The rule-based model approach. A Kappa model for hepatic stellate cells activation by TGFB1

Matthieu Bouguéon, Pierre Boutillier, Jérôme Feret, Octave Hazard, Nathalie Théret

▶ To cite this version:

HAL Id: hal-03388100

https://inria.hal.science/hal-03388100v2

Submitted on 24 Oct 2021

The rule-based model approach.

A Kappa model for hepatic stellate cells activation by TGFB1.

Matthieu Bouguéon,[1,2] Pierre Boutillier,[3] Jérôme Feret,[4,5*] Octave Hazard,[4,5,6] and Nathalie Théret[1,2*]

[1] *Univ Rennes, Inria, CNRS, IRISA, UMR 6074, Rennes, France*
[2] *Univ Rennes, Inserm, EHESP, Irset, UMR S1085, Rennes, France*
[3] *Nomadic Lab, 75013, Paris, 6 rue Germaine Richier, France*
[4] *Team Antique, Inria Paris, 75012, Paris, 2 rue Simonne Iff, France*
[5] *DI-ENS (ÉNS, CNRS, PSL University), École normale supérieure, 75005, Paris, 45 rue d'Ulm, France*
[6] *École Polytechnique, 91128, Palaiseau, Route de Saclay, France*

*Corresponding Authors: Jérôme Feret; jerome.feret@info.ens.psl.eu

Nathalie Théret; nathalie.theret@univ-rennes1.fr

Chapter 1

The rule-based model approach.

A Kappa model for hepatic stellate cells activation by TGFB1

In this chapter, we introduce Kappa, a site graph rewriting language. As a realistic case study, we model a population of hepatic stellate cells under the effect of the TGFB1 protein. Kappa offers a rule-centric approach, inspired from chemistry, where interaction rules locally modify the state of a system that is defined as a graph of components, connected or not. In this case study, the components will be occurrences of hepatic stellate cells in different states, and occurrences of the protein TGFB1. The protein TGFB1 induces different behaviors of hepatic stellate cells thereby contributing either to tissue repair or to fibrosis. Better understanding the overall behavior of the mechanisms that are involved in these processes is a key issue to identify markers and therapeutic targets likely to promote the resolution of fibrosis at the expense of its progression.

1.1. Introduction

1.1.1. Modeling systems of biochemical interactions

The description and the analysis of the large scale and highly combinatorial systems which emerge from some mechanistic models of Systems Biology are still out of scope of the state of the art. In such models, the individual behavior of proteins or other components, that may establish links and modify their capability of interaction, is driven by races against shared resources. Moreover, occurrences of proteins may form a large amount of distinct complexes. Concurrency between interactions at different time-scale induces non linear feedback loops that control the abundance of these complexes. Lastly, these systems involve interactions between very small molecules, as ions and ligands, and giant complexes as DNA strands, the ribosome, or the signalosome. Understanding how the collective behavior of population of proteins and other components emerges from interactions between individual proteins, remains a crucial and mainly open challenge.

While technological progresses provide quickly an ever increasing amount of details about the potential mechanistic interactions between the components of these systems, and at an affordable cost, the scientific community is far from a global understanding of how the macroscopic behavior of these systems emerges from these interactions. This is the holy grail of Systems Biology. Yet, this challenge is hopeless without the help of specific and innovative methods to describe these complex systems and analyze their properties. These methods must scale to the large amount of information that is published in the literature at an exponentially increasing rate.

1.1.2. Modeling languages

Formal languages have been widely used to describe models of mechanistic interactions between occurrences of proteins. They provide mathematical tools to encode interactions and to define rigorously the behavior of the systems they represent by the means of a choice of semantics, would they be qualitative, stochastic, or differential.

Languages as reaction networks and classical Petri nets [53] are based on multi-set rewriting. Applying an interaction consists in consuming some reactants while producing some products. Kinetic constants specify, according to the choice of semantics, either the speed, or the average frequencies of application of each kind of reactions. These languages are very convenient to model the behavior of small or medium-size interaction systems. Yet, they struggle to scale to large models because one name (or one placeholder in the case of Petri nets) is required for each distinct kind of complexes.

It is worth to make the distinction among agent-based and rule-based approaches. In agent-based approaches, each entity, would it be a process [22] or an object [37] has to contain the specification of all its potential behaviors. The evolution of the configurations of the different entities is synchronized by the means of communication rules that define the operational semantics of the model. There are usually very few rules. It is possible to restrict the behavior of an agent with respect to some conditions over the properties of some other agents to which this agent would be linked. Yet, some fictitious processes would then be required to fetch the necessary information. Such trick has been already used in the first models written in the π-calculus [77]. Nonetheless, in general, agent-based approaches lead to network of finite state processes [60]. Thanks to this, the behavior of these models can be studied by the means of symbolic model checking tools as PRISM [65].

Agent-based approaches fail to scale whenever occurrences of components admit too many distinct configurations or whenever their capabilities of interaction depend too much on the configurations of the components they are linked to. Such models
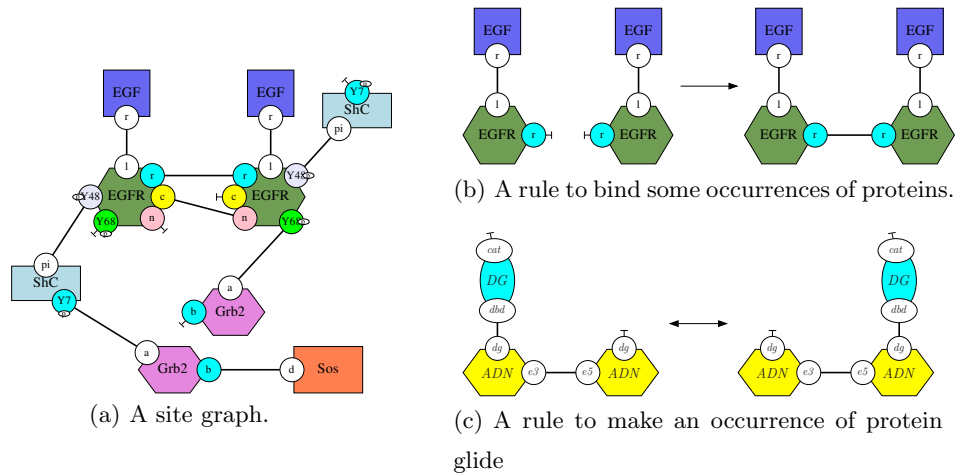
cannot be described, and *a fortiori* their behavior cannot be computed with such approaches.

Rule-based approaches consist in defining models by the means of interaction rules. Each rule specifies under which conditions over the configurations of the different occurrences of agents an interaction may happen and what is the impact of applying this interaction. This way, the state of an agent does not define once for all the capabilities of interaction of this agent. The capabilities of interaction are within the rules. This way, it is no longer necessary to itemize exhaustively the set of all the configurations agents may take. Rules only describe the parts that matter in the interactions that they describe. As a matter of fact, rule-based approaches scale better and ease the versioning of models. Moreover, since it is not necessary to describe every capability of interaction of the occurrences of the components, they ease unbiased modeling when the conception of the model is not influenced by a specific goal.

Ambient-calculus [19, 20], bioambients-calculus [78], and brane-calculi [18] are particular cases of languages. They describe the behavior of hierarchies of compartments, which may be arbitrarily nested. Some agents in the compartments, or in the case of the brane-calculi, in the membranes of compartments, provide to their compartments some capabilities to move within the hierarchy of compartments and to fuse pairwise. Capabilities of interaction may depend on the relative localization of compartments within the hierarchy of compartments. Projective brane calculus [29] describes even more faithfully the organization of compartments within a cell, by making the description of the state of the system independent from the choice of the root of the hierarchy of compartments.

1.1.3. Kappa

Languages for site graph rewriting [28, 42, 2, 59] aim at describing in a transparent way networks of interaction between occurrences of components, by the means of a syntax that is inspired from chemistry.

(a) A site graph.

(b) A rule to bind some occurrences of proteins.

(c) A rule to make an occurrence of protein glide

Figure 1.1: In Fig. 1.1(a) is given a site graph. This is a biochemical complex made of two occurrences of the ligand protein *EGF*, two occurrences of the receptor *EGFR*, one occurrence of a scaffold protein (*Shc*), two occurrences of a transport protein (*Grb2*), and one occurrence of the protein *Sos*. In Fig. 1.1(b) is given an example of binding rule. Two occurrences of the receptor *EGFR*, when both activated by a binding with some occurrences of the ligand *EGF*, may bind to each other to form a dimer. The other interaction sites are omitted because they play no rule in this interaction. In Fig. 1.1(c) is given an example of movement rule. An occurrence of the enzyme Glycolase (*DG*) may glide along both directions (according to a random walk process) along a DNA strand.

1.1.3.1. Overview

In Kappa, each complex is described by a site graph. An example of site graph is given in Fig. 1.1(a). A site graph is made of some nodes that denote occurrences of some components. Each component is associated with a list of interaction sites. Sites may be free, or bound pairwise. Besides, some interaction sites may be tagged with a property, which may stand for an activation level. Interactions between occurrences of components may change the conformation of components. For instance, in the case of proteins, they may be folded and/or unfolded, which may hide or reveal some interaction sites. In Kappa, there is no explicit notion of three dimension structure. In contrast, the conditions for a site to be visible are specified in the description of the interactions themselves.

The behavior of a system that is written in Kappa is described by the means of context-free rewriting rules. In Fig.1.1(b) is shown a rule of dimer formation. Two occurrences of the receptor *EGFR* that are activated by some occurrences of the ligand *EGF* may bind to each other and form a dimer. In Fig. 1.1(c) is given a second rule that is taken from a model of DNA repair [63]. In this rule, an occurrence of an enzyme, the Glycolase (*DG*), may glide randomly in both directions along a DNA strand.

1.1.3.2. The semantics of Kappa

A rule may be seen from an intensional point of view, as a local transformation of the state of the system, or extensionally as a potentially infinite set of reactions which may be obtained by fully specifying the context of application of these rules. From this set of reactions, several semantics may be induced. These semantics may be qualitative, stochastic, or differential, as for reaction networks and Petri nets (quantitive semantics — that is to say stochastic or differential — require the use of rate constant). Yet, the stochastic semantics of a model that is described in Kappa may be executed directly, without ever generating the underlying network of reactions. This execution is based on the iteration of the following event loop (which corresponds to the application of Gillespie's algorithm [50]). Given the current state of the system, denoted as a site graph, the set of all the potential events that may happen next is computed. An event consists in applying a rule in the site graph at an occurrence of the left hand side of a rule. Each event has a propensity which is defined as the rate of the corresponding interaction rule. Then, the next event is drawn randomly with a probability that is defined proportionally to its propensity, while the time between two consecutive events is drawn according to the exponential law with the parameter equal to the sum of the propensity of all the available events. For the sake of scalability, it would not be reasonable to recompute the list of all the potential events after each rule application. This set can indeed be dynamically updated by accounting only the new potential events and the events that are no longer possible, due to the application of the last

event [30]. The actual simulator engine has been optimized thanks to the potential sharing between the patterns that occur in the left hand side of rules [10].

1.1.3.3. The Kappa ecosystem

Modeling platform. Kappa models can be designed in a dedicated modeling platform [12]. Rules can be specified in a text window while widgets provide access to most of the existing Kappa tools, including simulation, static analysis, and causal analysis.

The platform can be used online (for reasonable size models) or installed locally. Support for linking with python frontend/backend and exporting results in Jupiter notebooks is also provided.

Stochastic simulation. The stochastic simulator KaSim samples the trajectories of Kappa models faithfully according to their probability density distribution. As explained in Sect. 1.1.3.2, it relies on a representation of the state of the system as a site graph. The set of events that may be applied in the current state is computed dynamically [30]. The use of a dedicated category-based data-structure to describe and update this set while optimizing the benefit due to potential common regions among patterns [10] has speed up the simulator.

Additionally, the Kappa platform provides support for end-user interactions during the execution of a model [9]. In particular, the end-user can pause the simulation while observing the behavior of the model, specify modifications of the state of the model, and restart the simulation to observe the impact of his intervention.

Static analysis. Static analysis enhances the confidence in models. The static analyzer KaSa [31, 46, 11] computes and proves some structural properties about the complexes that may be formed in a given model, given an initial configuration. This tool uses the abstract interpretation framework [25] to approximate the computation of a least fix point over sets of patterns of interest. This way, it detects and proves which of these patterns may be reachable. As a matter of fact, any pattern of in-

terest that is not in the result of the analysis will never occur in a state that may be reached from the initial state. Due to the approximation, we cannot conclude about the patterns that are discovered by the analysis: they may – or not – occur in a given reachable state. This analysis is particularly useful to detect some relationships among the states of the sites of some agents. It can also be proved that some rules will never be applied in a given model. More information about how static analysis can help the modeling process is provided in [11].

Causal analysis. Causal analysis aims at extracting different scenarii of interest from a given simulation. While a simulation describes the behavior of a population of agents which may evolve back and forth, causal analysis aims at describing the evolution of some individuals while focussing on the computation steps that make progresses. Given an event of interest, causal analysis provides a set of minimal scenarii that are extracted from the simulation traces and that describe the events that were necessary to trigger an instance of the event of interest.

Causal analysis relies on two main ingredients. Firstly, scenarii are described as event structures. This way, concurrency between causally independent events is exploited and their interleaving orders is abstracted away. Secondly, operational research techniques are used to extract minimal sub-structures leading to the same ending event. This way, non necessary events are discarded. More information about the formal background of causal analysis may be found in [33].

Underlying network and model reduction. As explained in Sec. 1.1.3.2, a Kappa model induces a (potentially infinite) network of reactions. The tool KaDe [17] generates this network. Several export formats are available: the network may be exported in the DOTNET language [43, 8] or in SBML [56], or as a set of differential equations written in MAPLE [71], MATHEMATICA [87], MATLAB [70], or OCTAVE [38].

Model reduction may be used to simplify the underlying reaction network, the underlying system of differential equations, or even the underlying Markov chain. Exact model reduction techniques consist in discovering a change of variables. They

find out sets of quantities that can be exactly described while discarding the others. Such changes of variables may be detected directly at the level of the site graph structure, hence without even generating the underlying network of reactions. These reductions are based on the detection of symmetries [45] and the static inspection of the flow of information among different regions of complexes [47, 32, 48, 16].

Conservative methods, based on tropicalisation techniques, have also been proposed [7]. By exploiting separation between time- and concentration-scales, they permit to eliminate some variables, at the cost of numerical approximations. Here the exact behavior of the variables of interest is lost, but it is safely approximated by the means of intervals.

1.1.3.4. Main limitations

Kappa suffers from several limitations. For instance, the name of the interaction sites of an agent shall be pairwise distinct; also, in regards to geometry, Kappa does not offer any support either for describing the tridimensional structure of the complexes, or for describing their spatial distribution. Disallowing multiple occurrences of sites in a given agent greatly eases the detection of occurrences of patterns in graphs. Not only this is the cornerstone for an efficient stochastic simulation, but also it is the root of some algebraic constructions widely used in static analysis and model reduction. Some languages get rid of this limitation either directly as in BNGL [42] and møds [1], or indirectly by encoding them by the means of hyperlinks as in React(C) [59]. Nevertheless, it deeply impacts the efficiency of simulation engines. As far as geometry is concerned, some assumptions about the spatial conformation of agents may be implicitly encoded within rewriting rules. Some extensions of the language provide a syntax to describe the relative position of agents within complexes so as to restrict the potential events to those which satisfy some specific constraints [34].
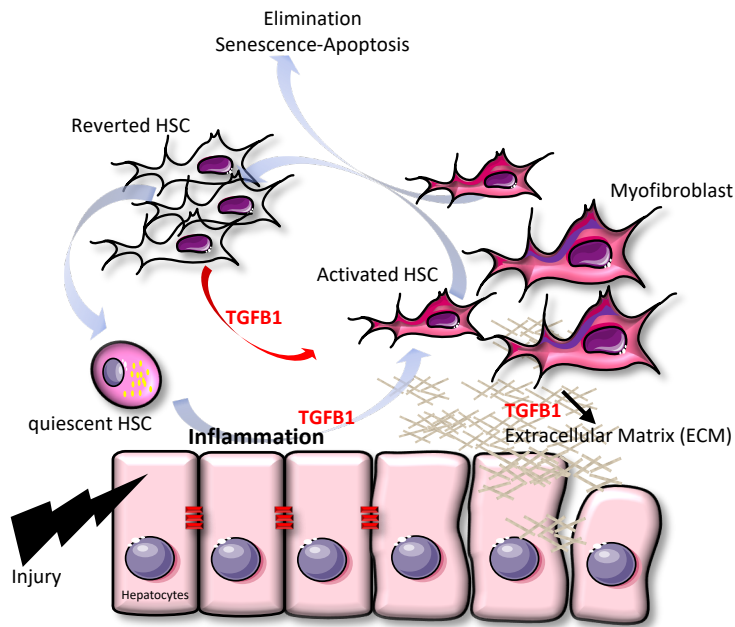
As far as the spatial distribution of complexes is concerned, the assumption is made that they are perfectly mixed. As a matter of fact, in the case of intracellular models,

crowding effects that may result from the accumulation of proteins in some specific regions of a cell, cannot be modeled. The same way, the gradients of proteins that may result from the action of a scaffold protein cannot be described (an occurrence of a scaffold protein holds some occurrences of proteins to maintain them in the same biochemical complex; in Kappa, no assumption is made about the position of these occurrences of proteins when they are released, even for a short amount of time). A partial alternative consists in encoding a grid of potential discrete positions. Then, some rewriting rules may be used to model the diffusion of proteins, which consists in making proteins glide from one position to an adjacent one. SpatialKappa [82] offers a transparent syntax for discrete diffusion of agents, based on this construction. Beside, the ML language [54] provides support for describing models of interactions between proteins with continuous motion. In Kappa, it is also possible to define a static finite hierarchy of compartments. Yet, the transport of occurrences of proteins by the means of vesicules cannot be modeled this way. The formal cell machinery [27] addresses this issue, but does not provide efficient simulation engines.

1.1.4. Modeling a population of hepatic stellate cells

In this chapter, we model in Kappa the behavior of a population of hepatic stellate cells. This is an interesting case study as it illustrates the flexibility of the language. It is worth noting that the system that is modeled is not, as it is usually the case, a population of proteins, but a population of cells interacting with some signaling proteins. Also the abstraction level of the model is tailored to cope with the amount of information that is available about the interaction and the combinatorial complexity of the execution of the underlying mechanisms. We provide in this section some biological context about the model.

Chronic liver diseases (CLDs) are long duration and slow progression pathologies

Figure 1.2: Dynamics of Hepatic Stellate Cells. Upon injury, damaged hepatocytes produce signal to induce inflammation that in turn promotes TGFB1-dependent activation of HSC. Activated HSC orchestrate tissue repair and are either eliminated through senescence and apoptosis or deactivated towards a transient reverted state that can be reactivated more rapidly. Upon repeated injuries, activated HSC progress toward a myofibroblast state that escapes to control, leading to fibrosis.

which represent a major public health issue in terms of economic cost [14]. CLDs are mainly associated with viral infections, alcoholic diseases and more recently with the non-alcoholic fatty liver diseases (NAFLD) due to the increasing frequency of metabolic syndromes (insulin resistance, type 2 diabetes and obesity). Chronic hepatitis is associated with the development of fibrosis, which results in the abnormal deposition of extracellular matrix rich in interstitial collagen and a severe dysfunction of liver functions. The terminal stage of fibrosis is cirrhosis, which constitutes the major risk of occurrence of Hepatocellular Carcinoma (HCC). The mortality linked to the complications of cirrhosis (hemorrhage, liver failure, cancer) leads to the death of a little more than one million people per year in the world.

The matrix microenvironment is therefore the major regulator of events related to the fibrosis-cirrhosis-cancer progression and Hepatic Stellate Cells (HSC) are the main actors for modifying the extracellular microenvironment (Fig. 1.2). In response to hepatic insults, HSC undergo a process of activation from quiescent vitamin A-rich cells in normal liver to proliferating, fibrogenic and contractile myofibroblasts [83]. Among the molecules that drive HSC activation, the transforming growth factor TGFB1 plays the major role. In addition to the deposition of fibrillar matrix components, activated HSC produce a wide variety of molecules involved in extracellular matrix remodeling, which in turn modulates the availability and signaling of TGFB1. Upon injury, HSC are activated to repair tissues and next are eliminated according to three mechanisms apoptosis, senescence and reversion leading to return to the healthy situation [61]. However, when injury persists, HSC remain activated with a myofibroblast phenotype, and extracellular matrix accumulates leading to fibrosis, cirrhosis and cancer. The understanding of the dynamics of HSC activation and regulation by TGFB1 is essential to identify markers and therapeutic targets likely to promote the resolution of fibrosis at the expense of its progression.

In this chapter, we developed a Kappa model to characterize the dynamics of HSC activation and the different states upon TGFB1 stimulation.

14

1.1.5. Outline

The rest of the chapter is organized in the following way. In Sect. 1.2, the main features of Kappa are informally explained in graphical representation (figures have been generated with the GKappa library [44]). In Sect. 1.3, the rules that model the behavior of the hepatic stellate cells are given and explained. The kinetic parameters are also documented. Some references to the literature are provided to justify both rules and parameters. In Sect. 1.4, the model is checked and simulated. Static analysis is used so as to increase the confidence on the model. Then the model is simulated under two scenarii. In the first one, the population of hepatic stellate cells responds to an acute inflammatory aggression. In the second one, the case of chronic inflammation is considered. In Sect. 1.5, the chapter is concluded. The current state of the model is discussed as well as future extensions of it.

1.2. Kappa

We give the syntax and the semantics of Kappa.

1.2.1. Site graphs

We introduce in this section the notion of site graphs. Site graphs will be used to describe not only the different states of the systems that we are modeling, but also the different patterns that will be used in the section 1.2.2 to describe, by the means of rewriting rules, the behavior of these systems.
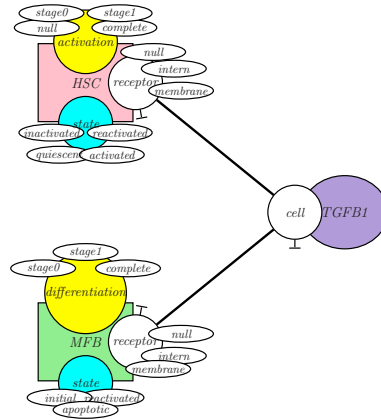
1.2.1.1. Signature

The definition of a model starts with its signature. This signature specifies the alphabet, that is to say all the ingredients that may be involved in this model. It may

be described graphically by the means of a *contact map*, as the one that is given in Fig. 1.3. A contact map is made of a list of nodes that specifies the different *kinds of agents* in the model. Each node has a name and is drawn with a specific shape. The notion of agent in Kappa is quite abstract. Agents can be used to encode not only instances of proteins, but also individual cells, depending on the granularity of the model. Moreover, in order to tune the scaling of a model, an agent may also stand for a fixed amount of occurrences of a given kind of protein, or a given kind of cell, all in a same configuration. Each agent is also fitted with a set of *interaction sites*, which are depicted around it by the means of named circles. In Kappa, a given kind of agent cannot bear two interaction sites with the same name. Lastly, each interaction site is fitted with a set of tags that may be used to denote its *activation level*, as the stage of differentiation of some cells, for instance. Activation states may also be used to describe the localisation of the occurrences of some agents or of some sites within a finite hierarchy of compartments. Interaction sites may also carry a *binding state*: sites with the symbol ⊣ may potentially remain free; the pairs of sites that may be bound pairwise are described in the contact map by the means of undirected arcs. In the contact map, a site may be bound to several others (we will explain later that in such a case, there is a competition for the binding to this interaction site). Moreover, a site may be bound to itself in the contact map (in such a case, the corresponding interaction sites of two occurrences of the agent may be bound together).

Example 1.1: *An example of contact map is given in Fig. 1.3. It describes the components of our model. This contact map introduces three kinds of agents: hepatic stellate cells HSC, myofibroblasts MFB, and occurrences of the transforming growth factor protein TGFB1. The model documents only one interaction site for the protein TGFB1. This site, which is called cell, enables the occurrences of the TGFB1 protein to bind either a hepatic stellate cell or a myofibroblast. The protein TGFB1 has also many other interaction sites and may exist in various forms (active, latent, degraded, . . .). Yet these considerations do not matter in the scope of this model. They are*

16

Figure 1.3: A contact map. This map specifies the signature of a model by itemizing the different kinds of agent, their interaction sites, the different activation states that each kind of sites may take, and the potential bindings between these interaction sites.

useful only when considering the extracellular matrix molecular network. Thus, we omit them for the sake of simplicity. In the model, we are interested in the state of three interaction sites of the hepatic stellate cells. Two of them describe the different forms of the cells and its different activation stages. We distinguish quiescent cells, activated cells, inactivated cells, and reactivated cells. Then within each form, there may be different activation stages. When the notion of activation stage does not make sense, the stage null is used. For instance, quiescent cells have no intermediary stages, thus their sites activation are always in the state null. Otherwise, the cell may be in an intermediary stage stage0 or stage1, or fully activated, which is written as complete. The third site, called receptor, is an abstraction of all the TGFB1 receptors in a hepatic stellate cell. This site receptor carries out a binding state. This site can be free or bound to an occurrence of the TGFB1 agent. A given cell may fix many occurrences of the TGFB1 protein. The sites of the occurrences of the TGFB1 protein may also be free. For scalability issues, we abstract this by a single interaction site per cell. Hence an agent TGFB1 does not stand for a single occurrence of the TGFB1 protein, but for the average amount of occurrences of this protein that are bound to a hepatic stellate
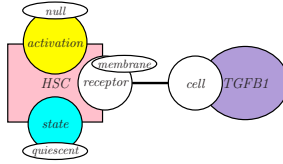
cell. The site receptor also carries out a localization, which ranges among null, intern, and membrane. The state null means that all the receptors of the cell are currently degraded. The state intern means that they all have been internalized. Lastly, the state membrane means that they are all available on the membrane of the cell.

As far as myofibroblasts are concerned, the model focuses on three interaction sites as well. Myofibroblasts carry out a state. We distinguish initial and apoptotic/senescent ones, and also the ones coming from the differentiation of a reactivated hepatic cell. Beside, there exist several differentiation stages between different states, which is encoded within a site called differentiation. Myofibroblasts may be in an intermediary stage, which is written as stage0 or stage1, or fully differentiated, which is written as complete. The third site, which is called receptor, copes for the receptors of an occurrence of myofibroblast. This site works exactly as the site receptor of the hepatic stellate cells does. Lastly, the agent TGFB1 has a unique interaction site. This site carries out a binding state. Indeed a pack of occurrences of the TGFB1 protein may be either free, or bound to the receptors of a hepatic stellate cell, or bound to the receptors of a myofibroblast.

1.2.1.2. Complexes

Kappa models describe the behavior of a soup of complexes. A complex is made of several occurrences of agents. Each occurrence of an agent is equipped with a set of interaction sites. Some sites carry out an activation state, but only one. Lastly, each occurrence of a site may be either free, or bound to exactly one other occurrence of a site. As opposed to the contact map, an occurrence of a site cannot be bound to itself in a complex. Additionally, an occurrence of a site cannot be bound to two distinct occurrences of sites. A complex forms a connected pattern, this means that it is always possible to go from a given occurrence of an agent to another one by following a potentially empty sequence of bonds.
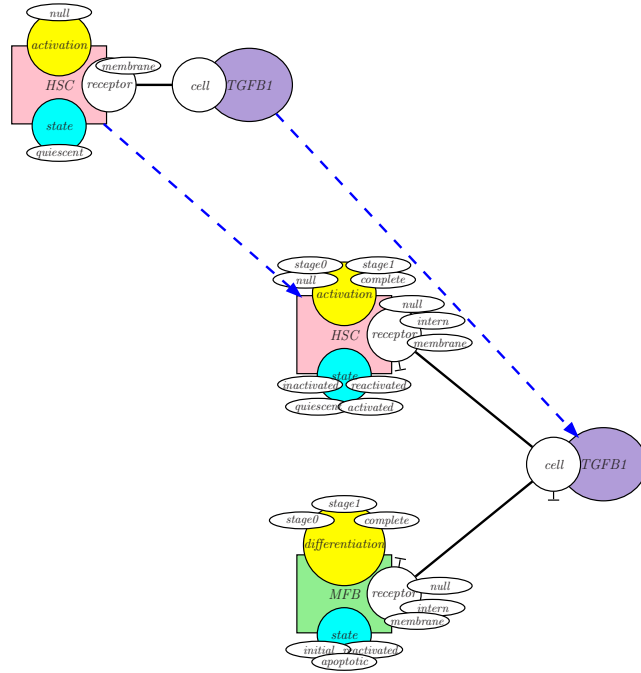
Example 1.2: *An example of complex is given in Fig. 1.4. This complex is made*

Figure 1.4: A complex. It contains several occurrences of agents. Each occurrence documents the set of its interaction sites. The sites that may carry out an activation state, have one. Moreover, each site that may carry a binding state is either free, or bound to another site.

of two occurrences of agents. The first one denotes an occurrence of hepatic stellate cell. The second one denotes a pack of TGFB1 proteins. The TGFB1 proteins are bound to the receptors of the hepatic stellate cell which is in its quiescent form (hence not activated yet). Lastly, the receptors of the hepatic stellate cell are located on the membrane of this cell.
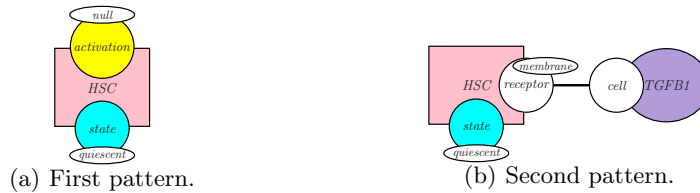
The signature constrains the set of complexes of a model. Not all the complexes that satisfy the syntax of Kappa are consistent with a signature. The contact map not only provides the list of all the potential kinds of agents with their interaction sites, but also it summarizes the potential states of the sites of each occurrence of agents. More precisely, each occurrence of an agent in a complex shall specify the same interaction sites as the corresponding agent in the contact map. Moreover, every site in a complex such that the corresponding site in the contact map admits at least one activation state, shall bear an activation state as well. This is the same for the binding states. These constraints ensure that the state of each occurrence of agents in a complex is fully specified. Three additional constraints ensure that the state of sites matches with the contact map: firstly an occurrence of site may carry an activation state only when the corresponding site in the contact map carries this activation state as well; secondly an occurrence of site may be free only when the corresponding site in the contact map may be free as well; thirdly, two occurrences of sites may be bound together only when the two corresponding sites may be bound in the contact map.

19

Figure 1.5: The unique projection from the complex of Fig. 1.4 into the contact map of Fig. 1.3. This projection is defined by mapping each occurrence of agents of the complex to the unique node corresponding to this agent in the contact map.

These constraints can be formalized by requiring that every complex can be projected onto the contact map, that is to say that the function which maps every occurrence of agents of a complex to the agent with the same name on the contact map is always a *homomorphism*. Said differently, the contact map may be understood as the folding of every complex of a model and every node of the contact map summarizes all the potential configurations of the occurrences of the corresponding agent.

Example 1.3: *In Fig. 1.5 is depicted the projection of the complex of Fig. 1.4 onto the contact map of Fig. 1.3.*

(a) First pattern.　　　　　(b) Second pattern.

Figure 1.6: Two connected patterns. They are made of occurrences of agents. Each occurrence of agents documents a subset of its interaction sites. Each site may carry out an internal state and/or a binding state (while remaining consistent with the content of the contact map). An occurrence of an interaction site may be free or bound to a specific occurrence of an interaction site.

1.2.1.3. Patterns

The behavior of complexes is defined by the means of rewriting rules. These rules specify not only the necessary conditions to trigger an interaction, but also the potential effects of these interactions. Before explaining more precisely what a rewriting rule is, it is necessary to introduce the notion of pattern. Indeed patterns are used to specify under which conditions a rule may be applied.

We focus the presentation on the description of connected patterns. More sophisticated patterns may be obtained by putting several connected patterns side by side. A *connected pattern* is a contiguous part of a complex. This way, it may be made of zero, one, or several occurrences of each kind of agents. Each occurrence of an agent may be associated with a set of interaction sites. Each occurrence of an interaction site may potentially bear an activation state. Lastly, each occurrence of an interaction site may be free or bound to exactly one other occurrence of an interaction site. The binding state of an occurrence of an interaction site may also remain unspecified.
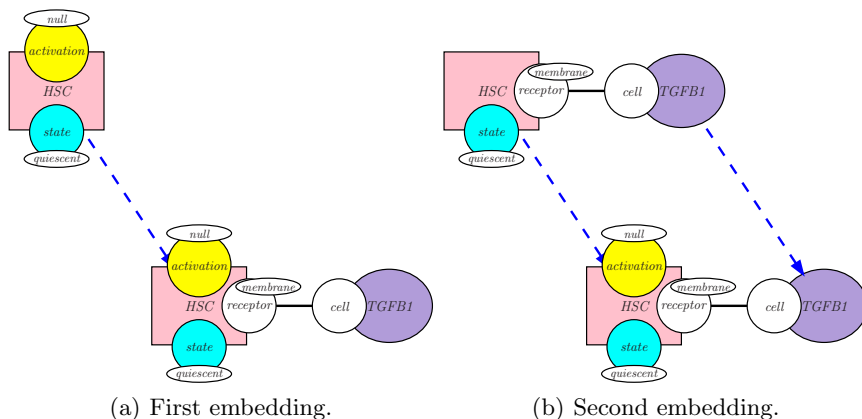
Example 1.4: *Two examples of connected patterns are given in Fig. 1.6. The first pattern (see in Fig. 1.6(a)) is made of a single occurrence of the agent HSC and documents the state of two sites. The site state is in the state quiescent whereas the site activation is in the state null. Neither the binding state, nor the state of the interaction site receptor is specified.*

The second pattern (see in Fig. 1.6(b)) is made of one occurrence of the agent HSC and one occurrence of the agent TGFB1 that are bound together via the site receptor of the first one and the site cell of the second one. The site receptor of the agent HSC is tagged with the string 'membrane' , which means that the receptors of this cell are on the membrane. It is also specified that the hepatic stellate cell must be in the quiescent state. The activation level of the cell is not documented.

As it was already the case for complexes, the contact map also contrains the patterns may be used in a model. This way, the occurrence of an agent in a pattern can only document the interaction sites which are associated to the unique occurrence of this agent in the contact map. An occurrence of an interaction site may bear a given activation state only if the corresponding site in the contact map is tagged with this activation state. An occurrence of an interaction site may be free only when the corresponding site in the contact map may be free as well. Lastly, two occurrences of sites may be bound together in a pattern only when there is a link between the two corresponding sites in the contact map. Said differently, as it was already the case for complexes, it shall be possible to project the pattern onto the contact map. This means that the function mapping each agent of a pattern to the unique corresponding agent in the contact map shall be a homomorphism.

1.2.1.4. Embeddings between patterns

A pattern may specify more or less information. It is indeed possible to insert some interaction sites in the occurrence of an agent that does not document all its interaction sites. Moreover, it is also possible to insert a new binding or activation state to an interaction site that misses one. It is even possible to bound a given site to the fresh occurrence of an agent. In all these cases, we will say that the initial pattern occurs in the second one, or equivalently, that the second pattern contains an occurrence of the first one. That is to say that the relationship between the occurrences of agents in the first pattern and the occurrences of agents in the second pattern induces an

(a) First embedding. (b) Second embedding.

Figure 1.7: Two embeddings between the patterns that were given in Fig. 1.6 and the complex that was given in Fig. 1.4.

embedding. An *embedding* from a pattern into another one is a function which maps each occurrence of agents in the first pattern to an occurrence of agents in the second pattern while preserving the structure of site graphs. This means that this mapping preserves the kinds of agents, the sites that are documented, the activation and binding states.

It is worth noting that complexes are particular patterns. In a complex, each occurrence of an agent documents all its sites, with an activation state and a binding state whenever they have one. This way, it is not possible to insert more information in a complex. A complex is a connected pattern that cannot be embedded in any other connected pattern.

Example 1.5: *Two examples of embeddings are given in Fig. 1.7. These are the only embeddings between the patterns that were given in Fig. 1.6 and the complex that was given in Fig. 1.4. In both embeddings, each occurrence of an agent in the pattern is mapped into the unique occurrence of this agent in the complex.*

It is worth noting that an embedding from a connected pattern into another pattern is fully defined by the image of one of its occurrences of agents. The other associations

may be retrieved by following the links between interaction sites and by using the fact that embeddings shall preserve links. This property highly eases the research of occurrences of a given pattern inside another pattern. Site graphs are *rigid* [32, 75].

1.2.2. Site graph rewriting

Patterns are used to speficy the potential behavior of Kappa models, by the means of rewriting rules. This is the objective of the current section.
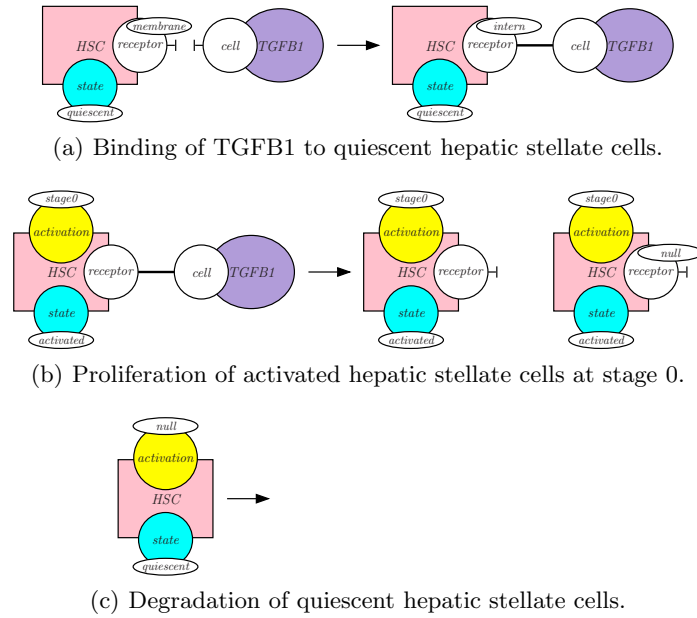
1.2.2.1. Interaction rules

Complexes may evolve by applying interaction rules. An *interaction rule* is defined as a pair of patterns with an implicit pairing relation between some occurrences of agents in the first pattern and some occurrences of agents in the second pattern. The first pattern specifies the local conditions under which an interaction may be triggered. The difference between both patterns specifies the transformation that is performed when this rule is applied. As a matter of fact, the second pattern in a rule shall be obtained from the first one by modifying the activation and/or the binding states of some interaction sites, inserting fresh occurrences of agents (in this case, the full interface of these occurrences shall be documented), and removing some occurrences of agents.

Example 1.6: *In Fig. 1.8, three examples of interaction rules are given.*

In Fig. 1.8(a), occurrences of the protein TGFB1 bind a hepatic stellate cell in its quiescent form. The receptors of this cell must be located on its membrane. The interation consists in establishing a bond between the site receptor of the agent HSC and the site cell of the agent TGFB1 while internalizing the receptors of the cell.

In Fig. 1.8(b), an activated hepatic stellate cell in activation stage 0 and connected to some TGFB1 proteins may proliferate. As a result, the TGFB1 proteins are consumed, which frees the receptors of the cell, and another cell is created. This cell is in

(a) Binding of TGFB1 to quiescent hepatic stellate cells.

(b) Proliferation of activated hepatic stellate cells at stage 0.

(c) Degradation of quiescent hepatic stellate cells.

Figure 1.8: Three examples of rules: in 1.8(a), a binding rule; in 1.8(b), a proliferation rule; and in 1.8(c), a degradation rule.

the same activation stage and is also activated. Moreover, its receptors are free and in the state null. We notice that the configuration of the newly produced cell is fully specified, as it is required when a fresh agent is created.

In Fig. 1.8(c), a quiescent hepatic cell in the activation state null may be degraded. Whenever this interaction rule is applied, an occurrence of hepatic stellate cell in this configuration is removed. There is no requirement on the receptors of the cell. They may be null, on the membrane, or internalized. Additionally, when the receptors are bound, the corresponding binding is released before degrading the cell, which also frees the occurrences of TGFB1 proteins potentially bound to the cell. This update has not to be specified explicitly in the rule: this is called a side effect.

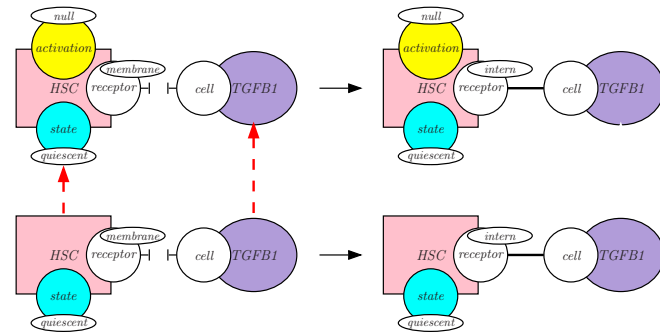## 1.2.2.2. Reactions induced by an interaction rule

As explained previously, the left hand side of a rule specifies under with context a given interaction may happen. It is then possible to insert further constraints for the conditions under which a rule may be applied, by refining the patterns that occur on the left hand side and on the right hand side of a rule in exactly the same way. A rule that cannot be refined further (without creating a new connected component) is called a *reaction-rule* [51].

Special care has to be taken about agent degradation. On the first hand, when refining the state of an agent to be degraded in the left hand side of a rule, the right hand side is not impacted (since there is no corresponding agent). On the second hand, agent degradation may cause side effects. Indeed, the state of an occurrence of agent to be degraded may be refined by binding one of its sites to the site of another occurrence of agent. In such a case, if the latter occurrence is not degraded, then its site becomes free in the right hand side of the rule. There is no pending bound in Kappa.
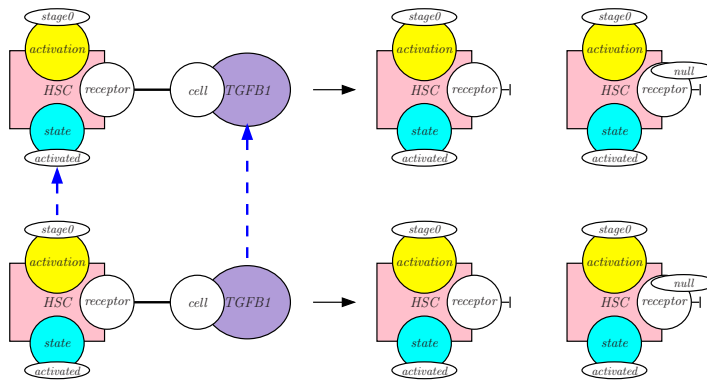
Example 1.7: *In Fig. 1.9, some rules are refined into reaction-rules. In Fig. 1.9(a) is shown a refinement of the rule that was given in Fig. 1.8(a). It is additionally specified that the hepatic stellate cell shall be on the state null of activation. In Fig. 1.9(b) is depicted the unique refinement of the rule that was given in Fig. 1.8(b) (since this was already a reaction, no further information can be inserted). In Fig. 1.9(c) is drawn a refinement of the rule that was given in Fig. 1.8(c). It is additionally specified that the receptors of the hepatic stelatte cell shall be internalized and bound to some occurrences of the TGFB1 proteins. These occurrences are released when the hepatic stellate cell is degraded, as a result of side effects.*

1.2.2.3. Underlying reaction network

A set of rules can be translated into a – potentially infinite – set of reaction-rules, by replacing each interaction rule by the set of the reaction-rules that can be obtained as

(a) Binding of TGFB1 on quiescent *HSC*s in activation level *null*.

(b) Proliferation of *HSC*s at stage 0.

(c) Degradation of *HSC*s bound to some occurrences of the *TGFB1* proteins.

Figure 1.9: Three examples of reaction-rules for the rules of Fig. 1.8.
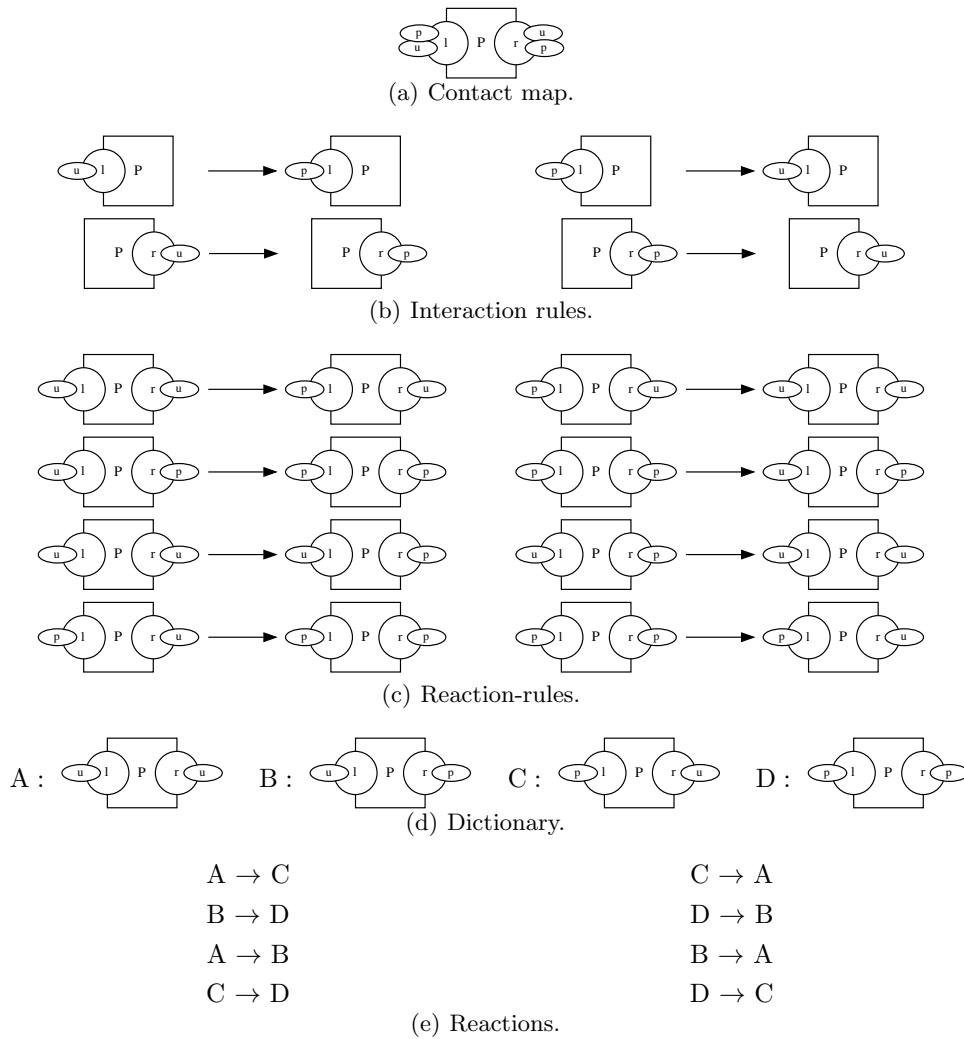
refinements of this rule. It is then enough to replace each complex by a name to get a proper (potentially infinite) reaction network, in which each rule is defined as a tuple of reactants and a tuple of products. This reaction network is defined uniquely up to the choice of the names of each complex. The behavior of a set of interaction rules may then be defined as the behavior of its underlying reaction network. Quantitative semantics require to assign rates to each rules, the rate of each reaction-rule being defined as the rate of the rule it has been generated from.

Example 1.8: *We conclude this section by describing the compilation of a toy model written in Kappa into a reaction network. We consider a model with only one kind of agent, a protein. This protein has two sites, l et r. Each occurrence of these sites may be phosphorylated, or not. The signature of the model is given in Fig. 1.10(a) by the means of a contact map. The phosphorylation and the dephosphorylation of each site of an occurrence of a protein is independent from the state of the other site in this occurrence, which is formalized in the four rules that are given in Fig. 1.10(b). This way, neither the phosphorylation rules, nor the dephosphorylation rules, document the state of the other site.*

The underlying reaction-rules are obtained by expanding the context of application of each interaction. This way, in our example, each rule gives birth to two reaction-rules, according to the phosphorylation state of the site that is not specified in the initial rule. These reaction-rules are given in Fig. 1.10(c).

The next step consists in naming the different kinds of complexes that are involved in the so-obtained reaction-rules. An occurrence of the protein with no phosphorylated site is called A, an occurrence of the protein with only the site r phosphorylated is called B, an occurrence of the protein with only the site l phosphorylated is called C, and an occurrence of the protein with both sites phosphorylated is called D. Named reactions are given in Fig. 1.10(e). They have been obtained by replacing each occurrence of complex with its name in the reaction-rules.

Defining the behavior of a model by the means of its underlying reaction networks

(a) Contact map.

(b) Interaction rules.

(c) Reaction-rules.

(d) Dictionary.

A → C	C → A
B → D	D → B
A → B	B → A
C → D	D → C

(e) Reactions.

Figure 1.10: A model made of a contact map and four interaction rules, and its compilation into a reaction network.

has been done to ease the presentation. The semantics of the language BNGL was initially implemented this way [43]. Yet, such a semantics is not so convenient in practice, since a Kappa model usually induces too many reactions. The semantics may be formalized directly by the means of a process calculus [31, 48] or in a categorical setting [33, 45]. The first approach provides a more operational perspective whereas the second one abstracts away more computational details. It is worth noting that usual categorical rewriting frameworks (by single push-out [67], double puch-out [23], or sesqui-pushout [24]) fail in modeling correctly side effects. Two known approaches solve this issue. It is possible to twist the definition of embeddings [33, 45] or to enrich graphs with constraints [6].
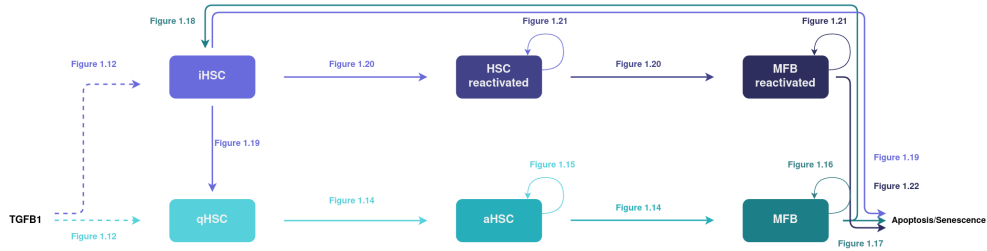
The simulator of Kappa directly applies rewriting rules in the graph that describes the state of the system [30, 10]. The underlying reaction network is never computed explicitely.

1.3. Model of activation of stellate cells

Now we describe in Kappa a model of the behavior of a population of hepatic stellate cells.

1.3.1. Overview of model

In Fig. 1.11 is sketched the potential behavior of an occurrence of hepatic stellate cell. This diagram itemizes the different transformation processes between the different forms of hepatic stellate cells. In particular, it describes how hepatic stellate cells are activated by the TGFB1, and differentiate into myofibroblasts which undergo different processes. An important point is the inactivation pathway leading to the generation of inactivated hepatic stellate cells which differ from the quiescent phenotype. Inactivated cells are more quickly reactivated by TGFB1 than quiescent ones, thereby amplifying

Figure 1.11: The behavior of hepatic stellate cells. TGFB1 protein induces the activation of quiescent hepatic stellate cells (qHSC) (green dashed arrow). The activated hepatic stellate cells (aHSC) differentiate to myobroblasts (MFB) (light green arrow). MFB have two potential behaviors (green arrow): either they enter in the apoptosis/senescence pathway or are inactivated to inactivated hepatic stellate cells (iHSC). iHSC have three potential behaviors: either they reverse into qHSC (light purple arrow), or they enter in the apoptosis/senescence pathway (light purple arrow), or they are activated by the protein TGFB1 (purple dashed arrow), leading to reactivated hepatic stellate cells (HSC reactivated) (light purple) that may differentiate to reactivated myobroblasts (MFB reactivated) (purple). The reactivated MFB have only one behavior, that is the apoptosis/senescence pathway (purple).

cell response in chronic diseases. The apoptotic/senescence pathway allows for the feedback control (to keep the explanation simple) of the activation pathway.

1.3.2. Some elements of biochemistry

Before describing the interaction rules, we give some reminders of basic elements of biochemistry. Our goal is to explain how rate constants are defined and computed. As often as possible, we try to define them with respect to the reaction half-time, that is the time after which on average, half of the reactants of a reaction have been consumed.

1.3.2.1. Reaction half-time

Let us consider a first-order reaction of the form $A \to B$. We assume that the time when each instance of the reaction is applied is drawn randomly according to an exponential law with parameter k. That is to say that given an occurrence of the component A, the probability that this occurrence has turned into an occurrence of the component B after time t is equal to $(1 - e^{-k \cdot t})$.

Assuming that A is in large quantity in the system, the expectation $E[A](t)$ of the quantity of A remaining in the system at time t, is defined by the following equation:

$$E[A](t) = E[A](0) \cdot e^{-k \cdot t}$$

where $E[A](0)$ denotes the initial quantity of A (we write it as an expectation for the sake of homogeneousness).

The half-time of the reaction, that is written $t_{1/2}$, is then the time so that the expectation of the quantity of the component A that has been consumed at time $t_{1/2}$ is equal to half of the initial quantity of this component. That is to say that:

$$\frac{E[A](0)}{2} = E[A](0) \cdot e^{-k \cdot t_{1/2}}.$$

It follows that:

$$k = \frac{ln\, 2}{t_{1/2}}.$$

In the case of a degradation reaction, that is to say, when the reaction has no product, the half-time of a reaction is also called its half-life time.

1.3.2.2. Conversion

In practice, reaction time is documented in the literature in various forms. Sometimes, it is documented as the time taken to transform all the occurrences of the component A into occurrences of the component B. Since there always remains a residual quantity,

we interpret this as the time $t_{99\%}$ so that the expectation of the quantity of the component A that has been consumed by the reaction at time $t_{99\%}$ is equal to 99% of the initial quantity of this component.

The conversion from $t_{99\%}$ to $t_{1/2}$ can be made thanks to the following reasoning. Given q between 0 and 1, we can define t_q the time so that the expectation of the component A that has been consumed is equal to the fraction q of the initial quantity of this component. The duration t_q is defined by the following equation:

$$(1-q) \cdot E[A](0) = E[A](0){\cdot}e^{-k{\cdot}t_q}.$$

It follows that:

$$\frac{ln\,2}{t_{1/2}} = \frac{-ln\,(1-q)}{t_q}.$$

Thus,

$$t_{1/2} = \frac{ln\,2}{ln\,\frac{1}{1-q}} \cdot t_q.$$

We can conclude that:

$$t_{1/2} = \frac{ln\,2}{ln\,100} \cdot t_{99\%}.$$

1.3.2.3. Production equilibrium

It often happens that a degradation rule is counter-balanced by a synthesis rule in order to maintain an expected average amount of components in stationary regime.

Let us consider two reactions, a degradation reaction $A \rightarrow .$ with a half-life time $t_{1/2}$ and a synthesis reaction $. \rightarrow A$ at a rate k. The goal is to set the rate constant k so that the expected average amount of the component A is equal to $E[A]_{eq}$ in stationary regime. This means that, when the quantity of A is equal to $E[A]_{eq}$, the overall propensity of the degradation rule shall be equal to the one of the synthesis rule. That is to say that:

$$k = \frac{ln\,2}{t_{1/2}} \cdot E[A]_{eq}.$$

33

It is worth noting that this parameterization does not enforce a rigid equilibrium. This ensures only the eventual behavior of the system in the absence of other mechanisms that could modify the quantity of the component A.

1.3.2.4. Erlang distributions

The exponential law is defined by only one parameter. As a consequence, the standard deviation of the time a given event may take, is fully defined by its average time. This is not always satisfying from a modeling point of view, since some processes may require time distributions with different standard deviations.

A solution to this issue consists in decomposing a given interaction into several intermediary steps, each of these being executed according to an exponential distribution. The resulting composite process satisfies a so called Erlang-distribution [41] that is defined by two parameters (the average time of each intermediary step and the number of these steps). For instance, we may consider the sequential composition of two steps the duration of each of which being defined by an exponential law with a same parameter, and compare it with a single process the duration of which is defined by an exponential law, twice as slow as each step of the composite process. Then the standard deviation for the time so that half of the quantity of the initial component has completed the two intemediary steps is less than the standard deviation of the time so that half of the quantity of the initial component has completed the single step process. Moreover, the time so that the expectation of the quantity of the initial component that has completed the two-step process is equal to 99% is shorter than the time that is defined the same way for the single step process.

We do not know how to define analytically the reaction half-time of the intermediary processes with respect to the overall completion time of the process. Instead, we fit these values empirically, by simulating the behavior of the intermediary interactions (without considering the rest of the model).
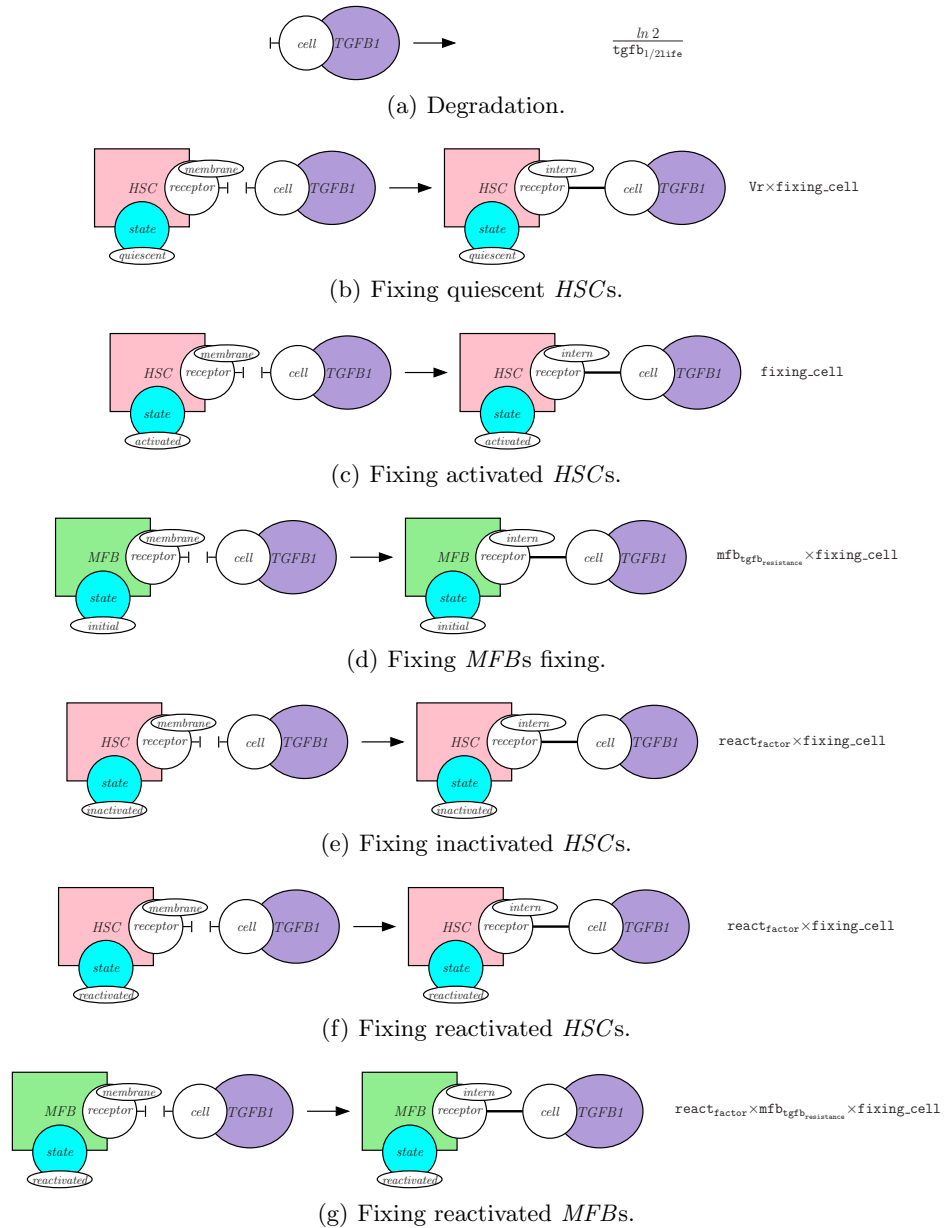
1.3.3. Interaction rules

We now itemize the interaction rules of our model. The rate constants are parameterized with some values essentially found in the literature. The value of these parameters is given after the description of the rules in Fig. 1.25 on page 51.
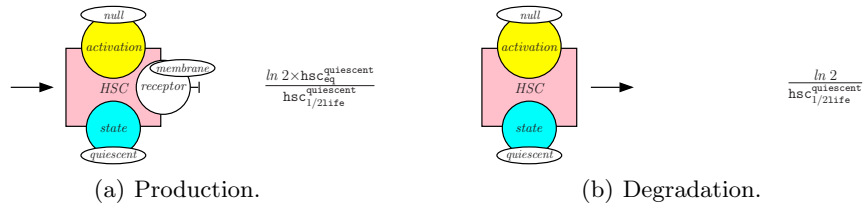
1.3.3.1. The behavior of TGFB1 proteins

In Fig. 1.12 is specified the behavior of TGFB1 proteins.

The degradation of the protein TGFB1 is described in Fig. 1.12(a). In general, the occurrences of the protein TGFB1 are spontaneously degraded only in their active form. Yet, in this model, only the active form of TGFB1 is involved, this is why the activation state of the occurrences of the protein TGFB1 is omitted. Inactive form of the protein TGFB1 plays an important role in the extracellular matrix. Additionally, this rule specifies that only the occurrences of the agent *TGFB1* that are not linked to any other agent may be degraded. The degradation rate is set according to the half-time of the protein.

The rules in Figs. 1.12(b), 1.12(c), 1.12(d), 1.12(e), 1.12(f), and 1.12(g) describe the interactions between TGFB1 and cells, the latter being in different states. We use several rules in order to give them different rates. Interactions with *TGFB1* agents are possible only when cell agents have a free *receptor* at their *membrane* and *TGFB1* agents are in their *active state* (which is omitted in our model) and not bound to any other agents. The receptors that bind TGFB1 are internalized leading to the state called *intern*. In order to preserve a pool of qHSC for cell renewal, we modulate the rule rate from (Fig. 1.12(b)) with a variable `Vr` that stands for the degree of renewal. This variable ranges between 0 and 1 according to the amount of cells that are currently bound to TGFB1 proteins. The more cells are bound to some TGFB1, the lower `Vr` will be, till reaching the value 0. Compared to hepatic stellate cells, myofibroblasts are less sensitive to the protein TGFB1 [80], therefore we reduced the rate of transformation induced by TGFB1 binding by $41,6\%$ ($41,6\%$

(a) Degradation.

(b) Fixing quiescent *HSC*s.

(c) Fixing activated *HSC*s.

(d) Fixing *MFB*s fixing.

(e) Fixing inactivated *HSC*s.

(f) Fixing reactivated *HSC*s.

(g) Fixing reactivated *MFB*s.

Figure 1.12: The behavior of the protein *TGFB1*.

$$\frac{ln\,2 \times \mathrm{hsc}_{eq}^{\mathrm{quiescent}}}{\mathrm{hsc}_{1/2\mathrm{life}}^{\mathrm{quiescent}}}$$

(a) Production.

$$\frac{ln\,2}{\mathrm{hsc}_{1/2\mathrm{life}}^{\mathrm{quiescent}}}$$

(b) Degradation.

Figure 1.13: The renewal of quiescent hepatic stellate cells.

= *HSC_proliferation/MFB_proliferation*), whatever the states of the *MFB* (*initial* or *reactivated*) are. Compared to the cells in the state qHSC and MFB, the cells in the state iHSC, HSC reactivated, and MFB reactivated are more sensitive to TGFB1 [80], that is why we increased the corresponding rate constant (4 fold time).

1.3.3.2. Renewal of quiescent HSCs

The quantity of quiescent HSC in the liver results from an equilibrium between the production and the degradation of these cells. While qHSCs have mesenchymal stem cell origin [76, 52], there is no information about the qHSC production and their renewal in normal and pathological livers. Upon TGFB1 stimulation, qHSC are transformed towards aHSC and the pool of qHSC is likely consumed. We introduced a variable `Vr` to preserve a residual pool of qHSC (see page 35). The degradation and production rules for hepatic stellate cells in their quiescent form are described in Fig. 1.13. The degradation of qHSC occurs when they have a *null* activation and no links with any other agents (see Fig. 1.13(b)). In Fig. 1.13(a), the state of the quiescent hepatic stellate cells which are synthetized has to be fully specified. It is written that their activation state is *null* and their receptors are free and on the membrane.

The rates of both rules are set according to the explanations that were given in Sec. 1.3.2 on page 31 so as to ensure the renewal rate of the hepatic stellate cells in their quiescent form and also the overall amount of them in regular regime.

1.3.3.3. Activation and differentiation

The activation of an occurrence of a hepatic stellate cell may happen when the three following conditions are satisfied: it shall be in a quiescent form (which is written as *quiescent*), its activation level shall be null (which is written as *null*), and it shall be activated by some TGFB1 proteins (Fig. 1.14(a)). This activation modifies the conformation of the hepatic stellate cell, which is now an activated one (which is written *activated*) in the first stage of activation (which is written as *stage0*).

Activation is a gradual process (Figs. 1.14(b) and 1.14(c)), the hepatic stellate cells enter in their second activation stage (which is written as *stage1*), and their third and last one (which is written as *complete*). During this process, they remain in their activated form.

Once in the last stage of activation, hepatic stellate cells may differentiate into myofibroblasts. This process is done in three stages. The first step consists in replacing a fully activated hepatic stellate cell into a myofibroblast (*MFB* agent) in initial state (which is written as *initial*) and in the first differentiation stage (which is written as *stage0*). The description of this first step requires four rules (Fig. 1.14(d)). The main reason is that the state of the cell receptors shall be maintained. Yet since *HSC* and *MFB* are two different agents, it is not possible to inherit information from the *HSC* agents that are consumed to the *MFB* agents that are created. Thus, the solution is to write one rule for each potential state of the interaction site *receptor*, and there are four of them.

After differentiation of HSC towards MFB, the latter undergoes two other steps of differentiation. The rule in Fig. 1.14(e) describes the passage from the initial stage into the second one (which is written *stage1*). Then, the rule in Fig. 1.14(f) describes the passage from the second stage into the last one (which is written *complete*).

The rates of activation and differentiation steps are defined by the means of a reaction half-time, following the guidelines that were given in Sec. 1.3.2. Each activation step shares the same rate, while each differentiation step shares another one.

(a) Initial step of activation.

(b) Second step of activation.

(c) Third step of activation.

(d) First step of differentiation.

(e) Second step of differentiation.

(f) Final step of differentiation.

Figure 1.14: Activation of hepatic stellate cells and formation of myofibroblasts.

(a) When in the initial stage.

(b) When in the second stage.

(c) When in the final stage.

Figure 1.15: Proliferation of activated hepatic stellate cells.

1.3.3.4. Proliferation of activated hepatic stellate cells

The activation of hepatic stellate cells by some occurrences of the *TGFB1* protein may induce their proliferation (Fig. 1.15).

There are three different rules according to the activation stage of the cells. As a result of proliferation, the occurrences of the *TGFB1* agent are consumed, and the occurrences of the *HSC* agent are duplicated. It is worth noting that we have assumed that the new occurrences are in the same state and in the same activation stage as the occurrences which have given them birth. But, their receptors are not operational yet (which is written as *null*).

The rate of the rules are computed from the half-time of the proliferation reaction.
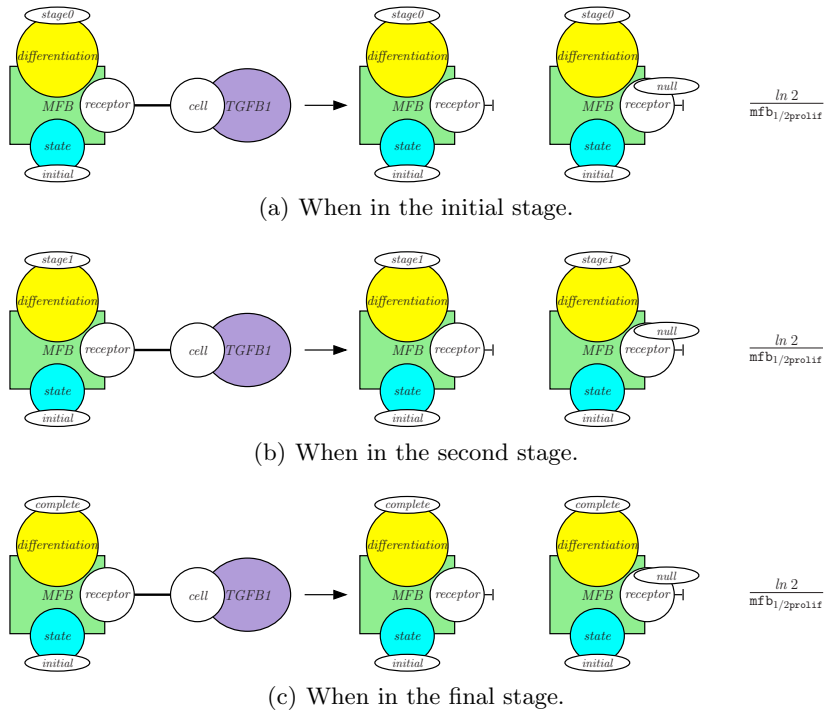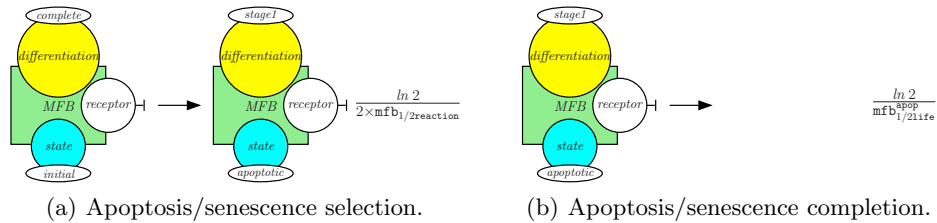
(a) When in the initial stage.

(b) When in the second stage.

(c) When in the final stage.

Figure 1.16: Proliferation of myofibroblasts.

1.3.3.5. Proliferation of myofibroblasts

The proliferation of myofibroblasts works exactly as the proliferation of activated hepatic stellate cells. Myofibroblasts must be in the *initial* state and activated by some occurrences of the TGFB1 protein. Occurrences of myofibroblasts may then be duplicated, conserving the state and the differentiation stage. The receptors of the newly created myofibroblasts are not operational yet (which is written as *null*). Proliferation speed is set by the means of the proliferation half-time of myofibroblasts.

1.3.3.6. Apoptosis and senescence of myofibroblasts

Upon the action of the TGFB1 protein, hepatic stellate cells are activated and differentiated into myofibroblats which are in charge of tissue repair. Moreover, when TGFB1 is consumed, some of these MFB are eliminated (around 50%) through the

(a) Apoptosis/senescence selection.   (b) Apoptosis/senescence completion.

Figure 1.17: Apoptosis/senescence pathway.

apoptosis/senescence pathway which is modeled in Fig. 1.17, the rest of them becomes inactivated (for 50%) through an inactivation pathway which is depicted in Fig. 1.18.
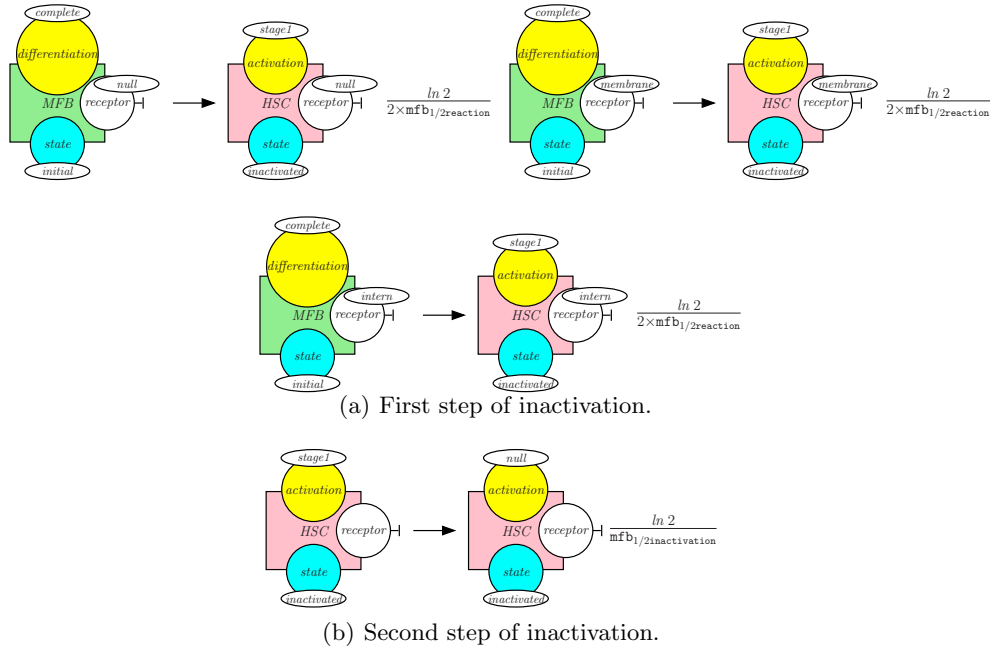
Due to the lack of information, the processes of apoptosis and senescence have been merged. The resulting pathway is made of two steps. The first step (Fig. 1.17(a)) consists in marking the myofibroblast for apoptosis/senescence. It requires the myofibroblast to be in its initial form (which is written *initial*), in the final stage of differentiation (which is written as *complete*), and its receptors to be free. As a result, the state is changed into *apoptosis*. We use a different scale for the differentiation level of the myofibroblasts that are marked for apoptosis, hence the differentiation level is set to *stage1*. The rate constant of the interaction accounts for the half-time reaction of myofibroblasts and the fact that only half of them follow this pathway.

The second step consists in the degradation of the myofibroblast (Fig. 1.17(b)). The degradation rate is computed from the half-life time of the myofibroblasts once they have entered the apoptosis/senescence pathway.

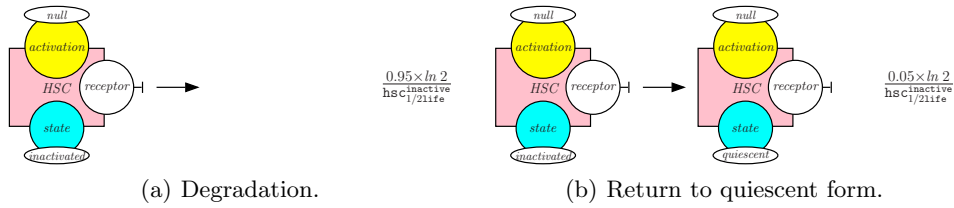## 1.3.3.7. Inactivation of myofibroblasts

Inactivation of myofibroblasts is a two-step process which is depicted in Fig. 1.18. This process turns them back to hepatic stellate cells. Yet, since the cells that come from the inactivation of myofibroblasts have a different behavior, we call them inactivated cells.

The first step (Fig. 1.18(a)) consists in turning occurrences of myofibroblasts into

(a) First step of inactivation.

(b) Second step of inactivation.

Figure 1.18: Inactivation pathway.

hepatic stellate cells in inactivated form (which is written *inactivated*). This requires the occurrences of myofibroblasts to be in their initial state (hence they have not entered the apoptosis pathway, and they cannot come from the reactivation of an inactivated cell) and in the final stage of differentiation. Moreover, binding with occurrences of the TGFB1 protein blocks this process, thus their receptors are assumed to be free. No further assumption is required on the state of the receptors of the occurrences of the myofibroblasts. As a result, the occurrences of myofibroblast are replaced with occurrences of hepatic stellate cell in the initial state and in the first stage of inactivation (which is written *stage1*). The state of the receptors is maintained. Since the agents *MFB* and *HSC* are different, we have to use three rules to model this (as we did already for describing the first step of differentiation of hepatic stellate cells (Fig. 1.14(d))). The rate of these rules are the same than the rules for apoptosis/senescence (Fig. 1.17(a)), since it stands for the behavior of the other half of the myofibroblasts.

(a) Degradation.   (b) Return to quiescent form.

Figure 1.19: Degradation and recycling of inactivated hepatic stellate cells.

The second step of the inactivation of hepatic stellate cells is described in Fig. 1.18(b). It still requires the cell not to be activated by the TGFB1 proteins. As a result, the cell enters the final stage of inactivation (which is encoded by the activation state *null*). The rate of the rule for this second step of inactivation is defined by the half-time of inactivation of hepatic stellate cells.

1.3.3.8. Behavior of inactivated hepatic stellate cells

Inactivated hepatic stellate cells are either slowly eliminated through degradation process, or returned to the quiescent form, or reactivated by a new *TGFB1* stimulation into myofibroblasts.

Inactivated HSC can be eliminated through two processes, apoptosis/senescence or reversing. To undergo those pathways, *HSC* agents must be inactivated (which is written *inactivated*) and not under the process of activation (which is written *null*). Moreover, these *HSC* agents must not be bound to *TGFB1* agents. We choose that 95% of iHSC will enter apoptosis/senescence (Fig. 1.19(a)), the remaining part will reverse into a quiescent state (Fig. 1.19(b)).

The 5% of iHSC undergoing reversion are similar to the one undergoing apoptosis/senescence and reverse into qHSC (Fig. 1.19(b)). The process of reversion forms similar qHSC than those previously described (Fig. 1.13). Nevertheless, the state of the receptor is conserved during reversing. The rates of degradation and return to quiescence rules depend on iHSC half-life time.

Upon a new TGFB1 stimulation, inactivated hepatic stellate cells may be react-

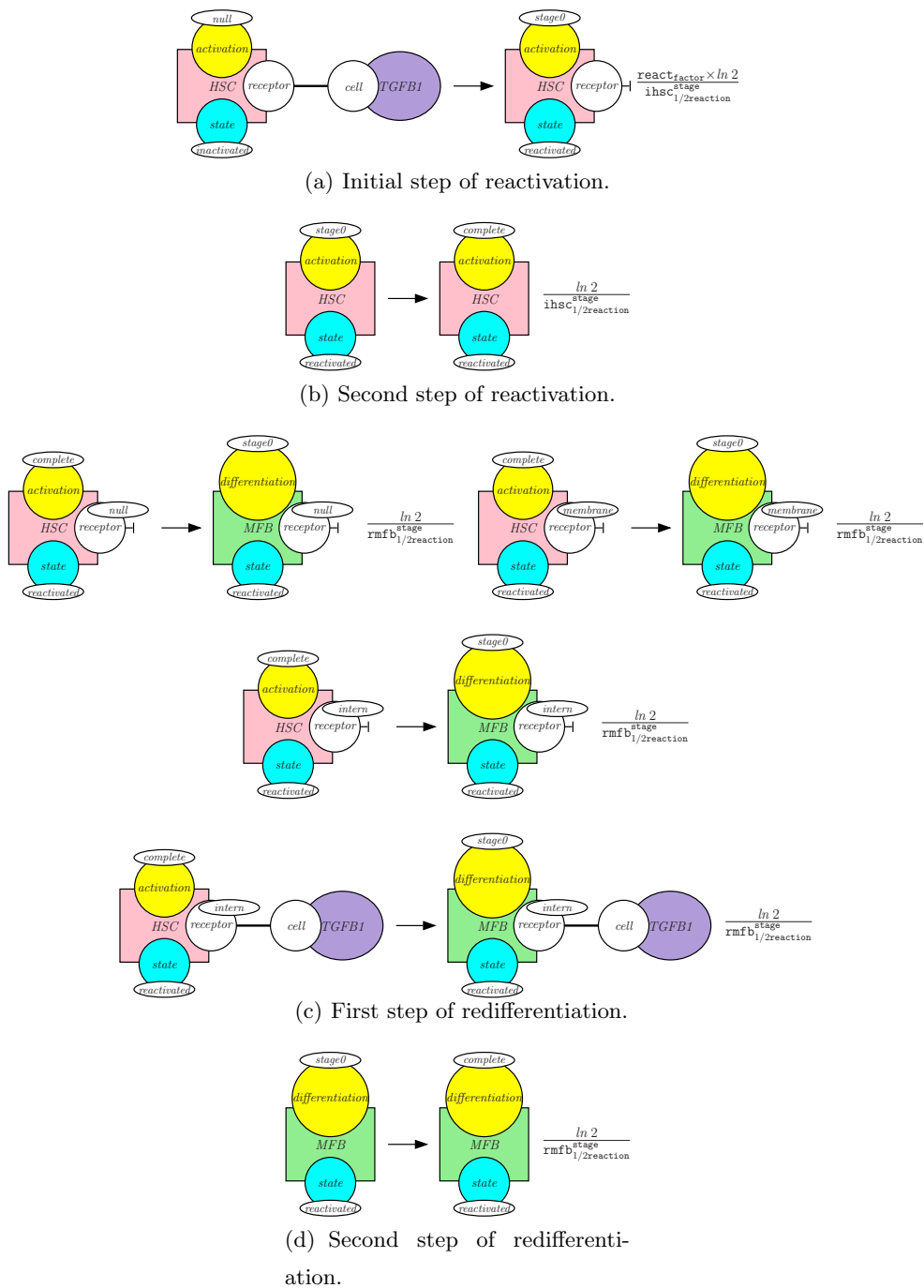(a) Initial step of reactivation.

(b) Second step of reactivation.

(c) First step of redifferentiation.

(d) Second step of redifferenti-
ation.

Figure 1.20: Redifferentiation of inactivated hepatic stellate cells.

ivated. This is a two-step process. In the first step, which is depicted in Fig. 1.20(a), the occurrences of the *TGFB1* agent are consumed and the conformation of the cell changes. They are now in the first stage of reactivation, which is formalized by the state *reactivated* and the activation level *stage0*. The second step, which is written in Fig. 1.20(b), puts the cell in the second stage of reactivation, its activation level is set to *complete*. The rate of the reactivation of iHSC is 4 time the rate of a *HSC* activation, that is why the parameter $\mathtt{ihsc}^{\mathtt{stage}}_{\mathtt{1/2reaction}}$ is divided by the parameter $\mathtt{react}_{\mathtt{factor}}$ (that is set to the value 4).

Reactivated hepatic stellate cells follow the same process of differentiation than activated ones leading to the formation of reactivated myofibroblasts (which is written as *reactivated*) as described in Figs. 1.20(c) and 1.20(d). Similar to the differentiation of initial myofibroblasts (Fig. 1.14), the description of the first step of redifferentiation requires four rules (Fig. 1.20(c)). Those four rules aim at conserving the state of the receptors during the change from occurrences of the agent *HSC* into occurrences of the agent *MFB*. The rate of the rules for *MFB reactivated* stage depends on the half-time of the reaction needed to their formation.

After differentiation of HSC towards MFB, the latter undergoes another step of differentiation. The rule in Fig. 1.20(d) describes the passage into the last stage (written *complete*). The rate of the complete differentiation rule depends on the half-time of the complete differentiation of the reactivated MFB.

1.3.3.9. Proliferation of reactivated cells

The proliferation of reactivated cells is formalized in Fig. 1.21. It works exactly as the proliferation of initial cells. Myofibroblasts and hepatic stellate cells must be in the state *reactivated* and bound to some occurrences of the protein TGFB1. Occurrences of *HSC* and *MFB* may then be duplicated, conserving the state of their sites *state* and *activation* for the formers and the state of their sites *state* and *differentiation* for the latters. The receptors of the newly created cells are not operational yet (which is
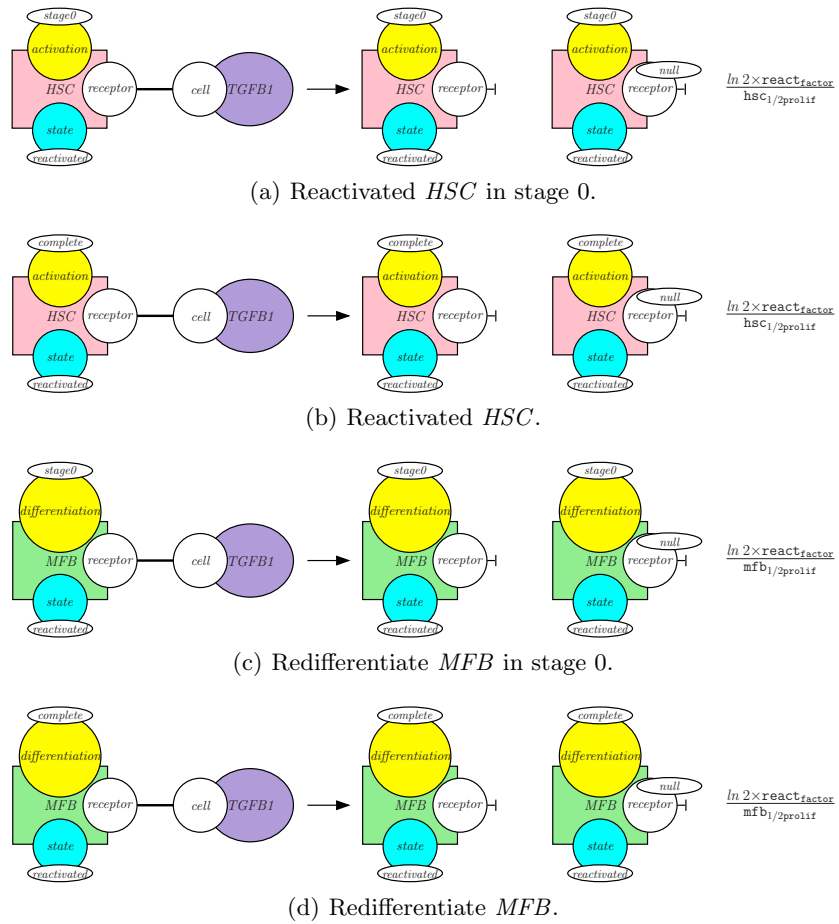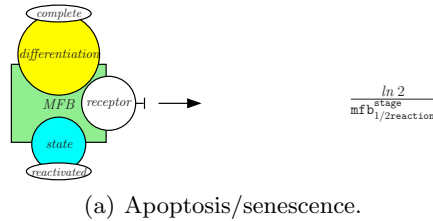
(a) Reactivated *HSC* in stage 0.

(b) Reactivated *HSC*.

(c) Redifferentiate *MFB* in stage 0.

(d) Redifferentiate *MFB*.

Figure 1.21: Proliferation of reactivated cells.

(a) Apoptosis/senescence.

Figure 1.22: Apoptosis/senescence of reactivated myofibroblasts.

written as *null*). Proliferation speed is set by the means of the proliferation half-time of each stage of activation and differentiation. In particular, the factor $\text{react}_{\text{factor}}$ accounts for the fact that reactivated cells proliferate more than firstly activated ones.

1.3.3.10. Degradation of reactivated MFB

Unlike Myofibroblasts, no information shows a potential inactivation for reactivated MFB. Starting from that, the only way for *MFB* in a *reactivated* to disappear is through apoptosis/senescence (Fig. 1.22). It requires the myofibroblasts to be in their reactivated form (which is written *reactivated*), in the final stage of differentiation (which is written as *complete*), and their receptors to be free. The reaction rate is computed from the half-life time of the myofibroblasts.

1.3.3.11. Behavior of receptors

The receptors of cells have also their own behavior which regulates the capability of cells to interact. This holds for the receptors of the protein TGFB1 as well. The behaviors of these receptors in hepatic stellate cells and in myofibroblasts is sketched in Fig. 1.23 by the means of transition systems which show the possible changes from one state to another, labelled with some rate constants. This behavior can be modeled by the means of four pairs of rules (Fig. 1.24) [84]. Those rules regulate the capability of the occurrences of the *HSC* and *MFB* agents to interact with occurrences of the *TGFB1* agent.

(a) The receptors of HSC.

(b) The receptors of MFB.

Figure 1.23: Receptors cycle.

(a) HSC receptors expression.

(b) MFB receptors expression.

(c) HSC receptors recycling.

(d) MFB receptors recycling.

(e) HSC internalized receptors degradation.

(f) MFB internalized receptors degradation.

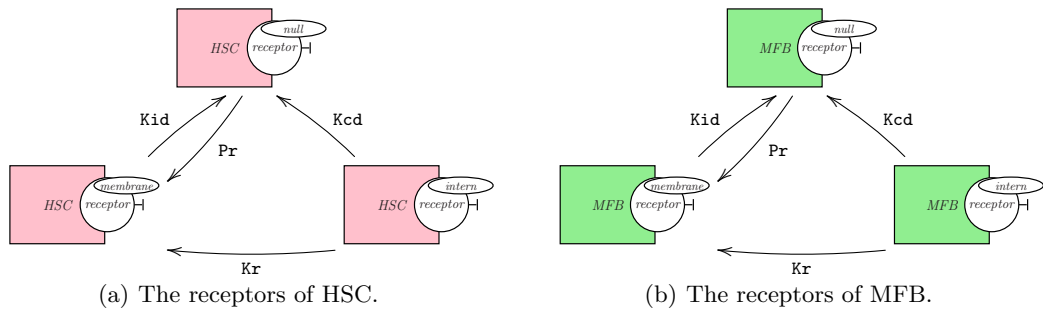(g) HSC externalized receptors degradation.
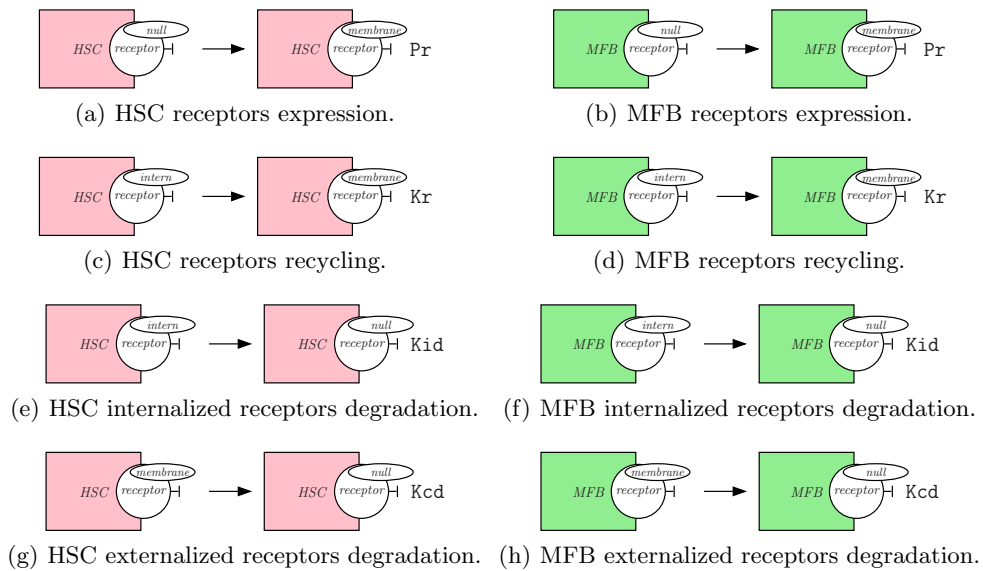

(h) MFB externalized receptors degradation.

Figure 1.24: Behavior of the receptors.

The first pair of rules describes the production of the receptors for *HSC* and *MFB* agents (Figs. 1.24(a) and 1.24(b)). Hepatic stellate cells and myofibroblasts may produce their receptors when they are not present (which is written *null*). Newly produced receptors are expressed at the membrane of the cells (which is written *membrane*). The rate of production of receptors is given by the parameter `Pr`.

The second pair of rules describes the recycling of the receptors from the pool of internalized receptors to the membrane (Fig. 1.24(c) and 1.24(d)). Interacting with their ligands may induce the internalization of the receptors (which is written *intern*). This internalization may lead to the recovery of the receptor permitting its expression at the membrane (which is written *membrane*). The rate of this process is given by the parameter `Kr`.

The third pair of rules describes the degradation of the receptors from the pool of internalized receptors (Figs. 1.24(e) and 1.24(f)). This internalization may also lead to the degradation of the receptors (which is written *null*). The rate of this degradation process is given by the parameter `Kid`.

The fourth pair of rules describes the degradation of the receptors lying on the membrane (Figs. 1.24(g) and 1.24(h)). Receptors have a turnover leading to their degradation even without interaction with the protein *TGFB1*. In that case, they may change from the state *membrane* into the state *null*. The rate of this degradation process is given by the parameter `Kcd`.

Definition	Symbol	Values	Acquisition	Reference
Environment				
Number of Disse's Space	Number_of_Disse_study	10	Calculated	[39]
Number of cells by Disse's Space	HSC_number_by_Disses	4	Calculated	[39]
Total number of cells	$hsc_{eq}^{quiescent}$	40	Calculated	
qHSC				
Half-life	$hsc_{1/2life}^{quiescent}$	90 days	Estimated	[62]
Half-activation time	$hsc_{1/2activation}$	0.17 hour	Calculated	[49]
Max of cell activated	Vr	0, 0.166 , 1	bibliography	[5]
aHSC				
Doubling time	$hsc_{1/2prolif}$	12.64 hours	Estimated	[80]
MFB				
Doubling time	$mfb_{1/2prolif}$	8.43 hours	Estimated	[80]
Half-life	$mfb_{1/2reaction}$	30 hours	Estimated	[62]
Apoptosis proportion		50 of total MFB	Bibliography	[40]
Inactivation proportion		50 of total MFB	Bibliography	[40]
MFB react				
Half-life	$mfb_{1/2reaction}$	25.29 hours	Estimated	
Doubling time	$\dfrac{mfb_{1/2reaction}^{stage}}{react_{factor}}$	1.58 hours	Estimated	[40]
iHSC				
Half-life	$hsc_{1/2life}^{inactive}$	90 days	Bibliography	[62]
Apoptosis proportion		95 of total iHSC	Estimated	[36], [66]
Quiescent return proportion		5 of total iHSC	Estimated	[36], [66]
Half-activation time	$\dfrac{ihsc_{1/2reaction}^{stage}}{react_{factor}}$	0.0425 hours	Estimated	[40]
TGFB1				
Half-life	$tgfb_{1/2life}$	5 min	Bibliography	[85]
Inflammatory input	TGFB_per_wave	100 * number of qHSC	Calculated	
Fixing time	fixing_cell	3 min	Estimated	
Number of receptor by cell	TGFB_factor	7730	Bilbiography	[69]
Receptor				
Recycling rate	Kr	0.5 hours	Bibliography	[84]
Production rate	Pr	0.066 hours	Bibliography	[84]
Degradation rate	Kcd	0.6 hours	Bibliography	[84]
Ligand induced degradation rate	Kid	0.066 hours	Bibliography	[84]
Time of transformation				[15]
aHSC transition stage	$hsc_{1/2reaction}^{stage}$	36.64 hours	Calculated	[5]
MFB differentiation stage	$mfb_{1/2reaction}^{stage}$	46.22 hours	Calculated	[5]
MFB inactivation stage	$mfb_{1/2inactivation}$	18.06 hours	Calculated	[40]
MFB apoptosis stage	$mfb_{1/2life}^{apop}$	3.62 hours	Estimated	
IHSC reactivation stage	$ihsc_{1/2reaction}^{stage}$	6.31 hours	Calculated	[40]
MFB second differentiation stage	$rmfb_{1/2reaction}^{stage}$	6.31 hours	Calculated	[40]
Factors				
IHSC reactivation stage	$react_{factor}$	4	Bibliography	[40]
MFB TGFB1 resistance	$mfb_{tgfb_{resistance}}$	0.416	Bibliography	[80]

Figure 1.25: Parameterization of the model.

1.3.4. Parameters

In the previous section we described the rules present in our model, the table in Fig. 1.25 contains the parameters used to define the rate of these rules. Most of these parameters were found in the literature; for some of them, some calculations and estimations were done. Sometimes, information was lacking, that is why for few

parameters we estimated their values using literature and our knowledge. As explained previously in Sec. 1.3.2, defining the reaction half-time of intermediary stages with respect to the overall completion time of the process is complicated. The computation of those reaction half-times has been made empirically.
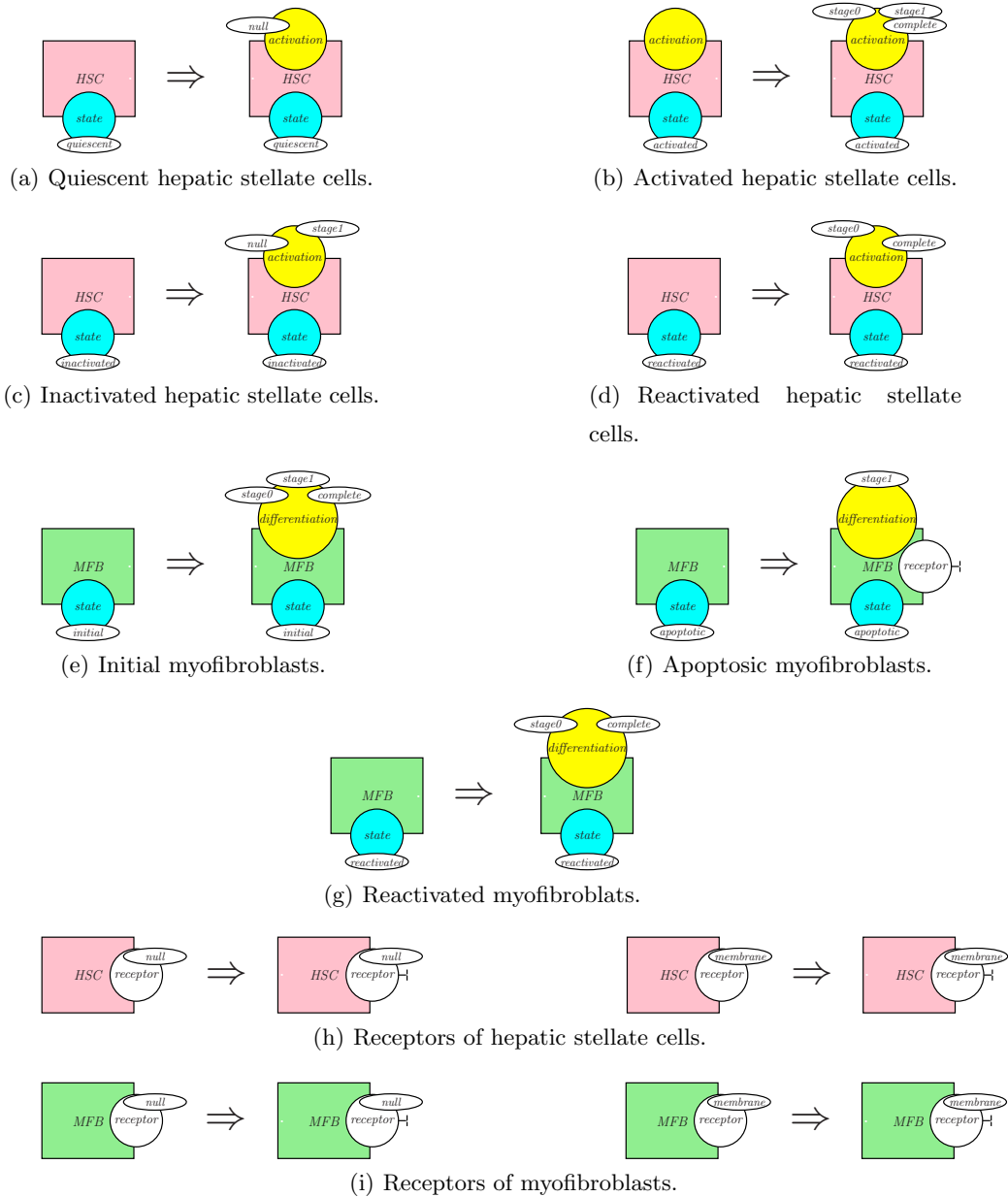
1.4. Results

1.4.1. Static analysis

Before sampling the trajectories of the model, we use the static analyzer KaSa [11] so as to check the structural invariants. The objective is two-fold. Firstly the analyzer may detect some rules that will never be applied in the model. If so, this would mean that some parts are missing and that the model should be completed accordingly. It may also be due to some typos that should be corrected. Secondly, we want to check whether the intended relationships among the state of interactions sites in the different cell conformations hold effectively.

The analysis takes about 0.08 seconds on a 2,3 GHz Intel Core i9 8 cores MacBook Pro. The analysis detects no unapplicable rule and infers the structural invariants that are shown in Fig. 1.26. These invariants take the form of some refinement lemmas. They are written as logical implications. The left hand side is made of a pattern that specifies some conditions about the conformation of a cell. The right hand side completes this pattern with some additional constraints. These constraints are necessary satisfied in every occurrence of the left hand side pattern in a state that the system may take during a potential execution. They take the form of sites that are decorated with an exhaustive list of the states that they may take.

In particular, the analysis detects and proves the following properties about the different stages of the different forms of cells.

- Quiescent hepatic stellate cells may be only in the stage *null* (Fig. 1.26(a));

52

(a) Quiescent hepatic stellate cells.

(b) Activated hepatic stellate cells.

(c) Inactivated hepatic stellate cells.

(d) Reactivated hepatic stellate cells.

(e) Initial myofibroblasts.

(f) Apoptosic myofibroblasts.

(g) Reactivated myofibroblats.

(h) Receptors of hepatic stellate cells.

(i) Receptors of myofibroblasts.

Figure 1.26: The result of static analysis. For each implication, every occurrence of the pattern on the left hand side of an implication in a reachable state necessarily satisfies the conditions described in the right hand side.

- Activated hepatic stellate cells may be only in the stages *stage0*, *stage1*, and *complete* (Fig. 1.26(b));

- Inactivated hepatic stellate cells may be only in the stages *null* and *stage1* (Fig. 1.26(c));

- Reactivated hepatic stellate cells may be only in the stages *stage0* and *complete* (Fig. 1.26(d));

- Myofibroblasts in their initial form may be only in the stages *stage0*, *stage1*, and *complete* (Fig. 1.26(e));

- Myofibroblasts on the way to apoptosis may be only in the stage *stage1* and their receptors may not be bound (Fig. 1.26(f));

- Reactivated myofibrolasts may be only in the stages *stage0* and *complete* (Fig. 1.26(g)).
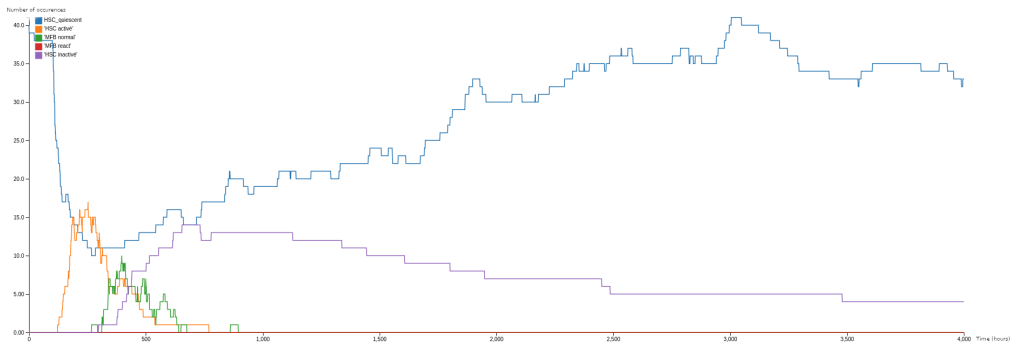
The analysis also discovers that the receptors that are either missing (*null*), or on the membrane (*membrane*) of the cells, are necessarily free both in the case of hepatic stellate cells (Fig. 1.26(h)) and myofibroblats (Fig. 1.26(i)).
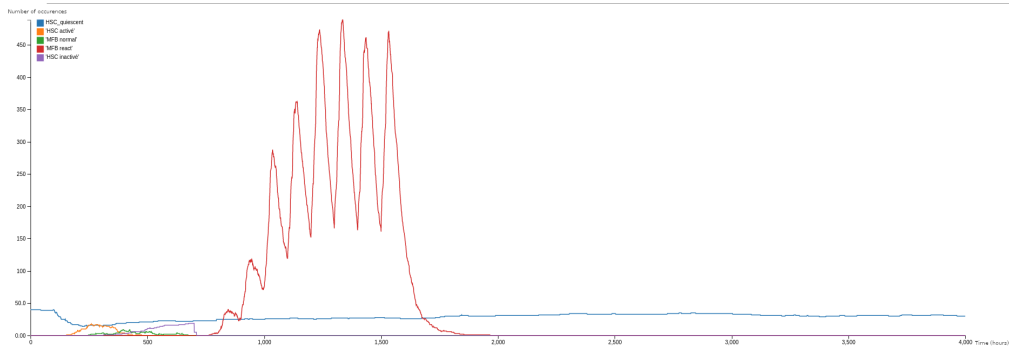
1.4.2. Underlying reaction network

The set of rules may be compiled into a reaction network or equivalently into a system of ordinary differential equations thanks to the tool KaDe [17]. This computation takes about 0.04 seconds on a 2,3 GHz Intel Core i9 8 cores MacBook Pro. The resulting system involves 56 kinds of complexes.

1.4.3. Simulations

Simulating our model provides the results given in Fig. 1.27. In Fig. 1.27(a), we mimic an acute inflammatory aggression by one input of TGFB1. This is indeed the regime under which we have calibrated our model thanks to the information available in the literature. In Fig. 1.27(b), chronic inflammation was created by multiple inputs of TGFB1 (10 times here), and the expected behavior was successfully reproduced by

(a) Response to an acute inflammatory aggression.

(b) Response to chronic inflammation.

Figure 1.27: Simulations of the model. Curves represent the time evolution of the number of occurrences of each form of cells in response to TGFB1 inputs. Time is expressed in hours. In Fig. 1.27(a), one input of TGFB1 at time 100. In Fig. 1.27(b), there are eleven inputs of TGFB1,first at time 100 then starting from time 700, each 100 hours there will be an input of TGFB till getting ten inputs.

our model. Each simulation takes around 14 seconds of CPU to simulate 4000 hours of biological time on a 2,3 GHz Intel Core i9 8 cores MacBook Pro. These curves show the evolution of the number of occurrences of each form of cells with respect to time (in hours). To parameterize the model, we choose to work in a volume corresponding to 10 Disse Spaces, containing among 40 quiescent HSC (calculated from [39]). Adding TGFB1 to the model at time 100 initializes the activation process, transforming quiescent HSC (qHSC, blue) towards activated HSC (aHSC, orange). After that, the newly activated HSC enter in differentiation process leading to the formation of MFB (green). When TGFB1 is completely consumed, MFB are either eliminated by apoptosis/senescence or inactivated (Fig. 1.27(a)). The inactivation of MFB leads to the formation of inactivated HSC (iHSC, purple). The iHSC are either eliminated by apoptosis/senescenceor or reverse towards quiescent HSC. Upon a new stimulation by TGFB1, iHSC can also reverse towards reactivated HSC first, then in reactivated MFB (red). Iterative inputs of TGFB1 in the model quickly favor the accumulation of reactivated myofibroblasts, the inactivation and apoptosis/senescence pathways being surpassed by MFB proliferation and reactivation. Moreover, TGFB1 iterative inputs induce a population switch with diminution of initial cells and augmentation of reactivated cells. After few inputs of TGFB1, some inactivated HSC remain but after more inputs, they completely disappear leading to the saturation of the environment by reactivated MFB 1.27(b). The timing for iterative inputs of TGFB1 is crucial. When too close, only MFP and HSC proliferation is observed. When too separate, much inactivated HSC are eliminated leading to decrease the inflammatory answer.

The primary results are encouraging, showing the dynamic of hepatic stellate cells in function of the TGFB1 inputs. Our model successfully describes the behavior of hepatic stellate cells shown in Sec 1.3.1 and respects the time link to this dynamic; activation process takes around 168 hours, differentiation around 336 hours, inactivation around 664 hours [5, 40]. However, some parts of the model need to be reviewed. Firstly the proliferation process is one of our main problems. Cells proliferation is lower than expected, for example the number of activated HSC should be 3 fold

higher than the number of cells undergoing activation process. It could come from a part that is missing to explain the dynamics of the model (as discussed in conclusion Sect. 1.5). It could also be solved by considering more intermediary stages in Erlang time-distributions (which would lower the standard deviation of their overall process durations, hence favoring cell synchronization, and amplifying the amplitude of their abundance peaks). Also, reactivated cells elimination should be homogenized with initial MFB apoptosis/senescence. Last but not least, the model is quite rigid. Some encoding artifacts have been used. For instance controlling HSC activation by a factor should be removed and the rules firing should control it without any help. Globally, this model needs to be perfected but the primary results are promising.

1.5. Conclusion

This chapter was devoted to the rule-based language Kappa. As a realistic case study, we developed a model for the activation of hepatic stellate cells by the protein TGFB1. We have used knowledge from the literature to calibrate the model according to its expected response to a single acute inflammatory aggression and we were able to reproduce the expected behavior in case of chronic inflammation (up to minor differences in response amplitudes). Beyond the benefit of formalizing executable models, Kappa offers a convenient syntax, close to biochemistry, which eases the modeling process, the potential updates of the model, and the documentation of it. In Sec. 1.3, the description of the Kappa interactions provides a practical road map to navigate among the different elements of the model.

Yet modeling requires a constant search for the most adequate abstraction trade-off. Models may be made arbitrarily precise when detailed information about biochemical mechanisms is available in the literature and when this amount of details does not overcome the *in silico* computational resources. In this model, we favored simplicity while sticking to the experimental observations. We aim to investigate further

different parts of the model and include more details gradually. During this, we will check that the overall behavior of the model is preserved and explore the impact of these updates in specific regimes. Rule-centric approaches make this empiric modeling approach easier.

More specifically, we plan to investigate further on two parts of the model. The first one is about the receptors of cells. With a view to simplifying, we have abstracted all the receptors of each occurrence of cells, by a single Kappa site, called *receptor*. Moreover each agent *TGFB1* indeed stands for a pack of occurrences of the protein TGFB1. The main motivation is to spare computation time. The impact on the model is that all the receptors of an occurrence of a cell are considered to be in the same state. This abstraction could be refined by modeling the receptors of each cell and each occurrence of the TGFB1 protein individually. Yet the cost may become prohibitive both with respect to memory (each occurrence of the TGFB1 protein would be described explicitly) and to computation time (binding between occurrences of the TGFB1 protein and their receptors would occur at a very fast scale and the simulation would have to account for very frequent instances of this event). The issue with memory could be solved easily by the means of counters. The total number of occurrences of free TGFB1 could be modeled as a numerical value (called 'token' in Kappa), and the number of receptors in each of the four different states (*null*, *membrane*, *intern* and free, and *intern* and bound) should be described by the means of some counters, as enabled by a recent extension of Kappa [13].

The second part concerns the memory of cells. As we have seen, activated and re-activated hepatic stellate cells exhibit a different behavior. This means that these cells have a memory and that the history of each cell impacts its further behavior. In the current version, this is modeled by introducing different forms of cells, and providing each form with different capabilities of interaction and rates. This modeling directly operates at the level of the phenotypes of the cells. It would be interesting to better understand where these different behaviors come from by modeling an abstraction of the protein content of the cells. This way, the behavior of each cell would directly

emerge from its protein content. Due to the difference of time-scales and the lack of details in the literature, the protein content cannot be modeled precisely. Instead it could be described by equipping each cell with some abstract counters [13]. The value of these counters could increase upon activation and decrease during cool-down phases, while influencing the capabilities of interaction of the cell.

In Kappa, basic elements are interactions. Their rates provide information about their time-distribution independently from the rest of the system. In the literature, information about reaction durations takes different forms. It can be specified as the time to complete a given ratio of a given process, or defined more phenotypically by the time period between peaks of concentrations. In Kappa, only exponential time-distributions can be assigned to an interaction. This means that the average time of a reaction and its standard deviation are fully entangled. More diverse time-distributions may emerge as a result of sequential composition of reactions. For instance, Erlang time-distributions [41] may be obtained by modeling a process as the sequential composition of k intermediary steps. Intermediary steps reduce the time-variability of the full process and eventually lead to a fixed duration. Yet, this comes with a computational overhead. From a parameterization point of view, rates of intermediary steps are more difficult to guess and must be data-fitted.

One critical point of modeling TGFB1-dependent activation of HSC is the identification of the parameters because of the lack of quantitative values. To overcome this issue, we merged information about HSC activation from both *in vivo* and *in vitro* experiments, however the difference in dynamics of activation/regulation between these two approaches has been already widely documented such as for gene expression [35]. Indeed, seeding quiescent HSC on plastic dishes induces HSC activation through molecular mechanisms that differ in part from mechanisms within liver tissue because of the presence of other cell types and microenvironments. An evolution of the model might be to integrate other cellular components and the biomechanical constraints that play a critical role in activation of HSC and in regulating myofibroblasts phenotype [86, 74]. Importantly, the phenotype of activated HSC is more complex than

59

initially reported and the recent development of single cells RNAseq analyses allows now to demonstrate the heterogeneity of activated HSC [64]. We need to get more information on phenotyping the different HSC species including the MFB states in order to better characterize the different reversibility pathways and to understand what contributes to the disequilibrium towards the disease progression.

While the biomechanical aspect is generally discarded in modeling cell activation-differentiation and cell signaling, modeling biological processes needs to take into account the physical constraints occurring in situ. Obviously extracellular matrix is the paradigm of such constraints and is the major regulator of cell responses [73, 21]. Extracellular matrix is not an inert material supporting cells within tissue but a plastic and complex network associating insoluble molecules such as collagen mostly arranged as supramolecular assemblies, glycoproteins such as fibronectin, and proetoglycans that consist in polypeptide backbone decorated by glycoaminoglycans that confer viscoelasticity and hydrophilic properties. This core matrisome characterized by [57] comprises among 300 proteins and is associated with ECM-affiliated proteins, ECM-regulators and secreted factors [81]. ECM composition and mechanical properties are specific of tissue and change during physiopathological processes such as development, inflammation, wound-healing, fibrosis and cancer. In the liver, the matrisome analyses were recently reviewed in [3] and showed the incomplete and heterogeneous description of this network. Because of the impossibility to catch a spatial and evolutive view of this network, models based on differential equations searched for driving molecules of the network behavior such as the core matrisome proteins, fibrin and collagen [68], and the ECM regulators [4, 58] in order to reduce the complexity.

Using Kappa language allows for the creation of an agent ECM that could interact with cellular and molecular agents. Different states could be attributed to ECM agent such as low, intermediate and high stiffness that control in turn the activation of HSC. While the relationship between ECM stiffness and HSC activation is known since a long time mainly by using 2D cell culture [86, 74], the quantitative evaluation of each molecule implication in stiffness remains to be clarified. Of course, the role

of several molecules in liver stiffness during fibrosis has been characterized and Lysyl Oxidases (LOX) that catalyze cross-linking of collagen and elastin are some of these major actors [79]. The development of 3D-multi cellular hepatic models and microfluidic organ-on-a-chip liver models might be useful to get quantitative data about the contribution of molecular agents in stiffness and activation of HSC [72, 26]. In line with this, our future challenge aims to integrate the Kappa model for extracellular matrix-dependent TGFB1 activation that we recently developed [11]. TGFB1 is synthesized as a latent form (LAP-TGFB1) associated with the Latent TGFB1 binding protein (LTBP1) that sequesters it within the extracellular matrix networks. The release of active TGFB1 depends on enzymatic activities but above all on mechanical strengths involving matrix components and membrane receptors [55]. As a regulatory loop, the activated HSC synthesize extracellular matrix components including TGFB1 and are involved in regulation of matrix plasticity thereby affecting TGFB1 activation. Integrating matrix components implicated in TGFB1-dependent HSC activation might improve the present model and allow us to identify new regulators of the equilibrium between repair and fibrosis.

Bibliography

[1] Jakob L. Andersen, Christoph Flamm, Daniel Merkle, and Peter F. Stadler. A software package for chemically inspired graph transformation. In Rachid Echahed and Mark Minas, editors, *Graph Transformation - 9th International Conference, ICGT 2016, in Memory of Hartmut Ehrig, Held as Part of STAF 2016, Vienna, Austria, July 5-6, 2016, Proceedings*, volume 9761 of *Lecture Notes in Computer Science*, pages 73–88. Springer, 2016. URL https://doi.org/10.1007/978-3-319-40530-8_5.

[2] Oana Andrei and Hélène Kirchner. A rewriting calculus for multigraphs with ports. *Electr. Notes Theor. Comput. Sci.*, 219:67–82, 2008. URL https://doi.org/10.1016/j.entcs.2008.10.035.

[3] Gavin Arteel and Alexandra Naba. The liver matrisome, looking beyond collagens. *JHEP Reports*, 2:100115, 04 2020. doi: 10.1016/j.jhepr.2020.100115.

[4] Wenrui Hao Avner Friedman. Mathematical modeling of liver fibrosis. *Mathematical Biosciences & Engineering*, 14(1):143–164, 2017.

[5] M.G. Bachem, D. Meyer, W. Schäfer, U. Riess, R. Melchior, K.M. Sell, and A.M. Gressner. The response of rat liver perisinusoidal lipocytes to polypeptide growth regulator changes with their transdifferentiation into myofibroblast-like cells in culture. *Journal of hepatology*, 18:40–52, 1993.

[6] Nicolas Behr and Jean Krivine. Compositionality of rewriting rules with conditions. *CoRR*, abs/1904.09322, 2019. URL http://arxiv.org/abs/1904.09322.

[7] Andreea Beica, Jérôme Feret, and Tatjana Petrov. Tropical abstraction of biochemical reaction networks with guarantees. In *Proceedings of SASB 2018, the*

Ninth International Workshop on Static Analysis and Systems Biology, Freiburg, Germany - August 28th, 2018, volume 350 of *Electronic Notes in Theoretical Computer Science*, pages 3–32. Elsevier, 2020. doi: 10.1016/j.entcs.2020.06.002. URL https://doi.org/10.1016/j.entcs.2020.06.002.

[8] Michael Blinov, James R. Faeder, Byron Goldstein, and William S. Hlavacek. Bionetgen: software for rule-based modeling of signal transduction based on the interactions of molecular domains. *Bioinformatics (Oxford, England)*, 20(17): 3289–3291, November 2004.

[9] Pierre Boutillier. The kappa simulator made interactive. In Luca Bortolussi and Guido Sanguinetti, editors, *Computational Methods in Systems Biology - 17th International Conference, CMSB 2019, Trieste, Italy, September 18-20, 2019, Proceedings*, volume 11773 of *Lecture Notes in Computer Science*, pages 296–301. Springer, 2019. doi: 10.1007/978-3-030-31304-3_16. URL https://doi.org/10.1007/978-3-030-31304-3_16.

[10] Pierre Boutillier, Thomas Ehrhard, and Jean Krivine. Incremental update for graph rewriting. In Hongseok Yang, editor, *Programming Languages and Systems - 26th European Symposium on Programming, ESOP 2017, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2017, Uppsala, Sweden, April 22-29, 2017, Proceedings*, volume 10201 of *Lecture Notes in Computer Science*, pages 201–228. Springer, 2017. URL https://doi.org/10.1007/978-3-662-54434-1_8.

[11] Pierre Boutillier, Ferdinanda Camporesi, Jean Coquet, Jérôme Feret, Kim Quyên Lý, Nathalie Théret, and Pierre Vignet. Kasa: A static analyzer for kappa. In Milan Ceska and David Safránek, editors, *Computational Methods in Systems Biology - 16th International Conference, CMSB 2018, Brno, Czech Republic, September 12-14, 2018, Proceedings*, volume 11095 of *Lecture Notes in Computer Science*, pages 285–291. Springer, 2018. URL https://doi.org/10.1007/978-3-319-99429-1_17.

[12] Pierre Boutillier, Mutaamba Maasha, Xing Li, Héctor F. Medina-Abarca, Jean Krivine, Jérôme Feret, Ioana Cristescu, Angus G. Forbes, and Walter Fontana. The kappa platform for rule-based modeling. *Bioinformatics*, 34(13):i583–i592, 2018. URL https://doi.org/10.1093/bioinformatics/bty272.

[13] Pierre Boutillier, Ioana Cristescu, and Jérôme Feret. Counters in kappa: Semantics, simulation, and static analysis. In Luís Caires, editor, *Programming Languages and Systems - 28th European Symposium on Programming, ESOP 2019, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2019, Prague, Czech Republic, April 6-11, 2019, Proceedings*, volume 11423 of *Lecture Notes in Computer Science*, pages 176–204. Springer, 2019. doi: 10.1007/978-3-030-17184-1_7. URL https://doi.org/10.1007/978-3-030-17184-1_7.

[14] Peter Byass. The global burden of liver disease: A challenge for methods and for public health. *BMC medicine*, 12:159, 09 2014. doi: 10.1186/s12916-014-0159-5.

[15] calculator.net, 2008-2021. URL https://www.calculator.net/half-life-calculator.htm.

[16] Ferdinanda Camporesi, Jérôme Feret, and Jonathan Hayman. Context-sensitive flow analyses: A hierarchy of model reductions. In Ashutosh Gupta and Thomas A. Henzinger, editors, *Computational Methods in Systems Biology - 11th International Conference, CMSB 2013, Klosterneuburg, Austria, September 22-24, 2013. Proceedings*, volume 8130 of *Lecture Notes in Computer Science*, pages 220–233. Springer, 2013. doi: 10.1007/978-3-642-40708-6_17. URL https://doi.org/10.1007/978-3-642-40708-6_17.

[17] Ferdinanda Camporesi, Jérôme Feret, and Kim Quyên Lý. Kade: A tool to compile kappa rules into (reduced) ODE models. In Jérôme Feret and Heinz Koeppl, editors, *Computational Methods in Systems Biology - 15th International Conference, CMSB 2017, Darmstadt, Germany, September 27-29, 2017, Pro-*

ceedings, volume 10545 of *Lecture Notes in Computer Science*, pages 291–299. Springer, 2017. doi: 10.1007/978-3-319-67471-1_18. URL https://doi.org/10. 1007/978-3-319-67471-1_18.

[18] Luca Cardelli. Brane calculi. In Vincent Danos and Vincent Schächter, editors, *Computational Methods in Systems Biology, International Conference, CMSB 2004, Paris, France, May 26-28, 2004, Revised Selected Papers*, volume 3082 of *Lecture Notes in Computer Science*, pages 257–278. Springer, 2004. URL https://doi.org/10.1007/978-3-540-25974-9_24.

[19] Luca Cardelli and Andrew D. Gordon. Mobile ambients. In Maurice Nivat, editor, *Foundations of Software Science and Computation Structure, First International Conference, FoSSaCS'98, Held as Part of the European Joint Conferences on the Theory and Practice of Software, ETAPS'98, Lisbon, Portugal, March 28 - April 4, 1998, Proceedings*, volume 1378 of *Lecture Notes in Computer Science*, pages 140–155. Springer, 1998. URL https://doi.org/10.1007/BFb0053547.

[20] Luca Cardelli and Andrew D. Gordon. Mobile ambients. *Theor. Comput. Sci.*, 240(1):177–213, 2000. doi: 10.1016/S0304-3975(99)00231-5. URL https://doi. org/10.1016/S0304-3975(99)00231-5.

[21] Ovijit Chaudhuri, J. Cooper-White, P. Janmey, D. Mooney, and V. Shenoy. Effects of extracellular matrix viscoelasticity on cellular behaviour. *Nature*, 584:535 – 546, 2020.

[22] Federica Ciocchetta and Jane Hillston. Bio-PEPA: A framework for the modelling and analysis of biological systems. *Theoretical Computer Science*, 410(33 – 34): 3065 – 3084, 2009. Concurrent Systems Biology: To Nadia Busi (1968–2007).

[23] Andrea Corradini, Ugo Montanari, Francesca Rossi, Hartmut Ehrig, Reiko Heckel, and Michael Löwe. Algebraic approaches to graph transformation - part I: basic concepts and double pushout approach. In Grzegorz Rozenberg, editor, *Hand-*

book of Graph Grammars and Computing by Graph Transformations, Volume 1: Foundations, pages 163–246. World Scientific, 1997.

[24] Andrea Corradini, Tobias Heindel, Frank Hermann, and Barbara König. Sesquipushout rewriting. In Andrea Corradini, Hartmut Ehrig, Ugo Montanari, Leila Ribeiro, and Grzegorz Rozenberg, editors, *Graph Transformations, Third International Conference, ICGT 2006, Natal, Rio Grande do Norte, Brazil, September 17-23, 2006, Proceedings*, volume 4178 of *Lecture Notes in Computer Science*, pages 30–45. Springer, 2006. URL https://doi.org/10.1007/11841883_4.

[25] Patrick Cousot and Radhia Cousot. Abstract interpretation: A unified lattice model for static analysis of programs by construction or approximation of fixpoints. In Robert M. Graham, Michael A. Harrison, and Ravi Sethi, editors, *Conference Record of the Fourth ACM Symposium on Principles of Programming Languages, Los Angeles, California, USA, January 1977*, pages 238–252. ACM, 1977. URL https://doi.org/10.1145/512950.512973.

[26] Marie Cuvellier, Frédéric Ezan, Hugo Oliveira, Sophie Rose, Jean-Christophe Fricain, Sophie Langouët, Vincent Legagneux, and Georges Baffet. 3d culture of heparg cells in gelma and its application to bioprinting of a multicellular hepatic model. *Biomaterials*, 269:120611, 2021. ISSN 0142-9612. doi: https://doi.org/10.1016/j.biomaterials.2020.120611. URL https://www.sciencedirect.com/science/article/pii/S0142961220308577.

[27] Troels Christoffer Damgaard, Espen Højsgaard, and Jean Krivine. Formal cellular machinery. *Electr. Notes Theor. Comput. Sci.*, 284:55–74, 2012. doi: 10.1016/j.entcs.2012.05.015. URL https://doi.org/10.1016/j.entcs.2012.05.015.

[28] Vincent Danos and Cosimo Laneve. Formal molecular biology. *Theoretical Computer Science*, 325(1):69 – 110, 2004. ISSN 0304-3975. doi: http://dx.doi.org/10.1016/j.tcs.2004.03.065. URL http://www.sciencedirect.com/science/article/pii/S0304397504002336. Computational Systems Biology.

[29] Vincent Danos and Sylvain Pradalier. Projective brane calculus. In Vincent Danos and Vincent Schächter, editors, *Computational Methods in Systems Biology, International Conference, CMSB 2004, Paris, France, May 26-28, 2004, Revised Selected Papers*, volume 3082 of *Lecture Notes in Computer Science*, pages 134–148. Springer, 2004. URL https://doi.org/10.1007/978-3-540-25974-9_11.

[30] Vincent Danos, Jérôme Feret, Walter Fontana, and Jean Krivine. Scalable simulation of cellular signaling networks. In Zhong Shao, editor, *Programming Languages and Systems, 5th Asian Symposium, APLAS 2007, Singapore, November 29-December 1, 2007, Proceedings*, volume 4807 of *Lecture Notes in Computer Science*, pages 139–157. Springer, 2007. URL https://doi.org/10.1007/978-3-540-76637-7_10.

[31] Vincent Danos, Jérôme Feret, Walter Fontana, and Jean Krivine. Abstract interpretation of cellular signalling networks. In Francesco Logozzo, Doron A. Peled, and Lenore D. Zuck, editors, *Verification, Model Checking, and Abstract Interpretation, 9th International Conference, VMCAI 2008, San Francisco, USA, January 7-9, 2008, Proceedings*, volume 4905 of *Lecture Notes in Computer Science*, pages 83–97. Springer, 2008. URL https://doi.org/10.1007/978-3-540-78163-9.

[32] Vincent Danos, Jérôme Feret, Walter Fontana, Russell Harmer, and Jean Krivine. Abstracting the differential semantics of rule-based models: Exact and automated model reduction. In *Proceedings of the 25th Annual IEEE Symposium on Logic in Computer Science, LICS 2010, 11-14 July 2010, Edinburgh, United Kingdom*, pages 362–381. IEEE Computer Society, 2010. ISBN 978-0-7695-4114-3. URL https://doi.org/10.1109/LICS.2010.44.

[33] Vincent Danos, Jérôme Feret, Walter Fontana, Russell Harmer, Jonathan Hayman, Jean Krivine, Christopher D. Thompson-Walsh, and Glynn Winskel. Graphs, rewriting and pathway reconstruction for rule-based models. In Deepak D'Souza, Telikepalli Kavitha, and Jaikumar Radhakrishnan, editors, *IARCS Annual Conference on Foundations of Software Technology and Theoretical*

Computer Science, FSTTCS 2012, December 15-17, 2012, Hyderabad, India, volume 18 of *LIPIcs*, pages 276–288. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2012. URL https://doi.org/10.4230/LIPIcs.FSTTCS.2012.276.

[34] Vincent Danos, Ricardo Honorato-Zimmer, Sebastian Jaramillo-Riveri, and Sandro Stucki. Rigid geometric constraints for Kappa models. In *SASB'12: Post-Proceedings of the 3rd International Workshop on Static Analysis and Systems Biology*, volume 313 of *ENTCS*, pages 23–46. Elsevier, 2015.

[35] Samuele De Minicis, Ekihiro Seki, Hiroshi Uchinami, Johannes Kluwe, Yonghui Zhang, David Brenner, and Robert Schwabe. Gene expression profiles during hepatic stellate cell activation in culture and in vivo. *Gastroenterology*, 132:1937–46, 06 2007. doi: 10.1053/j.gastro.2007.02.033.

[36] Brenda de Oliveira da Silva, Letícia Ferrreira Ramos, and Karen CM Moraes. Molecular interplays in hepatic stellate cells: apoptosis, senescence, and phenotype reversion as cellular connections that modulate liver fibrosis. *Cell biology international*, 41(9):946–959, 2017.

[37] Lorenzo Dematté, Corrado Priami, and Alessandro Romanel. The blenx language: A tutorial. In Marco Bernardo, Pierpaolo Degano, and Gianluigi Zavattaro, editors, *Formal Methods for Computational Systems Biology: 8th International School on Formal Methods for the Design of Computer, Communication, and Software Systems, SFM 2008 Bertinoro, Italy, June 2-7, 2008 Advanced Lectures*, pages 313–365, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg. URL http://dx.doi.org/10.1007/978-3-540-68894-5$_$9.

[38] John W. Eaton, David Bateman, Soren Hauberg, and Rik Wehbring. *GNU Octave version 4.0.0 manual: a high-level interactive language for numerical computations*. Free Software Foundation, 2015. URL http://www.gnu.org/software/octave.

[39] Avner Ehrlich, Daniel Duche, Gladys Ouedraogo, and Yaakov Nahmias. Challenges and opportunities in the design of liver-on-chip microdevices. *Annual Review of Biomedical Engineering*, 21(1):219–239, 2019. doi: 10.1146/annurev-bioeng-060418-052305. URL https://doi.org/10.1146/annurev-bioeng-060418-052305. PMID: 31167098.

[40] Adil El Taghdouini, Mustapha Najimi, Pau Sancho-Bru, Etienne Sokal, and Leo van Grunsven. In vitro reversion of activated primary human hepatic stellate cells. *Fibrogenesis and tissue repair*, 8:14, 08 2015. doi: 10.1186/s13069-015-0031-z.

[41] Agner Krarup Erlang. *Sandsynlighedsregning og Telefonsamtaler*. 1909. URL https://books.google.fr/books?id=xBJZQwAACAAJ.

[42] James R. Faeder, Michael L. Blinov, Byron Goldstein, and William S. Hlavacek. Rule-based modeling of biochemical networks. *Complexity*, 10(4):22–41, 2005. ISSN 1099-0526. doi: 10.1002/cplx.20074. URL http://dx.doi.org/10.1002/cplx.20074.

[43] James R. Faeder, Michael L. Blinov, and William S. Hlavacek. Rule-based modeling of biochemical systems with bionetgen. *Methods Mol Biol*, 500:113–67, 2009.

[44] Jérôme Feret. Gkappa. URL https://github.com/Kappa-Dev/GKappa.

[45] Jérôme Feret. An algebraic approach for inferring and using symmetries in rule-based models. *Electr. Notes Theor. Comput. Sci.*, 316:45–65, 2015. URL https://doi.org/10.1016/j.entcs.2015.06.010.

[46] Jérôme Feret and Kim Quyên Lý. Reachability analysis via orthogonal sets of patterns. *Electr. Notes Theor. Comput. Sci.*, 335:27–48, 2018. URL https://doi.org/10.1016/j.entcs.2018.03.007.

[47] Jérôme Feret, Vincent Danos, Jean Krivine, Russ Harmer, and Walter Fontana. Internal coarse-graining of molecular systems. *PNAS*, 2009.

[48] Jérôme Feret, Heinz Koeppl, and Tatjana Petrov. Stochastic fragments: A framework for the exact reduction of the stochastic semantics of rule-based models. *Int. J. Software and Informatics*, 7(4):527–604, 2013. URL http://www.ijsi.org/ch/reader/view_abstract.aspx?file_no=i173.

[49] Scott L Friedman, Glenn Yamasaki, and Linda Wong. Modulation of transforming growth factor beta receptors of rat lipocytes during the hepatic wound healing response. enhanced binding and reduced gene expression accompany cellular activation in culture and in vivo. *Journal of Biological Chemistry*, 269(14): 10551–10558, 1994.

[50] Daniel T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *The Journal of Physical Chemistry*, 81(25):2340–2361, 1977.

[51] Russ Harmer, Vincent Danos, Jérôme Feret, Jean Krivine, and Walter Fontana. Intrinsic information carriers in combinatorial dynamical systems. *Chaos*, 20, September 2010. URL http://link.aip.org/link/CHAOEH/v20/i3/p037108/s1.

[52] Dieter Häussinger and Claus Kordes. Space of disse: a stem cell niche in the liver. *Biological chemistry*, 401(1):81–95, 2019.

[53] Monika Heiner and Ina Koch. Petri net based model validation in systems biology. In Jordi Cortadella and Wolfgang Reisig, editors, *Applications and Theory of Petri Nets 2004: 25th International Conference, ICATPN 2004, Bologna, Italy, June 21–25, 2004. Proceedings*, pages 216–237. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004. URL http://dx.doi.org/10.1007/978-3-540-27793-4$_$13.

[54] Tobias Helms, Tom Warnke, Carsten Maus, and Adelinde M. Uhrmacher. Semantics and efficient simulation algorithms of an expressive multilevel modeling language. *ACM Trans. Model. Comput. Simul.*, 27(2):8:1–8:25, 2017. doi: 10.1145/2998499. URL https://doi.org/10.1145/2998499.

[55] Boris Hinz. The extracellular matrix and transforming growth factor-β1: Tale of a strained relationship. *Matrix Biology*, 47:54–65, 2015. ISSN 0945-053X. doi:

https://doi.org/10.1016/j.matbio.2015.05.006. URL https://www.sciencedirect.com/science/article/pii/S0945053X15001055.

[56] Michael Hucka, Frank T. Bergmann, Stefan Hoops, Sarah M. Keating, Sven Sahle, James C. Schaff, Lucian P. Smith, and Darren J. Wilkinson. The systems biology markup language (sbml): Language specification for level 3 version 1 core, 2010.

[57] Richard Hynes and Alexandra Naba. Overview of the matrisome–an inventory of extracellular matrix constituents and functions. *Cold Spring Harbor perspectives in biology*, 4:a004903, 09 2011. doi: 10.1101/cshperspect.a004903.

[58] Yu-Fang Jin, Hai-Chao Han, Jamie Berger, Qiuxia Dai, and Merry Lindsey. Combining experimental and mathematical modeling to reveal mechanisms of macrophage-dependent left ventricle remodeling. *BMC systems biology*, 5:60, 05 2011. doi: 10.1186/1752-0509-5-60.

[59] Mathias John, Cédric Lhoussaine, Joachim Niehren, and Cristian Versari. Biochemical reaction rules with constraints. In *Programming Languages and Systems - 20th European Symposium on Programming, ESOP 2011, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2011, Saarbrücken, Germany, March 26-April 3, 2011. Proceedings*, volume 6602 of *Lecture Notes in Computer Science*, pages 338–357. Springer, 2011. URL http://dx.doi.org/10.1007/978-3-642-19718-5.

[60] Ozan Kahramanogullari and Luca Cardelli. An intuitive modelling interface for systems biology. *Int. J. Software and Informatics*, 7(4):655–674, 2013. URL http://www.ijsi.org/ch/reader/view_abstract.aspx?file_no=i177.

[61] Tatiana Kisseleva and David Brenner. Molecular and cellular mechanisms of liver fibrosis and its regression. *Nature reviews. Gastroenterology & hepatology*, 18(3): 151–166, March 2021. ISSN 1759-5045. doi: 10.1038/s41575-020-00372-7. URL https://doi.org/10.1038/s41575-020-00372-7.

[62] Tatiana Kisseleva, Min Cong, YongHan Paik, David Scholten, Chunyan Jiang, Chris Benner, Keiko Iwaisako, Thomas Moore-Morris, Brian Scott, Hidekazu Tsukamoto, Sylvia M. Evans, Wolfgang Dillmann, Christopher K. Glass, and David A. Brenner. Myofibroblasts revert to an inactive phenotype during regression of liver fibrosis. *Proceedings of the National Academy of Sciences*, 109 (24):9448–9453, 2012. ISSN 0027-8424. doi: 10.1073/pnas.1201840109. URL https://www.pnas.org/content/109/24/9448.

[63] Agnes Köhler, Jean Krivine, and Jakob Vidmar. A rule-based model of base excision repair. In Pedro Mendes, Joseph O. Dada, and Kieran Smallbone, editors, *Computational Methods in Systems Biology - 12th International Conference, CMSB 2014, Manchester, UK, November 17-19, 2014, Proceedings*, volume 8859 of *Lecture Notes in Computer Science*, pages 173–195. Springer, 2014. URL https://doi.org/10.1007/978-3-319-12982-2_13.

[64] Oliver Krenkel, Jana Hundertmark, Thomas Ritz, Ralf Weiskirchen, and Frank Tacke. Single cell rna sequencing identifies subsets of hepatic stellate cells and myofibroblastsin liver fibrosis. *Cells*, 8:503, 05 2019. doi: 10.3390/cells8050503.

[65] Marta Z. Kwiatkowska, Gethin Norman, and David Parker. PRISM 4.0: Verification of probabilistic real-time systems. In Ganesh Gopalakrishnan and Shaz Qadeer, editors, *Computer Aided Verification - 23rd International Conference, CAV 2011, Snowbird, UT, USA, July 14-20, 2011. Proceedings*, volume 6806 of *Lecture Notes in Computer Science*, pages 585–591. Springer, 2011. URL https://doi.org/10.1007/978-3-642-22110-1_47.

[66] Xiao Liu, Jun Xu, David Brenner, and Tatiana Kisseleva. Reversibility of liver fibrosis and inactivation of fibrogenic myofibroblasts. *Current pathobiology reports*, 1:209–214, 09 2013. doi: 10.1007/s40139-013-0018-7.

[67] Michael Löwe. Algebraic approach to single-pushout graph transformation.

Theor. Comput. Sci., 109(1&2):181–224, 1993. URL https://doi.org/10.1016/0304-3975(93)90068-5.

[68] Andrea Malandrino, Michael Mak, Xavier Trepat, and Roger D. Kamm. Nonelastic remodeling of the 3d extracellular matrix by cell-generated forces. *bioRxiv*, 2017. doi: 10.1101/193458. URL https://www.biorxiv.org/content/early/2017/09/27/193458.

[69] Joan Massague and Betsy Like. Cellular receptors for type beta transforming growth factor. ligand binding and affinity labeling in human and rodent cell lines. *Journal of Biological Chemistry*, 260(5):2636–2645, 1985.

[70] MATLAB. *version 9.2*. The MathWorks Inc., Natick, Massachusetts, 2017.

[71] Michael B. Monagan, Keith O. Geddes, K. Michael Heal, George Labahn, Stefan M. Vorkoetter, James McCarron, and Paul DeMarco. *Maple 10 Programming Guide*. Maplesoft, 2005.

[72] Ehsanollah Moradi, Sasan Jalili-Firoozinezhad, and Mehran Solati-Hashjin. Microfluidic organ-on-a-chip models of human liver tissue. *Acta Biomaterialia*, 116: 67–83, 2020. ISSN 1742-7061. doi: https://doi.org/10.1016/j.actbio.2020.08.041. URL https://www.sciencedirect.com/science/article/pii/S1742706120305110.

[73] Jonathon M. Muncie and Valerie M. Weaver. Chapter one - the physical and biochemical properties of the extracellular matrix regulate cell fate. In Eveline S. Litscher and Paul M. Wassarman, editors, *Extracellular Matrix and Egg Coats*, volume 130 of *Current Topics in Developmental Biology*, pages 1–37. Academic Press, 2018. doi: https://doi.org/10.1016/bs.ctdb.2018.02.002. URL https://www.sciencedirect.com/science/article/pii/S0070215318300346.

[74] Abby Olsen, Steven Bloomer, Erick Chan, Marianna Gaca, Penelope Georges, Bridget Sackey, Masayuki Uemura, Paul Janmey, and Rebecca Wells. Hepatic stellate cells require a stiff environment for myofibroblastic differentiation. *Amer-*

ican journal of physiology. Gastrointestinal and liver physiology, 301:G110–8, 04 2011. doi: 10.1152/ajpgi.00412.2010.

[75] Tatjana Petrov, Jérôme Feret, and Heinz Koeppl. Reconstructing species-based dynamics from reduced stochastic rule-based models. In Oliver Rose and Adelinde M. Uhrmacher, editors, *Winter Simulation Conference, WSC '12, Berlin, Germany, December 9-12, 2012*, pages 225:1–225:15. WSC, 2012. ISBN 978-1-4673-4779-2. URL https://doi.org/10.1109/WSC.2012.6465241.

[76] Mark F. Pittenger, Dennis E. Discher, Bruno M. Péaultand Donald G. Phinney, Joshua M. Hare, and Arnold I. Caplan. Mesenchymal stem cell perspective: cell biology to clinical progress. npj regen med. 2019.

[77] Aviv Regev, William Silverman, and Ehud Shapiro. Representation and simulation of biochemical processes using the pi-calculus process algebra. In R. B. Altman, A. K. Dunker, L. Hunter, and T. E. Klein, editors, *Pacific Symposium on Biocomputing, Volume 6*, pages 459–470, Singapore, 2001.

[78] Aviv Regev, E. M. Panina, William Silverman, Luca Cardelli, and E. Y. Shapiro. Bioambients: An abstraction for biological compartments. *TCS*, 325(1):141–167, 2004.

[79] Takaoki Saneyasu, Riaz Akhtar, and Takao Sakai. Molecular cues guiding matrix stiffness in liver fibrosis. *BioMed Research International*, 2016:1–11, 01 2016. doi: 10.1155/2016/2646212.

[80] Bernd Schnabl, Cynthia A. Bradham, Brydon L. Bennett, Anthony M. Manning, Branko Stefanovic, and David A. Brenner. Tak1/jnk and p38 have opposite effects on rat hepatic stellate cells. *Hepatology*, 34(5):953–963, 2001. ISSN 0270-9139. doi: https://doi.org/10.1053/jhep.2001.28790. URL https://www.sciencedirect.com/science/article/pii/S0270913901529845.

[81] Xinhao Shao, Isra N Taha, Karl R Clauser, Yu (Tom) Gao, and Alexandra Naba. MatrisomeDB: the ECM-protein knowledge database. *Nucleic Acids Research*,

48(D1):D1136–D1144, 10 2019. ISSN 0305-1048. doi: 10.1093/nar/gkz849. URL https://doi.org/10.1093/nar/gkz849.

[82] Donald Stewart. Spatial biomodelling, 2010. Master thesis, School of Informatics, University of Edinburgh.

[83] Takuma Tsuchida and Scott L Friedman. Mechanisms of hepatic stellate cell activation. *Nature reviews. Gastroenterology & hepatology*, 14(7):397–411, July 2017. ISSN 1759-5045. doi: 10.1038/nrgastro.2017.38. URL https://doi.org/10.1038/nrgastro.2017.38.

[84] Jose Vilar, Ronald Jansen, and Chris Sander. Signal processing in the tgf-β superfamily ligand-receptor network. *PLoS computational biology*, 2:e3, 02 2006. doi: 10.1371/journal.pcbi.0020003.

[85] Laiage M Wakefield, Thomas S Winokur, Robin S Hollands, Karen Christopherson, Arthur D Levinson, Michael B Sporn, et al. Recombinant latent transforming growth factor beta 1 has a longer plasma half-life in rats than active transforming growth factor beta 1, and a different tissue distribution. *The Journal of clinical investigation*, 86(6):1976–1984, 1990.

[86] Rebecca Wells. The role of matrix stiffness in hepatic stellate cell activation and liver fibrosis. *Journal of clinical gastroenterology*, 39:S158–61, 05 2005. doi: 10.1097/01.mcg.0000155516.02468.0f.

[87] Inc. Wolfram Research. *Mathematica*. Wolfram Research, Inc., 2017.