Fig. 2.8 of a resource competition macro-transition system using non-overlapping interval parameters.

one must take into account fictitious behaviors that may arise from a stochastic system, such as reaction networks.

4.7 Conclusion

In this Chapter, significant efforts were placed on highlighting the abstraction process for various object types from a concrete system. This was done in order to emphasize how the abstraction process may impact the analysis of a system of interest. Importantly, it should become clear at this point how one builds an abstract transition system, which consists of a hierarchy of abstractions from the simplest abstract object in our classical framework: intervals. More specifically, at each layer of abstraction we derive the respective abstract function that allows one to traverse from a concrete object to an abstract object (see Fig. 4.7).

Additionally, these abstract functions underlie each of the Galois connections that we have established. Consequently, we have direct correspondances between each concrete and abstract domains, which permits us to coarse-grain a concrete transition system with varying degrees of granularity, while remaining sound. As such, this leads to the classical framework detailed in this Chapter and its goals, which is the ability to select, design, and build an abstract framework which can be tailored to highlight some particular features of interest.

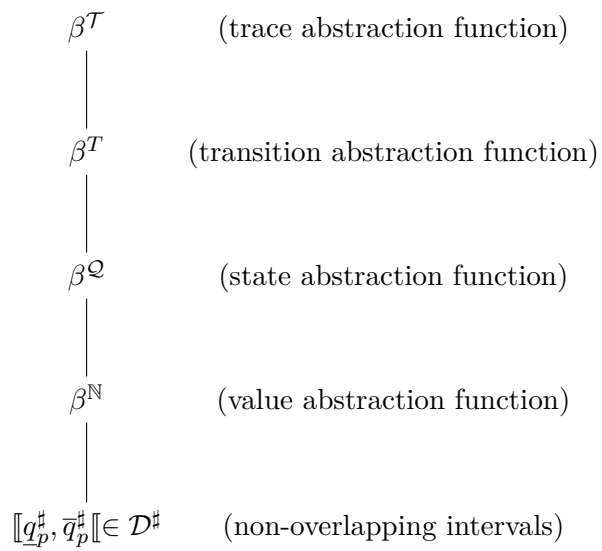


Figure 4.7: A hierarchy composed of the classical abstract interval domain (bottom) and each concrete element abstraction function derived in Chapter 4.

Chapter 5

An Abstract Domain with Overlapping Intervals

Objectives

In this Chapter, we extend the interval framework derived in Chapter 4 to include overlapping intervals. Additionally, we design a coarse-graining framework to abstract the quantities of chemical species using reference intervals. After which, we construct a hierarchy of abstraction processes to automatically coarse-grain a concrete transition system for reaction networks. Finally, we instantiate the derived framework on two case studies.

Contents

5.1	Characterizing overlapping intervals	85
5.2	Abstractions of quantities of chemical components using reference intervals	92
5.3	Chemical state abstraction using reference macro-states	103
5.4	Macro-transitions between regions of chemical states	109
5.5	Characterizing behaviors for a macro-transition system	114
5.6	Approximation of the collecting semantics	120
5.7	Non-deterministic macro-transition systems with minimal effort	122
	5.7.1 Revisiting the BD model II	122
	5.7.2 Revisiting the Resource Competition System II	123
5.8	Conclusion	125

In Chapter 4, we derived the abstract semantics for an abstract transition system using the classical interval lattice domain. The goal was to highlight the abstraction process and show how this process can lead to varying degrees of fictitious behaviors, such as in the case of relational and non-relational abstractions. We instantiated the classical framework on two case studies, the BD model and the competition system, which revealed that our abstractions were not sufficient in pinpointing the correct dynamics of the underlying reaction networks. This is because the behaviors we consider are stochastic, and those behaviors abstracted near interval borders may lead us to consider additional fictitious behaviors. Namely, the structural abstractions we consider in the classical framework permit an ease-of-access between regions of states to behavioral oscillations near regional boundaries. Thus, these subsets of behaviors may not necessarily reflect the general, dynamical trend of a system. Consequently, we consider additional abstract behaviors at the sake of remaining sound. To overcome this obstacle, in this Chapter we derive a more general abstract semantics which extends the classical framework to include overlapping intervals. Additionally, this new abstract semantics is parameterized by the stochastic reaction networks of Chapter 2. It is also important to note that our abstract semantics for reaction networks abstracts away information about probabilities. This is because the probability measure of a fictitious behavior is too high. Thus, in Chapter 6 we refine the abstract semantics of this Chapter with the probabilistic information computed from an original reaction network.

We chose to extend the abstraction process to include overlapping intervals to introduce a minimal effort system to chemical behaviors in order to restrict their ability to traverse between regions of concrete states (see Fig. 5.1). Traversing from one region to another requires a chemical behavior’s ability to pass through and exit via a bound of a region shared by two overlapping intervals: there is a buffer region represented by the area contained between the upper bound (exclusive) of the first interval and the lower bound (inclusive) of the second interval. Thus, coarse-graining a chemical transition system generates an abstract transition system denoting transitions between overlapping (hyper-)regions of quantities of chemical species. From hereon, a coarse-grained chemical transition system is referred to as a macro-transition system. Here, the macro-transition system for a given chemical reaction network over-approximates the collection of all the chemical behaviors; and, in this scenario we also consider overlapping intervals for each quantity of a chemical component. Thus, our abstractions seek to limit oscillations near interval borders and thus reduce the number of fictitious macro-behaviors that can be generated during the abstraction process. In retrospect, we expect that our minimal effort system can assist in pinpointing the correct dynamics of the underlying reaction system of interest. In addition to generating more precise abstractions, a second goal seeks to compare the models derived from an instantiation of our framework to its corresponding logical model. In Chapter 1 and 2 we discussed about logical models, their derivation and their applications, in addition to the type of biological conclusions that can be made from them. In Chapter 2, we observed that even on simple case studies, we were not able to highlight intrinsic properties of the given stochastic reaction networks. Consequently, in order to better understand the underlying structure of logical models, we use the macro-transition system derived from our

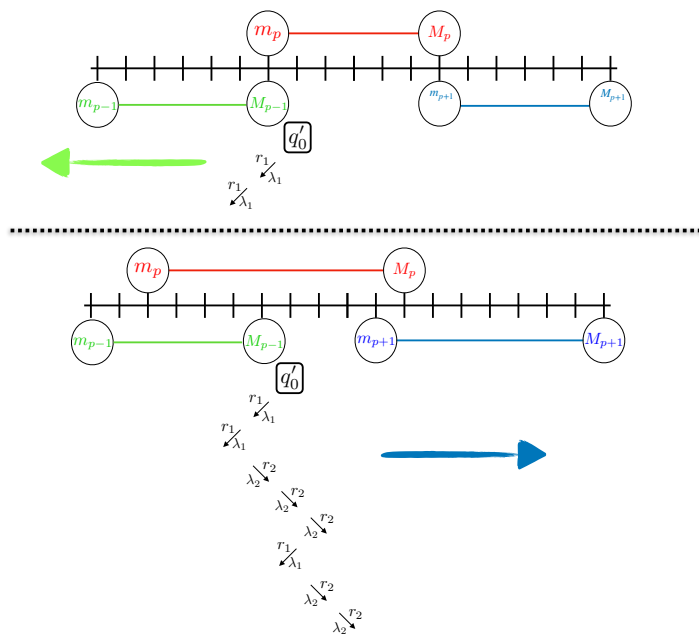


Figure 5.1: A minimal effort system for reaction network dynamics. On the top diagram, three consecutive non-overlapping intervals are implemented to coarse-grain the values in a chemical behavior. Given the reference interval $\llbracket m_p, M_p \rrbracket$, the initial sampling of the stochastic sequence can trigger an interval change, since the value is near the interval borders M_{p-1} and m_p . At this level of abstraction, the system dynamic appears to proceed leftwards (top green arrow). In the bottom diagram, three bordering overlapping intervals are used to coarse-grain the stochastic dynamics. Continuing on the previous chemical behavior reveals a rightward shift (bottom blue arrow) in this system's dynamics. However, in this scenario since intervals overlap, the value does not change intervals along its sequence because although it enters in the green interval, it does not exit the red one.

abstraction framework since they are structurally similar to the state transition system of a hand-written logical model. That is, we use interval representations for each chemical variable, observe macro-transitions between regions of states made from interval products, and focus on phenotypic, time-independent observations representing biological properties. Two additional features arise from our formally derived models: (i) In general, transitions between macro-states are reversible, whereas in logical models macro-state transitions are uni-directional and towards of the systems stable fixpoint [36]; and (ii) probabilities between macro-transitions can be recovered by injecting the stochastic behaviors from a reaction network into its abstraction, which is not a feature in the logical modeling scenario. This approach differs to that done, for example, in [34] where the authors impose a CTMC on the Boolean state space, whereas in our work probabilities are derived from the physical basis governing chemical kinetics. Thus, we provide a formal tool which can aid in the assessment of the behaviors that emerge from logical-type models.

In the following sections, we derive the abstract semantics to automatically derive a macro-transition system from a reaction network. Our method of accomplishing this is similar to the steps taken in Chapter 4. Namely, we derive each of the abstract objects we require in face of the chemical content of Chapter 2. Importantly, we characterize overlapping intervals and use intervals to guide the abstraction of chemical quantities. This is possible thanks to the abstraction functions we derive which are parameterized by (or, sensitive to) information regarding the potential interval value observed by each quantity of a chemical species in a chemical behavior. Also, for each abstract object we establish a Galois connection between a chemical object and its abstract counterpart. We focus on the relational aspect of the abstraction process, since we established in Chapter 4 that a non-relational abstraction results in fictitious behaviors that can potentially lead to dubious conclusions. Finally, we conclude with the two case studies considered in Chapter 2 in light of the overlapping intervals we characterized.

5.1 Characterizing overlapping intervals

We begin by generalizing the intervals definition in Def. 4.1, and of Sect. 4.1, so that we can consider intervals that may overlap, or not.

Definition 5.1 (Intervals). *We consider a family $(\underline{q}_p^\#, \bar{q}_p^\#)_{1 \leq p \leq n}$ of n pairs of values in $\mathbb{N} \cup \{+\infty\}$ where n is a natural number in \mathbb{N} such that the following properties are all satisfied:*

1. *for every natural number p between 1 and n , $\underline{q}_p^\# < \bar{q}_p^\#$;*
2. *for every natural number p between 2 and n , $\underline{q}_p^\# \leq \bar{q}_{p-1}^\#$;*
3. *and $\underline{q}_1^\# = 0$ and $\bar{q}_n^\# = +\infty$.*

We denote the set of intervals $\{(q_p^\#, \bar{q}_p^\#) \mid 1 \leq p \leq n\}$ by $\mathcal{I}^\#$.

An interval $(\underline{q}_p^\#, \bar{q}_p^\#)$ denotes the set of values $\{x \in \mathbb{N} \mid \underline{q}_p^\# \leq x < \bar{q}_p^\#\}$. Alternatively, we can express an interval as $\llbracket \underline{q}_p^\#, \bar{q}_p^\# \llbracket$. There are finitely many of them. By Cond. 1, intervals are

well-formed. Additionally, there may be a region where intervals may overlap. That is, given two bordering intervals, by Cond. 2 the upper bound of the first interval is greater than or equal to the lower bound of the second interval. Moreover, by Cond. 3, the lower bound of the first interval is 0 ($q_1^\sharp = 0$) and the upper bound of the last interval is unbounded ($\bar{q}_n^\sharp = +\infty$). A valid abstraction for a natural number value is an interval in the domain \mathcal{I}^\sharp that contains this value. There may be several such intervals. To select the appropriate interval, we rely on a reference interval to compute the nearest one. The reference interval intends to abstract the previous value for the quantity of this chemical component. We will detail more on this computation in Sect. 5.2. Beforehand, and in order to motivate this idea, since the abstraction of a value can belong to several intervals, we would like to distinguish the selected interval from several potential candidates.

The purpose of the following proposition is to characterize a region of values (e.g., an interval) for a scenario exhibited by two ordered intervals that border after a chemical behavior exited the lower interval into a higher one.

Proposition 5.1. *Let i_{t-1} and i_t be two natural numbers between 1 and n such that $i_{t-1} < i_t$. Then:*

$$\left(\llbracket \bar{q}_{i_{t-1}}^\sharp, +\infty \rrbracket \cap \llbracket q_{i_t}^\sharp, \bar{q}_{i_t}^\sharp \rrbracket \right) \setminus \left(\bigcup_{i_{t-1} < k < i_t} \llbracket q_k^\sharp, \bar{q}_k^\sharp \rrbracket \right) = \llbracket \bar{q}_{i_{t-1}}^\sharp, \bar{q}_{i_t}^\sharp \rrbracket. \quad (5.1)$$

Each hand side of Eq. 5.1 describe a region of values. Please note carefully that we distinguish between an interval's index at a previous discrete time step i_{t-1} and position in space $i_t - 1$. On the left hand side, the interval $\llbracket \bar{q}_{i_{t-1}}^\sharp, +\infty \rrbracket$ denotes the region above the upper bound of the abstract interval $\llbracket q_{i_{t-1}}^\sharp, \bar{q}_{i_{t-1}}^\sharp \rrbracket$ that was exited. The interval $\llbracket q_{i_t}^\sharp, \bar{q}_{i_t}^\sharp \rrbracket$ is the region denoted by the abstract interval that has been entered. The new abstract interval is supposed to be the least one, hence, the value shall not be in an interval denoted by an abstract interval in between as formalised by the expression $\bigcup_{i_{t-1} < k < i_t} \llbracket q_k^\sharp, \bar{q}_k^\sharp \rrbracket$. The right hand side describes the interval between the upper bound (included) of the previous interval position, $\bar{q}_{i_{t-1}}^\sharp$, and the upper bound (excluded) of the level that we have just entered, $\bar{q}_{i_t}^\sharp$.

Proof. Let us prove the equality of Proposition 5.1.

1. Let us prove the inclusion from left to right.

Let q be a value in $\left(\llbracket \bar{q}_{i_{t-1}}^\sharp, +\infty \rrbracket \cap \llbracket q_{i_t}^\sharp, \bar{q}_{i_t}^\sharp \rrbracket \right) \setminus \left(\bigcup_{i_{t-1} < k < i_t} \llbracket q_k^\sharp, \bar{q}_k^\sharp \rrbracket \right)$.

We want to prove that $q \in \llbracket \bar{q}_{i_{t-1}}^\sharp, \bar{q}_{i_t}^\sharp \rrbracket$.

- (a) Let us prove that $q \geq \bar{q}_{i_{t-1}}^\sharp$.

By assumption, the inequality $i_{t-1} < i_t$ holds.

We consider two cases.

- i. We assume that $i_t = i_{t-1} + 1$.

We know that:

$$q \in \llbracket \bar{q}_{i_{t-1}}^\sharp, +\infty \rrbracket.$$

It follows that:

$$q \geq \bar{q}_{i_{t-1}}^{\#}.$$

Since, by assumption:

$$i_{t-1} = i_t - 1.$$

We conclude that $q \geq \bar{q}_{i_t-1}^{\#}$.

ii. We assume that $i_t > i_{t-1} + 1$.

We have $i_{t-1} < i_t - 1 < i_t$.

It follows from $q \notin \bigcup_{i_{t-1} < k < i_t} \llbracket \underline{q}_k^{\#}, \bar{q}_k^{\#} \rrbracket$ that the following statement:

$$q \notin \llbracket \underline{q}_{i_t-1}^{\#}, \bar{q}_{i_t-1}^{\#} \rrbracket. \quad (5.2)$$

holds.

We know that $q \in \llbracket \underline{q}_{i_t}^{\#}, \bar{q}_{i_t}^{\#} \rrbracket$.

It follows that:

$$q \geq \underline{q}_{i_t}^{\#}.$$

By assumption on increasing lower bounds, we get that:

$$\underline{q}_{i_t}^{\#} > \underline{q}_{i_t-1}^{\#}.$$

It follows from both previous inequalities that:

$$q > \underline{q}_{i_t-1}^{\#}. \quad (5.3)$$

By 5.2 and 5.3, we conclude that $q > \bar{q}_{i_t-1}^{\#}$.

In both cases, $q \geq \bar{q}_{i_t-1}^{\#}$.

(b) Let us prove that $q < \bar{q}_{i_t}^{\#}$.

By assumption, we have:

$$q \in \llbracket \underline{q}_{i_t}^{\#}, \bar{q}_{i_t}^{\#} \rrbracket.$$

We conclude that $q < \bar{q}_{i_t}^{\#}$.

Thus, $q \in \llbracket \bar{q}_{i_t-1}^{\#}, \bar{q}_{i_t}^{\#} \rrbracket$.

2. Let us prove the inclusion from right to left.

Let q be a state in $\llbracket \bar{q}_{i_t-1}^{\#}, \bar{q}_{i_t}^{\#} \rrbracket$.

Let us prove that $q \in \left(\llbracket \bar{q}_{i_t-1}^{\#}, +\infty \rrbracket \cap \llbracket \underline{q}_{i_t}^{\#}, \bar{q}_{i_t}^{\#} \rrbracket \right) \setminus \left(\bigcup_{i_{t-1} < k < i_t} \llbracket \underline{q}_k^{\#}, \bar{q}_k^{\#} \rrbracket \right)$.

(a) Let us prove that $q \in \llbracket \bar{q}_{i_t-1}^{\#}, +\infty \rrbracket$.

We have assumed that $i_{t-1} < i_t$.

Since the previous inequality ranges over integers, it follows that:

$$i_{t-1} \leq i_t - 1.$$

The following inequality follows by assumption on increasing upper bounds:

$$\bar{q}_{i_{t-1}}^\# < \bar{q}_{i_t-1}^\#.$$

Since, $q \geq \bar{q}_{i_t-1}^\#$, we conclude the following inequality:

$$q > \bar{q}_{i_{t-1}}^\#$$

which means that $q \in \left[\bar{q}_{i_{t-1}}^\#, +\infty \right[$.

(b) Let us prove that $q \in \left[\underline{q}_{i_t}^\#, \bar{q}_{i_t}^\# \right]$.

i. Let us prove that $q \geq \underline{q}_{i_t}^\#$.

By assumption on intervals, the inequality $\bar{q}_{i_{t-1}}^\# \geq \underline{q}_{i_t}^\#$ holds.

By assumption, the inequality $q \geq \bar{q}_{i_{t-1}}^\#$ is satisfied.

We conclude that: $q \geq \underline{q}_{i_t}^\#$.

ii. Let us prove that $q < \bar{q}_{i_t}^\#$.

We already know it, since, by assumption $q \in \left[\bar{q}_{i_{t-1}}^\#, \bar{q}_{i_t}^\# \right]$.

Thus, $q \in \left[\underline{q}_{i_t}^\#, \bar{q}_{i_t}^\# \right]$.

(c) Let us prove that $q \notin \bigcup_{i_{t-1} < k < i_t} \left[\underline{q}_k^\#, \bar{q}_k^\# \right]$.

Let k be a natural number such that $i_{t-1} < k < i_t$.

We want to prove that $q \notin \left[\underline{q}_k^\#, \bar{q}_k^\# \right]$.

Since k and i_t are both integers, it follows that the following inequality:

$$k \leq i_t - 1$$

holds.

By assumption that upper bounds are increasing, the following inequality:

$$\bar{q}_k^\# < \bar{q}_{i_t-1}^\#$$

holds.

Since $q \in \left[\bar{q}_{i_t-1}^\#, \bar{q}_{i_t}^\# \right]$, we deduce that:

$$q \geq \bar{q}_{i_t-1}^\#.$$

The following inequality follows from both previous inequalities:

$$q > \bar{q}_k^\#$$

We conclude that $q \notin \llbracket \underline{q}_k^\#, \bar{q}_k^\# \rrbracket$.

We conclude that $q \notin \bigcup_{i_{t-1} < k < i_t} \llbracket \underline{q}_k^\#, \bar{q}_k^\# \rrbracket$.

We conclude that: $q \in \left(\llbracket \underline{q}_{i_{t-1}}^\#, +\infty \rrbracket \cap \llbracket \underline{q}_{i_t}^\#, \bar{q}_{i_t}^\# \rrbracket \right) \setminus \left(\bigcup_{i_{t-1} < k < i_t} \llbracket \underline{q}_k^\#, \bar{q}_k^\# \rrbracket \right)$. \square

In the next proposition, we also express the scenario when a chemical behavior enters new interval that is in a lower position (e.g., the reverse case of Prop. 5.1).

Proposition 5.2. *Let i_{t-1} and i_t be two natural numbers between 1 and n such that $i_{t-1} > i_t$.*

$$\left(\llbracket 0, \underline{q}_{i_{t-1}}^\# \rrbracket \cap \llbracket \underline{q}_{i_t}^\#, \bar{q}_{i_t}^\# \rrbracket \right) \setminus \left(\bigcup_{i_t < k < i_{t-1}} \llbracket \underline{q}_k^\#, \bar{q}_k^\# \rrbracket \right) = \llbracket \underline{q}_{i_t}^\#, \underline{q}_{i_t+1}^\# \rrbracket \quad (5.4)$$

Respectively, each hand side of Eq. 5.4 also describes a region of values. On the left hand side, the interval $\llbracket 0, \underline{q}_{i_{t-1}}^\# \rrbracket$ denotes the region below the lower bound of the abstract interval $\llbracket \underline{q}_{i_{t-1}}^\#, \bar{q}_{i_{t-1}}^\# \rrbracket$ that was exited. The interval $\llbracket \underline{q}_{i_t}^\#, \bar{q}_{i_t}^\# \rrbracket$ is the region denoted by the abstract interval that just entered. The new abstract interval is supposed to be the greatest one, hence, the value shall not be in an interval denoted by an abstract interval in between as formalised by the expression $\bigcup_{i_t < k < i_{t-1}} \llbracket \underline{q}_k^\#, \bar{q}_k^\# \rrbracket$. The right hand side describes the interval between the lower bound (included) of the current interval position, $\underline{q}_{i_t}^\#$, and the lower bound (excluded) of the previous upper interval, $\underline{q}_{i_t+1}^\#$.

Proof. Let us prove the equality of Proposition 5.2.

1. Let us prove the inclusion from left to right.

Let q be a value in $\left(\llbracket 0, \underline{q}_{i_{t-1}}^\# \rrbracket \cap \llbracket \underline{q}_{i_t}^\#, \bar{q}_{i_t}^\# \rrbracket \right) \setminus \left(\bigcup_{i_t < k < i_{t-1}} \llbracket \underline{q}_k^\#, \bar{q}_k^\# \rrbracket \right)$.

We want to prove that $q \in \llbracket \underline{q}_{i_t}^\#, \underline{q}_{i_t+1}^\# \rrbracket$.

- (a) Let us prove that $q < \underline{q}_{i_t+1}^\#$.

By assumption, the inequality $i_{t-1} > i_t$ holds.

We consider two cases.

- i. We assume that $i_t = i_{t-1} - 1$.

We know that:

$$q \in \llbracket 0, \underline{q}_{i_{t-1}}^\# \rrbracket.$$

It follows that:

$$q < \underline{q}_{i_{t-1}}^\#.$$

Since, by assumption:

$$i_{t-1} = i_t + 1.$$

We conclude that $q < \underline{q}_{i_t+1}^\#$.

ii. We assume that $i_t < i_{t-1} - 1$.

We have: $i_{t-1} > i_t + 1 > i_t$.

It follows from $q \notin \bigcup_{i_t < k < i_{t-1}} \llbracket \underline{q}_k^\#, \bar{q}_k^\# \rrbracket$ that the following statement:

$$q \notin \llbracket \underline{q}_{i_t+1}^\#, \bar{q}_{i_t+1}^\# \rrbracket. \quad (5.5)$$

holds.

We know that $q \in \llbracket \underline{q}_{i_t}^\#, \bar{q}_{i_t}^\# \rrbracket$.

It follows that:

$$q < \bar{q}_{i_t}^\#.$$

By assumption on intervals, we get that:

$$\underline{q}_{i_t+1}^\# \leq \bar{q}_{i_t}^\#.$$

It follows from both previous inequalities that:

$$q < \underline{q}_{i_t+1}^\#. \quad (5.6)$$

By 5.5 and 5.6, we conclude that $q < \underline{q}_{i_t+1}^\#$.

In both cases, $q < \underline{q}_{i_t+1}^\#$.

(b) Let us prove that $\underline{q}_{i_t}^\# \leq q$.

By assumption, we have:

$$q \in \llbracket \underline{q}_{i_t}^\#, \bar{q}_{i_t}^\# \rrbracket.$$

We conclude that $\bar{q}_{i_t}^\# \leq q$.

Thus, $q \in \llbracket \underline{q}_{i_t}^\#, \underline{q}_{i_t+1}^\# \rrbracket$.

2. Let us prove the inclusion from right to left.

Let q be a state in $\llbracket \underline{q}_{i_t}^\#, \underline{q}_{i_t+1}^\# \rrbracket$.

Let us prove that $q \in \left(\llbracket 0, \underline{q}_{i_{t-1}}^\# \rrbracket \cap \llbracket \underline{q}_{i_t}^\#, \bar{q}_{i_t}^\# \rrbracket \right) \setminus \left(\bigcup_{i_t < k < i_{t-1}} \llbracket \underline{q}_k^\#, \bar{q}_k^\# \rrbracket \right)$.

(a) Let us prove that $q \in \llbracket 0, \underline{q}_{i_{t-1}}^\# \rrbracket$.

We have assumed that: $i_{t-1} > i_t$.

Since the previous inequality ranges over integers, it follows that:

$$i_{t-1} \geq i_t + 1.$$

The following inequality follows by assumption on decreasing lower bounds:

$$\underline{q}_{i_{t-1}}^\# > \underline{q}_{i_t+1}^\#.$$

Since, $q < \bar{q}_{i_t+1}^\sharp$, we conclude the following inequality:

$$q < \underline{q}_{i_t-1}^\sharp$$

which means that $q \in \llbracket 0, \underline{q}_{i_t-1}^\sharp \llbracket$.

(b) Let us prove that $q \in \llbracket \underline{q}_{i_t}^\sharp, \bar{q}_{i_t}^\sharp \llbracket$.

i. Let us prove that $q < \bar{q}_{i_t}^\sharp$.

By assumption on intervals, the inequality $\underline{q}_{i_t+1}^\sharp \leq \bar{q}_{i_t}^\sharp$ holds.

By assumption, the inequality $q < \underline{q}_{i_t+1}^\sharp$ is satisfied.

We conclude that: $q < \bar{q}_{i_t}^\sharp$.

ii. Let us prove that $q \geq \underline{q}_{i_t}^\sharp$.

We already know it, since, by assumption $q \in \llbracket \underline{q}_{i_t}^\sharp, \underline{q}_{i_t+1}^\sharp \llbracket$.

Thus, $q \in \llbracket \underline{q}_{i_t}^\sharp, \bar{q}_{i_t}^\sharp \llbracket$.

(c) Let us prove that $q \notin \bigcup_{i_t < k < i_{t-1}} \llbracket \underline{q}_k^\sharp, \bar{q}_k^\sharp \llbracket$.

Let k be a natural number such that $i_{t-1} > k > i_t$.

We want to prove that $q \notin \llbracket \underline{q}_k^\sharp, \bar{q}_k^\sharp \llbracket$.

Since k and i_t are both integers, it follows that the following inequality:

$$k \geq i_t + 1$$

holds.

By assumption that lower bounds are increasing, the following inequality:

$$\underline{q}_k^\sharp > \underline{q}_{i_t+1}^\sharp$$

Since $q \in \llbracket \underline{q}_{i_t}^\sharp, \underline{q}_{i_t+1}^\sharp \llbracket$, we deduce that:

$$q < \underline{q}_{i_t+1}^\sharp.$$

The following inequality follows from both previous inequalities:

$$q < \underline{q}_k^\sharp.$$

We conclude that $q \notin \llbracket \underline{q}_k^\sharp, \bar{q}_k^\sharp \llbracket$.

We conclude that $q \notin \bigcup_{i_t < k < i_{t-1}} \llbracket \underline{q}_k^\sharp, \bar{q}_k^\sharp \llbracket$.

We conclude that: $q \in \left(\llbracket 0, \underline{q}_{i_t-1}^\sharp \llbracket \cap \llbracket \underline{q}_{i_t}^\sharp, \bar{q}_{i_t}^\sharp \llbracket \right) \setminus \left(\bigcup_{i_t < k < i_{t-1}} \llbracket \underline{q}_k^\sharp, \bar{q}_k^\sharp \llbracket \right)$.

□

Now that we have extended and characterized our interval domain to include overlapping intervals, we will take advantage of this abstract object as the basis for a collection of abstract functions we will derive to obtain abstract elements from objects containing chemical information. These flavors of abstractions follow a similar to routine to those derivations made in Chapter 4. In Sect. 5.2, we derive the first of these abstract functions that permit us to map the values of chemical quantities into an interval.

5.2 Abstractions of quantities of chemical components using reference intervals

The goal of this section is to derive the value abstraction function that we will use to map the value of a chemical component to an abstract interval. In order to do so, we use the interval domain defined in Sect. 5.1 and further parameterize the derived value abstraction function with a reference interval. Thus, this abstract function will keep track of the quantity of a chemical species with respect to a given reference interval when computing interval assignments.

In the explanation of Def. 5.1, we mentioned that the abstraction of the quantity of a chemical species may result in several interval options if one uses overlapping intervals. Additionally, in Props. 5.1 - 5.2 we characterized the region of values that correspond to an interval change in the situation where bordering intervals may overlap. Thus, we would like to design our abstractions to take into account the interval options, use a reference interval to guide the abstraction of a value, and ensure that the selected interval is correct.

Often, we remark the nearest interval as the one for which the distance between the indices of two intervals that is minimal. For example, a common computation we will conduct is, given a reference interval and a set of potential intervals where an abstracted value may habituate, we identify the interval in the candidate set that has its index nearest to the index of the reference interval. Using the results of the proof of Lemma 3.1 allows us to compute for any reference interval its projection onto a non-empty interval set. It does so by supplying the value projection function of Def. 3.5 with a reference interval and a set of intervals of interest. Accordingly, we integrate this feature into the following interval choice function in order to automatically choose the nearest interval.

Definition 5.2 (Interval choice function). *Let $\llbracket q_p^\#, \bar{q}_p^\# \rrbracket \in \mathcal{I}^\#$ be a reference interval. The interval choice function maps the reference interval to the index of the nearest interval in a non-empty convex set of interval indexes:*

$$\mathcal{H}_{\llbracket q_p^\#, \bar{q}_p^\# \rrbracket}^{\mathbb{N}} = \begin{cases} \text{Conv}(\mathbb{N}) \setminus \{\emptyset\} & \rightarrow \mathbb{N} \\ Y & \mapsto \text{proj}_{\mathbb{N}}(p, Y). \end{cases}$$

In Def. 5.2, the domain and co-domain of the function $\mathcal{H}_{\llbracket q_p^\#, \bar{q}_p^\# \rrbracket}^{\mathbb{N}}$ are, respectively, the set of non-empty convex intervals of natural numbers and the set of natural numbers. The interval choice function is parameterized by a reference abstract interval and inputs a non-empty set of

abstract intervals. Given these two pieces of information, the output is the index of the interval that belongs to the input set and which is nearest to the reference.

The reason we derived the interval choice function is to guide the abstraction of the values of chemical components. Here, we use reference intervals in order to ensure that the interval assignment of a value, among several possible candidates, is nearest to the given reference. We use this strategy to track whether the quantity of a species has been displaced from a reference interval, thus potentially entering a new one, or not. For this reason, we derive the value abstraction function and parameterize it with a reference interval.

Definition 5.3 (Value abstraction function). *Let $\llbracket q_p^\#, \bar{q}_p^\# \rrbracket \in \mathcal{I}^\#$ be a reference interval. The value abstraction function maps each value $x \in \mathbb{N}$ to the unique abstract interval given by:*

$$\bar{\beta}_{\llbracket q_p^\#, \bar{q}_p^\# \rrbracket}^{\mathbb{N}} = \begin{cases} \mathbb{N} & \rightarrow & \mathcal{I}^\# \\ x & \mapsto & \llbracket q_{p'}^\#, \bar{q}_{p'}^\# \rrbracket \quad \text{where } p' = \mathcal{H}_{\llbracket q_p^\#, \bar{q}_p^\# \rrbracket}^{\mathbb{N}} \left(\left\{ k \in \llbracket 1, n \rrbracket \mid q_k^\# \leq x < \bar{q}_k^\# \right\} \right). \end{cases}$$

In Def. 5.3, the domain and co-domain of the function $\bar{\beta}_{\llbracket q_p^\#, \bar{q}_p^\# \rrbracket}^{\mathbb{N}}$ are, respectively, the set of values and the set of intervals. Also, it is well-formed thanks to the hypotheses in Def. 5.1. More precisely, the existence of an interval in $\mathcal{I}^\#$ containing the quantity of a chemical component follows from the fact that the elements of $\mathcal{I}^\#$ forms a covering of \mathbb{N} , then thanks to the monotonicity of lower and upper bounds of the intervals, the set of intervals that contain a given value are contiguous elements in the domain. The interval choice function (see Def. 5.2) picks the nearest to the reference interval. Additionally, the value abstraction function in Def. 5.3 differs from Def. 4.2 since the former is both contextualized by information about reaction networks and reference intervals.

Example 5.1. Let $\mathcal{I}^\# = \{\llbracket 0, 6 \rrbracket, \llbracket 3, 9 \rrbracket, \llbracket 5, +\infty \rrbracket\}$ be the set of intervals and $\llbracket 0, 6 \rrbracket$ be the reference interval.

We would like to abstract the values 3, 6, and 10 an interval in $\mathcal{I}^\#$. Thus, applying the value abstraction function (see Def. 5.3) onto each value gives $\bar{\beta}_{\llbracket 0, 6 \rrbracket}^{\mathbb{N}}(3) = \llbracket 0, 6 \rrbracket$, $\bar{\beta}_{\llbracket 0, 6 \rrbracket}^{\mathbb{N}}(6) = \llbracket 3, 9 \rrbracket$, and $\bar{\beta}_{\llbracket 3, 9 \rrbracket}^{\mathbb{N}}(10) = \llbracket 5, +\infty \rrbracket$.

We would like to highlight at this point the structural similarity and difference between of our interval abstractions to those of logical models. The main similarity at this level of abstraction is the interval representation for each chemical species, yet the interval type is different. In our value abstraction, an interval denotes a range of values for a given chemical species. A benefit is that intervals can be fine-tuned to encapsulate each quantity of a chemical species with more or less accuracy, and may be done to highlight a certain property of interest such as the mechanism of formation and quantity of some protein complexes under steady state dynamics. Indeed, the goal of the upcoming value concretization function in Def. 5.5 is to recover a concrete information from its abstract representation. In contrast, an interval in a logical model is of a qualitative nature and represents an activity level of a chemical species. Although in some scenarios this

simplification can capture emergent phenotypic properties of interest, such as important stages of a cell division process [13], one generally loses a lot of information regarding variations in the copy numbers diverse bio-molecules. In the aforementioned case, drastic assumptions are made to rank and marginalize such variations by considering the activity among classes of reactions that may be more likely than others. Yet, it is difficult to verify how these assumptions may impact the dynamics of this logical model.

At this point, we are able to obtain intervals from values by the corresponding abstraction function. Now, we would like to follow a similar procedure to that done in Chapter 4 in order to derive a Galois connection between intervals and values in the context of reaction networks. The first function we will derive is the value set abstraction function which will generate for us sets of intervals from sets of values.

Definition 5.4 (Value set abstraction). *Let $\llbracket q_p^\sharp, \bar{q}_p^\sharp \rrbracket \in \mathcal{I}^\sharp$ be a reference interval. The value set abstraction function maps a set of values to the set of abstract intervals:*

$$\bar{\alpha}_{\llbracket q_p^\sharp, \bar{q}_p^\sharp \rrbracket}^{\mathbb{N}} = \begin{cases} \wp(\mathbb{N}) & \rightarrow & \wp(\mathcal{I}^\sharp) \\ X & \mapsto & \left\{ \bar{\beta}_{\llbracket q_p^\sharp, \bar{q}_p^\sharp \rrbracket}^{\mathbb{N}}(x) \mid x \in X \right\}. \end{cases}$$

In Def. 5.4, the domain and co-domain of the function $\bar{\alpha}_{\llbracket q_p^\sharp, \bar{q}_p^\sharp \rrbracket}^{\mathbb{N}}$ are, respectively, the powerset of the set of values and the powerset of the set of intervals. This set abstraction function is parameterized by a reference interval and inputs a set of values. After which, the interval reference is propagated to $\bar{\beta}_{\llbracket q_p^\sharp, \bar{q}_p^\sharp \rrbracket}^{\mathbb{N}}$ (see Def. 5.3) which is used to abstract each value in the input set. However, at this level of representation it remains unclear how the value set abstraction function carefully assigns each value to its appropriate interval. Thus, a more specific version of the value set abstraction function is given in the following proposition. Namely, it uses the characterization in given in Prop. 5.1 to reveal that in order to abstract a value, its value must be compared to the boundaries of the given reference interval, and then one corrects its interval assignment by considering its neighboring intervals.

Proposition 5.3. *Let $X \in \wp(\mathbb{N})$ be a set of values. Let $\llbracket \underline{q}_{i_{t-1}}^\sharp, \bar{q}_{i_{t-1}}^\sharp \rrbracket$ be a reference interval. Let $X_1, X_2, X_3 \subseteq X$ such that:*

$$\begin{aligned} X_1 &= \left\{ x \in X \mid \bar{q}_{i_{t-1}}^\sharp \leq x \right\}; \\ X_2 &= \left\{ x \in X \mid x < \underline{q}_{i_{t-1}}^\sharp \right\}; \\ X_3 &= \left\{ x \in X \mid \underline{q}_{i_{t-1}}^\sharp \leq x < \bar{q}_{i_{t-1}}^\sharp \right\}. \end{aligned}$$

Then:

$$\bar{\alpha}_{\llbracket \underline{q}_{i_{t-1}}^\sharp, \bar{q}_{i_{t-1}}^\sharp \rrbracket}^{\mathbb{N}}(X) = Y_1 \cup Y_2 \cup Y_3$$

where:

$$\begin{aligned} Y_1 &= \{ \llbracket \underline{q}_{i_t}^\sharp, \bar{q}_{i_t}^\sharp \rrbracket \in \mathcal{I}^\sharp \mid i_t \in \llbracket 1, n \rrbracket, \exists x \in X_1 : i_{t-1} < i_t \wedge \bar{q}_{i_{t-1}}^\sharp \leq x < \bar{q}_{i_t}^\sharp \} \\ Y_2 &= \{ \llbracket \underline{q}_{i_t}^\sharp, \bar{q}_{i_t}^\sharp \rrbracket \in \mathcal{I}^\sharp \mid i_t \in \llbracket 1, n \rrbracket, \exists x \in X_2 : i_{t-1} > i_t \wedge \underline{q}_{i_t}^\sharp \leq x < \underline{q}_{i_{t+1}}^\sharp \} \\ Y_3 &= \begin{cases} \{ \llbracket \underline{q}_{i_{t-1}}^\sharp, \bar{q}_{i_{t-1}}^\sharp \rrbracket \} & \text{if } X_3 \neq \emptyset, \\ \emptyset & \text{if } X_3 = \emptyset. \end{cases} \end{aligned}$$

Proof. Let $X \in \wp(\mathbb{N})$ be a set of values. Let $\llbracket \underline{q}_p^\sharp, \bar{q}_p^\sharp \rrbracket \in \mathcal{I}^\sharp$ be a reference interval. Let $X_1, X_2, X_3 \subseteq X$ such that:

$$\begin{aligned} X_1 &= \{ x \in \mathbb{N} \mid \bar{q}_{i_{t-1}}^\sharp \leq x \}; \\ X_2 &= \{ x \in \mathbb{N} \mid x < \underline{q}_{i_{t-1}}^\sharp \}; \\ X_3 &= \{ x \in \mathbb{N} \mid \underline{q}_{i_{t-1}}^\sharp \leq x < \bar{q}_{i_{t-1}}^\sharp \}. \end{aligned}$$

We have:

$$X = X_1 \cup X_2 \cup X_3.$$

It follows that:

$$\bar{\alpha}_{\llbracket \underline{q}_{i_{t-1}}^\sharp, \bar{q}_{i_{t-1}}^\sharp \rrbracket}^{\mathbb{N}}(X) = \bar{\alpha}_{\llbracket \underline{q}_{i_{t-1}}^\sharp, \bar{q}_{i_{t-1}}^\sharp \rrbracket}^{\mathbb{N}}(X_1) \cup \bar{\alpha}_{\llbracket \underline{q}_{i_{t-1}}^\sharp, \bar{q}_{i_{t-1}}^\sharp \rrbracket}^{\mathbb{N}}(X_2) \cup \bar{\alpha}_{\llbracket \underline{q}_{i_{t-1}}^\sharp, \bar{q}_{i_{t-1}}^\sharp \rrbracket}^{\mathbb{N}}(X_3).$$

1. Let us prove that:

$$\bar{\alpha}_{\underline{q}_{i_t}^\sharp}^{\mathbb{N}}(X_1) = \{ \llbracket \underline{q}_{i_t}^\sharp, \bar{q}_{i_t}^\sharp \rrbracket \in \mathcal{I}^\sharp \mid i_t \in \llbracket 1, n \rrbracket, \exists x \in X_1 : i_{t-1} < i_t \wedge \bar{q}_{i_{t-1}}^\sharp \leq x < \bar{q}_{i_t}^\sharp \}.$$

By definition:

$$\bar{\alpha}_{\llbracket \underline{q}_{i_{t-1}}^\sharp, \bar{q}_{i_{t-1}}^\sharp \rrbracket}^{\mathbb{N}}(X_1) = \left\{ \bar{\beta}_{\llbracket \underline{q}_{i_{t-1}}^\sharp, \bar{q}_{i_{t-1}}^\sharp \rrbracket}^{\mathbb{N}}(x) \mid x \in X_1 \right\}.$$

Let be x an element of X_1 .

We denote as i_t the integer between 1 and n such that $\bar{\beta}_{\llbracket \underline{q}_{i_{t-1}}^\sharp, \bar{q}_{i_{t-1}}^\sharp \rrbracket}^{\mathbb{N}}(x) = \llbracket \underline{q}_{i_t}^\sharp, \bar{q}_{i_t}^\sharp \rrbracket$.

We have:

- (a) $\bar{q}_{i_{t-1}}^\sharp \leq x$;
- (b) $x \in \llbracket \underline{q}_{i_t}^\sharp, \bar{q}_{i_t}^\sharp \rrbracket$ (by Definition of $\bar{\beta}_{\llbracket \underline{q}_{i_{t-1}}^\sharp, \bar{q}_{i_{t-1}}^\sharp \rrbracket}^{\mathbb{N}}$);
- (c) for every $k \in \llbracket 1, n \rrbracket$ such that $i_{t-1} < k$, we have $x \notin \llbracket \underline{q}_k^\sharp, \bar{q}_k^\sharp \rrbracket$ (by Definition of $\bar{\beta}_{\llbracket \underline{q}_{i_{t-1}}^\sharp, \bar{q}_{i_{t-1}}^\sharp \rrbracket}^{\mathbb{N}}$);

It follows by Prop. 5.1 that:

$$x \in \llbracket \bar{q}_{i_{t-1}}^\sharp, \bar{q}_{i_t}^\sharp \rrbracket.$$

We can conclude that there exists $i_t \in \llbracket 1, n \rrbracket$ such that $i_{t-1} < i_t$ and $\bar{q}_{i_{t-1}}^\sharp \leq x < \bar{q}_{i_t}^\sharp$.

Since, $\llbracket \bar{q}_{i_{t-1}}^\sharp, \bar{q}_{i_t}^\sharp \rrbracket \subseteq X_1$, we get that:

$$\bar{\alpha}_{\llbracket \underline{q}_{i_{t-1}}^\sharp, \bar{q}_{i_{t-1}}^\sharp \rrbracket}^{\mathbb{N}}(X_1) = \left\{ \bar{\beta}_{\llbracket \underline{q}_{i_{t-1}}^\sharp, \bar{q}_{i_{t-1}}^\sharp \rrbracket}^{\mathbb{N}}(x) \mid \exists i_t \in \llbracket 1, n \rrbracket : i_{t-1} < i_t \wedge \bar{q}_{i_{t-1}}^\sharp \leq x < \bar{q}_{i_t}^\sharp \right\}.$$

That is to say:

$$\bar{\alpha}_{\llbracket \underline{q}_{i_{t-1}}^\#, \bar{q}_{i_{t-1}}^\# \rrbracket}^{\mathbb{N}}(X_1) = \left\{ \llbracket \underline{q}_{i_t}^\#, \bar{q}_{i_t}^\# \rrbracket \in \mathcal{I}^\# \mid i_t \in \llbracket 1, n \rrbracket, \exists x \in X_1 : i_{t-1} < i_t \wedge \bar{q}_{i_{t-1}}^\# \leq x < \underline{q}_{i_t}^\# \right\}.$$

2. Let us prove that:

$$\bar{\alpha}_{\underline{q}_{i_t}^\#}^{\mathbb{N}}(X_2) = \left\{ \llbracket \underline{q}_{i_t}^\#, \bar{q}_{i_t}^\# \rrbracket \in \mathcal{I}^\# \mid i_t \in \llbracket 1, n \rrbracket, \exists x \in X_2 : i_{t-1} > i_t \wedge \underline{q}_{i_t}^\# \leq x < \underline{q}_{i_{t+1}}^\# \right\}.$$

By definition:

$$\bar{\alpha}_{\llbracket \underline{q}_{i_{t-1}}^\#, \bar{q}_{i_{t-1}}^\# \rrbracket}^{\mathbb{N}}(X_2) = \left\{ \bar{\beta}_{\llbracket \underline{q}_{i_{t-1}}^\#, \bar{q}_{i_{t-1}}^\# \rrbracket}^{\mathbb{N}}(x) \mid x \in X_2 \right\}.$$

Let be x an element of X_2 .

We denote as i_t the integer between 1 and n such that $\bar{\beta}_{\llbracket \underline{q}_{i_{t-1}}^\#, \bar{q}_{i_{t-1}}^\# \rrbracket}^{\mathbb{N}}(x) = \llbracket \underline{q}_{i_t}^\#, \bar{q}_{i_t}^\# \rrbracket$.

We have:

- (a) $x < \underline{q}_{i_{t-1}}^\#$;
- (b) $x \in \llbracket \underline{q}_{i_t}^\#, \bar{q}_{i_t}^\# \rrbracket$ (by Definition of $\bar{\beta}_{\llbracket \underline{q}_{i_{t-1}}^\#, \bar{q}_{i_{t-1}}^\# \rrbracket}^{\mathbb{N}}$);
- (c) for every $k \in \llbracket 1, n \rrbracket$ such that $i_{t-1} > k$, we have $x \notin \llbracket \underline{q}_k^\#, \bar{q}_k^\# \rrbracket$ (by Definition of $\bar{\beta}_{\llbracket \underline{q}_{i_{t-1}}^\#, \bar{q}_{i_{t-1}}^\# \rrbracket}^{\mathbb{N}}$);

It follows by Prop. 5.2 that:

$$x \in \llbracket \underline{q}_{i_t}^\#, \underline{q}_{i_{t+1}}^\# \rrbracket.$$

We can conclude that there exists $i_t \in \llbracket 1, n \rrbracket$ such that $i_{t-1} < i_t$ and $\bar{q}_{i_t}^\# \leq x < \underline{q}_{i_{t+1}}^\#$.

Since, $\llbracket \underline{q}_{i_t}^\#, \underline{q}_{i_{t+1}}^\# \rrbracket \subseteq X_2$, we get that:

$$\bar{\alpha}_{\llbracket \underline{q}_{i_{t-1}}^\#, \bar{q}_{i_{t-1}}^\# \rrbracket}^{\mathbb{N}}(X_2) = \left\{ \bar{\beta}_{\llbracket \underline{q}_{i_{t-1}}^\#, \bar{q}_{i_{t-1}}^\# \rrbracket}^{\mathbb{N}}(x) \mid \exists i_t \in \llbracket 1, n \rrbracket : i_{t-1} < i_t \wedge \bar{q}_{i_t}^\# \leq x < \underline{q}_{i_{t+1}}^\# \right\}.$$

That is to say:

$$\bar{\alpha}_{\llbracket \underline{q}_{i_{t-1}}^\#, \bar{q}_{i_{t-1}}^\# \rrbracket}^{\mathbb{N}}(X_2) = \left\{ \llbracket \underline{q}_{i_t}^\#, \bar{q}_{i_t}^\# \rrbracket \in \mathcal{I}^\# \mid i_t \in \llbracket 1, n \rrbracket, \exists x \in X_2 : i_{t-1} > i_t \wedge \bar{q}_{i_t}^\# \leq x < \underline{q}_{i_{t+1}}^\# \right\}.$$

3. Finally, we compute $\bar{\alpha}_{\llbracket \underline{q}_{i_{t-1}}^\#, \bar{q}_{i_{t-1}}^\# \rrbracket}^{\mathbb{N}}(X_3)$ by case analysis.

- (a) if $X_3 = \emptyset$, then $\bar{\alpha}_{\llbracket \underline{q}_{i_{t-1}}^\#, \bar{q}_{i_{t-1}}^\# \rrbracket}^{\mathbb{N}}(X_3) = \emptyset$.
- (b) Otherwise,

- Let $x \in X_3$.

We have:

$$\bar{\beta}_{\llbracket \underline{q}_{i_{t-1}}^\#, \bar{q}_{i_{t-1}}^\# \rrbracket}^{\mathbb{N}}(x) = \llbracket \underline{q}_{i_{t-1}}^\#, \bar{q}_{i_{t-1}}^\# \rrbracket.$$

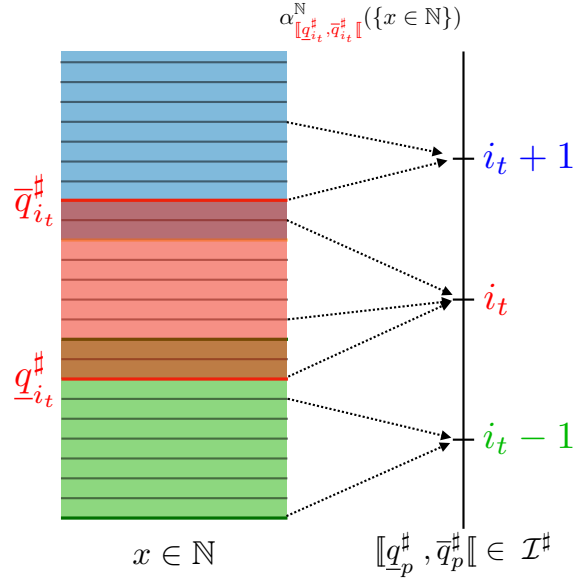


Figure 5.2: Abstraction of sets of values. On the left hand side are three sets of values depicted by three colored regions (green, red, and blue). Each line under each region depicts a value, and bordering regions share common values. On the right, we display the index of an interval along a vertical line, and the collection of these indices instantiated on the generic construct $\llbracket q_p^{\sharp}, \bar{q}_p^{\sharp} \rrbracket$ make up the set of intervals obtained from abstracting each region from the left. Each dashed arrow is a mapping of a region to an interval. The mapping from sets of values to sets of intervals is accomplished by the value set abstraction function of Prop. 5.3, which is parameterized by the reference interval $\llbracket q_{i_t}^{\sharp}, \bar{q}_{i_t}^{\sharp} \rrbracket$. This is done by comparing each value of each region on the left to the boundary values of the reference interval, and assigning this value to its corresponding interval.

Thus,

$$\bar{\alpha}_{\llbracket q_{i_t-1}^{\sharp}, \bar{q}_{i_t-1}^{\sharp} \rrbracket}^{\mathbb{N}}(X_3) \subseteq \left\{ \llbracket q_{i_t-1}^{\sharp}, \bar{q}_{i_t-1}^{\sharp} \rrbracket \right\}$$

- Since X_3 is not empty, we can pick x an element in it.

We have:

$$\bar{\beta}_{\llbracket q_{i_t-1}^{\sharp}, \bar{q}_{i_t-1}^{\sharp} \rrbracket}^{\mathbb{N}}(x) = \llbracket q_{i_t-1}^{\sharp}, \bar{q}_{i_t-1}^{\sharp} \rrbracket.$$

Thus:

$$\llbracket q_{\llbracket q_{i_t-1}^{\sharp}, \bar{q}_{i_t-1}^{\sharp} \rrbracket}^{\sharp}, \bar{q}_{\llbracket q_{i_t-1}^{\sharp}, \bar{q}_{i_t-1}^{\sharp} \rrbracket}^{\sharp} \rrbracket \in \bar{\alpha}_{\llbracket q_{i_t-1}^{\sharp}, \bar{q}_{i_t-1}^{\sharp} \rrbracket}^{\mathbb{N}}(X_3).$$

It follows that: $\bar{\alpha}_{\llbracket q_{i_t-1}^{\sharp}, \bar{q}_{i_t-1}^{\sharp} \rrbracket}^{\mathbb{N}}(X_3) = \left\{ \llbracket q_{i_t-1}^{\sharp}, \bar{q}_{i_t-1}^{\sharp} \rrbracket \right\}$

□

In Fig. 5.2, we illustrate how the three sets of values (depicted by the green, red, and blue regions) may be abstracted. The value set abstraction function of Prop. 5.3 is parameterized by the (red) reference interval $\llbracket q_{i_t}^{\sharp}, \bar{q}_{i_t}^{\sharp} \rrbracket$. In this scenario, we consider the abstraction of two sets of values, one above and below the reference interval, and one set equal to the reference. Note that each bordering region share some common values. Further, each independent dashed arrow

is an mapping by $\bar{\alpha}_{\llbracket \underline{q}_{i_t}^\#, \bar{q}_{i_t}^\# \rrbracket}^{\mathbb{N}}$ of each region on the left to an interval on the right. The collection of each interval (instantiated on $\llbracket \underline{q}_p^\#, \bar{q}_p^\# \rrbracket$) forms the set of intervals obtained after the mapping. Thus, each interval assignment is achieved by comparing the value within each region to the reference interval, and correcting the respective interval assignment by either Prop. 5.1 or 5.2 (which not shown in this diagram for the sake of simplicity).

Example 5.2.

Let $\mathcal{I}^\# = \{\llbracket 0, 6 \rrbracket, \llbracket 3, 9 \rrbracket, \llbracket 5, +\infty \rrbracket\}$ be the set of intervals and $\llbracket 0, 6 \rrbracket$ be the reference interval.

We would like to abstract the concrete set $\mathcal{X} = \{2, 6, 10\}$. Accordingly, inputting this concrete set into the value set abstraction function outputs $\bar{\alpha}_{\llbracket 0, 6 \rrbracket}^{\mathbb{N}}(\mathcal{X}) = \{\llbracket 0, 6 \rrbracket, \llbracket 3, 9 \rrbracket, \llbracket 5, +\infty \rrbracket\}$.

The second function we consider is the interval set concretization function to obtain sets of values from sets of intervals.

Definition 5.5 (Abstract interval set concretization). *Let $\llbracket \underline{q}_p^\#, \bar{q}_p^\# \rrbracket \in \mathcal{I}^\#$ be a reference interval. The abstract interval set concretization function maps a set of abstract intervals to the set of values:*

$$\bar{\gamma}_{\llbracket \underline{q}_p^\#, \bar{q}_p^\# \rrbracket}^{\mathcal{I}^\#} = \left\{ \begin{array}{ll} \wp(\mathcal{I}^\#) & \rightarrow \wp(\mathbb{N}) \\ Y & \mapsto \left\{ x \in \mathbb{N} \mid \bar{\beta}_{\llbracket \underline{q}_p^\#, \bar{q}_p^\# \rrbracket}^{\mathbb{N}}(x) \in Y \right\} \end{array} \right.$$

In Def. 5.5, the domain and co-domain for the function $\bar{\gamma}_{\llbracket \underline{q}_p^\#, \bar{q}_p^\# \rrbracket}^{\mathcal{I}^\#}$ are, respectively, the powerset of the set of intervals and the powerset of the set of values. Similarly to Def. 5.4, the interval set concretization function is also parameterized with a reference interval. This function outputs, from an input set of intervals, the set of values with the property that abstracting each value by the value abstraction function (see Def. 5.3) belongs to the input set. Since each value has more than one interval representation, then there can be several sets of values from each interval. Thus, we invoke Prop. 5.1 and take the set union to filter each concrete set, to ensure that our interval assignments are correct in the context of overlapping intervals. This mechanistic process becomes more apparent in the following proposition.

Proposition 5.4. *Let $Y \in \wp(\mathcal{I}^\#)$ be a set of abstract intervals. Let $\llbracket \underline{q}_{i_{t-1}}^\#, \bar{q}_{i_{t-1}}^\# \rrbracket$ be a reference interval. Let $Y_1, Y_2, Y_3 \subseteq Y$ such that:*

$$\begin{aligned} Y_1 &= \left\{ \llbracket \underline{q}_{i_t}^\#, \bar{q}_{i_t}^\# \rrbracket \in Y \mid i_{t-1} < i_t \right\}; \\ Y_2 &= \left\{ \llbracket \underline{q}_{i_t}^\#, \bar{q}_{i_t}^\# \rrbracket \in Y \mid i_{t-1} > i_t \right\}; \\ Y_3 &= \left\{ \llbracket \underline{q}_{i_t}^\#, \bar{q}_{i_t}^\# \rrbracket \in Y \mid i_{t-1} = i_t \right\}. \end{aligned}$$

Then:

$$\bar{\gamma}_{\llbracket \underline{q}_{i_{t-1}}^\#, \bar{q}_{i_{t-1}}^\# \rrbracket}^{\mathcal{I}^\#}(Y) = X_1 \cup X_2 \cup X_3$$

where:

$$\begin{aligned} X_1 &= \bigcup_{\llbracket \underline{q}_{i_t}^\sharp, \bar{q}_{i_t}^\sharp \rrbracket \in Y_1} \llbracket \bar{q}_{i_{t-1}}^\sharp, \bar{q}_{i_t}^\sharp \rrbracket; \\ X_2 &= \bigcup_{\llbracket \underline{q}_{i_t}^\sharp, \bar{q}_{i_t}^\sharp \rrbracket \in Y_2} \llbracket \underline{q}_{i_t}^\sharp, \underline{q}_{i_{t+1}}^\sharp \rrbracket; \\ X_3 &= \begin{cases} \llbracket \underline{q}_{i_t}^\sharp, \bar{q}_{i_t}^\sharp \rrbracket & \text{if } Y_3 \neq \emptyset, \\ \emptyset & \text{if } Y_3 = \emptyset. \end{cases} \end{aligned}$$

Proof. Let $Y \in \wp(\mathcal{D}^\sharp)$ be a set of intervals. Let $\llbracket \underline{q}_{i_{t-1}}^\sharp, \bar{q}_{i_{t-1}}^\sharp \rrbracket$ be a reference interval. Let $Y_1, Y_2, Y_3 \subseteq Y$ the three following sets:

$$\begin{aligned} Y_1 &= \left\{ \llbracket \underline{q}_{i_t}^\sharp, \bar{q}_{i_t}^\sharp \rrbracket \in \mathcal{I}^\sharp \mid i_{t-1} < i_t \right\}; \\ Y_2 &= \left\{ \llbracket \underline{q}_{i_t}^\sharp, \bar{q}_{i_t}^\sharp \rrbracket \in \mathcal{I}^\sharp \mid i_{t-1} > i_t \right\}; \\ Y_3 &= \left\{ \llbracket \underline{q}_{i_t}^\sharp, \bar{q}_{i_t}^\sharp \rrbracket \in \mathcal{I}^\sharp \mid i_{t-1} = i_t \right\}. \end{aligned}$$

We have:

$$Y = Y_1 \cup Y_2 \cup Y_3.$$

It follows that:

$$\bar{\gamma}_{\llbracket \underline{q}_{i_{t-1}}^\sharp, \bar{q}_{i_{t-1}}^\sharp \rrbracket}^{\mathcal{I}^\sharp}(Y) = \bar{\gamma}_{\llbracket \underline{q}_{i_{t-1}}^\sharp, \bar{q}_{i_{t-1}}^\sharp \rrbracket}^{\mathcal{I}^\sharp}(Y_1) \cup \bar{\gamma}_{\llbracket \underline{q}_{i_{t-1}}^\sharp, \bar{q}_{i_{t-1}}^\sharp \rrbracket}^{\mathcal{I}^\sharp}(Y_2) \cup \bar{\gamma}_{\llbracket \underline{q}_{i_{t-1}}^\sharp, \bar{q}_{i_{t-1}}^\sharp \rrbracket}^{\mathcal{I}^\sharp}(Y_3).$$

1. Let us prove that:

$$\bar{\gamma}_{\underline{q}_{i_t}^\sharp}^{\mathcal{I}^\sharp}(Y_1) = \bigcup_{\llbracket \underline{q}_{i_t}^\sharp, \bar{q}_{i_t}^\sharp \rrbracket \in Y_1} \llbracket \bar{q}_{i_{t-1}}^\sharp, \bar{q}_{i_t}^\sharp \rrbracket.$$

• Let us prove that:

$$\bar{\gamma}_{\underline{q}_{i_t}^\sharp}^{\mathcal{I}^\sharp}(Y_1) \subseteq \bigcup_{\llbracket \underline{q}_{i_t}^\sharp, \bar{q}_{i_t}^\sharp \rrbracket \in Y_1} \llbracket \bar{q}_{i_{t-1}}^\sharp, \bar{q}_{i_t}^\sharp \rrbracket.$$

By definition:

$$\bar{\gamma}_{\llbracket \underline{q}_{i_{t-1}}^\sharp, \bar{q}_{i_{t-1}}^\sharp \rrbracket}^{\mathcal{I}^\sharp}(Y_1) = \left\{ x \in \mathbb{N} \mid \bar{\beta}_{\llbracket \underline{q}_{i_{t-1}}^\sharp, \bar{q}_{i_{t-1}}^\sharp \rrbracket}^{\mathbb{N}}(x) \in Y_1 \right\}.$$

Let be x be an element of $\bar{\gamma}_{\llbracket \underline{q}_{i_{t-1}}^\sharp, \bar{q}_{i_{t-1}}^\sharp \rrbracket}^{\mathcal{I}^\sharp}(Y_1)$.

We denote as i_t the integer between 1 and n such that $\bar{\beta}_{\llbracket \underline{q}_{i_{t-1}}^\sharp, \bar{q}_{i_{t-1}}^\sharp \rrbracket}^{\mathbb{N}}(x) = \llbracket \underline{q}_{i_t}^\sharp, \bar{q}_{i_t}^\sharp \rrbracket$.

We know that: $\llbracket \underline{q}_{i_t}^\sharp, \bar{q}_{i_t}^\sharp \rrbracket \in Y_1$.

Additionally, we have:

- (a) $\bar{q}_{i_{t-1}}^\sharp \leq x$;
- (b) $x \in \llbracket \underline{q}_{i_t}^\sharp, \bar{q}_{i_t}^\sharp \rrbracket$ (by Definition of $\bar{\beta}_{\llbracket \underline{q}_{i_{t-1}}^\sharp, \bar{q}_{i_{t-1}}^\sharp \rrbracket}^{\mathbb{N}}$);

- (c) for every $k \in \llbracket 1, n \rrbracket$ such that $i_{t-1} < k$, we have $x \notin \llbracket \underline{q}_{i_t}^\sharp, \bar{q}_{i_t}^\sharp \rrbracket$
 (by Definition of $\bar{\beta}_{\text{intex}\underline{q}_{i_{t-1}}^\sharp, \bar{q}_{i_{t-1}}^\sharp}^{\mathbb{N}}$);

It follows by Prop. 5.1 that: $x \in \llbracket \bar{q}_{i_{t-1}}^\sharp, \bar{q}_{i_t}^\sharp \rrbracket$.

We get that:

$$\bar{\gamma}_{\llbracket \underline{q}_{i_{t-1}}^\sharp, \bar{q}_{i_{t-1}}^\sharp \rrbracket}^{\mathcal{I}^\sharp}(Y_1) \subseteq \bigcup_{\llbracket \underline{q}_{i_t}^\sharp, \bar{q}_{i_t}^\sharp \rrbracket \in Y_1} \llbracket \bar{q}_{i_{t-1}}^\sharp, \bar{q}_{i_t}^\sharp \rrbracket.$$

- Let us prove that:

$$\bigcup_{\llbracket \underline{q}_{i_t}^\sharp, \bar{q}_{i_t}^\sharp \rrbracket \in Y_1} \llbracket \bar{q}_{i_{t-1}}^\sharp, \bar{q}_{i_t}^\sharp \rrbracket \subseteq \bar{\gamma}_{q_{i_t}^\sharp}^{\mathcal{I}^\sharp}(Y_1).$$

Let $\llbracket \underline{q}_{i_t}^\sharp, \bar{q}_{i_t}^\sharp \rrbracket \in Y_1$.

For $x \in \llbracket \bar{q}_{i_{t-1}}^\sharp, \bar{q}_{i_t}^\sharp \rrbracket$, we have $\bar{\beta}_{\llbracket \underline{q}_{i_{t-1}}^\sharp, \bar{q}_{i_{t-1}}^\sharp \rrbracket}^{\mathbb{N}}(x) = \llbracket \underline{q}_{i_t}^\sharp, \bar{q}_{i_t}^\sharp \rrbracket$.

Thus, $\llbracket \bar{q}_{i_{t-1}}^\sharp, \bar{q}_{i_t}^\sharp \rrbracket \subseteq \bar{\gamma}_{q_{i_t}^\sharp}^{\mathcal{I}^\sharp}(Y_1)$.

It follow that:

$$\bigcup_{\llbracket \underline{q}_{i_t}^\sharp, \bar{q}_{i_t}^\sharp \rrbracket \in Y_1} \llbracket \bar{q}_{i_{t-1}}^\sharp, \bar{q}_{i_t}^\sharp \rrbracket \subseteq \bar{\gamma}_{q_{i_t}^\sharp}^{\mathcal{I}^\sharp}(Y_1).$$

Thus:

$$\bar{\gamma}_{q_{i_t}^\sharp}^{\mathcal{I}^\sharp}(Y_1) = \bigcup_{\llbracket \underline{q}_{i_t}^\sharp, \bar{q}_{i_t}^\sharp \rrbracket \in Y_1} \llbracket \bar{q}_{i_{t-1}}^\sharp, \bar{q}_{i_t}^\sharp \rrbracket.$$

2. Let us prove that:

$$\bar{\gamma}_{q_{i_t}^\sharp}^{\mathcal{I}^\sharp}(Y_2) = \bigcup_{\llbracket \underline{q}_{i_t}^\sharp, \bar{q}_{i_t}^\sharp \rrbracket \in Y_2} \llbracket \underline{q}_{i_t}^\sharp, \underline{q}_{i_t+1}^\sharp \rrbracket.$$

- Let us prove that:

$$\bar{\gamma}_{q_{i_t}^\sharp}^{\mathcal{I}^\sharp}(Y_2) \subseteq \bigcup_{\llbracket \underline{q}_{i_t}^\sharp, \bar{q}_{i_t}^\sharp \rrbracket \in Y_2} \llbracket \underline{q}_{i_t}^\sharp, \underline{q}_{i_t+1}^\sharp \rrbracket.$$

By definition:

$$\bar{\gamma}_{\llbracket \underline{q}_{i_{t-1}}^\sharp, \bar{q}_{i_{t-1}}^\sharp \rrbracket}^{\mathcal{I}^\sharp}(Y_2) = \left\{ x \in \mathbb{N} \mid \bar{\beta}_{\llbracket \underline{q}_{i_{t-1}}^\sharp, \bar{q}_{i_{t-1}}^\sharp \rrbracket}^{\mathbb{N}}(x) \in Y_2 \right\}.$$

Let be x an element of $\bar{\gamma}_{\llbracket \underline{q}_{i_{t-1}}^\sharp, \bar{q}_{i_{t-1}}^\sharp \rrbracket}^{\mathcal{I}^\sharp}(Y_2)$.

We denote as i_t the integer between 1 and n such that $\bar{\beta}_{\llbracket \underline{q}_{i_{t-1}}^\sharp, \bar{q}_{i_{t-1}}^\sharp \rrbracket}^{\mathbb{N}}(x) = \llbracket \underline{q}_{i_t}^\sharp, \bar{q}_{i_t}^\sharp \rrbracket$.

We have:

- (a) $x < \underline{q}_{i_{t-1}}^\sharp$;
- (b) $x \in \llbracket \underline{q}_{i_t}^\sharp, \bar{q}_{i_t}^\sharp \rrbracket$ (by Definition of $\bar{\beta}_{\llbracket \underline{q}_{i_{t-1}}^\sharp, \bar{q}_{i_{t-1}}^\sharp \rrbracket}^{\mathbb{N}}$);
- (c) for every $k \in \llbracket 1, n \rrbracket$ such that $i_{t-1} < k$, we have $x \notin \llbracket \underline{q}_{i_t}^\sharp, \bar{q}_{i_t}^\sharp \rrbracket$
 (by Definition of $\bar{\beta}_{\llbracket \underline{q}_{i_{t-1}}^\sharp, \bar{q}_{i_{t-1}}^\sharp \rrbracket}^{\mathbb{N}}$);

It follows by Prop. 5.2 that $x \in \llbracket \underline{q}_{i_t}^\sharp, \underline{q}_{i_t+1}^\sharp \rrbracket$.

We can conclude that:

$$\overline{\gamma}_{q_{i_t}^\sharp}^{\mathcal{I}^\sharp}(Y_2) \subseteq \bigcup_{\llbracket \underline{q}_{i_t}^\sharp, \underline{q}_{i_t}^\sharp \rrbracket \in Y_2} \llbracket \underline{q}_{i_t}^\sharp, \underline{q}_{i_t+1}^\sharp \rrbracket.$$

- Let us prove that:

$$\bigcup_{\llbracket \underline{q}_{i_t}^\sharp, \underline{q}_{i_t}^\sharp \rrbracket \in Y_2} \llbracket \underline{q}_{i_t}^\sharp, \underline{q}_{i_t+1}^\sharp \rrbracket \subseteq \overline{\gamma}_{q_{i_t}^\sharp}^{\mathcal{I}^\sharp}(Y_2).$$

Let $\llbracket \underline{q}_{i_t}^\sharp, \underline{q}_{i_t}^\sharp \rrbracket \in Y_2$.

For $x \in \llbracket \underline{q}_{i_t}^\sharp, \underline{q}_{i_t+1}^\sharp \rrbracket$, we have $\overline{\beta}_{\llbracket \underline{q}_{i_t-1}^\sharp, \underline{q}_{i_t-1}^\sharp \rrbracket}^{\mathbb{N}}(x) = \llbracket \underline{q}_{i_t}^\sharp, \underline{q}_{i_t}^\sharp \rrbracket$.

Thus, $\llbracket \underline{q}_{i_t}^\sharp, \underline{q}_{i_t+1}^\sharp \rrbracket \subseteq \overline{\gamma}_{q_{i_t}^\sharp}^{\mathcal{I}^\sharp}(Y_2)$.

It follow that:

$$\bigcup_{\llbracket \underline{q}_{i_t}^\sharp, \underline{q}_{i_t}^\sharp \rrbracket \in Y_2} \llbracket \underline{q}_{i_t}^\sharp, \underline{q}_{i_t+1}^\sharp \rrbracket \subseteq \overline{\gamma}_{q_{i_t}^\sharp}^{\mathcal{I}^\sharp}(Y_2).$$

Thus:

$$\overline{\gamma}_{q_{i_t}^\sharp}^{\mathcal{I}^\sharp}(Y_2) = \bigcup_{\llbracket \underline{q}_{i_t}^\sharp, \underline{q}_{i_t}^\sharp \rrbracket \in Y_2} \llbracket \underline{q}_{i_t}^\sharp, \underline{q}_{i_t+1}^\sharp \rrbracket.$$

3. Finally,

(a) if $Y_3 = \emptyset$, then $\overline{\gamma}_{\llbracket \underline{q}_{i_t-1}^\sharp, \underline{q}_{i_t-1}^\sharp \rrbracket}^{\mathcal{I}^\sharp}(Y_3) = \emptyset$.

(b) Otherwise,

- Let $x \in \overline{\gamma}_{\llbracket \underline{q}_{i_t-1}^\sharp, \underline{q}_{i_t-1}^\sharp \rrbracket}^{\mathcal{I}^\sharp}(Y_3)$

We have:

$$\overline{\beta}_{\llbracket \underline{q}_{i_t-1}^\sharp, \underline{q}_{i_t-1}^\sharp \rrbracket}^{\mathbb{N}}(x) = \llbracket \underline{q}_{i_t-1}^\sharp, \underline{q}_{i_t-1}^\sharp \rrbracket.$$

Thus,

$$\overline{\gamma}_{\llbracket \underline{q}_{i_t-1}^\sharp, \underline{q}_{i_t-1}^\sharp \rrbracket}^{\mathcal{I}^\sharp}(Y_3) \subseteq \left\{ \llbracket \underline{q}_{i_t-1}^\sharp, \underline{q}_{i_t-1}^\sharp \rrbracket \right\}$$

- Since Y_3 is not empty, we can pick $\llbracket \underline{q}_{i_t}^\sharp, \underline{q}_{i_t}^\sharp \rrbracket$ an element in it.

By definition of Y_3 , we know that $i_t = i_{t-1}$.

We have:

$$\overline{\beta}_{\llbracket \underline{q}_{i_t-1}^\sharp, \underline{q}_{i_t-1}^\sharp \rrbracket}^{\mathbb{N}}(x) = \llbracket \underline{q}_{i_t-1}^\sharp, \underline{q}_{i_t-1}^\sharp \rrbracket.$$

Thus

$$\llbracket \underline{q}_{i_t-1}^\sharp, \underline{q}_{i_t-1}^\sharp \rrbracket \in \overline{\gamma}_{\llbracket \underline{q}_{i_t-1}^\sharp, \underline{q}_{i_t-1}^\sharp \rrbracket}^{\mathcal{I}^\sharp}(Y_3).$$

It follows that: $\overline{\gamma}_{\llbracket \underline{q}_{i_t-1}^\sharp, \underline{q}_{i_t-1}^\sharp \rrbracket}^{\mathcal{I}^\sharp}(Y_3) = \left\{ \llbracket \underline{q}_{i_t-1}^\sharp, \underline{q}_{i_t-1}^\sharp \rrbracket \right\}$.

□

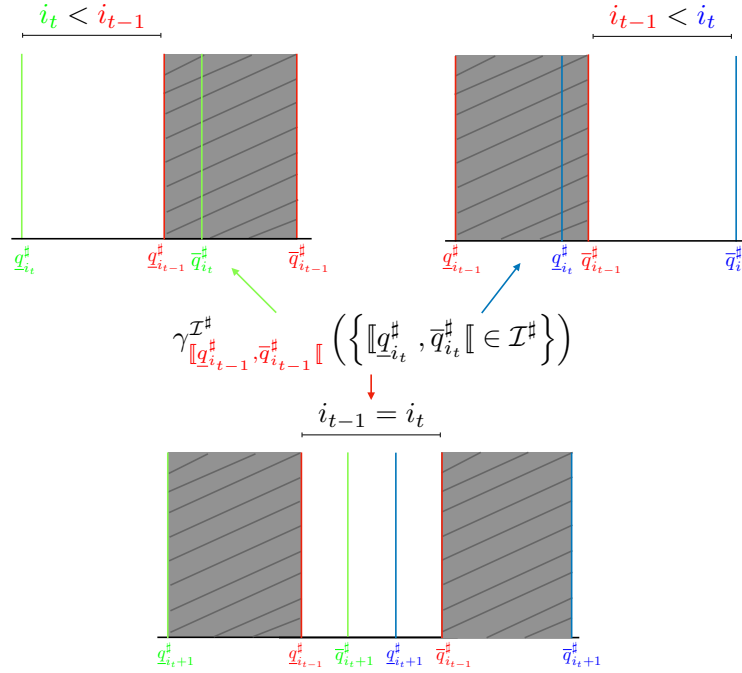


Figure 5.3: Concretization of sets of abstract intervals. entered in one time step. Thus, each diagram depicts a scenario for concretizing an entered interval whose index is either lower (top left), higher (top right) or equal (bottom) to the reference interval $(\llbracket q_{i_{t-1}}^{\#}, \bar{q}_{i_{t-1}}^{\#} \rrbracket)$. Accordingly, for each scenario we ensure that the respective set of intervals concretized by the interval set concretization function in Prop. 5.4 outputs the corresponding set of values pertain to the correct region of values (as characterized in Props. 5.1 - 5.2), thus omitting, for example, each gray regions.

In Fig. 5.3, we illustrate how the specific interval set concretization function of Prop. 5.4 handles the extraction of sets of values from overlapping intervals in an input set of intervals. Here, we display three possible scenarios and note that each interval is split into subsets of intervals depending on the value of the interval index with respect to the reference interval $\llbracket q_{i_{t-1}}^{\#}, \bar{q}_{i_{t-1}}^{\#} \rrbracket$. The first (top left, Fig. 5.3) corresponds to the concretization of the subset of intervals whose indices are smaller than the reference ($i_t < i_{t-1}$). In the second (top right, Fig. 5.3), we concretize the subset of intervals who indices are larger than the reference interval ($i_t > i_{t-1}$). The third (bottom center, Fig. 5.3), and final, concretizes the subset of intervals which indices equal to the reference ($i_t = i_{t-1}$). In each scenario, we ensure to remain consistent with the overlapping interval properties of Props. 5.1 - 5.2. That is, the gray regions of each interval scenario are removed as we require an interval to be exited completely.

Example 5.3.

Let $\mathcal{I}^{\#} = \{\llbracket 0, 6 \llbracket, \llbracket 3, 9 \llbracket, \llbracket 5, +\infty \llbracket\}$ be the set of intervals and $\llbracket 0, 6 \llbracket$ be the reference interval.

We would like to concretize the set of intervals $\mathcal{Y}^{\#} = \{\llbracket 0, 6 \llbracket, \llbracket 3, 9 \llbracket, \llbracket 5, +\infty \llbracket\}$ (note that $\mathcal{Y}^{\#} = \bar{\alpha}_{\llbracket 0, 6 \llbracket}^{\mathbb{N}}(\{2, 6, 10\})$, see Ex. 5.2). Then $\bar{\gamma}_{\llbracket 0, 6 \llbracket}^{\mathcal{I}^{\#}}(\mathcal{Y}^{\#}) = \mathbb{N}$.

Consequently, and as expected, the abstraction process $\bar{\gamma}_{\llbracket 0, 6 \llbracket}^{\mathcal{I}^{\#}} \circ \bar{\alpha}_{\llbracket 0, 6 \llbracket}^{\mathbb{N}}$ on the set $\{2, 6, 10\}$

has resulted in a concrete set less precise than the original, e.g., $\{2, 6, 10\} \subset [\overline{\gamma}_{[0,6]}^{\mathcal{I}^\#} \circ \overline{\alpha}_{[0,6]}^{\mathcal{N}}](\{2, 6, 10\})$.

In this Section, we introduced the interval domain with overlapping intervals with the goal of reducing fictitious abstractions that may emerge by the intrinsic stochastic properties of a reaction network. Additionally, we show how to derive intervals from the values of chemical components and how these abstractions can be guided by a reference interval. Indeed, the motivation of the reference interval abstractions is to track the directionality of a value involved in a reaction and to determine whether a value has exited a reference interval or not. Finally, we have derived a Galois connection between the values and interval domain. In comparison to the Galois connections we derived in Chapter 4, we focus our efforts on relational abstractions. Thus, we have at our disposal correspondances between the value (concrete) and interval (abstract) domains which ensures that our abstractions are sound by construct, meaning that we do not miss any component values.

5.3 Chemical state abstraction using reference macro-states

In Sect. 5.2, we introduced the process of abstracting values into intervals, especially those which overlap. In this Section, we would like to lift these type of operations onto chemical states, those which contain information regarding the quantity of a number of chemical species. To begin, we define the abstract counterpart to chemical states: macro-states.

Definition 5.6 (Macro-states). *Let \mathcal{V} be a set of variables. A macro-state is a function $q^\# : \mathcal{V} \rightarrow \mathcal{I}^\#$.*

The set of all macro-states is denoted $\overline{\mathcal{Q}}^\#$.

In Def. 5.6, a macro-state is a function $q^\#$ with a domain and co-domain, respectively, as the set of variables and the set of intervals. A macro-state can be interpreted as a vector, such that each respective element of the vector contains information regarding the interval value for a chemical species. Note that we use one abstract domain for each variable in order to ease the notation; yet, in practice one may consider several domains for each component. One can also regard a macro-state as a box (see Def. 3.6) delimited by the interval boundary values of each variable along its coordinate.

Note that a box corresponds to a convex set of macro-states, and is a prelude to the type of abstractions we would like to perform in this Chapter. That is, our goal is to abstract chemical states into macro-states while ensuring that each macro-state is nearest to a reference macro-state. Indeed, this would require a similar choice function to Def. 5.2. This motivates the following definition.

Definition 5.7 (Macro-state choice function). *Let $q^\# = [v \mapsto [\underline{q}_{i_v}^\#, \overline{q}_{i_v}^\#]]$ be a reference macro-*

state in $\overline{\mathcal{Q}}^\sharp$. The macro-state choice function is defined as:

$$\mathcal{H}_{q^\sharp}^{\mathcal{B}} = \begin{cases} \text{Box}(\llbracket 1, n \rrbracket^{\mathcal{V}}) \setminus \{\emptyset\} & \rightarrow \llbracket 1, n \rrbracket^{\mathcal{V}} \\ Y & \mapsto (\text{proj}_{\mathbb{N}}(i_v, \pi_v(Y)))_{v \in \mathcal{V}} \end{cases} \quad (5.7)$$

or, equivalently:

$$\mathcal{H}_{q^\sharp}^{\mathcal{B}} = \begin{cases} \text{Box}(\llbracket 1, n \rrbracket^{\mathcal{V}}) \setminus \{\emptyset\} & \rightarrow \llbracket 1, n \rrbracket^{\mathcal{V}} \\ Y & \mapsto \text{proj}_{\mathbb{N}^{\mathcal{V}}}([v \mapsto i_v], Y). \end{cases} \quad (5.8)$$

In Def. 5.7, the domain and co-domain of the function $\mathcal{H}_{q^\sharp}^{\mathcal{B}}$ are, respectively, the non-empty set of boxes and the set of functions from variables to interval indexes. The state choice function is parameterized by a reference macro-state (q^\sharp) and inputs a box set. The reference macro-state and box set are used as, respectively, source and target inputs to the state projection function of Def. 3.7. Thus, the state choice function can be interpreted in two manners. Either we can compute the projection of a state onto a box by reasoning coordinate wise (Eq. 5.7). Or, one can compute directly the projection (Eq. 5.8). In either way, the expressions are equivalent and the state choice function will output a macro-state that is both in the given box set and that is at minimal distance to the reference. We will use this function to guide the abstraction of chemical states.

In Fig. 5.4 we provide intuition as to how the macro-state choice function works. A projection of a given reference state (red) onto a box set is determined by minimizing the distance between the reference macro-state and each macro-state in the box set. Indeed, this computation is equivalent to an orthogonal projection of the reference macro-state and each macro-state in the box set onto its respective variable coordinate. Afterwhich, the distance between the index of each interval along their respective coordinates are computed, and the macro-state in the box with the smallest index distance per variable (depicted by the green lines in Fig. 5.4) to the reference is chosen.

Now, one can use the macro-state choice function to parameterize the following chemical state abstraction function, which permits one to compute macro-states from chemical states.

Definition 5.8 (Chemical state abstraction). *Let $q^\sharp \in \overline{\mathcal{Q}}^\sharp$ be a reference macro-state. The chemical state abstraction function maps each chemical state to the macro-state:*

$$\overline{\beta}_{q^\sharp}^{\mathcal{Q}} = \begin{cases} \mathcal{Q} & \rightarrow \overline{\mathcal{Q}}^\sharp \\ q & \mapsto \text{lift} \circ \left[\mathcal{H}_{q^\sharp}^{\mathcal{B}} \left(\prod_{v \in \mathcal{V}} \{p \in \llbracket 1, n \rrbracket \mid \underline{q}_p^\sharp \leq \pi_v(q) < \overline{q}_p^\sharp\} \right) \right]. \end{cases}$$

where *lift* maps each function from the set \mathcal{V} to the interval $\llbracket 1, n \rrbracket$ into the function $[v \mapsto \llbracket \underline{q}_{f(v)}^\sharp, \overline{q}_{f(v)}^\sharp \rrbracket]$ from the set \mathcal{V} to abstract domain \mathcal{I}^\sharp .

In Def. 5.8, the domain and co-domain of the function $\overline{\beta}_{q^\sharp}^{\mathcal{Q}}$ are, respectively, the set of chemical states and the set of macro-states. The chemical state abstraction function is parameterized by

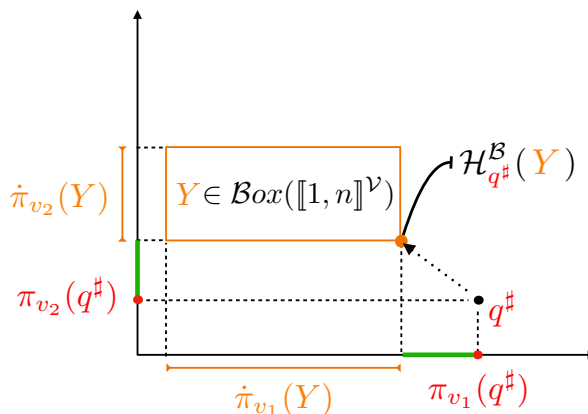


Figure 5.4: A projection of a reference macro-state to a box. The macro-state choice function $\mathcal{H}_{q^\#}^{\mathcal{B}}$ is parameterized by the reference macro-state $q^\#$ and inputs a box set, Y . The projection of this reference macro-state onto the box Y is equal to the nearest macro-state in the box. This distance is computed by calculating the component-wise distance between the interval index of the reference interval and each interval in the box, and then choosing the index with the smallest distance (depicted by the green lines). Note that in the diagram no difference is made between an interval and the index of this interval in the abstract domain.

reference a macro-state and inputs a chemical state. This information is propagated to the macro-state choice function (see Def. 5.7) which identifies from a set of macro-states (arising from the fact that there may be several macro-states for a chemical state) the one nearest to the reference. That is, each macro-state in the candidate set is composed of interval values for each variable and these intervals, along each respective coordinate, contains the corresponding component value of the abstracted chemical state. The auxiliary function lift is used to translate interval indexes into intervals.

Example 5.4. Let $\mathcal{I}^\# = \{\llbracket 0, 6 \rrbracket, \llbracket 3, 9 \rrbracket, \llbracket 5, +\infty \rrbracket\}$ be the set of intervals and $\mathcal{V} = \{A, B, C, AB, AC, ABC\}$ be the set of variables. Let $q^\# = [A \mapsto \llbracket 0, 6 \rrbracket, B \mapsto \llbracket 0, 6 \rrbracket, C \mapsto \llbracket 0, 6 \rrbracket, AB \mapsto \llbracket 0, 6 \rrbracket, BC \mapsto \llbracket 0, 6 \rrbracket, ABC \mapsto \llbracket 0, 6 \rrbracket]$ be the reference macro-state.

We would like to abstract the chemical state $q_\bullet = [A \mapsto 5, B \mapsto 10, C \mapsto 7, AB \mapsto 0, BC \mapsto 2, ABC \mapsto 1]$ with the chemical state abstraction function (see Def. 5.8) and the interval set $\mathcal{I}^\#$.

Thus, we obtain:

$$\bar{\beta}_{q^\#}^{\mathcal{Q}}(q_\bullet) = [A \mapsto \llbracket 0, 6 \rrbracket, B \mapsto \llbracket 5, +\infty \rrbracket, C \mapsto \llbracket 3, 9 \rrbracket, AB \mapsto \llbracket 0, 6 \rrbracket, BC \mapsto \llbracket 0, 6 \rrbracket, ABC \mapsto \llbracket 0, 6 \rrbracket].$$

We notice that in order to obtain the corresponding macro-state, each component value in the chemical state undergoes an interval inclusion check with each respective interval in the reference macro-state, and an interval assignment is computed according to how this value compares to the interval boundary values.

For us, a macro-state depicts a (hyper-)box given by the interval representation of each

chemical variable in a system. More specifically, a macro-state contains information of the interval value for a collection of variables and along its coordinate, which differs from the macro-states observed in logical models. Whereas in our formalizations each face of a macro-state considers a potential range of values, in the corresponding logical model each dimension is an activity level. This distinction will become important in Sect. 5.4 where we observe the ability for a macro-state to traverse from one macro-state to another. In the logical model, this ability to traverse is governed by the hand-written expression regulating the variables in the logical system. In our framework, a traversal is governed by the stochastic dynamics of a reaction network which permits one to pinpoint the direction a macro-state tends to habituate. A flaw, however, is that we consider additional macro-states that may not carry much information about the underlying reaction network behaviors. We plan in Chapter 6 to include information on the likelihood of habituating in one macro-state versus another, which should aid in alleviating the number of fictitious behaviors.

At this point we will begin to derive a Galois connection between the chemical state and macro-state domains. Thus, we begin by deriving the chemical state set abstraction function to map sets of chemical states to sets of macro-states.

Definition 5.9 (Chemical state set abstraction). *Let $q^\sharp \in \overline{\mathcal{Q}}^\sharp$ be a reference macro-state. The chemical state set abstraction function $\overline{\alpha}_{q^\sharp}^{\mathcal{Q}}$ maps a set of chemical states to the set of macro-states:*

$$\overline{\alpha}_{q^\sharp}^{\mathcal{Q}} = \begin{cases} \wp(\mathcal{Q}) & \rightarrow & \wp(\overline{\mathcal{Q}}^\sharp) \\ X & \mapsto & \{\overline{\beta}_{q^\sharp}^{\mathcal{Q}}(q) \mid q \in X\}. \end{cases}$$

In Def. 5.9, the domain and co-domain of $\overline{\alpha}_{q^\sharp}^{\mathcal{Q}}$ are, respectively, the powerset of the set of chemical states and the powerset of the set of macro-states. The chemical state set abstraction function is parameterized by a reference macro-state, q^\sharp , and inputs a set of chemical states. Accordingly, this function outputs a set of macro-states by lifting the chemical state abstraction function (see Def. 5.8) over each chemical state in the input set.

In the following proposition, we derive a specific expression for Def. 5.9 to abstract a given set of chemical states, which is accomplished by ensuring that for each macro-state obtained, their resulting interval values are elements of the set containing the respective chemical component value that has been abstracted by the value set abstraction function (see Def. 5.3).

Proposition 5.5. *Let $X \in \wp(\mathcal{Q})$ be a set of chemical states. Let $q^\sharp \in \overline{\mathcal{Q}}^\sharp$ be a reference macro-state. Then:*

$$\overline{\alpha}_{q^\sharp}^{\mathcal{Q}}(X) = \left\{ q^{\sharp'} \in \overline{\mathcal{Q}}^\sharp \mid \exists q \in X, \forall v \in \mathcal{V} : q^{\sharp'}(v) \in \overline{\alpha}_{q^\sharp(v)}^{\mathbb{N}}(\{q(v)\}) \right\}.$$

Proof. Let $X \in \wp(\mathcal{Q})$ be a set of chemical states. Let $q^\sharp \in \overline{\mathcal{Q}}^\sharp$ be a reference macro-state.

$$\begin{aligned} \overline{\alpha}_{q^\sharp}^{\mathcal{Q}}(X) &= \left\{ \overline{\beta}_{q^\sharp}^{\mathcal{Q}}(q) \mid q \in X \right\} \\ &= \left\{ q^{\sharp'} \in \overline{\mathcal{Q}}^\sharp \mid \exists q \in X : \overline{\beta}_{q^\sharp}^{\mathcal{Q}}(q) = q^{\sharp'} \right\} \\ &= \left\{ q^{\sharp'} \in \overline{\mathcal{Q}}^\sharp \mid \exists q \in X, \forall v \in \mathcal{V} : \overline{\beta}_{q^\sharp}^{\mathcal{N}}(q(v)) = q^{\sharp'}(v) \right\} \\ &= \left\{ q^{\sharp'} \in \overline{\mathcal{Q}}^\sharp \mid \exists q \in X, \forall v \in \mathcal{V} : q^{\sharp'}(v) \in \overline{\alpha}_{q^\sharp}^{\mathcal{N}}(\{q(v)\}) \right\}. \end{aligned}$$

□

One immediate observation one should make of Prop. 5.5 is that the expression has built on the value set abstraction function of Prop. 5.3. Thus, the process of abstracting a set of chemical states narrows down to performing interval abstraction of each value in each chemical state.

Example 5.5. Let $\mathcal{I}^\sharp = \{\llbracket 0, 6\llbracket, \llbracket 3, 9\llbracket, \llbracket 5, +\infty\llbracket\}$ be the set of intervals and $\mathcal{V} = \{A, B, C, AB, AC, ABC\}$ be the set of variables. Let $q^\sharp = [A \mapsto \llbracket 0, 6\llbracket, B \mapsto \llbracket 0, 6\llbracket, C \mapsto \llbracket 0, 6\llbracket, AB \mapsto \llbracket 0, 6\llbracket, BC \mapsto \llbracket 0, 6\llbracket, ABC \mapsto \llbracket 0, 6\llbracket$ be the reference macro-state.

We would like to abstract with the chemical state set abstraction function (see Prop. 5.5) the set containing the following chemical states:

$$\begin{aligned} q_\bullet &= [A \mapsto 5, B \mapsto 10, C \mapsto 7, AB \mapsto 0, BC \mapsto 2, ABC \mapsto 1], \\ q_\diamond &= [A \mapsto 1, B \mapsto 6, C \mapsto 2, AB \mapsto 0, BC \mapsto 7, ABC \mapsto 0]. \end{aligned}$$

Thus, one obtains $\overline{\alpha}_{q^\sharp}^{\mathcal{Q}}(\{q_\bullet, q_\diamond\}) = \{q_\bullet^\sharp, q_\diamond^\sharp\}$ where:

$$\begin{aligned} q_\bullet^\sharp &= [A \mapsto \llbracket 0, 6\llbracket, B \mapsto \llbracket 5, +\infty\llbracket, C \mapsto \llbracket 3, 9\llbracket, AB \mapsto \llbracket 0, 6\llbracket, BC \mapsto \llbracket 0, 6\llbracket, ABC \mapsto \llbracket 0, 6\llbracket], \\ q_\diamond^\sharp &= [A \mapsto \llbracket 0, 6\llbracket, B \mapsto \llbracket 3, 9\llbracket, C \mapsto \llbracket 0, 6\llbracket, AB \mapsto \llbracket 0, 6\llbracket, BC \mapsto \llbracket 3, 9\llbracket, ABC \mapsto \llbracket 0, 6\llbracket]. \end{aligned}$$

Accordingly, the following macro-state set concretization function will be used to obtain sets of chemical states from sets of macro-states.

Definition 5.10 (Macro-state set concretization). *Let $q^\sharp \in \overline{\mathcal{Q}}^\sharp$ be a reference macro-state. The macro-state set concretization function maps a set of macro-states to the set of chemical states:*

$$\overline{\gamma}_{q^\sharp}^{\mathcal{Q}} : \begin{cases} \wp(\overline{\mathcal{Q}}^\sharp) & \rightarrow & \wp(\mathcal{Q}) \\ Y & \mapsto & \left\{ q \in \mathcal{Q} \mid \overline{\beta}_{q^\sharp}^{\mathcal{Q}}(q) \in Y \right\}. \end{cases}$$

In Def. 5.10, the domain and co-domain of the function $\overline{\gamma}_{q^\sharp}^{\mathcal{Q}}$ are, respectively, the powerset of the set of macro-states and the powerset of the set of chemical states. The macro-state set concretization function is parameterized by a reference macro-state and inputs a set of macro-states. This function will output a set of chemical states with the property that abstracting each

chemical state by the chemical state abstraction function (see Def. 5.8) will retain its abstract counterpart in the input set.

Respectively, a more specific version of how a set of macro-states is concretized into a set of chemical states is given in the following proposition. In this case, a collection of concrete states are retrieved from each macro-state in a given abstract set by checking that for a given variable, each value in its chemical state belongs to a corresponding interval value of each macro-state in the given set.

Proposition 5.6. *Let $Y \in \wp(\overline{\mathcal{Q}}^\sharp)$ be a set of macro-states. Let $q^\sharp \in \overline{\mathcal{Q}}^\sharp$ be a reference macro-state.*

$$\overline{\gamma}_{q^\sharp}^{\mathcal{Q}^\sharp}(Y) = \bigcup_{q^{\sharp'} \in Y} \left\{ q \in \mathcal{Q} \mid \forall v \in \mathcal{V} : q(v) \in \overline{\gamma}_{q^{\sharp'}(v)}^{\mathcal{I}^\sharp} \left(\{q^{\sharp'}(v)\} \right) \right\}.$$

Proof. Let $Y \in \wp(\overline{\mathcal{Q}}^\sharp)$ be a set of macro-states. Let $q^\sharp \in \overline{\mathcal{Q}}^\sharp$ be a reference macro-state.

$$\begin{aligned} \overline{\gamma}_{q^\sharp}^{\mathcal{Q}^\sharp}(Y) &= \left\{ q \in \mathcal{Q} \mid \overline{\beta}_{q^\sharp}^{\mathcal{Q}}(q) \in Y \right\} \\ &= \left\{ q \in \mathcal{Q} \mid \exists q^{\sharp'} \in Y : \overline{\beta}_{q^\sharp}^{\mathcal{Q}}(q) = q^{\sharp'} \right\} \\ &= \left\{ q \in \mathcal{Q} \mid \exists q^{\sharp'} \in Y, \forall v \in \mathcal{V} : \overline{\beta}_{q^{\sharp'}(v)}^{\mathbb{N}}(q(v)) = q^{\sharp'}(v) \right\} \\ &= \left\{ q \in \mathcal{Q} \mid \exists q^{\sharp'} \in Y, \forall v \in \mathcal{V} : q(v) \in \overline{\gamma}_{q^{\sharp'}(v)}^{\mathcal{I}^\sharp} \left(\{q^{\sharp'}(v)\} \right) \right\} \\ &= \bigcup_{q^{\sharp'} \in Y} \left\{ q \in \mathcal{Q} \mid \forall v \in \mathcal{V} : q(v) \in \overline{\gamma}_{q^{\sharp'}(v)}^{\mathcal{I}^\sharp} \left(\{q^{\sharp'}(v)\} \right) \right\} \end{aligned}$$

□

Example 5.6. Let $\mathcal{I}^\sharp = \{\llbracket 0, 6\llbracket, \llbracket 3, 9\llbracket, \llbracket 5, +\infty\llbracket\}$ be the set of intervals and $\mathcal{V} = \{A, B, C, AB, AC, ABC\}$ be the set of variables. Let $q^\sharp = [A \mapsto \llbracket 0, 6\llbracket, B \mapsto \llbracket 0, 6\llbracket, C \mapsto \llbracket 0, 6\llbracket, AB \mapsto \llbracket 0, 6\llbracket, BC \mapsto \llbracket 0, 6\llbracket, ABC \mapsto \llbracket 0, 6\llbracket\}$ be the reference macro-state.

We would like to concretize with the macro-state set concretization function (see Prop. 5.6) the abstract set obtained in the result of Ex. 5.5. Namely, the abstract set of macro-states $\{q_\bullet^\sharp, q_\diamond^\sharp\}$ such that:

$$\begin{aligned} q_\bullet^\sharp &= [A \mapsto \llbracket 0, 6\llbracket, B \mapsto \llbracket 5, +\infty\llbracket, C \mapsto \llbracket 3, 9\llbracket, AB \mapsto \llbracket 0, 6\llbracket, BC \mapsto \llbracket 0, 6\llbracket, \\ &\quad ABC \mapsto \llbracket 0, 6\llbracket, \\ q_\diamond^\sharp &= [A \mapsto \llbracket 0, 6\llbracket, B \mapsto \llbracket 3, 9\llbracket, C \mapsto \llbracket 0, 6\llbracket, AB \mapsto \llbracket 0, 6\llbracket, BC \mapsto \llbracket 3, 9\llbracket, \\ &\quad ABC \mapsto \llbracket 0, 6\llbracket. \end{aligned}$$

Thus, we obtain that $\overline{\gamma}_{q^\sharp}^{\mathcal{Q}^\sharp}(\{q_\bullet^\sharp, q_\diamond^\sharp\})$ is equal to:

$$\left\{ \begin{array}{l} [A \mapsto \llbracket 0, 6\llbracket, B \mapsto \llbracket 9, +\infty\llbracket, C \mapsto \llbracket 6, 9\llbracket, AB \mapsto \llbracket 0, 6\llbracket, BC \mapsto \llbracket 0, 6\llbracket, ABC \mapsto \llbracket 0, 6\llbracket \\ [A \mapsto \llbracket 0, 6\llbracket, B \mapsto \llbracket 6, 9\llbracket, C \mapsto \llbracket 0, 6\llbracket, AB \mapsto \llbracket 0, 6\llbracket, BC \mapsto \llbracket 6, 9\llbracket, ABC \mapsto \llbracket 0, 6\llbracket \end{array} \right\}$$

The output set is a super-set of the original set of chemical states (see Ex. 5.5). For example,

note that the abstraction process $[\overline{\gamma}_{q^\#}^{Q^\#} \circ \overline{\alpha}_{q^\#}^Q](\{q_\bullet, q_\diamond\})$ such that:

$$\begin{aligned} q_\bullet &= [A \mapsto 5, B \mapsto 10, C \mapsto 7, AB \mapsto 0, BC \mapsto 2, ABC \mapsto 1], \\ q_\diamond &= [A \mapsto 1, B \mapsto 6, C \mapsto 2, AB \mapsto 0, BC \mapsto 7, ABC \mapsto 0]. \end{aligned}$$

has resulted in an unbounded solution space for the species B .

Similar to the chemical state set abstraction function of Prop. 5.5, the expression in Prop. 5.6 builds on the interval set concretization function derived in Prop. 5.4. This means that, for example, we can reason on the concretization of a macro-state in an abstract set component-wise and reconstruct the corresponding chemical state in a concrete set from the product of interval sets considered for each variable. Each interval set comes from the interval values of each variable in the macro-state.

The abstraction process $\overline{\gamma}_{q^\#}^{Q^\#} \circ \overline{\alpha}_{q^\#}^Q$ for an initial set of chemical states may consider a larger region of values as it traverses to the macro-state domain and then back to the chemical state domain. The consequence is an analysis that can be largely imprecise since we consider a bigger set of solutions for each chemical species in a reaction system. However, one possible way to retain a degree of precision is to impose intrinsic chemical constraints on the variables of an underlying system. For example, a common linear constraint is mass invariance, meaning that the molecular composition of certain chemical species are bounded by the number of diverse markers that form a species. This property was formalized in [1] to increase the precision of the proposed abstractions of a deterministic reaction system. In our work, we also considered the resource constraint which bounded the competition system of Chapter 2.

We have discussed in this Section the process of abstracting chemical states to macro-states, and how one can parameterize the chemical state abstract function with a reference macro-state. Further, this parameterization is guided by the state choice function which ensures that the macro-state that approximates a chemical state is nearest to a given reference. In our context, macro-states depict regions of values that arise from each interval representation of each chemical species considered in a reaction network. We also show that the chemical and macro-state domains form a Galois connection, and how each Galois function for states build on its predecessor functions derived for the value domains in Sect. 5.2. In the next section, Sect. 5.4, we will derive relations between macro-states, which give rise to macro-transitions. These macro-transitions will provide further insights as to how, for example, one variable can traverse from one region to another.

5.4 Macro-transitions between regions of chemical states

So far, we have derived two abstract elements: intervals, which are obtained from abstracting the values of variables; and, macro-states, which are generated from abstracted chemical states. In this Section, we will derive the third abstract object: macro-transitions, and further derive the necessary functions to obtain this object from chemical transitions (see Def. 2.4). Finally, we will

conclude by deriving a Galois connection between the chemical transition and macro-transition domains.

First, we define a macro-transition which shares a common structure to the abstract transitions in Def. 4.11 but further equipped with information from reaction networks.

Definition 5.11 (Macro-transitions). *An element $(q^\sharp, l, q^{\sharp'}) \in \overline{\mathcal{Q}}^\sharp \times \llbracket 1, \sigma \rrbracket \times \overline{\mathcal{Q}}^\sharp$ is called an abstract transition.*

The set of all abstract transitions is denoted \overline{T}^\sharp .

In Def. 5.11, a macro-transition is a relation between two macro-states. Each transition between macro-states is labeled by the index of an event l which denotes which reaction in the network has been triggered. An application of an event may change the interval value of a subset of variables and is reflected in the target macro-state. Underlying this interval change is a chemical transition that occurred, and its abstraction captures the ability for chemical state to traverse from one region to another. Thus, one obtains macro-transitions from chemical ones.

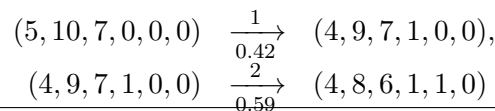
Definition 5.12 (Chemical transition abstraction). *Let $q^\sharp \in \overline{\mathcal{Q}}^\sharp$ be a reference macro-state. The chemical transition abstraction function maps each chemical transition (q, l, q') such that $\overline{\beta}_{q^\sharp}^{\mathcal{Q}}(q) = q^\sharp$ to a macro-transition:*

$$\overline{\beta}_{q^\sharp}^T = \begin{cases} \{(q, l, q') \in T \mid \overline{\beta}_{q^\sharp}^{\mathcal{Q}}(q) = q^\sharp\} & \rightarrow \{(q^{\sharp''}, l, q^{\sharp'}) \in \overline{T}^\sharp \mid q^{\sharp''} = q^\sharp\} \\ (q, l, q') & \mapsto (q^\sharp, l, \overline{\beta}_{q^\sharp}^{\mathcal{Q}}(q')). \end{cases}$$

In Def. 5.12, the chemical transition abstraction function is parameterized by a reference macro-state q^\sharp . The domain and co-domain of the function $\overline{\beta}_{q^\sharp}^T$ are, respectively, the set of the chemical transitions that starts in a state q such that $\overline{\beta}_{q^\sharp}^{\mathcal{Q}}(q) = q^\sharp$ and the set of the macro-transitions that starts in the macro-state q^\sharp . This way, the reference macro-state must be equal to the abstracted source chemical state of the input chemical transition. This will ensure that the abstracted target state of the same chemical transition is nearest to the reference, and will aid in pinpointing the direction of the dynamics of the underlying chemical transition system. Furthermore, the abstraction of the aforementioned target chemical state is accomplished using the chemical state abstraction function (see Def. 5.8).

Example 5.7. Let $\mathcal{I}^\sharp = \{\llbracket 0, 6 \llbracket, \llbracket 3, 9 \llbracket, \llbracket 5, +\infty \llbracket\}$ be the set of interval parameters and $\mathcal{V} = \{A, B, C, AB, AC, ABC\}$ be the set of variables. Let $q^\sharp = [A \mapsto \llbracket 0, 6 \llbracket, B \mapsto \llbracket 5, +\infty \llbracket, C \mapsto \llbracket 3, 9 \llbracket, AB \mapsto \llbracket 0, 6 \llbracket, BC \mapsto \llbracket 0, 6 \llbracket, ABC \mapsto \llbracket 0, 6 \llbracket$ be the reference macro-state.

We would like to abstract using the chemical transition abstraction function (see Def. 5.12) the following two chemical transitions obtained from a starting chemical state $q_1 = (5, 10, 7, 0, 0, 0)$ using the reaction rules in Ex. 5.1 and using the kinetic constants from [1] (all reaction rates have been set to 1):



Note that the probabilities have been computed using Def. 2.5.

As such, applying $\overline{\beta}_{q^\sharp}^T$ on these chemical transitions results in the following macro-transitions:

$$\begin{aligned} & (\llbracket 0, 6 \rrbracket, \llbracket 5, +\infty \rrbracket, \llbracket 3, 9 \rrbracket, \llbracket 0, 6 \rrbracket, \llbracket 0, 6 \rrbracket, \llbracket 0, 6 \rrbracket) \xrightarrow{\sharp} (\llbracket 0, 6 \rrbracket, \llbracket 5, +\infty \rrbracket, \llbracket 3, 9 \rrbracket, \llbracket 0, 6 \rrbracket, \llbracket 0, 6 \rrbracket, \llbracket 0, 6 \rrbracket), \\ & (\llbracket 0, 6 \rrbracket, \llbracket 5, +\infty \rrbracket, \llbracket 3, 9 \rrbracket, \llbracket 0, 6 \rrbracket, \llbracket 0, 6 \rrbracket, \llbracket 0, 6 \rrbracket) \xrightarrow{\sharp} (\llbracket 0, 6 \rrbracket, \llbracket 5, +\infty \rrbracket, \llbracket 3, 9 \rrbracket, \llbracket 0, 6 \rrbracket, \llbracket 0, 6 \rrbracket, \llbracket 0, 6 \rrbracket). \end{aligned}$$

We notice two things from our abstraction example. First, the macro-states do not change along each macro-transition. We notice two things from our abstraction example. First, the macro-states do not change along each macro-transition. This is specific to the two transitions that we have abstracted. In particular, we notice that in the second transition, the value of the variable B enter a new interval. Yet, since it does not exit the previous one, it does not induce a change of abstract value. This illustrate the minimal effort strategy that is introduced by the overlapping between intervals. Second, our abstraction of a chemical transition loses information regarding probabilities, thus macro-transitions have become non-deterministic.

In Sect. 5.3, we briefly discussed the ability of a macro-state traversing to another, and in Sects. 2.2 - 2.3 its comparison to the same feature in logical models. We will expand on these details now. Namely, a logical state in a logical model is driven by the hand-written expression that regulates each of the chemical variable in each state. Further, following René Thomas' principle [36], a logical state transition is uni-directional and towards a state configuration that is stable and reveals some phenotypic observation(s) that is deemed biologically plausible. Reverse directions in logical macro-transitions are possible, but this implodes the number of macro-states that one must consider and thus hampers the analysis of the transition state space. In our formal framework, we do not restrict the directionality of a macro-transition. Thus, bi-directional macro-transitions are plausible and are driven by the dynamics from a reaction network. However, as mentioned previously, we must also consider additional behaviors for the macro-states as a consequence. Additionally, our abstractions of chemical transitions loses information regarding probabilities. Thus, the resulting macro-transition system is non-deterministic. In order to circumvent the introduction of artifacts in our abstractions, we will examine the likelihood for a given pair of macro-transitions to occur in Chapter 6, and attempt to refine the macro-transition system with information pertaining to the actual dynamics of the underlying reaction network. This allows, for example, to evaluate as to whether some macro-transitions are more plausible than others, and thus omissions of certain macro-transitions can be guided by their association to low probabilities.

Similarly to the previous abstract domains, we will establish a Galois connection between the chemical transition and macro-transition domains. Thus, we derive the chemical transition set abstraction function to map sets of chemical transitions to sets of macro-transitions.

Definition 5.13 (Chemical transition set abstraction). *Let $q^\sharp \in \overline{\mathcal{Q}}^\sharp$ be a reference macro-state. The chemical transition set abstraction function maps a set of chemical transitions all starting*

in a state q such that $\bar{\beta}_{q^\sharp}^{\mathcal{Q}}(q) = q^\sharp$ to a set of macro-transitions:

$$\bar{\alpha}_{q^\sharp}^T = \begin{cases} \wp(\{(q, l, q') \in T \mid \bar{\beta}_{q^\sharp}^{\mathcal{Q}}(q) = q^\sharp\}) & \rightarrow \wp(\{(q^{\sharp\prime\prime}, l, q^{\sharp\prime}) \in \bar{T}^\sharp \mid q^{\sharp\prime\prime} = q^\sharp\}) \\ X & \mapsto \{\bar{\beta}_{q^\sharp}^T(q, l, q') \in \bar{T}^\sharp \mid (q, l, q') \in X\}. \end{cases}$$

In Def. 5.13, the chemical transition set abstraction function is parameterized by a reference macro-state q^\sharp . The domain and co-domain of the function $\bar{\alpha}_{q^\sharp}^T$ are, respectively, the powerset of the set of chemical transitions (q, l, q') such that $\bar{\beta}_{q^\sharp}^{\mathcal{Q}}(q) = q^\sharp$ and the powerset of the set of macro-transitions that start from the macro-state q^\sharp . Thus, each chemical transition in the input set is abstracted by the chemical transition abstraction function of Def. 5.12.

In the following proposition, a more specific version of Def. 5.13 reveals that abstracting a given set of chemical transitions equals a set composed of macro-transitions such that each macro-transition's source and target macro-states are elements belonging to abstract sets obtained from lifting the chemical state set abstraction function (see Def. 5.5) onto the distinct, respective, sets containing the source and target chemical states of each chemical transition in the given set.

Proposition 5.7. *Let $q^\sharp \in \bar{\mathcal{Q}}^\sharp$ be a reference macro-state. Let $X \in \wp(T)$ be a set of chemical transitions (q, l, q') , such that $\bar{\beta}_{q^\sharp}^{\mathcal{Q}}(q) = q^\sharp$. Then:*

$$\bar{\alpha}_{q^\sharp}^T(X) = \{(q^\sharp, l, q^{\sharp\prime}) \in \bar{T}^\sharp \mid \exists q, q' \in \mathcal{Q} : (q, l, q') \in X \wedge q^{\sharp\prime} \in \bar{\alpha}_{q^\sharp}^{\mathcal{Q}}(\{q'\})\}.$$

Proof. Let $q^\sharp \in \bar{\mathcal{Q}}^\sharp$ be a reference macro-state. Let $X \in \wp(T)$ be a set of chemical transitions (q, l, q') , such that $\bar{\beta}_{q^\sharp}^{\mathcal{Q}}(q) = q^\sharp$.

$$\begin{aligned} \bar{\alpha}_{q^\sharp}^T(X) &= \{\bar{\beta}_{q^\sharp}^T(q, l, q') \in \bar{T}^\sharp \mid (q, l, q') \in X\} \\ &= \{(q^{\sharp\prime\prime}, l, q^{\sharp\prime}) \in \bar{T}^\sharp \mid \exists (q, l, q') \in X : \bar{\beta}_{q^\sharp}^T(q, l, q') = (q^{\sharp\prime\prime}, l, q^{\sharp\prime})\} \\ &= \{(q^{\sharp\prime\prime}, l, q^{\sharp\prime}) \in \bar{T}^\sharp \mid \exists q, q' \in \mathcal{Q} : (q, l, q') \in X \wedge \bar{\beta}_{q^\sharp}^{\mathcal{Q}}(q) = q^{\sharp\prime\prime} \wedge \bar{\beta}_{q^\sharp}^{\mathcal{Q}}(q') = q^{\sharp\prime}\} \\ &= \{(q^\sharp, l, q^{\sharp\prime}) \in \bar{T}^\sharp \mid \exists q, q' \in \mathcal{Q} : (q, l, q') \in X \wedge \bar{\beta}_{q^\sharp}^{\mathcal{Q}}(q') = q^{\sharp\prime}\} \\ &= \{(q^\sharp, l, q^{\sharp\prime}) \in \bar{T}^\sharp \mid \exists q, q' \in \mathcal{Q} : (q, l, q') \in X \wedge q^{\sharp\prime} \in \bar{\alpha}_{q^\sharp}^{\mathcal{Q}}(\{q'\})\}. \end{aligned}$$

□

Accordingly, we derive the macro-transition set concretization function responsible for mapping sets of macro-transitions to sets of chemical transitions.

Definition 5.14 (Macro-transition set concretization). *Let $q^\sharp \in \bar{\mathcal{Q}}^\sharp$ be a reference macro-state. The macro-transition set concretization function maps a set of macro-transitions to the set of chemical transitions:*

$$\bar{\gamma}_{q^\sharp}^{T^\sharp} = \begin{cases} \wp(\{(q^{\sharp\prime\prime}, l, q^{\sharp\prime}) \in \bar{T}^\sharp \mid q^{\sharp\prime\prime} = q^\sharp\}) & \rightarrow \wp(\{(q, l, q') \in T \mid \bar{\beta}_{q^\sharp}^\mathcal{Q}(q) = q^\sharp\}) \\ Y & \mapsto \{(q, l, q') \in T \mid \bar{\beta}_{q^\sharp}^T(q, l, q') \in Y\}. \end{cases}$$

In Def. 5.14, the macro-transition set concretization function is parameterized by a reference macro-state q^\sharp . The domain and co-domain of the function $\bar{\gamma}_{q^\sharp}^{T^\sharp}$ are, respectively, the powerset of the set of the macro-transitions that starts in the state q^\sharp and the powerset of the set of chemical transitions that start in a state q such that $\bar{\beta}_{q^\sharp}^\mathcal{Q}(q) = q^\sharp$. The function takes as an input a set of macro-transitions. The resulting output concrete set is composed with those chemical transitions with the property that their abstraction is an element of the input set.

In the following proposition, this becomes more evident as concretizing a given set of macro-transitions output the union of the sets of chemical transitions which are composed of chemical transitions whose source and target chemical states are elements of the sets composed of, respectively, the source and target macro-states of each macro-transition in the input set after having been concretized by the macro-state set concretization function (see Prop. 5.6).

Proposition 5.8. *Let $q^\sharp \in \bar{\mathcal{Q}}^\sharp$ be a reference macro-state.*

Let $Y \in \wp(\{(q^{\sharp\prime\prime}, l, q^{\sharp\prime}) \in \bar{T}^\sharp \mid q^{\sharp\prime\prime} = q^\sharp\})$ be a set of macro-transitions. Then:

$$\bar{\gamma}_{q^\sharp}^{T^\sharp}(Y) = \bigcup_{q^{\sharp\prime} \in \bar{\mathcal{Q}}^\sharp} \{(q, l, q') \in T \mid (q^\sharp, l, q^{\sharp\prime}) \in Y \wedge q \in \bar{\gamma}_{q^\sharp}^\mathcal{Q}(\{q^\sharp\}) \wedge q' \in \bar{\gamma}_{q^\sharp}^\mathcal{Q}(\{q^{\sharp\prime}\})\}.$$

Proof. Let $Y \in \wp(\{(q^{\sharp\prime\prime}, l, q^{\sharp\prime}) \in \bar{T}^\sharp \mid q^{\sharp\prime\prime} = q^\sharp\})$ be a set of macro-transitions. Let $q^\sharp \in \bar{\mathcal{Q}}^\sharp$ be a reference macro-state and $q \in \mathcal{Q}$ be a chemical state such that $\bar{\beta}_{q^\sharp}^\mathcal{Q}(q) = q^\sharp$.

$$\begin{aligned} \bar{\gamma}_{q^\sharp}^{T^\sharp}(Y) &= \{(q, l, q') \in T \mid \bar{\beta}_{q^\sharp}^T(q, l, q') \in Y\} \\ &= \{(q, l, q') \in T \mid \exists q^{\sharp\prime\prime}, q^{\sharp\prime} \in \bar{\mathcal{Q}}^\sharp : (q^{\sharp\prime\prime}, l, q^{\sharp\prime}) \in Y \wedge \bar{\beta}_{q^\sharp}^T(q, l, q') = (q^{\sharp\prime\prime}, l, q^{\sharp\prime})\} \\ &= \{(q, l, q') \in T \mid \exists q^{\sharp\prime} \in \bar{\mathcal{Q}}^\sharp : (q^\sharp, l, q^{\sharp\prime}) \in Y \wedge \bar{\beta}_{q^\sharp}^T(q, l, q') = (q^\sharp, l, q^{\sharp\prime})\} \\ &= \{(q, l, q') \in T \mid \exists q^{\sharp\prime} \in \bar{\mathcal{Q}}^\sharp : (q^\sharp, l, q^{\sharp\prime}) \in Y \wedge \bar{\beta}_{q^\sharp}^\mathcal{Q}(q) = q^\sharp \wedge \bar{\beta}_{q^\sharp}^\mathcal{Q}(q') = q^{\sharp\prime}\} \\ &= \{(q, l, q') \in T \mid \exists q^{\sharp\prime} \in \bar{\mathcal{Q}}^\sharp : (q^\sharp, l, q^{\sharp\prime}) \in Y \wedge q \in \bar{\gamma}_{q^\sharp}^\mathcal{Q}(\{q^\sharp\}) \wedge q' \in \bar{\gamma}_{q^\sharp}^\mathcal{Q}(\{q^{\sharp\prime}\})\} \\ &= \bigcup_{q^{\sharp\prime} \in \bar{\mathcal{Q}}^\sharp} \{(q, l, q') \in T \mid (q^\sharp, l, q^{\sharp\prime}) \in Y \wedge q \in \bar{\gamma}_{q^\sharp}^\mathcal{Q}(\{q^\sharp\}) \wedge q' \in \bar{\gamma}_{q^\sharp}^\mathcal{Q}(\{q^{\sharp\prime}\})\}. \end{aligned}$$

□

For the two Galois connection functions belonging to the chemical transition and macro-transition domains, we did not provide clear examples. The reason is because the abstraction process $\bar{\gamma}_{q^\sharp}^{T^\sharp} \circ \bar{\alpha}_{q^\sharp}^T$ can result in an intractable number of chemical transitions arising from traversing to the macro-transition domain and then back to the chemical transition domain. More

specifically, this abstraction process first generates, from an initial set of chemical transitions, a set of macro-transitions using the expression in Prop. 5.7. Consequently, the resulting set contains macro-transitions each denoting a potential transition from one region of chemical states to another. Afterwhich, we plug this abstract set into Prop. 5.8, which generates the corresponding collection of sets of chemical transitions such that each chemical transition is a result of piecing together the source (resp. target) chemical states related by a reaction type and whose species values are delimited by the source (resp. target) regions of each macro-transitions in the abstract set. That is, if there exists a reaction type that relates a value in the source region to the target region, then this chemical transition is recovered. Thus, one must consider such a relation for each chemical species involved in a reaction network.

In this Section, we introduced the notion of macro-transitions, and showed how this abstract object can be derived from chemical transitions using the chemical transition abstraction function. Afterwhich, we derived a Galois connection between the macro-transition and chemical transition domains. In Sect. 5.5 we will characterize the final abstract element we will consider for reaction networks: macro-traces. Macro-traces are the result of piecing together each abstract objects derived up to this point, and describe the abstract behaviors of a macro-transition system.

5.5 Characterizing behaviors for a macro-transition system

The final abstract object we will derive are macro-traces. Macro-traces correspond to the abstract behaviors that can be generated from reaction networks, thus this abstract element shares a structure similar its concrete counterpart: chemical traces (see Def. 2.6); and, are defined as follows.

Definition 5.15 (Macro-traces). *Let k be an element in \mathbb{N} . A macro-trace of size k is an element $(q_0^\sharp, (q_i^\sharp, l_i, q_i^{\sharp'})_{1 \leq i \leq k}) \in \overline{\mathcal{Q}}^\sharp \times T^{\sharp k}$ such that $q_i^\sharp = q_{i-1}^{\sharp'}$ for every integer i such that $1 \leq i \leq k$.*

The set of all macro-traces is denoted $\overline{\mathcal{T}}^\sharp$.

In Def. 5.15, a macro-trace carries two components. The first component describes an initial macro-state. An initial macro-state contains all the initial interval values for each chemical species in a reaction network. The second component describes sequences of macro-transitions that follow from an initial macro-state. Note that the property $q_i^\sharp = q_{i-1}^{\sharp'}$ can also be written as $pre(t_i^\sharp) = post(t_{i-1}^\sharp)$ for each macro-transition t_i^\sharp belong to a macro-trace. Here, a single macro-trace describes an abstract behavior for a macro-transition system. All the abstract behaviors that can be obtained from an initial macro-state is contained in $\overline{\mathcal{T}}^\sharp$. A main distinction to chemical traces arises from the fact that macro-traces carry no information about probabilities. To obtain macro-traces, we can abstract chemical ones.

Definition 5.16 (Chemical trace abstraction function). *Let $q^\sharp \in \overline{\mathcal{Q}}^\sharp$ be a reference macro-state.*

The chemical trace abstraction function is defined as:

$$\overline{\beta}_{q^\sharp}^{\mathcal{T}} = \begin{cases} \mathcal{T} & \mapsto \overline{\mathcal{T}}^\sharp \\ (q'_0, ((q_i, l_i, q'_i), \mu_i)_{1 \leq i \leq k}) & \mapsto (q_0^\sharp, (q_i^\sharp, l'_i, q_i^{\sharp'})_{1 \leq i \leq k}) \end{cases}$$

maps each chemical trace $(q'_0, ((q_i, l_i, q'_i), \mu_i)_{1 \leq i \leq k})$ to the macro-trace that is defined inductively as follows:

1. $\overline{\beta}_{q^\sharp}^{\mathcal{T}}(q'_0, ()) = (\overline{\beta}_{q^\sharp}^{\mathcal{Q}}(q_0^{\sharp'}), ());$

2. By induction,

$$\overline{\beta}_{q^\sharp}^{\mathcal{T}}(q'_0, ((q_i, l_i, q'_i), \mu_i)_{1 \leq i \leq k}) = (q_0^{\sharp'}, (q_i^\sharp, l'_i, q_i^{\sharp'})_{1 \leq i < k}) \widehat{\ } \overline{\beta}_{q_{k-1}^{\sharp'}}^{\mathcal{T}}((q_k, l_k, q'_k), \mu_k).$$

where $(q_0^{\sharp'}, (q_i^\sharp, l'_i, q_i^{\sharp'})_{1 \leq i < k}) = \overline{\beta}_{q^\sharp}^{\mathcal{T}}(q'_0, (q_i, l_i, q'_i)_{1 \leq i < k})$.

In Def. 5.16, $q^\sharp \in \overline{\mathcal{Q}}^\sharp$ be a reference macro-state. The domain and co-domain of the function $\overline{\beta}_{q^\sharp}^{\mathcal{T}}$ are, respectively, the set of chemical traces and the set of macro-traces. The chemical trace abstraction function inputs a chemical trace and, from its initial chemical state, abstracts each chemical state along the sequence inductively. It does so accordingly. If the trace is composed of only an initial chemical state (Cond. 1), then this state will be set to its initial abstract state while taking into account the initial reference macro-state q^\sharp . Otherwise, if there is at least one transition in the second component of a chemical trace (Cond. 2), then the chemical transition abstraction function (see Def. 5.12) will zip through each transition in the input chemical trace and abstract inductively each chemical transition. Note that at each chemical transition abstraction, the reference macro-state is updated with the abstracted target macro-state of the previously abstracted chemical transition. This is the reason that the chemical trace abstraction function is not parameterized with a single macro-state. Additionally, this function construct allows one to better abstract chemical behaviors such as to hone in on to the directionality of the chemical behaviors to be abstracted as they can give us dynamic insights as to which particular regions of each chemical states habituate.

Example 5.8. Let $\mathcal{D}_1^\sharp = \{\llbracket 0, 5 \llbracket, \llbracket 5, 10 \llbracket, \llbracket 10, +\infty \llbracket\}$ be a set of non-overlapping intervals and $\mathcal{D}_2^\sharp = \{\llbracket 0, 6 \llbracket, \llbracket 2, 8 \llbracket, \llbracket 4, +\infty \llbracket\}$ be a set of overlapping intervals. We will show how the chemical trace abstraction function (see Def. 5.16) abstracts a chemical trace. Given the following chemical trace for the BD model discussed in Sect. 2.2:

$$3 \xrightarrow{0.87} 4 \xrightarrow{0.83} 5 \xrightarrow{0.20} 4 \xrightarrow{0.83} 5 \xrightarrow{0.20} 4 \xrightarrow{0.83} 5 \xrightarrow{0.80} 6.$$

With the choice of non-overlapping intervals, \mathcal{D}_1^\sharp , we obtain the following abstract trace:

$$\llbracket 0, 5 \llbracket \xrightarrow{r_1}^\sharp \llbracket 0, 5 \llbracket \xrightarrow{r_1}^\sharp \llbracket 5, 10 \llbracket \xrightarrow{r_2}^\sharp \llbracket 0, 5 \llbracket \xrightarrow{r_1}^\sharp \llbracket 5, 10 \llbracket \xrightarrow{r_2}^\sharp \llbracket 0, 5 \llbracket \xrightarrow{r_1}^\sharp \llbracket 5, 10 \llbracket \xrightarrow{r_1}^\sharp \llbracket 5, 10 \llbracket$$

whereas with the choice of overlapping intervals, \mathcal{D}_2^\sharp we obtain the following one (we set the initial reference macro-state as $\llbracket 0, 6 \rrbracket$):

$$\llbracket 0, 6 \rrbracket \xrightarrow{r_1^\sharp} \llbracket 0, 6 \rrbracket \xrightarrow{r_1^\sharp} \llbracket 0, 6 \rrbracket \xrightarrow{r_2^\sharp} \llbracket 0, 6 \rrbracket \xrightarrow{r_1^\sharp} \llbracket 0, 6 \rrbracket \xrightarrow{r_2^\sharp} \llbracket 0, 6 \rrbracket \xrightarrow{r_1^\sharp} \llbracket 0, 6 \rrbracket \xrightarrow{r_1^\sharp} \llbracket 2, 8 \rrbracket.$$

We notice that with the second choice of intervals, the system remains in initial macro-state almost all the trace, except for the last transition. By emphasizing the effort to go back and forth between neighboring intervals, the abstraction has abstracted away the oscillations near the interval borders. This is not the case with the first choice of intervals. Please observe that no chemical behavior has been lost in the process. Indeed, each choice of sampling intervals comes with a different interpretation of macro-traces, which may highlight or hide some information accordingly.

In our framework, a macro-trace corresponds to sequences of macro-transitions that begin from an initial macro-state. Namely, underlying each macro-trace are several possible chemical traces that one can use to pinpoint some general, behavioral trend for a given reaction network. Also, thanks to the structure of the chemical trace abstraction function (see Def. 5.16), we abstract chemical traces inductively and ensure that each chemical state abstraction has been carefully parameterized by a respective reference macro-state at each step along the sequence. Thus, our macro-traces carry information regarding the range of quantities each chemical species will exhibit as a consequence to the reaction dynamics of a system. In the logical counterpart, a sequence from the transition state space carries information regarding the activity evolution of some number of chemical variables. Indeed, our macro-traces and logical traces share similarly in that they capture sequences of events and where time has been discretized. Yet, our macro-traces carry additional information since in a sequence a transition may proceed in two directions, rather than in one direction observed in logical traces. Additionally, the long-term dynamics that are observed in our macro-traces are governed by the steady state dynamics computed from a reaction network. For example, if a macro-trace stabilizes onto an interval that contains a steady state value, then we can be more confident that the macro-trace dynamics converged. In the logical case, a logical trace converging to an attractor point would have to be confirmed by expert knowledge and should relay a phenotypic observation resulting from the combination of different chemical species activity levels.

At this point, we will begin to derive a Galois connection between the macro-traces and chemical traces domain. In the following definition we derive the chemical trace set abstraction function which is responsible for generating sets of macro-traces from sets of chemical traces.

Definition 5.17 (Chemical trace set abstraction). *Let $q^\sharp \in \overline{\mathcal{Q}}^\sharp$ be a reference macro-state. The chemical trace set abstraction function maps a set of chemical traces to the set of macro-transitions:*

$$\overline{\alpha}_{q^\sharp}^{\mathcal{T}} = \begin{cases} \wp(\mathcal{T}) & \rightarrow & \wp(\overline{\mathcal{T}}^\sharp) \\ X & \mapsto & \{ \overline{\beta}_{q^\sharp}^{\mathcal{T}}(\tau) \in \overline{\mathcal{T}}^\sharp \mid \tau \in X \}. \end{cases}$$

In Def. 5.17, $q^\sharp \in \overline{\mathcal{Q}}^\sharp$ is a reference macro-state. The domain and co-domain of the function $\overline{\alpha}_{q^\sharp}^{\mathcal{T}}$ are, respectively, the powerset of the set of chemical traces and the powerset of the set of macro-traces. The chemical trace set abstraction function inputs a set of chemical traces and outputs those macro-traces obtained by lifting the chemical trace abstraction function of Def. 5.16 onto each chemical trace in the input set.

Accordingly, the following proposition explicitly abstracts a given set of chemical traces to a set of macro-traces composed of macro-traces that are obtained by ensuring that each component of each macro-trace (see Def. 5.15) belong to the abstracted concrete sets containing each corresponding chemical trace component counterpart, those of which have been abstracted by their respective abstraction functions.

Proposition 5.9. *Let q^\sharp be a reference macro-state. Let $X \in \wp(\mathcal{T})$ be a set of chemical traces.*

We introduce X_0 as the set of the chemical traces in X of size 0 and $X_{\geq 1}$ as the set of the chemical traces in X of size greater or equal to 1.

Then:

$$\overline{\alpha}_{q^\sharp}^{\mathcal{T}}(X) = Y_0 \cup Y_{\geq 1}.$$

where:

$$\begin{aligned} \bullet Y_0 &= \left\{ (q'_0, ()) \mid q'_0 \in \overline{\alpha}_{q^\sharp}^{\mathcal{Q}}(\{q'_0 \in \mathcal{Q} \mid (q'_0, ()) \in X_0\}) \right\}; \\ \bullet Y_{\geq 1} &= \left\{ (q_0^\sharp, (q_i^\sharp, l_i, q_i^{\sharp'})_{1 \leq i \leq k}) \in \overline{\mathcal{T}}^\sharp \mid \exists (q'_0, (t_i, \mu_i)_{1 \leq i \leq k}) \in X_{\geq 0} : \right. \\ &\quad \left. (q_0^\sharp, (q_i^\sharp, l_i, q_i^{\sharp'})_{1 \leq i < k}) \in \overline{\alpha}_{q^\sharp}^{\mathcal{T}}(\{(q'_0, (t_i)_{1 \leq i < k})\}) \wedge (q_k^\sharp, l_k, q_k^{\sharp'}) \in \overline{\alpha}_{q_{k-1}^{\sharp'}}^{\mathcal{T}}(\{t_k\}) \right\}. \end{aligned}$$

Proof. Let q^\sharp be a reference macro-state. Let $X \in \wp(\mathcal{T})$ be a set of chemical traces.

We introduce X_0 as the set of the chemical traces in X of size 0 and $X_{\geq 1}$ as the set of the chemical traces in X of size greater or equal to 1.

We have: $X = X_0 \cup X_{\geq 0}$.

Then: $\overline{\alpha}_{q^\sharp}^{\mathcal{T}}(X) = \overline{\alpha}_{q^\sharp}^{\mathcal{T}}(X_0) \cup \overline{\alpha}_{q^\sharp}^{\mathcal{T}}(X_{\geq 1})$.

1. Let us prove that $\overline{\alpha}_{q^\sharp}^{\mathcal{T}}(X_0) = \left\{ (q'_0, ()) \mid q'_0 \in \overline{\alpha}_{q^\sharp}^{\mathcal{Q}}(\{q'_0 \in \mathcal{Q} \mid (q'_0, ()) \in X_0\}) \right\}$.

We have:

$$\begin{aligned} \overline{\alpha}_{q^\sharp}^{\mathcal{T}}(X_0) &= \left\{ \overline{\beta}_{q^\sharp}^{\mathcal{T}}(q'_0, (t_i, \mu_i)_{1 \leq i \leq k}) \in \overline{\mathcal{T}}^\sharp \mid (q'_0, (t_i, \mu_i)_{1 \leq i \leq k}) \in X_0 \right\} \\ &= \left\{ \overline{\beta}_{q^\sharp}^{\mathcal{T}}(q'_0, ()) \in \overline{\mathcal{T}}^\sharp \mid (q'_0, ()) \in X_0 \right\} \\ &= \left\{ (\overline{\beta}_{q^\sharp}^{\mathcal{Q}}(q'_0, (t_i, \mu_i)_{1 \leq i \leq k})) \in \overline{\mathcal{T}}^\sharp \mid (q'_0, (t_i, \mu_i)_{1 \leq i \leq k}) \in X_0 \right\} \\ &= \left\{ (q'_0, ()) \mid q'_0 \in \overline{\alpha}_{q^\sharp}^{\mathcal{Q}}(\{q'_0 \in \mathcal{Q} \mid (q'_0, ()) \in X_0\}) \right\}; \end{aligned}$$

2. Let us prove that:

$$\begin{aligned} \overline{\alpha}_{q^\sharp}^{\mathcal{T}}(X_{\geq 0}) &= \left\{ (q_0^\sharp, (q_i^\sharp, l_i, q_i^{\sharp'})_{1 \leq i \leq k}) \in \overline{\mathcal{T}}^\sharp \mid \exists (q'_0, (t_i, \mu_i)_{1 \leq i \leq k}) \in X_{\geq 0} : \right. \\ &\quad \left. (q_0^\sharp, (q_i^\sharp, l_i, q_i^{\sharp'})_{1 \leq i < k}) \in \overline{\alpha}_{q^\sharp}^{\mathcal{T}}(\{(q'_0, (t_i)_{1 \leq i < k})\}) \wedge (q_k^\sharp, l_k, q_k^{\sharp'}) \in \overline{\alpha}_{q_{k-1}^{\sharp'}}^{\mathcal{T}}(\{t_k\}) \right\}. \end{aligned}$$

We have:

$$\begin{aligned}
 \bar{\alpha}_{q^\sharp}^{\mathcal{T}}(X_{\geq 0}) &= \left\{ \bar{\beta}_{q^\sharp}^{\mathcal{T}}(q'_0, (t_i, \mu_i)_{1 \leq i \leq k}) \in \bar{\mathcal{T}}^\sharp \mid (q'_0, (t_i, \mu_i)_{1 \leq i \leq k}) \in X_{\geq 0} \right\} \\
 &= \left\{ (q_0^{\sharp'}, (t_i^\sharp)_{1 \leq i \leq k}) \in \bar{\mathcal{T}}^\sharp \mid \exists (q'_0, (t_i, \mu_i)_{1 \leq i \leq k}) \in X_{\geq 0} : \right. \\
 &\quad \left. \bar{\beta}_{q^\sharp}^{\mathcal{T}}(q'_0, (t_i, \mu_i)_{1 \leq i \leq k}) = (q_0^{\sharp'}, (t_i^\sharp)_{1 \leq i \leq k}) \right\} \\
 &= \left\{ (q_0^{\sharp'}, (q_i^\sharp, l_i, q_i^{\sharp'})_{1 \leq i \leq k}) \in \bar{\mathcal{T}}^\sharp \mid \exists (q'_0, (t_i, \mu_i)_{1 \leq i \leq k}) \in X_{\geq 0}, \right. \\
 &\quad \left. \begin{aligned}
 (q_0^{\sharp'}, (q_i^\sharp, l_i, q_i^{\sharp'})_{1 \leq i < k}) &= \bar{\beta}_{q^\sharp}^{\mathcal{T}}(q'_0, (t_i)_{1 \leq i < k}) \\
 \wedge (q_k^\sharp, l_k, q_k^{\sharp'}) &= \bar{\beta}_{q_{k-1}^{\sharp'}}^{\mathcal{T}}(t_k) \end{aligned} \right\}. \\
 &= \left\{ (q_0^{\sharp'}, (q_i^\sharp, l_i, q_i^{\sharp'})_{1 \leq i \leq k}) \in \bar{\mathcal{T}}^\sharp \mid \exists (q'_0, (t_i, \mu_i)_{1 \leq i \leq k}) \in X_{\geq 0} : \right. \\
 &\quad \left. \begin{aligned}
 (q_0^{\sharp'}, (q_i^\sharp, l_i, q_i^{\sharp'})_{1 \leq i < k}) &\in \bar{\alpha}_{q^\sharp}^{\mathcal{T}}(\{(q'_0, (t_i)_{1 \leq i < k})\}) \\
 \wedge (q_k^\sharp, l_k, q_k^{\sharp'}) &\in \bar{\alpha}_{q_{k-1}^{\sharp'}}^{\mathcal{T}}(\{t_k\}) \end{aligned} \right\}.
 \end{aligned}$$

□

The abstraction of sets of chemical traces is straightforward. Given an input set of chemical traces, the specific chemical trace abstraction function in Prop. 5.9 proceeds by induction. The traces of size 0 are abstracted by the abstraction of the set of their initial states by Prop. 5.5. Then each trace of size greater or equal to 1 is abstracted by abstracting inductively its prefix, while its final transition is abstracted by Prop. 5.7 with the final state of the prefix as reference macro-state. Note that we keep the relation between prefixes and final transitions.

Now we define the macro-trace set concretization function to concretize sets of macro-traces to sets of chemical traces.

Definition 5.18 (Macro-trace set concretization). *Let q^\sharp be a reference macro-state. The macro-trace concretization function maps a set of macro-traces to the set of chemical traces:*

$$\bar{\gamma}_{q^\sharp}^{\mathcal{T}} = \begin{cases} \wp(\bar{\mathcal{T}}^\sharp) & \rightarrow & \wp(\mathcal{T}) \\ Y & \mapsto & \{ \tau \in \mathcal{T} \mid \bar{\beta}_{q^\sharp}^{\mathcal{T}}(\tau) \in Y \}. \end{cases}$$

In Def. 5.18, q^\sharp is a reference macro-state. The domain and co-domain of the function $\bar{\gamma}_{q^\sharp}^{\mathcal{T}}$ are, respectively, the powerset of the set of macro-traces and the powerset of the set of chemical traces. The macro-trace set concretization function inputs a set of macro-traces and outputs the set of chemical traces such that each chemical trace in the output set has the property that its abstraction by the chemical trace abstraction function (see Def. 5.16) belongs to the input set.

In the following proposition, concretization of a given set of macro-traces breaks down to piecing together chemical traces from their constituents, those of which that can be constructed by pieces obtained from the concretization of the abstract sets, and using the appropriate functions, composed each of the abstract components of each macro-trace in the given set.

Proposition 5.10. *Let $q^\sharp \in \overline{\mathcal{Q}}^\sharp$ be a reference macro-state. Let $Y \in \wp(\overline{\mathcal{T}}^\sharp)$ be a set of macro-traces. Let $q_0^\sharp, q^\sharp \in \overline{\mathcal{Q}}^\sharp$ be two reference macro-states.*

$$\begin{aligned} \overline{\gamma}_{q^\sharp}^{\mathcal{T}^\sharp}(Y) &= \left\{ (q'_0, ()) \in \mathcal{T} \mid \exists q_0^\sharp \in \overline{\mathcal{Q}}^\sharp : (q_0^\sharp, ()) \in Y \wedge q'_0 \in \overline{\gamma}_{q_0^\sharp}^{\mathcal{Q}^\sharp} \left(\{q_0^\sharp\} \right) \right\} \\ &\cup \left\{ (q'_0, (t_i)_{1 \leq i < k}) \frown t_k \in \mathcal{T} \mid \exists (q_i^\sharp, l_i, q_i^{\sharp'})_{1 \leq i < k} \in \overline{\mathcal{T}}^\sharp, t_k^\sharp \in \overline{\mathcal{T}}^\sharp : \right. \\ &\quad (q'_0, ((q_i^\sharp, l_i, q_i^{\sharp'}))_{1 \leq i < k}) \frown t_k^\sharp \in Y \wedge t_k \in \overline{\gamma}_{q_{k-1}^{\sharp'}}^{\mathcal{T}^\sharp} \left(\{t_k^\sharp\} \right) \\ &\quad \left. \wedge (q'_0, (t_i)_{1 \leq i < k}) \in \overline{\gamma}_{q^\sharp}^{\mathcal{T}^\sharp} \left(\{ (q_0^\sharp, ((q_i^\sharp, l_i, q_i^{\sharp'}))_{1 \leq i < k}) \} \right) \right\}. \end{aligned}$$

Proof. Let $q^\sharp \in \overline{\mathcal{Q}}^\sharp$ be a reference macro-state. Let $Y \in \wp(\overline{\mathcal{T}}^\sharp)$ be a set of macro-traces.

$$\begin{aligned} \overline{\gamma}_{q^\sharp}^{\mathcal{T}^\sharp}(Y) &= \left\{ (q'_0, (t_i)_{1 \leq i \leq k}) \in \mathcal{T} \mid \overline{\beta}_{q^\sharp}^{\mathcal{T}}(q'_0, (t_i)_{1 \leq i \leq k}) \in Y \right\} \\ &= \left\{ (q'_0, (t_i)_{1 \leq i \leq k}) \in \mathcal{T} \mid \exists (q_0^{\sharp'}, (t_i^\sharp)_{1 \leq i \leq k}) \in Y : \right. \\ &\quad \left. \overline{\beta}_{q^\sharp}^{\mathcal{T}}(q'_0, (t_i)_{1 \leq i \leq k}) = (q_0^{\sharp'}, (t_i^\sharp)_{1 \leq i \leq k}) \right\} \\ &= \left\{ (q'_0, ()) \in \mathcal{T} \mid \exists q_0^\sharp \in \overline{\mathcal{Q}}^\sharp : (q_0^\sharp, ()) \in Y \wedge q'_0 = \overline{\beta}_{q^\sharp}^{\mathcal{Q}}(q_0^\sharp) \right\} \\ &\cup \left\{ (q'_0, (t_i)_{1 \leq i < k}) \frown t_k \in \mathcal{T} \mid \exists (q_0^{\sharp'}, ((q_i^\sharp, l_i, q_i^{\sharp'}))_{1 \leq i < k}) \in \overline{\mathcal{T}}^\sharp, t_k^\sharp \in \overline{\mathcal{T}}^\sharp : \right. \\ &\quad (q_0^{\sharp'}, ((q_i^\sharp, l_i, q_i^{\sharp'}))_{1 \leq i < k}) \frown t_k^\sharp \in Y \wedge \overline{\beta}_{q_{k-1}^{\sharp'}}^{\mathcal{T}}(t_k) = t_k^\sharp \\ &\quad \left. \wedge \overline{\beta}_{q^\sharp}^{\mathcal{T}}(q'_0, (t_i)_{1 \leq i < k}) = (q_0^{\sharp'}, (q_i^\sharp, l_i, q_i^{\sharp'}))_{1 \leq i < k} \right\} \\ &= \left\{ (q'_0, ()) \in \mathcal{T} \mid \exists q_0^\sharp \in \overline{\mathcal{Q}}^\sharp : (q_0^\sharp, ()) \in Y \wedge q'_0 \in \overline{\gamma}_{q_0^\sharp}^{\mathcal{Q}^\sharp} \left(\{q_0^\sharp\} \right) \right\} \\ &\cup \left\{ (q'_0, (t_i)_{1 \leq i < k}) \frown t_k \in \mathcal{T} \mid \exists (q_0^{\sharp'}, ((q_i^\sharp, l_i, q_i^{\sharp'}))_{1 \leq i < k}) \in \overline{\mathcal{T}}^\sharp, t_k^\sharp \in \overline{\mathcal{T}}^\sharp : \right. \\ &\quad (q_0^{\sharp'}, ((q_i^\sharp, l_i, q_i^{\sharp'}))_{1 \leq i < k}) \frown t_k^\sharp \in Y \wedge t_k \in \overline{\gamma}_{q_{k-1}^{\sharp'}}^{\mathcal{T}^\sharp} \left(\{t_k^\sharp\} \right) \\ &\quad \left. \wedge (q'_0, (t_i)_{1 \leq i < k}) \in \overline{\gamma}_{q^\sharp}^{\mathcal{T}^\sharp} \left(\{ (q_0^{\sharp'}, ((q_i^\sharp, l_i, q_i^{\sharp'}))_{1 \leq i < k}) \} \right) \right\}. \end{aligned}$$

□

The specific macro-trace set concretization function derived in Prop. 5.10 considers two cases. The first case corresponds to when a macro-trace inside the set to be concretized is constituted of a single macro-state (e.g., the second component of the macro-trace is empty). Thus, concretizing this abstract set gives the concrete set containing chemical traces of length one, specifically those traces that contain in its first component the of initial chemical states stemming from the initial macro-state of the macro-trace. The second case of the specific concretization function unwinds in a forward manner each macro-trace of at least length two in the prescribed abstract set and reconstructs from the initial macro-state, and for each macro-trace, those chemical traces that can be extracted. This is done by splitting the initial macro-transition of a macro-trace, and concretizing the set containing this macro-transition using Prop. 5.8 in order to obtain the potential set of chemical transitions. This process is repeated until the macro-trace terminal is reached, and along the way the set of chemical traces are reconstructed by appending those chemical transitions that are obtained throughout each iterative cycle.

Indeed, the abstraction process $\overline{\gamma}_{q^\sharp}^{\mathcal{T}^\sharp} \circ \overline{\alpha}_{q^\sharp}^{\mathcal{T}}$ traverses from the domain of chemical traces to the domain of macro-traces, and back. Consequently, and as one observed in Sect. 4.4, this process incurs a loss in accuracy since the analysis results in a final set of chemical traces that is

larger than an original set. Nevertheless, a Galois connection between macro-traces and chemical traces ensure that no chemical behaviors are lost, and that one can design a tailored abstract interpretation to highlight and pinpoint diverse chemical behaviors of interest.

In this Section, we have derived our notion of macro-traces for the macro-transition system of Sect. 5.4, which pertain to the abstract behaviors of a reaction network. Additionally, we show how we can obtain macro-traces from chemical traces and also discussed how this process abstracts away information about probabilities associated to each chemical trace. Finally, we derive a Galois connection between the macro-traces and chemical traces domain, which allow us to recover chemical traces following the abstraction process. Yet, we do not recover probabilities. In Chapter 6, we detail how to recover these probabilities.

In Sect. 5.6, we will derive an approximation of the chemical collecting semantics in order to generate automatically the set of all chemical and abstract behaviors from a reaction network, similarly to what was done in Sect. 4.5. These results will rely heavily on the expressions derived in this Section in order to generate each respective collection of traces.

5.6 Approximation of the collecting semantics

In this Section, the objective is to derive a collection of operators in order to over-approximate the behaviors of a reaction network, in the case of overlapping intervals.

Firstly we introduce $\mathcal{Q}_{\mathcal{R},0} \subseteq \mathcal{Q}$ a set of potential initial states and $T' \subseteq T$ a set of potential transitions. We also denote as $\mathcal{T}_{\mathcal{Q}_{\mathcal{R},0}}$ the set of traces $\{(q, ()) \mid q \in \mathcal{Q}_{\mathcal{R},0}\}$, that is the set of the traces made of a single state and no transition. The collecting semantics $\mathcal{T}_{\mathcal{Q}_{\mathcal{R},0},T'}$ has been defined in Sect. 4.5.1.

We derive the macro-trace elongation operator that can be used to extend a macro-trace in a set by a macro-transition. We use macro-state which maps each variable to the first interval as reference. We denote it as 0^\sharp .

Definition 5.19 (Elongation of macro-traces). *The macro-trace elongation operator is defined as:*

$$\overline{\mathbb{F}}^\sharp: \begin{cases} \wp(\overline{\mathcal{T}}^\sharp) & \rightarrow & \wp(\overline{\mathcal{T}}^\sharp) \\ Y & \mapsto & \overline{\alpha}_{0^\sharp}^{\mathcal{T}} \left(\mathbb{F} \left(\overline{\gamma}_{0^\sharp}^{\mathcal{T}}(Y) \right) \right). \end{cases}$$

In Def. 5.19, the domain and co-domain of the function $\overline{\mathbb{F}}^\sharp$ are both the powerset of the set of macro-traces. The macro-trace elongation operator inputs a set of macro-traces and outputs an elongated abstract set of macro-traces in the following three steps. First, an input set of macro-traces is concretized by the macro-state set concretization function of Prop. 5.10. Thus, from each macro-trace in the input set, a collection of filtered sets of chemical traces will be extracted. Second, each chemical trace in the concrete set from the first step will be elongated thanks to chemical trace elongation operator in Def. 4.20. Third, and finally, the elongated set from the second step will be abstracted by chemical trace set abstraction function in Prop. 5.9. Thus, the output set of the macro-trace elongation operator will be an abstract set larger than

the input abstract set, since this process considers the abstraction of elongated chemical traces. An approximate version of Def. 5.19 is enclosed in the following proposition.

Proposition 5.11. *Let $Y \in \wp(\mathcal{T})$ be a set of macro-traces. We have:*

$$\overline{\mathbb{F}}^\sharp \subseteq Y \cup \left\{ \tau^\sharp \frown (q^\sharp, l, q^\sharp) \in \overline{\mathcal{T}}^\sharp \mid \tau^\sharp \in Y \wedge (q^\sharp, l, q^\sharp) \in \overline{\alpha}_{q^\sharp}^T(T') \wedge \mathit{final}(\tau^\sharp) = q^\sharp \right\}.$$

Proof. Let $Y \in \wp(\overline{\mathcal{T}}^\sharp)$.

$$\begin{aligned} \overline{\mathbb{F}}^\sharp(Y) &= \overline{\alpha}_{0^\sharp}^T(\mathbb{F}(\overline{\gamma}_{0^\sharp}^T(Y))) \\ &= \overline{\alpha}_{0^\sharp}^T \left(\overline{\gamma}_{0^\sharp}^T(Y) \cup \left\{ \tau \frown (q, l, q') \in \mathcal{T} \mid \tau \in \overline{\gamma}_{0^\sharp}^T(Y) \wedge (q, l, q') \in T' \wedge \mathit{final}(\tau) = q \right\} \right) \\ &= \overline{\alpha}_{0^\sharp}^T \left(\overline{\gamma}_{0^\sharp}^T(Y) \right) \cup \overline{\alpha}_{0^\sharp}^T \left(\left\{ \tau \frown (q, l, q') \in \mathcal{T} \mid \tau \in \overline{\gamma}_{0^\sharp}^T(Y) \wedge (q, l, q') \in T' \wedge \mathit{final}(\tau) = q \right\} \right) \\ &= \overline{\alpha}_{0^\sharp}^T \left(\overline{\gamma}_{0^\sharp}^T(Y) \right) \cup \left\{ \overline{\beta}_{0^\sharp}^T(\tau \frown (q, l, q')) \in \overline{\mathcal{T}}^\sharp \mid \tau \in \overline{\gamma}_{0^\sharp}^T(Y) \wedge (q, l, q') \in T' \wedge \mathit{final}(\tau) = q \right\} \\ &= \overline{\alpha}_{0^\sharp}^T \left(\overline{\gamma}_{0^\sharp}^T(Y) \right) \cup \left\{ \overline{\beta}_{0^\sharp}^T(\tau) \frown \overline{\beta}_{\mathit{last}(\overline{\beta}_{0^\sharp}^T(\tau))}^T(q, l, q') \in \overline{\mathcal{T}}^\sharp \mid \tau \in \overline{\gamma}_{0^\sharp}^T(Y) \wedge (q, l, q') \in T' \wedge \mathit{final}(\tau) = q \right\} \\ &= \overline{\alpha}_{0^\sharp}^T \left(\overline{\gamma}_{0^\sharp}^T(Y) \right) \cup \left\{ \tau^\sharp \frown (q^\sharp, l, q^\sharp) \in \overline{\mathcal{T}}^\sharp \mid \exists \tau \in \overline{\gamma}_{0^\sharp}^T(Y), \exists (q, l, q') \in T' : \right. \\ &\quad \left. \overline{\beta}_{0^\sharp}^T(\tau) = \tau^\sharp \wedge \overline{\beta}_{q^\sharp}^T(q, l, q') = (q^\sharp, l, q^\sharp) \wedge \mathit{final}(\tau^\sharp) = q^\sharp \right\} \\ &= \overline{\alpha}_{0^\sharp}^T \left(\overline{\gamma}_{0^\sharp}^T(Y) \right) \cup \left\{ \tau^\sharp \frown (q^\sharp, l, q^\sharp) \in \overline{\mathcal{T}}^\sharp \mid \exists \tau \in \overline{\gamma}_{0^\sharp}^T(Y), \exists (q, l, q') \in T' : \right. \\ &\quad \left. \tau^\sharp \in \overline{\alpha}_{0^\sharp}^T(\{\tau\}) \wedge (q^\sharp, l, q^\sharp) \in \overline{\alpha}_{q^\sharp}^T(\{(q, l, q')\}) \wedge \mathit{final}(\tau^\sharp) = q^\sharp \right\}. \\ &= \overline{\alpha}_{0^\sharp}^T \left(\overline{\gamma}_{0^\sharp}^T(Y) \right) \cup \left\{ \tau^\sharp \frown (q^\sharp, l, q^\sharp) \in \overline{\mathcal{T}}^\sharp \mid \tau^\sharp \in \overline{\alpha}_{0^\sharp}^T \left(\overline{\gamma}_{0^\sharp}^T(Y) \right) \wedge (q^\sharp, l, q^\sharp) \in \overline{\alpha}_{q^\sharp}^T(T') \wedge \mathit{final}(\tau^\sharp) = q^\sharp \right\} \\ &\subseteq Y \cup \left\{ \tau^\sharp \frown (q^\sharp, l, q^\sharp) \in \overline{\mathcal{T}}^\sharp \mid \tau^\sharp \in Y \wedge (q^\sharp, l, q^\sharp) \in \overline{\alpha}_{q^\sharp}^T(T') \wedge \mathit{final}(\tau^\sharp) = q^\sharp \right\}. \end{aligned}$$

□

In the following, we denote $\overline{\mathbb{F}}^{\sharp,1}$ the function mapping every set of macro-traces $Y \subseteq \overline{\mathcal{T}}^\sharp$ into the set of macro-traces:

$$Y \cup \left\{ \tau^\sharp \frown (q^\sharp, l, q^\sharp) \in \overline{\mathcal{T}}^\sharp \mid \tau^\sharp \in Y \wedge (q^\sharp, l, q^\sharp) \in \overline{\alpha}_{q^\sharp}^T(T') \wedge \mathit{final}(\tau^\sharp) = q^\sharp \right\}.$$

Note that, by Prop. 5.11, the function $\overline{\mathbb{F}}^{\sharp,1}$ is an over-approximation of the function $\overline{\mathbb{F}}^\sharp$.

Similar to the chemical trace elongation operator, we can define the collection of all abstract behaviors for a macro-transition system using Tarski and Kleene results as, respectively:

$$\mathcal{T}_{\mathcal{Q}_{\mathcal{R},0},T'}^\sharp = \bigcap \left\{ Y \in \wp(\overline{\mathcal{T}}^\sharp) \mid \overline{\mathbb{F}}^{\sharp,1}(Y) \subseteq Y \wedge \overline{\alpha}_{0^\sharp}^T(\mathcal{T}_{\mathcal{Q}_{\mathcal{R},0}}) \subseteq Y \right\}. \quad (5.9)$$

and:

$$\mathcal{T}_{\mathcal{Q}_{\mathcal{R},0},T'}^\sharp = \bigcup_{n \in \mathbb{N}} \left\{ \left[\overline{\mathbb{F}}^{\sharp,1} \right]^n \left(\overline{\alpha}_{0^\sharp}^T(\mathcal{T}_{\mathcal{Q}_{\mathcal{R},0}}) \right) \right\}. \quad (5.10)$$

Thus, we have two strategies for computing the set of macro-traces from a reaction network, which we refer to as the *macro-behavior collecting semantics*. The set of macro-behaviors is by construction an over-approximation of the set of chemical behaviors, that is to say $\mathcal{T} \subseteq \overline{\gamma}_{0^\sharp}^T(\overline{\mathcal{T}}^\sharp)$.

At this point, we derived the macro-behavior collecting semantics, those which allow us to over-approximate the set of chemical traces for a reaction network. This is done by taking advantage of the fact that the domains of each of our operators are powersets, and hence complete lattices (see Def. 3.13); and, invoking the results of Tarski and Kleene in order to extract invariants of interest from this mathematical structure. Additionally, for the macro-behavior collecting semantics we were able to unfold the computation of macro-trace elongations by taking advantage of the Galois connection we derived in Sect. 5.5.

In Sect. 5.7, we consider a case study of an instantiation of our formal framework of this Chapter to coarse-grain the reaction networks of the BD model and the resource competition system introduced in Chapter 2. In this scope, we hope assess a non-deterministic macro-transition system using our interval domain that uses overlapping intervals.

5.7 Non-deterministic macro-transition systems with minimal effort

In this Section we instantiate the formal framework derived in this Chapter on the BD model from Sect. 2.2 and the resource competition system from Sect. 2.3. As we discussed extensively, the abstraction process for a given set of chemical behaviors induces a non-deterministic macro-transition system that carries no probabilities. Thus, our goal for investigating the non-deterministic version of the aforementioned systems is two-folded: i) to highlight the structural similarity of a macro-transition system to that of a logical model's; and, ii) to discuss the reason and how one introduces a minimal effort system onto each reaction system investigated.

5.7.1 Revisiting the BD model II

In Fig. 5.5 we display the non-deterministic version of Fig. 2.5. That is, we lift the formal framework derived in this Chapter in order to generate the BD macro-transition system using intervals that overlap. Accordingly, in our diagram each rectangular box represents an interval corresponding to the respective ranges of values for the molecule A . Underlying each rectangle are

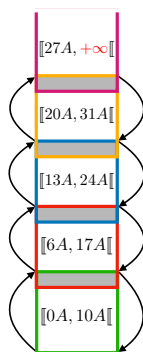


Figure 5.5: The non-deterministic version of Fig. 2.5 of a BD macro-transition system using overlapping interval parameters.

the chemical behaviors emerge from the BD model, and under the same set of assumptions made in Sect. 2.2. We note that the macro-transition system in Fig. 4.5 is induced by sampling various chemical behaviors from this reaction network, and lifting the chemical trace set abstract function (see Prop. 5.9) this set of chemical behaviors with the concrete trace set abstraction function (see Def. 4.18) with interval parameters $\mathcal{I}^\sharp = \{\llbracket 0, 10 \llbracket, \llbracket 6, 17 \llbracket, \llbracket 13, 24 \llbracket, \llbracket 20, 31 \llbracket, \llbracket 27, +\infty \llbracket$.

We notice that at this level of abstraction, since intervals do overlap then the coarse-graining may consider less oscillations near the interval borders. This, in retrospect, may reduce the number of fictitious behaviors into our analysis of this macro-transition system. Using overlapping intervals permit us to introduce a minimal effort system into our abstractions. Given the stochastic nature of the potential behaviors from a reaction network, then abstraction of these behaviors could highlight oscillations near interval borders for the values of a subset of chemical species involved in the reaction network. Indeed, these oscillations are a source that may contribute to some of the fictitious behaviors observed in Sect. 4.6.1 from the abstraction of the BD dynamics using intervals that do not overlap. Thus, one of the reasons we consider the overlapping intervals in Fig. 5.5 is to reduce the emphasis in our abstractions on these oscillatory behaviors. In this example, the effort system is depicted by the gray regions, and the size of each buffer region is four. As a consequence, the minimal effort system for the BD model is implemented to better pinpoint the trend underlying its dynamics. However, at this level of abstraction the BD macro-transition system is unlabeled, thus one cannot simply deduce the general direction of this system’s behaviors.

In Sect. 2.2, we derived the BD logical model in order to compare it to the formally derived one. Our goal, thus, is to use the formal model to assess the behaviors of the corresponding logical model. For example, in the logical BD model we observed that by following the assumption of René Thomas [36], the state transition system is unidirectional in its approach to the system’s focal point, corresponding to a stable interval. This, however, does not take into account the BD models ability to proceed bidirectionally, since this reaction system is composed of complementary reactions. In our macro-transition system in Fig. 5.5, we allow for reversible transitions between interval values of the molecule A . Additionally, using the intervals we characterized in Def. 5.1, we can tune these interval values to highlight regions of interest that may better approximate the underlying stochastic behaviors of a reaction network. Our non-deterministic abstractions lead to a macro-transition system that is more representative to the system it is abstracting, yet we are not able to deduce in a quantitative manner the dynamical trend of the underlying stochastic BD system. For example, we know which is steady state interval (since we computed the steady state value prior to the abstraction process, see Sect. 2.2), and could parse the macro-transition system and extract this interval value (which is $\llbracket 13, 24 \llbracket$) as one would do in the logical model counterpart. However, this would not be particularly insightful and, in light of a system with a larger number of variables, could be cumbersome.

5.7.2 Revisiting the Resource Competition System II

In Fig. 5.6 we display the non-deterministic version of Fig. 2.8. Each box depicts an abstract state and arises from the product of the respective interval set along each coordinate. Underlying each

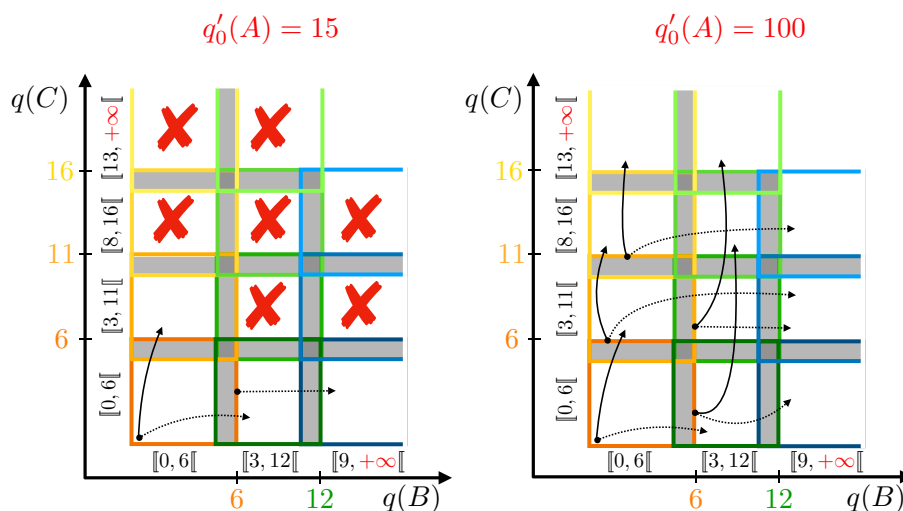


Figure 5.6: The non-deterministic version of Fig. 2.8 of a resource competition macro-transition system using overlapping interval parameters.

abstract state are the chemical behaviors that can emerge from the resource competition model, and under the same set of assumptions made in Sect. 2.3. Respectively, this macro-transition system is induced by sampling the chemical behaviors of the underlying reaction network, and abstracting by relational means the set containing these behaviors by the chemical trace set abstraction function (see Def. 5.17) using the interval parameters $\mathcal{I}_1^\sharp = \{\llbracket 0, 6 \rrbracket, \llbracket 3, 12 \rrbracket, \llbracket 9, +\infty \rrbracket\}$ for the molecule B and $\mathcal{I}_2^\sharp = \{\llbracket 0, 6 \rrbracket, \llbracket 3, 11 \rrbracket, \llbracket 8, 16 \rrbracket, \llbracket 13, +\infty \rrbracket\}$ for the molecule C . As a reminder, dashed arrow correspond to macro-transitions along the horizontal axis, while solid arrows are along the vertical axis. Additionally, the resource competition system compose reactions that increase the amount of molecules B and C , until the molecule A is depleted. Therefore, in the left diagram of Fig. 4.6, where resources are scarce, the red crosses correspond to unreachable regions. In the right scenario, resources are in abundance and thus more macro-transitions are a result. Thus, we obtain the same scenario as in Fig. 2.8 with the exception that we do not retain the likelihood of a macro-transition to occur prior to another.

The manner in which we consider a minimal effort system in the resource competition system differs from the BD system. In the latter system, we considered buffer regions between between two overlapping regions of values for the given variable, while in the former this effort policy is implemented by considering the subsequent region for an orthogonal macro-transition which has entered along the horizontal (resp. vertical) axis. This is done to take care of, for example, the values of each component that may be arbitrarily close to their upper bounds upon entering a new region that is of interest to an alternative component.

The structural similarities of the resource competition macro-transition system to its logical counterpart can be interpreted similar to what was done for the BD system in Sect. 5.7.1. That is, since our formal framework envelops the stochastic dynamics of the resource competition system, then we can say more about its reference reaction network representation than its logical model (see Sect. 2.3). Thus, we can fine-tune our abstractions, for example, to highlight the distinction between the initial availability of the resource pool, namely by the quantities of the molecule A .

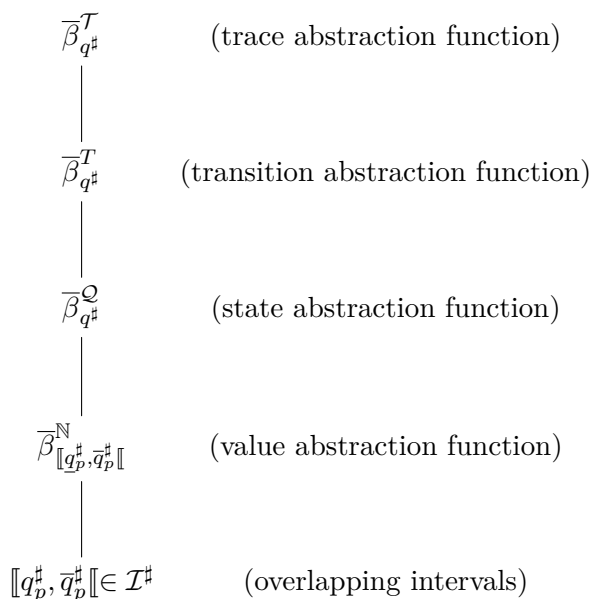


Figure 5.7: A hierarchy of the abstract functions derived in Chapter 5.

Additionally, we can impose constraints on this common resource to unmask the dynamics of the two variables modeled in the system, molecules B and C (see Sect. 2.3). Such a constraint would not be as straightforward to implement in the resource competition logical model.

5.8 Conclusion

To come into conclusion, in this Chapter we have managed to extend the classical interval introduced in Sect. 4 to include overlapping intervals. Our reason for this was to be able to gain flexibility in the ability to fine-tune interval abstractions to better approximate the behaviors of a reaction network of interest. Additionally, in order to abstract the quantities of variables we use reference intervals to guide the abstraction process. This construct was developed because, in light of overlapping intervals, it became possible for a value to be abstracted into several intervals. Thus, we wanted to ensure that the actual interval assignment of an abstracted value is the interval nearest to a given reference. Further, we have characterized interval assignments for chemical behaviors that contain quantities of variables that may traverse from one region of values to another (see Props. 5.1 and 5.2).

Consequently, we have derived a collection of abstract functions built on our new interval domain that allows us to traverse from an element containing chemical information to an abstract element containing its over-approximation (see Fig. 5.7). A subset of these functions are parameterized by either a reference interval or a macro-state. We use this information to pinpoint the evolution of the values of variables given by chemical behaviors, and ensure that such the reactions transforming their values are carefully abstracted by our framework. Additionally, our abstract functions allowed us to establish a Galois connection between each concrete and abstract element in this Chapter. Following several examples, we showed how one can use our formal framework to highlight certain behaviors belonging to stochastic reaction networks.

Further, it should now be clear that the abstractions from Chapter 4 and this Chapter lead to non-deterministic abstractions, since we lose information about the probabilities of the underlying system. In Chapter 6, we derive a formal scheme to re-introduce probabilities into our abstract systems.

Chapter 6

Chemical Goals: Propagating Probabilities to the Abstract Semantics

Objectives

In this Chapter, we formalize the notion of goals in order to re-introduce probabilities into the non-deterministic abstractions obtained in Chapters 4 and 5. We do so by carefully reasoning on the concrete domain, and propagating this information into our abstractions. Additionally, we provide results for a closed-form expression to compute such probabilities for a one-dimensional system, such as the BD model.

Contents

6.1	Relaying probabilities to macro-transitions	129
6.2	Case Study	134
6.2.1	Biased one-dimensional random walk with non-constant coefficients	134
6.2.2	BD model revisited	138
6.3	Conclusion	140

We discovered in Chapter 5 that abstracting a stochastic transition system results in a non-deterministic abstract transition system. Consequently, we obtain a structure that is similar to those transition systems observed in logical models; however, at this level of abstraction we are not able to deduce the trend of the underlying stochastic system. Thus, in this Chapter we will re-introduce probabilities into the abstract transition system obtained in Chapter 5 by reasoning on the behaviors that can emerge from a corresponding reaction network. We accomplish this task by introducing notions of target regions, which we often refer to as chemical goals that can be achieved by the quantities of chemical species via chemical behaviors.

6.1 Relaying probabilities to macro-transitions

In this Section, we refine the abstract semantics for reaction networks with some quantitative information to compare the likelihood of macro-transitions. This basically means that knowing that the system has just entered a new macro-state, and a given pair of potential macro-transitions, we would like to know with which probability will a macro-transition (in the pair) occur prior to the other. In order to compute these probabilities, we relate closely the abstract and chemical semantics of Chapter 5 by bounding the quantities of chemical species with intervals and then computing the probability of reaching a boundary value and exiting this value prior to another when starting from a position inside of the prescribed interval.

By construction of our abstract domain, macro-transitions are triggered when a given chemical species reaches a particular copy number. Accordingly, we introduce the notion target regions.

Definition 6.1 (Target region). *A target region is a set of chemical states of the form*

$$\{q \in \mathcal{Q}_{\mathcal{R}} \mid q(v) \sqsubseteq b\},$$

where v is a chemical species in $\mathcal{V}_{\mathcal{R}}$, \sqsubseteq a binary relation in the set $\{\leq, \geq\}$, and b a natural number in \mathbb{N} .

This target region is denoted as $g_{v, \sqsubseteq, b}$.

In Def. 6.1, a target region corresponds to a set of chemical states such that the quantity of a chemical species in some chemical state(s) has reached a particular threshold. For us, this threshold corresponds to the value of a boundary for a given interval bounding the value of a chemical species. We can also refer to a target region as a chemical goal.

For the remainder of this Section, we consider $q_{\bullet} \in \mathcal{Q}_{\mathcal{R}}$ to be a chemical state, \mathcal{G} a set of goals, and g a specific goal in the set \mathcal{G} . With this information, we want to define the probability that the system, when starting from the state q_{\bullet} , will enter the specific goal g before entering any other goals of the set \mathcal{G} . In order not to overcount the probabilities, we cut chemical traces as soon as they enter the goal g and ignore the behaviors that enter another region in the set \mathcal{G} before.

Definition 6.2 (Minimum successful traces). *We denote as $\chi_{(q_{\bullet}, \mathcal{G}, g)}$ the set of the chemical traces $(q'_0, ((q_i, r_i, q'_i), \mu_i)_{1 \leq i \leq k})$ such that the following conditions are satisfied:*

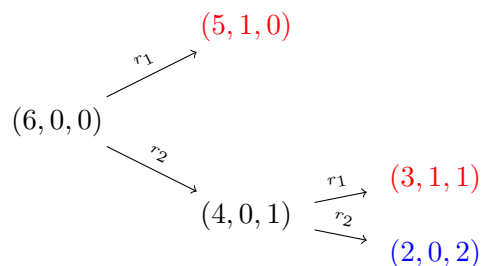


Figure 6.1: A sample transition system of the resource competition system reaching a target region. Note that for illustration purposes we do not show the complete set of reachable states. This transition system initiates from the initial chemical state $q'_0 = (6, 0, 0)$, composing the initial quantities of, respectively, the molecules A, B , and C . From this starting point several chemical traces can be generated via reaction r_1 or r_2 either until a goal along a trace has been achieved or until the common resource (molecules of A) become depleted. See in text for the complete reaction network of this system. In this example, we consider as a goal those trace who achieve in their final state a value of the molecule C equal to two, $q(C) = 2$ (indicated in blue). Alternatively, a second competing goal is when the trace terminates at the value of molecule B equal to one, $q(B) = 1$ (indicated in red).

1. $q'_0 = q_\bullet$;
2. $q'_k \in g$; and
3. $\forall i \in \mathbb{N}$, such that $0 \leq i < k$, $q'_i \notin \bigcup \mathcal{G}$.

In Def. 6.2, the set $\chi_{(q_\bullet, \mathcal{G}, g)}$ contains all the chemical traces that start in the state q_\bullet (Cond. 1), reach the target region g in their final state (Cond. 2), and have reached no other target regions before (Cond. 3). Thus, one can assess the content of successful traces in order to gain information about the which goals tend to be more habituated than others.

Example 6.1. We illustrate those minimum successful traces that can be obtained from the initial state $q_\bullet = (6, 0, 0)$ containing an initial copy number of the molecules of A, B , and C (e.g., the tuple $(q_\bullet(A), q_\bullet(B), q_\bullet(C))$) from the following reaction network:



As such, in the diagram depicted in Fig. 6.1, either one of two reactions can occur (along each branch) from the initial state which respectively updates the corresponding molecule copy number at each transition. A sequence of transitions from the initial state is a chemical trace. Note that the diagram does not depict the complete set of traces for the given system, but only the minimum successful traces. The minimal successful traces are those traces indicated on their terminal either blue or red corresponding to a goal. Here, two goals can be achieved. Either those traces which reach the goal $g_{C, \geq 2}$, that is to say the set $\{q \mid q(C) \geq 2\}$ (see Def. 6.1). A second goal is $g_{B, \geq 1}$ which corresponds to the set $\{q \mid q(B) \geq 1\}$. Thus, the set of goals in this example is $\mathcal{G} = \{g_{C, \geq 2}, g_{B, \geq 1}\}$. Namely, by Def. 6.2, we keep those traces τ

that have the properties that $final(\tau) \in g_{C,\geq,2}$ (final state drawn in blue) and exclude the traces such that $final(\tau) \in \mathcal{G} \setminus \{g_{C,\geq,2}\}$ (final state drawn in red).

We are now ready to compute probabilities of minimal successful traces.

Definition 6.3 (Probability to reach a specific goal first). *The probability that the system reaches the target region g before any other target regions in \mathcal{G} when starting in the state q_\bullet is defined as:*

$$P_g^{\mathcal{G}}(q_\bullet) = \sum_{\tau \in \chi(q_\bullet, \mathcal{G}, g)} P(\tau | q_\bullet).$$

Namely, the probability to reach a specific goal first is computed by summing the probability of all minimum successful traces that begin from the same initial state. Note that the probability of each minimum successful trace is computed similarly to that done in Def. 2.8.

The following proposition provides an easier way to compute this probability.

Proposition 6.1. *Let $g \in \mathcal{G}$ be a goal. The probabilities $P_g^{\mathcal{G}}(q)$ for every state $q \in \mathcal{Q}_{\mathcal{R}}$ are related by the following three conditions:*

1. $P_g^{\mathcal{G}}(q) = 1$ whenever $q \in g$;
2. $P_g^{\mathcal{G}}(q) = 0$ whenever $q \in \bigcup \mathcal{G} \setminus g$;
3. $P_g^{\mathcal{G}}(q) = \sum_{q \xrightarrow{r_i} q'} \mu_i(q) \cdot P_g^{\mathcal{G}}(q')$ whenever $q \notin \bigcup \mathcal{G}$.

Prop. 6.1 provides an iterative scheme that computes for every state q a sequence of values that converges from below to the value of $P_g^{\mathcal{G}}(q)$. It does so accordingly. By Cond. 1, the first case covers the scenario when a chemical state begins from within the goal of interest. By contrast, in Cond. 2 a chemical state begins in an alternate goal. The final case covers those chemical states that are not contained in any defined goal, thus by Cond. 3 one can compute recursively the probability to reach the prescribed goal by zipping through each transition, while computing the transition probability along the way, in each minimum successful trace. Indeed, computing the complete set of successful traces is usually not possible. Thus, one under-approximates the probability to reach the indicated goal by effectively summing each transition probability. By complementing, we can also obtain an upper bound to this probability. That is, $\sum_{g \in \mathcal{G}} P_g^{\mathcal{G}}(q) \leq 1$ is the probability sum of each successful trace to reach the goal $g \in \mathcal{G}$ from the chemical state q . Thus, $P_g^{\mathcal{G}}(q) \leq 1 - \sum_{g \in \mathcal{G} \setminus \{g\}} P_g^{\mathcal{G}}(q)$ is the probability to reach an alternate goal $g' \notin \mathcal{G}$. The result is an interval of probabilities that bound the exact probability to reach a new goal from a source goal.

Fig. 6.2 provides an intuition on how intervals of probabilities can be computed for pairs of macro-transitions. Here we portray the resource competition system of Sect. 2.3. A minimum successful trace τ will begin from an initial state belonging to a bounded region. By noting that the aforementioned reaction system the quantities of the molecules B and C can only increase, up until the quantities of molecule A deplete, then one can bound the instances of these molecules

by an upper bound. In this example, τ enters a new region along the horizontal axis (green point; $final(\tau)(B) > M_1$). The state abstraction of $final(\tau)$ would provide information about molecule B entering a new interval, but lose information about the exact location of molecule C . In order to introduce a minimal effort for C to reach its upper bound, we consider the next goal for the molecule B to be g_{B,\leq,M'_1} and the subsequent one for C to be g_{C,\leq,M'_2} . As such, in Fig. 6.2, we consider two possible macro-transitions starting from the (green) state that exited the initial region accordingly; and, compute an interval of probabilities using Prop. 6.1 for those macro-transitions that reach, respectively, the goals g_{B,\leq,M'_1} and g_{C,\leq,M'_2} .

We can go further by the means of matrix computations. We consider one dimension for each potential state. Each function that maps states to real numbers is interpreted as a vector, whereas each function that maps pairs of states to real numbers is interpreted as a matrix.

We define the cases vector B as follows:

$$B(q) = \begin{cases} 1 & \text{whenever } q \in g, \\ 0 & \text{otherwise.} \end{cases}$$

The cases vector corresponds to the base cases indicated in Prop. 6.1. Mathematically, it is a column vector that holds information regarding the prescribed goals of interest. Thus, it takes into account the instances when a chemical state belongs to a goal or not. Further, the identity matrix can be defined accordingly:

$$I(q, q') = \begin{cases} 1 & \text{whenever } q = q', \\ 0 & \text{otherwise.} \end{cases}$$

And, the transition matrix as:

$$A(q, q') = \begin{cases} 0 & \text{when } q \in \bigcup \mathcal{G}, \\ \sum_{q \xrightarrow{r_i} q'} \mu_i(q) \cdot P_g^{\mathcal{G}}(q') & \text{otherwise.} \end{cases}$$

Here, the transition matrix contains the transition probabilities for minimum successful traces.

Afterwhich, the sequence $(X_k)_{k \in \mathbb{N}}$ of column vectors that is defined by:

1. $X_0 = B$;
2. $X_{k+1} = A \cdot X_k + B$ for every $k \in \mathbb{N}$;

converges component-wise to the probability $P_g^{\mathcal{G}}(q)$. It follows that $P_g^{\mathcal{G}} = (\sum_{j \in \mathbb{N}} A^j) \cdot B$. Or even, $P_g^{\mathcal{G}} = (I - A)^{-1} \cdot B$, whenever the matrix $(I - A)$ is invertible.

The computation of the probabilities $P_g^{\mathcal{G}}(q)$ can be proceeded by using any available linear algebra library. This is indeed what the model checker PRISM [22, 16] is doing. Yet, having unfolded the computation offers several advantages. For instance, in our setting, the probabilities can be approximated from below by finitely approximating the formal expansion of the sums

of the powers of the sparse matrix A . Secondly, when dealing with high dimensional models, expressions with scalar coefficients can be symbolically simplified into expressions over interval coefficients [26] in order to eliminate some dimensions and to tune the trade-off between accuracy and efficiency. This would not be possible with a black box approach.

6.2 Case Study

In this Section, we target a more in-depth discussion of the probability computations that were accomplished for the BD model of Sect. 2.2. Indeed, the exact probabilities that we computed for both the non-overlapping and overlapping abstract transition systems were possible thanks to probabilistic expressions we discuss in this Section.

6.2.1 Biased one-dimensional random walk with non-constant coefficients

In the following, we aim at deriving for the BD model a closed form expression that permits us to compute the probability of, when bounded by an interval, to reach a prescribed goal prior to another.

The BD model is a particular instance of a one-dimensional random walk with non-equal non-constant parameters. In the rest of the Section, we consider $m < M$ two natural numbers such that $M - m \geq 2$, k_0 a number in the set $\{m \dots M\}$, and $(\mu_k)_{m < k < M}$ a family of real numbers indexed over the set $\{m + 1 \dots M - 1\}$ such that $0 \leq \mu_k \leq 1$ for every k in the set $\{m + 1 \dots M - 1\}$.

Definition 6.4. *We define the 1-dimensional random walk over the state space $\{m \dots M\}$ with the parameters $(\mu_k)_{m < k < M}$ and the initial state k_0 as the DTMC that starts in the state k_0 with probability 1, and at each state k proceeds as follows:*

1. *Whenever $k \in \{m, M\}$ then the state remains the same.*
2. *Otherwise, the system jump at state $k + 1$ with probability μ_k and at state $k - 1$ with probability $\mu_k - 1$.*

In the following proposition, we provide an explicit expression for the probability that the system reaches the state M before the state m .

Proposition 6.2. *In the one-dimensional random walk over the state space $\{m \dots M\}$ with the parameters $(\mu_k)_{m < k < M}$ and the initial state k_0 , the probability U_{k_0} to reach the state M before the state m is equal to:*

$$U_{k_0} = \frac{1 + \sum_{m < o < k_0} \prod_{m < l \leq o} \left(\frac{1}{\mu_l} - 1\right)}{1 + \sum_{m < o < M} \prod_{m < l \leq o} \left(\frac{1}{\mu_l} - 1\right)}.$$

Proof. For any integer k such that $m \leq k \leq M$, we denote as U_k the probability that the system reach the state M before the state m , knowing that the system starts in state k .

We have:

$$\begin{cases} U_m = 0, \\ U_M = 1, \\ U_k = \mu_k \cdot U_{k+1} + (1 - \mu_k) \cdot U_{k-1} \quad \text{whenever } m < k < M. \end{cases}$$

Thus, for k such that $m < k < M$, we have:

$$\begin{aligned} U_k &= \mu_k \cdot U_{k+1} + (1 - \mu_k) \cdot U_{k-1} \\ (\mu_k + (1 - \mu_k)) \cdot U_k &= \mu_k \cdot U_{k+1} + (1 - \mu_k) \cdot U_{k-1} \\ (1 - \mu_k) \cdot (U_k - U_{k-1}) &= \mu_k \cdot (U_{k+1} - U_k) \\ U_{k+1} - U_k &= \frac{1 - \mu_k}{\mu_k} \cdot (U_k - U_{k-1}) \\ U_{k+1} - U_k &= \left(\frac{1}{\mu_k} - 1 \right) \cdot (U_k - U_{k-1}) \end{aligned}$$

It follows that, for k such that $m < k < M$,

$$U_{k+1} - U_k = (U_{m+1} - U_m) \cdot \prod_{m < l \leq k} \left(\frac{1}{\mu_l} - 1 \right).$$

Then, for k such that $m < k \leq M$,

$$\begin{aligned} U_k &= U_m + \sum_{m \leq o < k} U_{o+1} - U_o \\ U_k &= U_m + (U_{m+1} - U_m) \cdot \sum_{m \leq o < k} \prod_{m < l \leq o} \left(\frac{1}{\mu_l} - 1 \right). \end{aligned}$$

Since $U_m = 0$ and $U_M = 1$, for k such that $m < k \leq M$, we get that:

$$\begin{aligned} U_k &= \frac{(U_{m+1} - U_m) \cdot \left(\sum_{m \leq o < k} \prod_{m < l \leq o} \left(\frac{1}{\mu_l} - 1 \right) \right)}{(U_{m+1} - U_m) \cdot \left(\sum_{m \leq o < M} \prod_{m < l \leq o} \left(\frac{1}{\mu_l} - 1 \right) \right)} \\ U_k &= \frac{\sum_{m \leq o < k} \prod_{m < l \leq o} \left(\frac{1}{\mu_l} - 1 \right)}{\sum_{m \leq o < M} \prod_{m < l \leq o} \left(\frac{1}{\mu_l} - 1 \right)} \\ U_k &= \frac{1 + \sum_{m < o < k} \prod_{m < l \leq o} \left(\frac{1}{\mu_l} - 1 \right)}{1 + \sum_{m < o < M} \prod_{m < l \leq o} \left(\frac{1}{\mu_l} - 1 \right)} \end{aligned}$$

□

By differentiating this expression, we study how this probability increases when one param-

eter $\mu_{k'}$ increases:

Proposition 6.3. *The probability that the system reaches the state M before the state m increases when the rates $(\mu_k)_{m < k < M}$ increase.*

Proof. Let us consider two integer k, k' such that $m < k < M$ and $m < k' < M$. Let us compute the derivative of the expression U_k with respect to $\mu_{k'}$.

Let us write:

$$U_k = \frac{P_k}{P_M}$$

where:

- $P_i = \sum_{m \leq o < i} T_o$;
- $T_o = \prod_{m < l \leq o} \left(\frac{1}{\mu_l} - 1 \right)$.

We have:

$$\frac{\delta T_k}{\delta \mu_{k'}} = \begin{cases} 0 & \text{whenever } o < k' \\ -\frac{T_k}{\mu_{k'}^2 \cdot \left(\frac{1}{\mu_{k'}} - 1 \right)} & \text{otherwise} \end{cases}$$

That is to say:

$$\frac{\delta T_k}{\delta \mu_{k'}} = \begin{cases} 0 & \text{whenever } o < k' \\ \frac{T_k}{\mu_{k'}^2 - \mu_{k'}} & \text{otherwise} \end{cases}$$

Then:

$$\frac{\delta P_i}{\delta \mu_{k'}} = \sum_{m \leq o < i} \frac{\delta T_o}{\delta \mu_{k'}}.$$

It follows that:

$$\frac{\delta P_i}{\delta \mu_{k'}} = \begin{cases} 0 & \text{when } k' > i \\ \frac{\sum_{k' < o < i} T_o}{\mu_{k'}^2 - \mu_{k'}} & \text{otherwise} \end{cases}$$

And:

$$\frac{\delta P_i}{\delta \mu_{k'}} = \begin{cases} 0 & \text{when } k' > i \\ \frac{P_i - P_{k'}}{\mu_{k'}^2 - \mu_{k'}} & \text{otherwise} \end{cases}$$

Also, we have:

$$U_k = \frac{P_k}{P_M}.$$

Thus:

$$\frac{\delta U_k}{\delta \mu_{k'}} = \frac{\frac{\delta P_k}{\delta \mu_{k'}} \cdot P_M - \frac{\delta P_M}{\delta \mu_{k'}} \cdot P_k}{P_M^2}.$$

It follows that:

$$\frac{\delta U_k}{\delta \mu_{k'}} = \begin{cases} \frac{-(P_M - P_{k'}) \cdot P_k}{\mu_{k'}^2 - \mu_{k'}} & \text{when } k' > q(A) \\ \frac{(P_k - P_{k'}) \cdot P_M - (P_M - P_{k'}) \cdot P_k}{\mu_{k'}^2 - \mu_{k'}} & \text{otherwise} \end{cases}$$

That is to say:

$$\frac{\delta U_k}{\delta \mu_{k'}} = \begin{cases} \frac{(P_M - P_{k'}) \cdot P_k}{\mu_{k'} - \mu_{k'}^2} & \text{when } k' > q(A) \\ \frac{P_{k'} \cdot (P_M - P_k)}{\mu_{k'} - \mu_{k'}^2} & \text{otherwise} \end{cases}$$

Given that:

- $0 < \mu_r(q) < 1$,
- $P_n \geq 0$ for any $n \in \mathbb{N}$,
- $P_m \leq P_{m+n}$ for any $m, n \in \mathbb{N}$,

we can conclude that:

$$\frac{\delta U_k}{\delta \mu_{k'}} \geq 0.$$

□

It follows from Prop. 6.3 that the probability of reaching the state M before the state m increases when the parameters $(\mu_k)_{m < k < M}$ increases. Thus we can get a lower bound on the probability to reach the state M before the state m by replacing each parameter by a lower bound on them, and an upper bound to this probability by replacing these parameters by an upper bound on them.

Proposition 6.4. *Let $\underline{\mu}$ and $\bar{\mu}$ be two real numbers such that $0 < \underline{\mu} \leq \bar{\mu} < 1$ and for every integer k such that $m < k < M$, $\underline{\mu} \leq k \leq \bar{\mu}$.*

Then, for every integer k such that $m \leq k \leq M$, we have:

$$g(m, k, M, \underline{\mu}) \leq U_k \leq g(m, k, M, \bar{\mu})$$

$$\text{where: } g(m, k, M, \lambda) = \begin{cases} \frac{(\frac{1}{\lambda} - 1)^{k-m} - 1}{(\frac{1}{\lambda} - 1)^{M-m} - 1} & \text{whenever } \lambda \neq \frac{1}{2} \\ \frac{k-m}{M-m} & \text{whenever } \lambda = \frac{1}{2} \end{cases}$$

Proof. By Prop. 6.3, we have:

$$\frac{\sum_{m \leq o < k} \prod_{m < l \leq o} \left(\frac{1}{\underline{\mu}} - 1 \right)}{\sum_{m \leq o < M} \prod_{m < l \leq o} \left(\frac{1}{\underline{\mu}} - 1 \right)} \leq U_k \leq \frac{\sum_{m \leq o < k} \prod_{m < l \leq o} \left(\frac{1}{\bar{\mu}} - 1 \right)}{\sum_{m \leq o < M} \prod_{m < l \leq o} \left(\frac{1}{\bar{\mu}} - 1 \right)}$$

Then:

$$\frac{\sum_{m \leq o < k} \left(\frac{1}{\underline{\mu}} - 1 \right)^{o-m}}{\sum_{m \leq o < M} \left(\frac{1}{\underline{\mu}} - 1 \right)^{o-m}} \leq U_k \leq \frac{\sum_{m \leq o < k} \left(\frac{1}{\bar{\mu}} - 1 \right)^{o-m}}{\sum_{m \leq o < M} \left(\frac{1}{\bar{\mu}} - 1 \right)^{o-m}}$$

Thus:

$$\frac{\sum_{0 \leq o < k-m} \left(\frac{1}{\underline{\mu}} - 1 \right)^o}{\sum_{0 \leq o < M-m} \left(\frac{1}{\underline{\mu}} - 1 \right)^o} \leq U_k \leq \frac{\sum_{0 \leq o < k-m} \left(\frac{1}{\bar{\mu}} - 1 \right)^o}{\sum_{0 \leq o < M-m} \left(\frac{1}{\bar{\mu}} - 1 \right)^o}$$

We conclude that:

$$g(m, k, M, \underline{\mu}) \leq U_k \leq g(m, k, M, \bar{\mu})$$

$$\text{where: } g(m, k, M, \lambda) = \begin{cases} \left(\frac{1}{\lambda}-1\right)^{k-m} - 1 & \text{whenever } \lambda \neq \frac{1}{2} \\ \left(\frac{1}{\lambda}-1\right)^{M-m} - 1 & \\ \frac{k-m}{M-m} & \text{whenever } \lambda = \frac{1}{2} \end{cases} \quad \square$$

In Prop. 6.4, the computation of the lower and upper bounds on the probability to reach the state M before the state m comes down to the same computation as in the case when the stochastic rates μ_k are all equal. Each bound is then expressed as the quotient between two sums of power of a same number. This quotient can be simplified further. This gives rise to two different expressions depending on whether this number is equal to 1, or not. The former case is obtained in the case of a fair random walk where each rate μ_k is equal to $\frac{1}{2}$.

6.2.2 BD model revisited

In this Section, we will instantiate the results obtained in Sect. 6.2.1 to the BD model. First, we instantiate the closed expression for a one-dimensional system of Prop. 6.2.

Proposition 6.5. *Let $m, M \in \mathbb{N}$. In the birth- and death-process over the state space $\mathcal{Q}_{\mathcal{R}}$ with the parameters $(\mu_k)_{m < k < M}$ and the initial state q , the probability U_q to reach the state $[A \rightarrow M]$ before the state $[A \rightarrow m]$ is equal to:*

$$U_q = \frac{m! + \sum_{m < o < q(A)} (o!) \cdot \left(\frac{k_{A'}}{k_A}\right)^{o-m}}{m! + \sum_{m < o < M} (o!) \cdot \left(\frac{k_{A'}}{k_A}\right)^{o-m}}.$$

Proof. Let $q \in \mathcal{Q}_{\mathcal{R}}$. For any integer $q(A)$ such that $m \leq q(A) \leq M$, we denote as U_q the probability that the system reach the state $[A \rightarrow M]$ before the state $[A \rightarrow m]$, knowing that the system starts in state q . We have:

$$U_q = \frac{1 + \sum_{m < o < q(A)} \prod_{m < l \leq o} \left(\frac{l \cdot k_{A'}}{k_A}\right)}{1 + \sum_{m < o < M} \prod_{m < l \leq o} \left(\frac{l \cdot k_{A'}}{k_A}\right)}$$

Then:

$$\frac{1 + \sum_{m < o < q(A)} \frac{o!}{m!} \cdot \left(\frac{k_{A'}}{k_A}\right)^{o-m}}{1 + \sum_{m < o < M} \frac{o!}{m!} \cdot \left(\frac{k_{A'}}{k_A}\right)^{o-m}}$$

Thus,

$$\frac{m! + \sum_{m < o < q(A)} (o!) \cdot \left(\frac{k_{A'}}{k_A}\right)^{o-m}}{m! + \sum_{m < o < M} (o!) \cdot \left(\frac{k_{A'}}{k_A}\right)^{o-m}}.$$

□

Thus, the expression in Prop. 6.5 allows us to compute the probability of eventually exiting a bounded region. Namely, this is possible due to the nature of the probability functions of the

BD model. That is, both probability functions of the BD model are naturally monotonic (see Fig. 6.3), which allow us to compute an interval of probabilities that encapsulates the likelihood of belonging to a given region, or even exiting it as we previously described in the case study of Sect. 2.2. This becomes more clear in the following proposition.

Indeed, we use in the following proposition that since the BD probability functions are monotonic, we can get an upper bound on U_q by over-estimating the probability of synthesis events, and a lower bound by under-estimating the probability of synthesis events. To achieve this, we bound the number of occurrences of A between m and M .

Proposition 6.6. *Let $m, M \in \mathbb{N}$. In the birth- and death-process over the state space $\mathcal{Q}_{\mathcal{R}}$, let $\mu_M = \frac{k_A}{k_A + M \cdot k_{A'}}$ and $\mu_m = \frac{k_A}{k_A + m \cdot k_{A'}}$ such that for each state q such that $m < q(A) < M$, $\mu_M \leq \frac{k_A}{k_A + q(A) \cdot k_{A'}} \leq \mu_m$. Then, for every state q such that $m < q(A) < M$, we have:*

$$h\left(m, q(A), M, \frac{M \cdot k_{A'}}{k_A}\right) \leq U_q \leq h\left(m, q(A), M, \frac{m \cdot k_{A'}}{k_A}\right)$$

$$\text{where: } h(m, q(A), M, \lambda) = \begin{cases} \frac{\lambda^{q(A)-m} - 1}{\lambda^{M-m} - 1} & \text{whenever } \lambda \neq 1 \\ \frac{q(A) - m}{M - m} & \text{whenever } \lambda = 1 \end{cases}$$

Proof. By Prop. 6.3, we have

$$\frac{1 + \sum_{m < o < q(A)} \prod_{m < l \leq o} \left(\frac{M \cdot k_{A'}}{k_A}\right)}{1 + \sum_{m < o < M} \prod_{m < l \leq o} \left(\frac{M \cdot k_{A'}}{k_A}\right)} \leq U_q \leq \frac{1 + \sum_{m < o < q(A)} \prod_{m < l \leq o} \left(\frac{m \cdot k_{A'}}{k_A}\right)}{1 + \sum_{m < o < M} \prod_{m < l \leq o} \left(\frac{m \cdot k_{A'}}{k_A}\right)}$$

Then:

$$\frac{1 + \sum_{m < o < q(A)} \left(\frac{M \cdot k_{A'}}{k_A}\right)^{o-m}}{1 + \sum_{m < o < M} \left(\frac{M \cdot k_{A'}}{k_A}\right)^{o-m}} \leq U_q \leq \frac{1 + \sum_{m < o < q(A)} \left(\frac{m \cdot k_{A'}}{k_A}\right)^{o-m}}{1 + \sum_{m < o < M} \left(\frac{m \cdot k_{A'}}{k_A}\right)^{o-m}}$$

Thus:

$$\frac{\sum_{m \leq o < q(A)} \left(\frac{M \cdot k_{A'}}{k_A}\right)^{o-m}}{\sum_{m \leq o < M} \left(\frac{M \cdot k_{A'}}{k_A}\right)^{o-m}} \leq U_q \leq \frac{\sum_{m \leq o < q(A)} \left(\frac{m \cdot k_{A'}}{k_A}\right)^{o-m}}{\sum_{m \leq o < M} \left(\frac{m \cdot k_{A'}}{k_A}\right)^{o-m}}$$

It follows that:

$$\frac{\sum_{0 \leq o < q(A)-m} \left(\frac{M \cdot k_{A'}}{k_A}\right)^o}{\sum_{0 \leq o < M-m} \left(\frac{M \cdot k_{A'}}{k_A}\right)^o} \leq U_q \leq \frac{\sum_{0 \leq o < q(A)-m} \left(\frac{m \cdot k_{A'}}{k_A}\right)^o}{\sum_{0 \leq o < M-m} \left(\frac{m \cdot k_{A'}}{k_A}\right)^o}.$$

We conclude that:

$$h\left(m, q(A), M, \frac{M \cdot k_{A'}}{k_A}\right) \leq U_q \leq h\left(m, q(A), M, \frac{m \cdot k_{A'}}{k_A}\right)$$

where:

$$h(m, q(A), M, \lambda) = \begin{cases} \frac{\lambda^{q(A)-m} - 1}{\lambda^{M-m} - 1} & \text{whenever } \lambda \neq 1 \\ \frac{q(A) - m}{M - m} & \text{whenever } \lambda = 1. \end{cases}$$

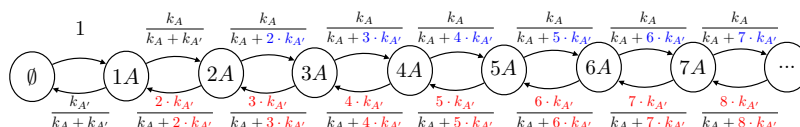


Figure 6.3: A DTMC of the BD model with non-constant transition probabilities.

□

In Prop. 6.6, the lower and upper bounds over the quantity of A are propagated by monotonicity to get a lower and an upper bounds to the probability of reaching the state M before the state m . Each of this bound is indeed a quotient between two sums of successive powers of a same number, which can be simplified further. Note that the case when the number is equal to 1 is specific.

Consequently, the series of derivations given by Props. 6.5 - 6.6 permits a user to compute probabilities to exit regions and further assign probability intervals to macro-transitions in order to capture its tendency. This derivation is possible for the one-dimensional BD system, yet remains difficult for the systems with higher dimensions. However, for systems composed of higher dimensions, we can compute interval probabilities by sampling our expression given in Prop. 6.1.

6.3 Conclusion

In this Chapter, we introduced the notion of chemical goals which allow us to carefully relate a concrete transition system to an abstract one in order to re-introduce probabilities into a non-deterministic process. Namely, a chemical goal corresponds to a reachable set of regions that may be observed by some chemical behaviors. We quantify these behaviors by observing the first instance at which the quantity of a chemical component achieves a prescribed value of interest. Accordingly, we derive a mathematical structure to under-approximate (resp. over-approximate) the probability of exiting from a bounded region, thus giving us the derived interval of probabilities of interest. We can use these computed quantities in order to observe the tendency between pairs of macro-transitions in our macro-transition system. That is, the likelihood of the direction of a particular macro-transition.

Finally, we further characterized the BD process discussed during the first case study of Sect. 2.2. Here, we derived and characterized a series of functions that permits us to compute exact probabilities of exiting a bounded region. This allows one to coarse-grain with confidence the BD system dynamics at a lower level of representation, while retaining a general description of its dynamics. More importantly, it provides an example of how one can carefully reason on the concrete transition system in order to refine the abstraction process and increase its precision.

Chapter 7

Conclusion

In this manuscript, we have discussed extensively the design of approximation methods for stochastic reaction networks. The abstractions we develop in the work are strongly motivated by techniques borrowed from AI, a theory of approximation of mathematical structures. Specifically, in order to abstract a concrete transition system, one must first define each of the mathematical objects that characterize the concrete behaviors with the simplest element being discrete values. We refer to this aforementioned system as the concrete semantics. After which, we design the abstract semantics, which is an over-approximation of the concrete one. In the abstract, we derive the abstract counterpart to each concrete element, with the most simplest being intervals. We show how an interval in the abstract is a sound approximation of a concrete value. In the end, we end up with a hierarchy of abstract functions to traverse from the concrete domain into the abstract one. Additionally, we use Galois connections in order to go back and forth from the two domains. We derive the notion of macro-transition systems. We have proposed a generic framework to coarse-grain stochastic reaction networks by sampling the quantity of each kind of molecules within a set of intervals. Instead of neglecting unlikely transitions between abstract regions of states, we compute conservative bounds on their probability. More specifically, whereas in the transition state space of logical models often focus on unidirectional dynamics towards a focal point, the type of macro-transitions we characterize are bidirectional and for each transition label interval probabilities. Thus, the interval probabilities provide insight on the likelihood that a macro-transition will traverse to an abstract region. Thus, in our formally derived framework we can make quantitative arguments regarding the general trend of the underlying stochastic reaction system. Our goal is indeed to check whether we can derive as accurate models as hand-written ones while ensuring a formal relationship between the potential behaviors in the initial and the derived models. We expect to gain new insights to understand the underlying assumptions behind logical modeling. Getting formal — but accurate — coarse-grained models requires a specific treatment of boundary effects. It is indeed important not to amplify the importance of some unlikely behaviors. In particular we ensure that every chemical transition between abstract regions of states corresponds to a minimal number of steps. For transitions induced by reverse reactions, we use overlapping intervals. Namely, we generalize the classical interval lattice interval domain for the purpose of obtaining a more accurate coarse-graining of the dynamics of the stochastic chemical transition system. Consequently, when a chemical state can be arbitrary close to the boundary of an interval, we examine the capacity to cross the next interval instead of just entering it. This induces a non-standard interpretation of the dynamics of the coarse-grained systems. Fewer trajectories are considered in the abstract, while soundness is still ensured (by construction). As in a non-deterministic setting, every concrete behavior is reflected in the abstract, but additionally an upper bound on their probability is computed. Hopefully, rare events are assigned a small upper bound on their probability to occur.

An additional advantage of our framework is the ability to perform and combine numerical abstractions, such as finite expansions of infinite increasing series and include their overall impact as a unique bound on the numerical errors made on the computation of probability values. However, scaling the framework to more complex systems would require one to formally parse

intricate relations between numerous variables. For example, to deal with higher dimensional models, symbolic simplification of expressions [26] is possible. Another avenue of thought is to use exact model reduction methods based on the structure of the components of an initial reaction network [14]. Yet, in practice, exact model reduction techniques are not very efficient, especially in a stochastic setting [15]. Still, in our context, we can be more optimistic since, on the one hand not all the properties of the underlying stochastic system have to be preserved and because on the other hand, we can admit numerical approximations on probability values as any other sources of numerical imprecision and include them in the computation of sound over-approximations.

Our motivation was to be able to explain them thoroughly and to focus on minimal difficulties that occur pervasively in models. Ideally we would like to target bigger — but still reasonable — models such as the one for the early events of the EGFR cascade presented in [6]. These models already cope with around three hundred kinds of molecular species. To scale up to this kind of model, we will restrict our study to the competition between pairs of macro-transitions. Yet special care will have to be taken to deal with the denominator of probability functions. These denominators involve the sum over the propensities of each potential event which make them particularly tricky to abstract. Instead of using numerical approaches, we plan to use marginalization to isolate independent subnetworks and reduce the number of terms in denominators accordingly. Yet, here again perfectly independent reaction sub-networks are very unlikely to occur, thus we plan to propose a relaxed version, at the cost of including an additional component in the computation of bounds of probability values.

Bibliography

- [1] Wassim Abou-Jaoudé, Denis Thieffry, and Jérôme Feret. “Formal derivation of qualitative dynamical models from biochemical networks”. *Biosystems* 149 (2016), pp. 70–112 (cit. on pp. 15, 26, 109, 110).
- [2] U. Alon. *An Introduction to Systems Biology: Design Principles of Biological Circuits*. Chapman & Hall/CRC Mathematical and Computational Biology. Taylor & Francis, 2006 (cit. on p. 3).
- [3] Virgile Andreani. “Modelling and Efficient Characterization of Enzyme-Mediated Response to Antibiotic Treatments”. PhD thesis (cit. on p. 4).
- [4] Ezio Bartocci and Pietro Lió. “Computational Modeling, Formal Analysis, and Tools for Systems Biology”. *PLoS Computational Biology* 12.1 (2016), pp. 1–22 (cit. on pp. 2, 14).
- [5] Andreea Beica. “Abstractions of biochemical reaction networks”. PhD thesis. Paris Sciences et Lettres (ComUE), 2019 (cit. on p. 5).
- [6] Michael L. Blinov, James R. Faeder, Byron Goldstein, and William S. Hlavacek. “A network model of early events in epidermal growth factor receptor signaling that accounts for combinatorial complexity”. *Biosystems* 83.2 (2006), pp. 136–151 (cit. on pp. 5, 143).
- [7] C. Chaouiya. “Petri net modelling of biological networks”. *Briefings in Bioinformatics* 8.4 (2007), pp. 210–219 (cit. on p. 6).
- [8] Claudine Chaouiya, Elisabeth Remy, Brigitte Mossé, and Denis Thieffry. “Qualitative Analysis of Regulatory Graphs: A Computational Tool Based on a Discrete Formal Framework”. *Positive Systems*. Ed. by Luca Benvenuti, Alberto Santis, and Lorenzo Farina. Vol. 294. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 119–126 (cit. on p. 7).
- [9] Julia Chifman, Seda Arat, Zhiyong Deng, Erica Lemler, James C. Pino, Leonard A. Harris, et al. “Activated Oncogenic Pathway Modifies Iron Network in Breast Epithelial Cells: A Dynamic Modeling Perspective”. *PLoS Computational Biology* 13.2 (2017). Ed. by Anand R. Asthagiri, e1005352 (cit. on p. 7).
- [10] P. Cousot and R. Cousot. “Abstract interpretation: a unified lattice model for static analysis of programs by construction or approximation of fixpoints”. In: *Proceedings of the 4th ACM SIGACT-SIGPLAN Symposium on Principles of Programming Languages* (1977), pp. 238–252 (cit. on pp. 5, 50).
- [11] Patrick Cousot. *Principles of abstract interpretation*. Cambridge, Massachusetts: The MIT Press, 2021. 819 pp. (cit. on p. 50).
- [12] Vincent Danos, Jérôme Feret, Walter Fontana, and Jean Krivine. “Scalable Simulation of Cellular Signaling Networks”. 4807 (2007). Ed. by Zhong Shao. Red. by David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, et al., pp. 139–157 (cit. on p. 5).
- [13] A. Faure, A. Naldi, C. Chaouiya, and D. Thieffry. “Dynamical analysis of a generic Boolean model for the control of the mammalian cell cycle”. *Bioinformatics* 22.14 (2006), e124–e131 (cit. on pp. 8, 94).
- [14] J. Feret, V. Danos, J. Krivine, R. Harmer, and W. Fontana. “Internal coarse-graining of molecular systems”. *Proceedings of the National Academy of Sciences* 106.16 (2009), pp. 6453–6458 (cit. on p. 143).
- [15] Jérôme Feret, Heinz Koepl, and Tatjana Petrov. “Stochastic Fragments: A Framework for the Exact Reduction of the Stochastic Semantics of Rule-Based Models”. *Int. J. Softw. Informatics* 7.4 (2013), pp. 527–604 (cit. on pp. 5, 9, 143).
- [16] V. Forejt, M. Kwiatkowska, G. Norman, and D. Parker. “Automated Verification Techniques for Probabilistic Systems”. *Formal Methods for Eternal Networked Software Systems (SFM’11)*. Ed. by M. Bernardo and V. Issarny. Vol. 6659. LNCS. Springer, 2011, pp. 53–113 (cit. on p. 133).

-
- [17] Daniel T Gillespie. “A general method for numerically simulating the stochastic time evolution of coupled chemical reactions”. *Journal of Computational Physics* 22.4 (1976), pp. 403–434 (cit. on p. 19).
- [18] Daniel T. Gillespie. “Exact stochastic simulation of coupled chemical reactions”. *J. Phys. Chem.* 81.25 (1977), pp. 2340–2361 (cit. on p. 6).
- [19] S.A. Kauffman. “Metabolic stability and epigenesis in randomly constructed genetic nets”. *Journal of Theoretical Biology* 22.3 (1969), pp. 437–467 (cit. on p. 8).
- [20] Stephen Cole Kleene. *Introduction to Metamathematics*. Vol. 19. North Holland, 1952 (cit. on p. 74).
- [21] Ina Koch, Björn H. Junker, and Monika Heiner. “Application of Petri net theory for modelling and validation of the sucrose breakdown pathway in the potato tuber”. *Bioinformatics* 21.7 (2005), pp. 1219–1226 (cit. on p. 7).
- [22] M. Kwiatkowska, G. Norman, and D. Parker. “PRISM 4.0: Verification of Probabilistic Real-time Systems”. *Proc. 23rd International Conference on Computer Aided Verification (CAV’11)*. Ed. by G. Gopalakrishnan and S. Qadeer. Vol. 6806. LNCS. Springer, 2011, pp. 585–591 (cit. on p. 133).
- [23] D.A. McQuarrie and J.D. Simon. *Physical Chemistry: A Molecular Approach*. G - Reference, Information and Interdisciplinary Subjects Series. University Science Books, 1997 (cit. on pp. 4, 5).
- [24] Hannah R. Meredith, Virgile Andreani, Helena R. Ma, Allison J. Lopatkin, Anna J. Lee, Deverick J. Anderson, et al. “Applying ecological resistance and resilience to dissect bacterial antibiotic responses”. *Sci. Adv.* 4.12 (2018), eaau1873 (cit. on p. 4).
- [25] R. Milo and R. Phillips. *Cell Biology by the Numbers*. CRC Press, 2015 (cit. on p. 3).
- [26] Antoine Miné. “Symbolic Methods to Enhance the Precision of Numerical Abstract Domains”. *Verification, Model Checking, and Abstract Interpretation, 7th International Conference, VMCAI 2006, Charleston, SC, USA, January 8-10, 2006, Proceedings*. Ed. by E. Allen Emerson and Kedar S. Namjoshi. Vol. 3855. Lecture Notes in Computer Science. Springer, 2006, pp. 348–363 (cit. on pp. 134, 143).
- [27] Chris J. Myers. *Engineering genetic circuits*. Chapman & Hall/CRC mathematical and computational biology series. Boca Raton: CRC Press, 2010. 278 pp. (cit. on p. 5).
- [28] Chris J. Myers. *Engineering Genetic Circuits*. 0th ed. Chapman and Hall/CRC, 2016 (cit. on p. 5).
- [29] Jignesh H. Parmar and Pedro Mendes. “A computational model to understand mouse iron physiology and disease”. *PLoS Comput Biol* 15.1 (2019). Ed. by Andrew D. McCulloch, e1006680 (cit. on p. 4).
- [30] James Lyle Peterson. *Petri net theory and the modeling of systems*. Englewood Cliffs, N.J: Prentice-Hall, 1981. 290 pp. (cit. on p. 6).
- [31] C. A. Petri. “Fundamentals of a Theory of Asynchronous Information Flow”. *Information Processing, Proceedings of the 2nd IFIP Congress 1962, Munich, Germany, August 27 - September 1, 1962*. North-Holland, 1962, pp. 386–390 (cit. on p. 6).
- [32] Venkatramana N. Reddy, Michael N. Liebman, and Michael L. Mavrovouniotis. “Qualitative analysis of biochemical reaction systems”. *Computers in Biology and Medicine* 26.1 (1996), pp. 9–24 (cit. on p. 6).
- [33] Xavier Rival and Kwangkeun Yi. *Introduction to static analysis: an abstract interpretation perspective*. Cambridge, Massachusetts: The MIT Press, 2020. 299 pp. (cit. on p. 50).
- [34] Gautier Stoll, Eric Viara, Emmanuel Barillot, and Laurence Calzone. “Continuous time boolean modeling for biological signaling: application of Gillespie algorithm”. *BMC Syst Biol* 6.1 (2012), p. 116 (cit. on p. 85).
- [35] Alfred Tarski. “A lattice-theoretical fixpoint theorem and its applications.” *Pacific Journal of Mathematics* 5.2 (1955), pp. 285–309 (cit. on p. 74).
- [36] René Thomas. “Boolean formalization of genetic control circuits”. *Journal of Theoretical Biology* 42.3 (1973), pp. 563–585 (cit. on pp. 8, 20, 85, 111, 123).
- [37] Ionela Zevedei-Oancea and Stefan Schuster. “Topological analysis of metabolic networks based on Petri net theory”. *In Silico Biol* 3.3 (2003), pp. 323–345 (cit. on p. 6).

RÉSUMÉ

Les paradigmes de modélisation pour la biologie des systèmes jouent un rôle important dans l'étude de la fonction orchestrée de divers systèmes biologiques. En outre, ils permettent d'étudier un système *in-silico* afin d'obtenir des informations mécanistiques à partir des trajectoires résultant du modèle. Un enjeu majeur pour dériver une représentation idéale pour un processus de système est de fixer le compromis descriptif entre la simplicité et la précision. D'une part, les modèles trop simples ont tendance à ne reproduire que les connaissances a priori. D'autre part, les modèles trop descriptifs donnent lieu à des comportements trop difficiles à analyser, voire à calculer. Dans les deux cas, l'acquisition de nouvelles connaissances est entravée. C'est pourquoi il est sans doute important, lors de la sélection des outils de mesurer l'impact de la sélection des modèles sur la capture des phénomènes biologiques, en particulier ceux qui peuvent être vérifiés.

Dans ce manuscrit, nous présentons un cadre formel pour dériver automatiquement des modèles discrets de systèmes biologiques à partir de réseaux de réactions stochastiques. Pour ce faire, nous utilisons des techniques offertes par Abstract Interpretation pour évaluer les comportements résultant des modèles logiques, un outil de modélisation populaire en biologie des systèmes. Malgré le succès des modèles logiques pour résumer les observations expérimentales et prédire les propriétés locales du système, leurs hypothèses de modélisation sous-jacentes restent souvent implicites. Au lieu de cela, les modèles à gros grains que nous obtenons traitent de tous les comportements de la sémantique stochastique des réseaux de réaction initiaux, qui est explicitement définie. Plus précisément, l'espace d'état du réactionnel est divisé en régions abstraites et les transitions non déterministes entre les régions abstraites sont dérivées de manière conservatrice. En outre, nous récupérons les probabilités de transitions du réseau réactionnel de référence, de sorte que les limites du réseau réactionnel non déterministe sont calculées de manière conservatrice. Il est important de souligner que nous pouvons utiliser ce cadre pour évaluer, par le biais du modèle formellement dérivé, les comportements du modèle logique qui l'accompagne. En d'autres termes, le travail établi dans cette thèse ouvre une voie pour évaluer les modèles qui sont naturellement discrets, tout en ouvrant la voie à l'établissement de techniques de réduction de modèles plus efficaces pour les systèmes de réaction.

MOTS CLÉS

Interprétation abstraite, modélisation logique, méthodes formelles, réseaux de réaction stochastiques

ABSTRACT

Modeling paradigms for Systems Biology plays an important role in investigating the orchestrated function of various biological systems. Additionally, they permit one to study a system *in-silico* in order to gain mechanistic insights from the trajectories resulting from the model. A common battle to derive an ideal representation for a system process is the descriptive trade-off between the simplicity and accuracy. On one hand, too simple models are prone to reproduce only a priori knowledge. On the other hand, too descriptive models result in behaviors too difficult to parse, or even yet compute. In both cases, gaining new insights is hampered. Thus, it is arguably important in generating tools to measure the impact model selection has in capturing biological phenomena, especially those which can be verified.

In this manuscript, we report a formal framework to automatically derive discrete models of biological systems from stochastic reaction networks. To do so, we utilize techniques offered by Abstract Interpretation (AI) to assess the behaviors resulting from logical models, a popular Systems Biology modeling tool. Albeit the success of logical models in recapitulating experimental observations and predicting local system properties, their underlying modeling assumptions are often kept implicit. Instead, the coarse-grain models that we obtain deal with all the behaviors of the stochastic semantics of the initial reaction networks, which is explicitly defined. More precisely, the state space of the reaction network is split into abstract regions and non-deterministic transitions between abstract regions are derived conservatively. Also, we recover the probabilities of transitions from the reference reaction network, so that bounds to the probability of unlikely behaviors can be computed. Importantly, we emphasize how one can use this framework to assess, via the formally derived model, the behaviors of the accompanying logical model of each reaction system. Namely, the work established in this thesis bares an avenue to assess those models which are naturally discrete, while also paving a path towards establishing more efficient model reduction techniques for stochastic, combinatorial systems.

KEYWORDS

Abstract interpretation, logical modelling, formal methods, stochastic reaction networks