Extracting Proofs from Tabled Proof Search

Dale Miller, Alwen Tiu

Extracting Proofs from Tabled Proof Search[⋆]

Dale Miller[1] and Alwen Tiu[2]

[1] INRIA-Saclay & LIX/École Polytechnique
[2] Research School of Computer Science, The Australian National University &
School of Computer Engineering, Nanyang Technological University

Abstract. We consider the problem of model checking specifications involving co-inductive definitions such as are available for bisimulation. A proof search approach to model checking with such specifications often involves state exploration. We consider four different tabling strategies that can minimize such exploration significantly. In general, tabling involves storing previously proved subgoals and reusing (instead of reproving) them in proof search. In the case of co-inductive proof search, tables allow a limited form of loop checking, which is often necessary for, say, checking bisimulation of non-terminating processes. We enhance the notion of tabled proof search by allowing a limited deduction from tabled entries when performing table lookup. The main problem with this enhanced tabling method is that it is generally unsound when co-inductive definitions are involved and when tabled entries contain unproved entries. We design a proof system with tables and show that by managing tabled entries carefully, one would still be able to obtain a sound proof system. That is, we show how one can extract a post-fixed point from a tabled proof for a co-inductive goal. We then apply this idea to the technique of bisimulation "up-to" commonly used in process algebra.

1 Introduction

Model checking and theorem proving are usually considered two distinct techniques in formal verification: the former is concerned mainly with satisfiability in a given model while the latter is concerned mainly with provability (e.g., validity in all models). Viewed algorithmically, model checking can be loosely characterized as a model exploration technique (e.g., explorations of states in a transition systems, worlds in a Kripke structure, etc). We adopt this view here. When inference and proof are enriched to contain flexible treatments of (least and greatest) fixed points, model checking can be seen as deduction. As such, the model checkers can be expected to output *proof certificates* justifying their completed state explorations in a manner similar to what one might expect to have output from automatic or interactive theorem provers.

[⋆] We thank the anonymous referees for their helpful comments. The first author has been supported by the ERC Advanced Grant ProofCert and the second author has been supported by the ARC Discovery Grant DP110103173.

In this paper, formal proofs will be based on the Linc sequent calculus [15, 17] (see Section 2), which generalizes Gentzen's sequent calculus LJ for intuitionistic logic with induction and co-induction. We shall also focus on the Bedwyr model checking implementation of part of Linc [1], particularly the form of *tabled deduction* that is implemented in that system. Bedwyr has been most successfully applied to domains where model checking is performed on syntactically rich domains (involving expressions taken from process calculi and programming languages) instead of more simple state-like domains comprised of tuples of booleans, small integers, etc.

1.1 Model checking as proof search

In this paper, we address the problem of integrating (co-)inductively proved theorems with model checking and we will use bisimulation as a specific and important example. In the setting of Linc, bisimulation is defined as the greatest fixed point of the following recursive definition.

$$\begin{aligned}\mathrm{bisim}(P,Q) \overset{\nu}{=} {} & [\forall P' \forall A.\ P \xrightarrow{A} P' \supset \exists Q'.\ Q \xrightarrow{A} Q' \wedge \mathrm{bisim}(P',Q')] \wedge \\ & [\forall Q' \forall A.\ Q \xrightarrow{A} Q' \supset \exists P'.\ P \xrightarrow{A} P' \wedge \mathrm{bisim}(Q',P')]\end{aligned}$$

Bedwyr's proof search mechanism will turn this definition into a state exploration procedure. Such a direct and intimate connection between bisimulation defined as a logical formula and an algorithmic state exploration algorithm provides at least two important novelties. First, logical encodings may clarify some aspects of the theories being encoded: for example, the difference between late bisimulation and open bisimulation for the π-calculus can be explained as the distinction between intuitionistic and classical logic, i.e., the presence (or absence) of the excluded middle principle applied to the equality of names [16]. Second, since model checking can be seen as building Linc proofs, a checker should be able to output a formal *proof certificate*: for example, successful proof search in Bedwyr for a query concerning bisimilarity of two processes or satisfiability of a modal formulas by a process yields a Linc proof which can be extracted and checked independently.

Although such logical encoding of state exploration techniques is in principle straightforward, naive proof search techniques can yield inefficient algorithmic search and proof certificates that are unacceptably large. One way to address these problems is to use *tabled deduction* so that proved subgoals can be shared and not reproved. For example, Bedwyr stores certain (sub)goals that have been proved and attempts to reuse them when proving other (sub)goals. The tabling of proved subgoals is not, however, sufficient to deal with model checking of potentially non-terminating systems. For example, to prove bisimilarity of simple processes such as $!a$ and $!(a+a)$, a naive unfolding of the processes will not terminate (of course, checking bisimilarity is undecidable so no fixed strategy will yield bounded search in all cases). A more clever approach to showing bisimulation is the *bisimulation up-to* technique [11], which can employ additional information about bisimulation in order to reduce the size of the relation (the table) needed to demonstrate bisimilarity.

1.2 Four tabling strategies

In this paper, we examine how tabled deduction can be used to build a bisimulation as well as a bisimulation-up-to. In particular, we examine four tabling strategies in model checking that allow building smaller witnesses (ultimately, proof certificates) of relationships on possibly non-terminating processes. In each case, the main technical difficulties involves extracting an independently checkable proof certificate: obviously, such extraction guarantees soundness of the tabling method. As a case study, we show how bisimulation up-to techniques for process calculi can be encoded in proof search in one of the tabling strategies, and show how proof certificates can be generated.

Since we view model checking as a certain process for building a proof, at any particular moment, the state of that process can be abstracted to be roughly two items: the *partial proof* and the *table*. The first of these is a tree structure of nodes that is labeled by atomic formulas. Nodes are either *leaf* nodes or *interior* nodes and both of these classes can be further divided between *open* and *closed*. A *closed leaf node* is one that has been proved and an *open leaf node* is one for which no proof has yet been found. A *closed interior node* is one all of whose descendant leaf nodes are closed and an *open interior node* is one with some descendant leaf node that is open. The second component of the model checker's state, the table, is a set of node occurrences. We shall always allow a table to contain closed leaf occurrences from the associated partial proof. We shall also use the term *history atom* to describe a formula that labels an interior node.

Two independent choices are available in describing a tabling strategy: the first choice is between allowing or not allowing history atoms into the table and the second is allowing the table to infer an atom by simply checking its membership in the table or by allowing a deduction from tabled atoms and some assumed set of theories. As an example of this latter choice, consider a table that contains the atomic statements $(\mathrm{bisim}(p_1, p_2))$ and $(\mathrm{bisim}(p_2, p_3))$. If the table is only used to infer its members, we can infer these two atoms. If we have proved elsewhere (using a proof assistant that understands (co-)induction) that the $(\mathrm{bisim}(\cdot, \cdot))$ relationship is transitive, then a table that incorporates that theorem could also conclude $(\mathrm{bisim}(p_1, p_3))$. More formally, if R is the set of atoms in a table and T is a set of theories, then we are allowed to infer the atomic formula G from this table if formula $(R \wedge T) \to G$ is provable. In this paper, we shall assume that T is a set of hereditary Harrop (hH) formulas (formulas containing only conjunction, implication, and universal quantifiers: these formulas subsume Horn clauses and are basis of λProlog [7]). While in most of our examples, such hH formulas will form a simple, decidable theory, we shall not assume a priori that theories are, in fact, decidable.

We identify the following four tabling strategies.

I History atoms are not tabled; the table only infers its members.

II History atoms are not tabled; the table uses theories to infer additional atoms.
III History atoms can be tabled; the table only infers its members.
IV History atoms can be tabled; the table uses theories to infer additional atoms.

The first two strategies yield proof certificates that simply use the cut rule: these two strategies are always sound as long as the theory (in Strategy II) is known to be valid, i.e., proved elsewhere using (co-)inductive techniques. Actually, Strategy I collapses into Strategy II if the empty set is an allowed theory. Soundness of these two strategies is not difficult to establish and it follows the work presented in [8]. Strategy III is sound only when the tabled entries are co-inductive predicates: furthermore, a proof certificate can always be constructed and it will be essentially a post-fixed point found within the table. When tabled entries are restricted to co-inductive predicates, the last strategy corresponds to the bisimulation up-to technique, as described in, say [10]. In this case, the theory $\mathcal{T}$ that is used to expand the table no longer corresponds to a lemma in the meta logic. They instead encode functions on relations, and soundness of a tabled proof in this case depends on the soundness of these functions, i.e., whether they allow one to construct a post-fixed point of a co-inductive definition. We shall focus on strategy III and IV in this paper, but the soundness results for Strategy I and II can be found in [9, Appendix B].

There is significant precedent in the literature related to the use of history atoms to capture aspects of co-inductive proofs, notably works on *cyclic proofs* for logics with induction and co-induction [14, 4]. In particular, proof search strategies similar to strategy III above have also been used in cyclic theorem provers [3] and tabling methods in co-inductive logic programming (see e.g. [5, 13]). Soundness of cyclic proofs (inductive or co-inductive) is not difficult to establish semantically and there are well known syntactic criteria for cyclic proof systems to be sound, e.g., the notion of a *progressing trace* that dates back to work on modal μ-calculus [18] and its first-order extensions [14]. However, there are two main distinguishing features of our work compared to these related work:

First, we do not justify the soundness of cyclic proofs via semantics but instead we translate cyclic proofs into a more standard proof system that uses explicit (co-)induction rules, e.g., the logic Linc or higher-order logic, for which the issue of soundness has been well established and for which there is a well developed proof theory. Such translation is in general difficult: Sprenger and Dam in [14] provide such a general translation but it requires annotations of fixed point operators with ordinals. For annotation-free cyclic proof systems such as that of Brotherston [4], the translation from cyclic proofs to proofs with explicit (co-)induction rules remains an open problem. While our cyclic proof system (for strategy III) does not introduce explicit ordinal annotations, the kind of cyclic structures allowed in that proof system is much simpler than in [14, 4] and forbids cross-branch cycles and mutually recursive definitions. We are thus able to give simple constructions of proofs with explicit (co-)induction rules.

Second, our strategy IV has no counterpart in literature of cyclic proofs. The interpretation of such a cyclic proof is not a straightforward construction of post fixed points since the circularity induced by applications of the theory component in this strategy does not obey the notion of progressing traces underlying existing cyclic proof systems mentioned above. Of course semantic soundness for such applications is known in the literature of bisimulation up-to [10]; our work can be seen as a formal logical formulation of the soundness criteria in [10].

We note that strategies I to III have been implemented in the current development version of Bedwyr, and a preliminary version of strategy IV is being developed at the Parsifal team at INRIA. An example in [9, Section A] illustrates the use of strategy IV to prove bisimilarity of two non-terminating processes, something which is not possible with other strategies.

In Section 2, we present the proof system for intuitionistic logic that we use in the rest of this paper. In Section 3, we present a proof system which uses tables. The four tabling strategies outlined above are differentiated in this tabled system by a function that filters appropriate elements of the tables and the theories that are assumed in the proof. Soundness of strategy III is proved in Section 4, where we show how to construct a post fixed point from tabled entries. In Section 5 we show how to interpret theories as up-to functions and tabled entries as a post fixed point "up-to". In Section 6, we show how compositions of up-to functions can be encoded as compositions of logical theories. We then show, via a permutation argument, that up-to functions can be freely and soundly composed, provided certain conditions related to how these theories permute over each other hold. In Section 7, we discuss further work. The appendix of the companion paper [9] contains several proofs that are omitted in the main text.

2 Backgrounds

We give an overview of the logical framework used as the foundation of this work, i.e., the logic Linc [15], and the bisimulation up-to techniques [11, 10].

The Linc logic is essentially a version of Church's Simple Theory of Types with the following differences. (i) Linc is based on intuitionistic provability (described here using a two-sided sequent calculus similar to Gentzen's LJ proof system). (ii) The type of quantified variables are restricted to those not containing the type of propositions (i.e., the type o in Church's notation): thus, Linc does not allow predicate quantification. (iii) Linc also contains *free equality*, i.e., equality in the term model, and *inductive* and *co-inductive* definitions as *logical connectives* and these will be given introduction rules in the sequent calculus. (iv) Finally, Linc also contains the ∇-quantifier (see, for example, [16]) but we can safely ignore it in this paper.

Each predicate symbol in Linc is given a designation as either *undefined*, *inductive* or *co-inductive*. An undefined predicate is the usual one in first-order intuitionistic logic, i.e., its interpretation in a model is allowed to be an arbitrary subset of the domain of interpretation. To each (co-)inductive predicate p, we associate a *definition*, i.e., a formula possibly containing occurrences of p. Formally, we write $p\,\vec{x} \overset{\mu}{=} D\ p\ \vec{x}$ to denote an inductive definition of p. Here D is an abstraction, containing no occurrences of p, that is applied to p and variables

$$\frac{\{\Gamma[\rho] \longrightarrow C[\rho]\}_{\rho \in \mathbb{U}(s,t)}}{s = t, \Gamma \longrightarrow C}\ \mathrm{eq}\mathcal{L} \qquad \frac{}{\Gamma \longrightarrow t = t}\ \mathrm{eq}\mathcal{R}$$

$$\frac{B\,S\,\vec{y} \longrightarrow S\,\vec{y} \quad \Gamma, S\,\vec{t} \longrightarrow C}{\Gamma, p\,\vec{t} \longrightarrow C}\ \mathrm{I}\mathcal{L},\ p\,\vec{x} \overset{\mu}{=} B\,p\,\vec{x} \qquad \frac{\Gamma \longrightarrow B\,p\,\vec{t}}{\Gamma \longrightarrow p\,\vec{t}}\ \mathrm{I}\mathcal{R},\ p\,\vec{x} \overset{\mu}{=} B\,p\,\vec{x}$$

$$\frac{B\,p\,\vec{t}, \Gamma \longrightarrow C}{p\,\vec{t}, \Gamma \longrightarrow C}\ \mathrm{CI}\mathcal{L},\ p\,\vec{x} \overset{\nu}{=} B\,p\,\vec{x} \qquad \frac{\Gamma \longrightarrow S\,\vec{t} \quad S\,\vec{x} \longrightarrow B\,S\,\vec{x}}{\Gamma \longrightarrow p\,\vec{t}}\ \mathrm{CI}\mathcal{R},\ p\,\vec{x} \overset{\nu}{=} B\,p\,\vec{x}$$

Fig. 1. The Linc inference rules for equality and the least and greatest fixed points

$\vec{x}$. We shall require that p occurs strictly positively in $D\ p\ \vec{x}$. A co-inductive definition is similarly defined, with $\overset{\nu}{=}$ replacing $\overset{\mu}{=}$. We write $p\,\vec{x} \overset{\triangle}{=} D\ p\ \vec{x}$ to denote either an inductive or a co-inductive definition.

In Section 1.1, the definition of bisimulation illustrates this scheme by setting the schema variable D to be the λ-term with abstractions $\lambda\mathrm{bisim}\lambda P \lambda Q$ and with its body being the entire right-hand-side of the definition. Further restrictions are needed, e.g., restrictions on mutual recursions between inductive and co-inductive definitions, to guarantee cut-elimination; see [15, 17] for details.

We consider terms as equal modulo α-conversion and assume the usual notion of capture-avoiding substitutions for λ-calculus. The application of a substitution θ to a term t is written $t[\theta]$. This notation extends to application of substitutions to multisets of formulas, i.e., $\Gamma[\theta] = \{B[\theta] \mid B \in \Gamma\}$. The inference rules of Linc are those for LJ plus the rules for equality and fixed points that are given in Figure 1. In eq$\mathcal{L}$, the expression $\mathbb{U}(s,t)$ is used to denote a complete set of unifiers for s and t. Since equality has introduction rules, it is a logical connective and not a predicate. The rules for the introduction of inductive predicates on the right or co-inductive predicates on the left are given by familiar unfolding rules while the introduction of inductive predicates on the left or co-inductive predicates on the right are given by the corresponding induction or co-induction principles. In this latter case, the predicate variable S in those inference rules correspond to the invariant (pre-fixed point) or co-inductive invariant (post-fixed point). Notice that unfolding inductive predicates on the left and co-inductive predicates on the right are admissible (sound) inference rules.

We shall often need to restrict ourselves to the "level 0/1 fragment" [1] of Linc. To define this fragment, we assume that every predicate symbol is either inductive or co-inductive and is assigned a *level* of 0 or 1. A formula is *level-0* if it contains no predicates of level 1 and contains no occurrences of implication or universal quantifier. Level-1 formulas satisfy the following grammar:

$$F ::= \bot \mid \top \mid t = s \mid p\,\vec{t} \mid \exists x.F \mid \forall x.F \mid G \supset F \mid F \wedge F \mid F \vee F.$$

where G ranges over level-0 formulas and p ranges over level-0 or level-1 predicates. A definition $p\,\vec{x} \overset{\triangle}{=} B$ is a level-0 (level-1) definition if both p and B are level-0 (resp. level-1) formulas.

Bisimulation up-to [11] refers to a technique for proving bisimilarity of processes that aims at reducing the size of the relation one needs to construct to prove bisimilarity. Bisimulation is a binary relation $\mathcal{R}$ that satisfies some closure properties w.r.t. the transition system generated by processes, as shown in the diagram on the left below. The up-to technique modifies this definition to allow P' and Q' to be related by a larger relation $\mathcal{F}(\mathcal{R})$, defined via an *up-to function* $\mathcal{F}$, as shown in the diagram on the right below.

$$\begin{tikzcd}
P \arrow[d, "\alpha"'] & \mathcal{R} & Q \arrow[d, "\alpha"] \\
P' & \mathcal{R} & Q'
\end{tikzcd}
\qquad\qquad
\begin{tikzcd}
P \arrow[d, "\alpha"'] & \mathcal{R} & Q \arrow[d, "\alpha"] \\
P' & \mathcal{F}(\mathcal{R}) & Q'
\end{tikzcd}$$

Let $\mathbb{B}$ be the function on binary relations defined by

$$\begin{aligned}
\mathbb{B}(R) = \{\langle P, Q\rangle \mid\ & [\forall P' \forall A.\ P \xrightarrow{A} P' \supset \exists Q'.\ Q \xrightarrow{A} Q' \wedge R(P', Q')] \wedge \\
& [\forall Q' \forall A.\ Q \xrightarrow{A} Q' \supset \exists P'.\ P \xrightarrow{A} P' \wedge R(Q', P')]\}
\end{aligned}$$

Then bisimilarity, denoted by $\sim$, is defined as the greatest fixed point of $\mathbb{B}$. The left-diagram above shows that $\mathcal{R} \subseteq \mathbb{B}(\mathcal{R})$, i.e., that $\mathcal{R}$ is a post-fixed point of $\mathbb{B}$. Since $\mathbb{B}$ is monotone, the Knaster-Tarski fixed point theorem implies that $\mathcal{R}$ is included in $\sim$. The right-diagram, on the other hand, only proves that $\mathcal{R} \subseteq \mathbb{B}(\mathcal{F}(\mathcal{R}))$ and in general this does not establish $\mathcal{R}$ as a post-fixed point of $\mathbb{B}$, so one needs to prove that the function $\mathcal{F}$ is *sound*, i.e., for every $\mathcal{R}$, if $\mathcal{R} \subseteq \mathbb{B}(\mathcal{F}(\mathcal{R}))$ then $\mathcal{R} \subseteq \sim$. This up-to technique is not limited to bisimulation and it can be used with other co-inductive definitions [10].

3 Tabled deduction presented as a proof system

When inductive and co-inductive predicates are not used, tabled deduction is easily justified using the cut inference rules of sequent calculus [8]. For example, proving $A \wedge B$ from assumptions Γ can proceed as follows:

$$\dfrac{\begin{array}{c}\Xi_A\\ \Gamma \longrightarrow A\end{array} \qquad \dfrac{\dfrac{}{A, \Gamma \longrightarrow A}\ init \qquad \begin{array}{c}\Xi_B\\ A, \Gamma \longrightarrow B\end{array}}{A, \Gamma \longrightarrow A \wedge B}\ \wedge R}{\Gamma \longrightarrow A \wedge B}\ cut$$

Here, A is both proved by the subproof Ξ_A and is an assumption in the subproof Ξ_B of B from Γ.

When co-inductive predicates are present, one way to establish a co-inductive goal, say bisimulation, is to allow a form of *circular proofs*. In a circular proof, a branch in the proof tree is allowed to close when there is a 'loop', i.e., the sequent at the leaf of the branch matches another sequent lower in the tree. This is a familiar notion in fixed point logics and conditions that guarantee soundness for such circular proofs are known: e.g., the notion of a *progressing trace* in [4]. Such conditions include forbidding loops across minor premises of an inference rule, and every loop must be 'guarded', i.e., there must be an unfolding of a co-inductive atom in the loop. These kind of conditions are too strong, however, to encode up-to techniques for bisimulation. A commonly used up-to technique for bisimulation, say for CCS, is the up-to context technique, which uses the up-to function $\mathcal{F}(\mathcal{R}) = \{(C[P], C[Q]) \mid (P, Q) \in \mathcal{R}\}$, where C is a process context. So, for example, to establish $P{+}Q \sim R{+}Q$, one can simplify this first to the problem of checking $P \sim R$ via the up-to function $\mathcal{F}$. This kind of simplification via up-to context is exploited in [2], for example, to obtain a better bisimulation checking algorithm. An example of using "up-to context" is given in [9, Section A].

To capture bisimulation up-to, we need to encode up-to functions as logical theories, and use them to simplify a goal, before doing loop checking. This leads to inconsistency if done naively, even when the theories are valid. For example, since the processes $a.0$ and $b.0$ are not bisimilar, the formula $\mathrm{bisim}(a, b) \supset \bot$ should be provable. Now consider the following circular proof:

$$\dfrac{\dfrac{}{\mathrm{bisim}(a,b) \supset \bot \longrightarrow \mathrm{bisim}(a,b)}\;\mathit{loop} \qquad \dfrac{}{\mathrm{bisim}(a,b) \supset \bot, \bot \longrightarrow \mathrm{bisim}(a,b)}\;\bot\mathcal{L}}{\mathrm{bisim}(a,b) \supset \bot \longrightarrow \mathrm{bisim}(a,b)}\;\supset\!\mathcal{L}$$

where the leftmost leaf is the same as the root sequent. If this were admitted as a proof, then one can prove $\bot$. Indeed, this kind of loop is forbidden in sound circular proof systems [4, 14] and is an example of a non-progressing loop. Unfortunately, as we mentioned above, forbidding circular proofs outright leads to a restricted system where bisimulation up-to algorithms cannot be encoded directly. An important part of the design of the tabled proof system is to rule out unsound loops while still being able to encode up-to techniques. This involves a careful management of tabled entries from which we deduce good loops.

In our tabled proof system, we capture the notion of a loop in a derivation by extending sequents with *history contexts*. We consider only tabling of atoms and universally quantified atoms. We distinguish three types of co-inductive atoms: *proved atoms*, *history atoms*, and *open atoms*. Only the first two types of atoms can appear in a table. Open atoms can only appear in the goal formula (i.e., the formula on the right-hand side of a sequent) or a theory, and are used to indicate atoms that are yet to be proved or disproved. When atoms occur in sequents, the history atoms will be annotated with $\circ$ while open atoms are annotated with $*$. History atoms and open atoms are syntactic devices used only in the tabled proof system; they have no meaning inside Linc. As the name suggests, history atoms are those encountered during proof search, for which a co-inductive rule has been applied. If a predicate symbol is co-inductively defined, then its history atoms are used to establish a post-fixed point. We consider only history atoms that are co-inductive.[3] A formula is $*$-free (resp., history free) if it has no occurrences of open atoms (resp., history atoms). Given a set $\mathcal{P}$ of formulas, we denote with $\mathcal{P}^\circ$ the set of history atoms in $\mathcal{P}$. Given a predicate p, we denote with $\mathcal{P} \setminus p$ the set $\mathcal{P}$ with all atoms of the form $p\,\vec{t}$ removed.

Sequents have for form $\mathcal{P}; \mathcal{T}; \Gamma \longrightarrow C; \mathcal{P}'$, where Γ is a set of level-0 $*$-free and history-free formulas; C is a level-1 history-free formula; $\mathcal{P}$ and $\mathcal{P}'$ are multisets of $*$-free atoms or universally quantified atoms; and $\mathcal{T}$ is a *theory*, i.e.,

[3] Inductive history atoms can be added, and their use would be to table *disproved* atomic goals. We leave the treatment of inductive history atoms to future work.

a set of closed formulas. The set $\mathcal{P}$ and $\mathcal{P}'$ are bookkeeping devices essentially. Operationally, the sequent can be understood as follows: in the beginning of proof search for the sequent, $\mathcal{P}$ contains the current table entries, and when proof search concludes successfully, $\mathcal{P}'$ contains the new table entries generated by the proof search.

Depending on the tabling strategy, theories can be lemmas (provable in, say, Linc) or rewriting rules on open atoms (which correspond to up-to functions), or a mixture of both. When no history atoms are present, the informal reading of such a sequent is as follows: assuming $\mathcal{T}$ and $\mathcal{P}$ are provable in Linc, then $\Gamma \longrightarrow C$ is provable in Linc and its proof contains subproofs of atoms in $\mathcal{P}'$. When history atoms are present, the interpretation of the sequent is more complicated. Roughly, assuming we only have one co-inductively defined predicate symbol, say p, and the only history atoms are those of p, then $\mathcal{P}^\circ \cup (\mathcal{P}')^\circ$ forms a post fixed point of (the operator associated with) p. The precise interpretation will be given when we formally prove the soundness result for each strategy.

The inference rules involving these richer sequents are given in Figure 2. We consider only unification problems that have most general unifiers, e.g., first-order or higher-order pattern unification: in this way, eq$\mathcal{L}$ has at most one premise. In branching rules, the accumulated history or proved atoms on the right-hand side of a sequent in one branch are passed on to the other branch. When using this proof system for proof search, this set will be populated deterministically in a depth-first search strategy. The most interesting rule is ν_R. Here, reading the rule upwards, one replaces the co-inductive predicate p with p^*, and add $p^\circ \vec{t}$ to the history context on the left to allow it to be used to detect loops. When proof search is done, the history context on the right will be populated with history atoms. The intention is that these history atoms will form a post-fixed point (up-to) of p; hence when the proof search concludes, we replace each history atom p° on the right with p, signifying that every element in the post-fixed point is contained in the largest fixed point of p.

Notice that our sequent calculus does not have explicit structural rules (contraction and weakening) since these rules have been internalized in other rules. We have also omitted the cut rule. We currently do not know whether cut is admissible, but it is not important for this work as we only are interested in soundness. Notice also that if a sequent has a non-empty left-hand (Γ) context, then it can be the conclusion of only left-introduction rules: furthermore, since Γ can only contain level-0 formulas, there is no need for left introduction rules for implications and universal quantifiers.

Let $p_1, \ldots, p_n$ be the set of all co-inductive predicates that are defined in the logic. We denote by $\mathcal{L}$ the set $\{\forall \vec{x_1}(p_1^\circ \vec{x_1} \supset p_1^* \vec{x_1}), \ldots, \forall \vec{x_n}(p_n^\circ \vec{x_n} \supset p_n^* \vec{x_n})\}$. That is, $\mathcal{L}$ allows one to backchain from an open atom to a history atom. Adding $\mathcal{L}$ as theories to the tabled proof system allows one to loop on co-inductive atoms.

The function $\mathbb{S}$ used in Figure 2 is determined by the tabling strategies:

$$\begin{array}{rlrl}
\text{Strategy I:} & \mathbb{S}(\mathcal{P}, \mathcal{T}) = \mathcal{P} \setminus \mathcal{P}^\circ & \text{Strategy III:} & \mathbb{S}(\mathcal{P}, \mathcal{T}) = \mathcal{P} \cup \mathcal{L} \\
\text{Strategy II:} & \mathbb{S}(\mathcal{P}, \mathcal{T}) = (\mathcal{P} \setminus \mathcal{P}^\circ) \cup \mathcal{T} & \text{Strategy IV:} & \mathbb{S}(\mathcal{P}, \mathcal{T}) = \mathcal{P} \cup \mathcal{T}
\end{array}$$

$$\frac{\mathbb{S}(\mathcal{P},\mathcal{T}) \vdash_I A}{\mathcal{P};\mathcal{T};\cdot \longrightarrow A;\cdot}\;\mathit{init} \qquad \frac{}{\mathcal{P};\mathcal{T};\bot,\Gamma \longrightarrow B;\cdot}\;\bot\mathcal{L} \qquad \frac{}{\mathcal{P};\mathcal{T};\cdot \longrightarrow \top;\cdot}\;\top\mathcal{R}$$

$$\frac{\mathcal{P};\mathcal{T};B,C,\Gamma \longrightarrow D;\mathcal{P}'}{\mathcal{P};\mathcal{T};B\wedge C,\Gamma \longrightarrow D;\mathcal{P}'}\;\wedge\mathcal{L} \qquad \frac{\mathcal{P};\mathcal{T};\cdot \longrightarrow B;\mathcal{P}_1 \quad \mathcal{P},\mathcal{P}_1;\mathcal{T};\cdot \longrightarrow C;\mathcal{P}'}{\mathcal{P};\mathcal{T};\cdot \longrightarrow B\wedge C;\mathcal{P}',\mathcal{P}_1}\;\wedge\mathcal{R}$$

$$\frac{\mathcal{P};\mathcal{T};B,\Gamma \longrightarrow D;\mathcal{P}_1 \quad \mathcal{P},\mathcal{P}_1;\mathcal{T};C,\Gamma \longrightarrow D;\mathcal{P}'}{\mathcal{P};\mathcal{T};B\vee C,\Gamma \longrightarrow D;\mathcal{P}',\mathcal{P}_1}\;\vee\mathcal{L} \qquad \frac{\mathcal{P};\mathcal{T};\cdot \longrightarrow B_i;\mathcal{P}'}{\mathcal{P};\mathcal{T};\cdot \longrightarrow B_1\vee B_2;\mathcal{P}'}\;\vee\mathcal{R}$$

$$\frac{\mathcal{P};\mathcal{T};B \longrightarrow C;\mathcal{P}'}{\mathcal{P};\mathcal{T};\cdot \longrightarrow B\supset C;\mathcal{P}'}\;\supset\mathcal{R} \qquad \frac{\mathcal{P};\mathcal{T};\cdot \longrightarrow B[y/x];\mathcal{P}'}{\mathcal{P};\mathcal{T};\cdot \longrightarrow \forall x.B;\mathcal{P}'}\;\forall\mathcal{R}$$

$$\frac{\mathcal{P};\mathcal{T};B[y/x],\Gamma \longrightarrow C;\mathcal{P}'}{\mathcal{P};\mathcal{T};\exists x.B,\Gamma \longrightarrow C;\mathcal{P}'}\;\exists\mathcal{L} \qquad \frac{\mathcal{P};\mathcal{T};\cdot \longrightarrow B[t/x];\mathcal{P}'}{\mathcal{P};\mathcal{T};\cdot \longrightarrow \exists x.B;\mathcal{P}'}\;\exists\mathcal{R}$$

$$\frac{\mathcal{P};\mathcal{T};\Gamma[\rho] \longrightarrow C[\rho];\mathcal{P}'}{\mathcal{P};\mathcal{T};s=t,\Gamma \longrightarrow C;\mathcal{P}'}\;\mathrm{eq}\mathcal{L}, \rho = \mathrm{mgu}(s,t) \qquad \frac{}{\mathcal{P};\mathcal{T};\cdot \longrightarrow t=t;\cdot}\;\mathrm{eq}\mathcal{R}$$

$$\frac{}{\mathcal{P};\mathcal{T};s=t,\Gamma \longrightarrow C;\cdot}\;\mathrm{eq}\mathcal{L}, s \text{ and } t \text{ not unifiable.}$$

$$\frac{\mathcal{P};\mathcal{T};B\,p\,\vec{t},\Gamma \longrightarrow C;\mathcal{P}'}{\mathcal{P};\mathcal{T};p\,\vec{t},\Gamma \longrightarrow C;\mathcal{P}'}\;\mathrm{def}\mathcal{L} \qquad \frac{\mathbb{S}(\mathcal{P},\mathcal{T}) \not\vdash_I p\,\vec{t} \qquad \mathcal{P};\mathcal{T};\cdot \longrightarrow B\,p\,\vec{t};\mathcal{P}'}{\mathcal{P};\mathcal{T};\cdot \longrightarrow p\,\vec{t};\mathcal{P}',\forall\vec{x}.p\,\vec{t}}\;\mathrm{def}\mathcal{R}$$

$$\frac{\mathbb{S}(\mathcal{P},\mathcal{T}) \not\vdash_I p^*\,\vec{t} \qquad \mathcal{P},p^\circ\vec{t};\mathcal{T};\cdot \longrightarrow B\,p^*\,\vec{t};\mathcal{P}'}{\mathcal{P};\mathcal{T};\cdot \longrightarrow p^*\,\vec{t};\mathcal{P}',p^\circ\,\vec{t}}\;\nu_R^*$$

$$\frac{\mathbb{S}(\mathcal{P}\setminus p^\circ,\mathcal{T}) \not\vdash_I p\,\vec{t} \qquad (\mathcal{P}\setminus p^\circ),p^\circ\vec{t};\mathcal{T};\cdot \longrightarrow B\,p^*\,\vec{t};\mathcal{P}'}{\mathcal{P};\mathcal{T};\cdot \longrightarrow p\,\vec{t};\mathcal{P}'[p/p^\circ],p\,\vec{t}}\;\nu_R$$

Fig. 2. Inference rules for the tabled proof system. In def$\mathcal{L}$ and def$\mathcal{R}$, $p\,\vec{x} \overset{\triangle}{=} B\,p\,\vec{x}$, and in ν_R^* and ν_R, $p\,\vec{x} \overset{\nu}{=} B\,p\,\vec{x}$ and $\vec{t}$ are ground terms.

We shall refer to these functions as, respectively, $\mathbb{S}_1$, $\mathbb{S}_2$, $\mathbb{S}_3$ and $\mathbb{S}_4$. The proof systems for these strategies are defined as follows: the proof systems $\mathcal{TD}_1$ and $\mathcal{TD}_2$ are proof systems obtained by using, respectively, $\mathbb{S}_1$ and $\mathbb{S}_2$, and whose rules include all the inference rules in Figure 2 except ν_R^* and ν_R. The proof systems $\mathcal{TD}_3$ and $\mathcal{TD}_4$ are proofs systems obtained using, respectively, $\mathbb{S}_3$ and $\mathbb{S}_4$, and whose rules include all the inference rules in Figure 2 except def$\mathcal{R}$.

The relation $\vdash_I$ refers to the deducibility relation of intuitionistic logic (without fixed points). When $\mathcal{T}$ is restricted to formulas containing just $\supset$, $\wedge$, and $\forall$ the relation $\vdash_I$ is implemented by λProlog [7].

4 Constructing post-fixed point from tables

In the following, given two lists of terms $\vec{s} = s_1, \ldots, s_n$ and $\vec{t} = t_1, \ldots, t_n$, we write $\vec{s} = \vec{t}$ to denote the formula $(s_1 = t_1) \wedge (s_2 = t_2) \wedge \cdots \wedge (s_n = t_n)$. For simplicity, we shall assume that all co-inductive predicates have the same arity. We denote by $\mathcal{P}^\bullet$ the set $\mathcal{P} \setminus \mathcal{P}^\circ$.

Theorem 1. *Suppose $\mathcal{P}; \mathcal{T}; \Gamma \longrightarrow C; \mathcal{P}'$ is derivable in $\mathcal{TD}_3$, where Γ is history-free and C contains no negative occurrences of history atoms. Let $\{p_1, \ldots, p_n\}$ be the set of co-inductive predicates occuring in $\mathcal{P}, C$ and $\mathcal{P}'$. Then there exist invariants $S_1, \ldots, S_n$ such that*

- *the sequent $(\mathcal{P}^\bullet, \Gamma \longrightarrow C[S_1/p_1^*, \ldots, S_n/p_n^*])$ is derivable in* Linc*,*
- *for each $B \in (\mathcal{P}')^\bullet$, the sequent $(\mathcal{P}^\bullet \longrightarrow B)$ is derivable in* Linc*, and*
- *for each $p_i^\circ \vec{t} \in (\mathcal{P}')^\circ$, where $p_i\,\vec{x} \overset{\nu}{=} D_i\,p_i\,\vec{x}$, the sequent $(\mathcal{P}^\bullet \longrightarrow D_i\,S_i\,\vec{t})$ is derivable in* Linc.

Proof. (Outline.) Given sequent $\mathcal{P}; \mathcal{T}; \Gamma \longrightarrow C; \mathcal{P}'$, the abstraction S_i

$$S_i = \lambda\vec{x}. \bigwedge(\mathcal{P})^\bullet \wedge \bigvee\{(\vec{x} = \vec{t}) \mid p_i^\circ\,\vec{t} \in \mathcal{P} \cup \mathcal{P}'\},$$

forms a post-fixed point of the definition of p_i, i.e., $D\ S_i\ \vec{x} \longrightarrow S_i\,\vec{x}$. □

5 Co-inductive tabling modulo theories

In a naive algorithm for bisimulation checking, one can construct a bisimulation set by progressively unfolding transitions from a given pair of processes, until one arrives at stuck processes or encounters a previously seen pair of processes. This is very similar to how proof search with strategy III works. The up-to techniques add to this the possibility of simplifying the continuations of a pair of processes, before doing the loop checking. For example, a typical simplification rule is the context closure, e.g., when one encounters a new pair to be checked $((P \mid R), (Q \mid R))$, instead of unfolding these, we simplify it to (P, Q) and proceed. This kind of simplification before loop checking is in general unsound; see [11] for an example. An important line of research in the up-to techniques is in characterizing sound simplification rules.

To capture up-to techniques in our tabling proof system, we need a mechanism to apply simplification to an open co-inductive goal before doing loop checking. This can be done simply by backchaining on the co-inductive goal. Since open co-inductive goals are marked with $*$, to be able to backchain on them, we need to allow $*$-atoms in the theory component of a sequent. However, when the theory $\mathcal{T}$ contains $*$-atoms, it is not possible in general to construct a post fixed point from tabled entries as they are no longer closed under fixed point unfolding. This is because the theory $\mathcal{T}$ may allow one to deduce $*$-atoms that have not been encountered during proof search (hence those particular atoms would not have been unfolded). Soundness in this case is conditional on an additional statement, which happens to coincide with the (logical interpretation) of the soundness condition for up-to techniques [10].

To simplify the presentation, we shall restrict to one co-inductive definition in the following. We shall refer to this definition simply as $p\vec{x} \overset{\nu}{=} D\,p\,\vec{x}$. So we have only one kind of history atoms and one kind of $*$ atoms, i.e., those of the form $p^\circ\vec{t}$ and $p^*\vec{t}$. The set $\mathcal{L}$ in this case contains exactly one formula, i.e., $\forall\vec{x}(p^\circ\vec{x} \supset p^*\vec{x})$.

To formalize the up-to techniques, we need to quantify over relations and functions. Thus we introduce HOLinc, the extension of Linc that contains higher-order quantifiers. In other words, the logic we have now is an intuitionistic higher-order logic (i.e., the intuitionistic version of Church's Simple Theory of Types) with fixed points and (free) equality. The latter two can be encoded in higher-order logic, so we essentially only work within higher-order logic.

Definition 1. *An up-to theory is a set $\mathcal{T}$ of higher-order hereditary Harrop (hH) formulas such that the head of each clause is of the form $p^*\vec{t}$. We assume that $\mathcal{L} \subseteq \mathcal{T}$, and the only place where history atoms occur in $\mathcal{T}$ is in this subset.*

Definition 2. *If $\mathcal{T}$ is an up-to theory, it induces the function*

$$\mathcal{F}_{\mathcal{T}} = \lambda\mathcal{R}\lambda\vec{x}.\forall q. \bigwedge \mathcal{T}[q/p^*, \mathcal{R}/p^\circ] \supset q\vec{x}.$$

In more informal set-theoretic notation, $\mathcal{F}_{\mathcal{T}}$ can be written as:

$$\mathcal{F}_{\mathcal{T}}(\mathcal{R}) = \{\vec{x} \mid \forall q \big(\bigwedge \mathcal{T}[q/p^*, \mathcal{R}/p^\circ] \supset q\vec{x}\big) \text{ is provable in HOLinc. } \}$$

The adequacy of this encoding of up-to functions is the result of the completeness of goal-directed proof for hH fragment of higher-order logic; see [7].

Definition 3. *An up-to theory $\mathcal{T}$ is sound if the following formula, named $Snd(\mathcal{T})$ holds: $\forall\mathcal{R}.(\forall\vec{x}.(\mathcal{R}\vec{x} \supset D\ (\mathcal{F}_{\mathcal{T}}\mathcal{R})\ \vec{x})) \supset (\forall\vec{x}.\mathcal{R}\vec{x} \supset p\ \vec{x})$.*

Theorem 2. *Suppose $\mathcal{P}; \mathcal{T}; \Gamma \longrightarrow C; \mathcal{P}'$ is derivable in $\mathcal{TD}_4$. Then there exists an invariant S such that*

- *the sequent $(Snd(\mathcal{T}), \mathcal{P}^\bullet, \Gamma \longrightarrow C[\mathcal{F}_{\mathcal{T}}S/p^*])$ is derivable in* HOLinc*,*
- *for each $B \in (\mathcal{P}')^\bullet$, the sequent $(\mathcal{P}^\bullet \longrightarrow B)$ is derivable in* HOLinc*, and*
- *for each $p\vec{t} \in (\mathcal{P}')^\circ$, the sequent $(\mathcal{P}^\bullet \longrightarrow D\,(\mathcal{F}_{\mathcal{T}}S)\,\vec{t})$ is derivable in* HOLinc.

Proof. (Outline.) Given $\mathcal{P}; \mathcal{T}; \Gamma \longrightarrow C; \mathcal{P}'$, the abstraction $\lambda\vec{x}. \bigvee\{(\vec{x} = \vec{u}) \mid p^\circ\vec{u} \in \mathcal{P} \cup \mathcal{P}'\}$ can be shown to be a post-fixed point "up-to" $\mathcal{F}_{\mathcal{T}}$. □

Corollary 1. *Let $\mathcal{T}$ be an up-to theory. If $\cdot; \mathcal{T}; \cdot \longrightarrow B; \mathcal{P}$ is derivable in $\mathcal{TD}_4$, for some $\mathcal{P}$, then $Snd(\mathcal{T}) \longrightarrow B$ is derivable in* HOLinc.

Thus strategy IV is sound for, say, bisimulation checking if one can discharge the assumption $Snd(\mathcal{T})$, a task that can often be tedious to do. We are currently developing some of these proofs in the (higher-order version of the) theorem prover Abella, which is an interactive prover based on a logic similar to Linc.

6 Compositions of up-to functions

One important line of research in the up-to techniques in bisimulation is that of compositions of up-to functions. More precisely, one is interested in characterizing when the composition of two sound up-to functions gives rise to a sound up-to function. Such results allow one to combine simple functions to form powerful sound composite functions. We show next that composition of up-to functions can be defined via a notion of composition of up-to theories.

Definition 4. *Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be up-to theories. Their composition, written $\mathcal{T}_1 \circ \mathcal{T}_2$, is defined as $\mathcal{T}_1 \circ \mathcal{T}_2 = \mathcal{T}_1[F/p^\circ]$ where $F = \lambda\vec{x}.(\forall q. \bigwedge \mathcal{T}_2[q/p^*] \supset q\,\vec{x})$.*

The following lemma states that this definition of composition of theories is adequate, i.e., it respects the composition of logical up-to functions.

Lemma 1. *$\mathcal{F}_{\mathcal{T}_1} \circ \mathcal{F}_{\mathcal{T}_2}$ and $\mathcal{F}_{\mathcal{T}_1 \circ \mathcal{T}_2}$ define the same function.*

In practice, up-to techniques are often used by interleaving applications of several up-to functions. However, proving that such interleaving is sound is obviously more complicated than proving soundness of restricted compositions. In the logical encodings, interleaving of two theories $\mathcal{T}_1$ and $\mathcal{T}_2$ can be captured simply by joining the theories, i.e., $\mathcal{T}_1 \cup \mathcal{T}_2$. We show next that soundness of tabled proof search in the up-to theory $\mathcal{T}_1 \cup \mathcal{T}_2$ can be reduced to soundness of proof search under their composition $\mathcal{T}_1 \circ \mathcal{T}_2$, under certain conditions.

To prove the following results, it is convenient to view a theory as an inference rule. This is straightforward when the theories are Horn clauses. The Horn clause

$$\forall\vec{x}.(A_1 \wedge \cdots \wedge A_n) \supset p^*\vec{t} \quad \text{can be written as the rule} \quad \frac{A_1 \quad \cdots \quad A_n}{p^*\vec{t}} \ ,$$

where $A_1, \ldots, A_n$ are atoms and where $\vec{x}$ become schematic variables of the inference rule. Let $\mathcal{D}(\mathcal{T})$ denote the set of inference rules for a given Horn theory $\mathcal{T}$. Then $\mathcal{P}, \mathcal{T} \vdash_I p^*\,\vec{t}$ holds iff there is a derivation of $p^*\vec{t}$ from $\mathcal{P}$ in the inference system $\mathcal{D}(\mathcal{T})$. We say that an inference rule r_1 *permutes over* another inference rule r_2 iff every derivation of $p^*\vec{t}$, for any $\vec{t}$, where r_2 appears immediately above r_1 can be transformed into another derivation of $p^*\vec{t}$ where r_1 appears above r_2. Given $\mathcal{D}(\mathcal{T}_1)$ and $\mathcal{D}(\mathcal{T}_2)$, we say that $\mathcal{D}(\mathcal{T}_1)$ permutes over $\mathcal{D}(\mathcal{T}_2)$ iff every rule of $\mathcal{D}(\mathcal{T}_1 \setminus \mathcal{T}_2)$ permutes over every rule of $\mathcal{D}(\mathcal{T}_2 \setminus \mathcal{T}_1)$.

Lemma 2. *Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be two Horn up-to theories such that $\mathcal{D}(\mathcal{T}_2)$ permutes over $\mathcal{D}(\mathcal{T}_1)$. Then $(\mathcal{P}, \mathcal{T}_1, \mathcal{T}_2 \vdash_I p^*\vec{t})$ iff $(\mathcal{P}, \mathcal{T}_1 \circ \mathcal{T}_2 \vdash_I p^*\vec{t})$, for every set of $*$-free atoms $\mathcal{P}$ and every $\vec{t}$.*

Theorem 3. *Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be two Horn up-to theories such that $\mathcal{T}_1 \circ \mathcal{T}_2$ is sound and that $\mathcal{D}(\mathcal{T}_2)$ permutes over $\mathcal{D}(\mathcal{T}_1)$. If $\cdot; \mathcal{T}_1, \mathcal{T}_2; \cdot \longrightarrow B; \mathcal{P}$ is derivable in $\mathcal{TD}_4$, for some $\mathcal{P}$, then $Snd(\mathcal{T}_1 \circ \mathcal{T}_2) \longrightarrow B$ is derivable in* HOLinc.

Proof. This follows from Theorem 2 and Lemma 2. □

Note that Theorem 3 does not imply that $\mathcal{F}_{\mathcal{T}_1 \cup \mathcal{T}_2}$ is sound given that $\mathcal{F}_{\mathcal{T}_1 \circ \mathcal{T}_2}$ is sound; it only implies that, for the purpose of proving a co-inductive goal in the tabled proof system, one can freely combine $\mathcal{T}_1$ and $\mathcal{T}_2$ without losing soundness. This is useful in practice where one could combine different up-to techniques freely but only need to prove soundness for a restricted form of composition.

Below, we shall use $\sim$ to denote the predicate *bisim*.

Example 1. Consider the CCS example again. Let $\mathcal{T}_1$ be the up-to theory formalizing context closure, and let $\mathcal{T}_2$ be the up-to theory formalizing reflexive and transitive closure. The inference rules of $\mathcal{D}(\mathcal{T}_1)$ are the rules $\{b, re, tr\}$ and the rules of $\mathcal{D}(\mathcal{T}_2)$ are $\{b, cng\}$, where b, re, tr, cng are as follows:

$$\frac{s \sim^\circ t}{s \sim^* t}\,b \qquad \frac{}{t \sim^* t}\,re \qquad \frac{s \sim^* u \quad u \sim^* t}{s \sim^* t}\,tr \qquad \frac{s \sim^* t}{C[s] \sim^* C[t]}\,cng$$

and where $C[\;]$ is a process context. It can be easily shown that cng permutes up over re and tr, for example:

$$\frac{\dfrac{s \sim^* u \quad u \sim^* t}{s \sim^* t}\,tr}{C[s] \sim^* C[t]}\,cng \quad\rightsquigarrow\quad \frac{\dfrac{s \sim^* u}{C[s] \sim^* C[u]}\,cng \quad \dfrac{u \sim^* t}{C[u] \sim^* C[t]}\,cng}{C[s] \sim^* C[t]}\,tr$$

So we can freely mix $\mathcal{T}_1$ and $\mathcal{T}_2$ in proving particular instances of bisimilarity, but we only need to prove soundness of the composition $\mathcal{T}_1 \circ \mathcal{T}_2$.

If the up-to theory $\mathcal{T}$ contains occurrences of the co-inductive predicate p, then we can consider using previously proved facts, say $\mathcal{T}'$, about p to prove subgoals of the form $p\,\vec{t}$. The use of lemmas is orthogonal to the soundness condition for up-to techniques, as stated in the following theorem.

Theorem 4. *Let $\mathcal{U}$ be a set of $*$-free and history-free formulas that are valid in* HOLinc. *Suppose $\mathcal{P}; \mathcal{U}, \mathcal{T}; \Gamma \longrightarrow C; \mathcal{P}'$ is derivable in $\mathcal{TD}_4$. Then there exists an invariant S such that*

- *the sequent $(Snd(\mathcal{T}), \mathcal{P}^\bullet, \Gamma \longrightarrow C[\mathcal{F}_\mathcal{T} S/p^*])$ is derivable in* HOLinc*,*
- *for each $B \in (\mathcal{P}')^\bullet$, the sequent $(\mathcal{P}^\bullet \longrightarrow B)$ is derivable in* HOLinc*, and*
- *for each $p\vec{t} \in (\mathcal{P}')^\circ$, the sequent $\mathcal{P}^\bullet \longrightarrow D\,(\mathcal{F}_\mathcal{T} S)\,\vec{t})$ is derivable in* HOLinc.

Proof. This proof follows the proof of Theorem 2, except we use the following invariant: given sequent $\mathcal{P}; \mathcal{U}, \mathcal{T}; \Gamma \longrightarrow C; \mathcal{P}'$, define $S = \lambda \vec{x}. \bigwedge \mathcal{U} \wedge \bigvee \{(\vec{x} = \vec{u}) \mid p^\circ \vec{u} \in \mathcal{P} \cup \mathcal{P}'\}$. □

The composition result (Theorem 3) can be slightly modified to take into account uses of lemmas. As we shall see later, this leads to a rather pleasant result concerning compositions with up-to bisimilarity.

Lemma 3. *Let $\mathcal{U}$ be a set of Horn clauses which are also lemmas of* HOLinc. *Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be two Horn up-to theories such that $\mathcal{D}(\mathcal{U} \cup \mathcal{T}_2)$ permutes over $\mathcal{D}(\mathcal{U} \cup \mathcal{T}_1)$. Then $(\mathcal{P}, \mathcal{U}, \mathcal{T}_1, \mathcal{T}_2 \vdash_I p^* \vec{t})$ iff $(\mathcal{U}, \mathcal{P}, \mathcal{T}_1 \circ \mathcal{T}_2 \vdash_I p^* \vec{t})$, for every set of $*$-free atoms $\mathcal{P}$ and every $\vec{t}$.*

Theorem 5. *Let $\mathcal{U}$ be a set of lemmas of* HOLinc. *Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be two Horn up-to theories such that $\mathcal{T}_1 \circ \mathcal{T}_2$ is sound and that $\mathcal{D}(\mathcal{U} \cup \mathcal{T}_2)$ permutes over $\mathcal{D}(\mathcal{U} \cup \mathcal{T}_1)$. If $\cdot; \mathcal{U}, \mathcal{T}_1, \mathcal{T}_2; \cdot \longrightarrow B; \mathcal{P}$ is derivable in $\mathcal{TD}_4$, for some $\mathcal{P}$, then $Snd(\mathcal{T}_1 \circ \mathcal{T}_2) \longrightarrow B$ is derivable in* HOLinc.

Example 2. Let $\mathcal{T}_1$ be the theory encoding up-to bisimilarity and let $\mathcal{T}_2$ be the theory encoding up-to context-closure for CCS. The inference rules of $\mathcal{T}_1$ consist of the rule b (see Example 1) and the following rule:

$$\frac{s \sim u \quad u \sim^* v \quad v \sim t}{s \sim^* t}\ bs$$

The composition $\mathcal{T}_1 \circ \mathcal{T}_2$ is shown to be sound in, e.g., [10]. Since bisimilarity in CCS is closed under arbitrary contexts, we can prove the lemma below (left) in HOLinc: the inference rule corresponding to that lemma is on the right:

$$\forall C \forall x, y\ (x \sim y \supset C[x] \sim C[y]) \qquad \frac{s \sim t}{C[s] \sim C[t]}\ bcng$$

where C denotes a process context. Let $\mathcal{U}$ be a set of Horn lemmas that includes this lemma. We show that $\mathcal{D}(\mathcal{U} \cup \mathcal{T}_2)$ permutes over $\mathcal{D}(\mathcal{U} \cup \mathcal{T}_1)$. It is enough to show that the rule *cng* (see Example 1) permutes over *bs*:

$$\dfrac{\dfrac{s \sim u \quad u \sim^* v \quad v \sim t}{s \sim^* t}\ bs}{C[s] \sim^* C[t]}\ cng \qquad \leadsto$$

$$\dfrac{\dfrac{s \sim u}{C[s] \sim C[u]}\ bcng \quad \dfrac{s \sim^* u}{C[u] \sim^* C[v]}\ cng \quad \dfrac{v \sim t}{C[v] \sim C[t]}\ bcng}{C[s] \sim^* C[t]}\ bs$$

This shows that, rather surprisingly, we can apply the congruence rule first, before applying up-to bisimilarity, without losing soundness, even though the meta theory only allows one to apply congruence rules last. This can potentially lead to a shorter proof as the congruence rule allows simplification of processes.

7 Conclusion and Future work

We have shown a range of strategies for incorporating tables into proof search, where the most advanced strategy allows us to capture the up-to techniques for bisimilarity. For all strategies, we show that tabled proofs can be soundly interpreted as a proper proof in the same logic and formal proof certificates can be constructed from each successful proof search. Our encoding of up-to techniques also enables us to derive a new result in the composition of up-to techniques, allowing one to freely compose up-to techniques while only needing to prove soundness of a limited form of composition.

Orthogonal to all these strategies is the question of whether one should allow quantified formulas (existentially or universally) in the table. Such a possibility can arise if for example one can prove a goal $(p\ a\ X)$ for any X, e.g., simply because X is not used in the definition of p. Then a natural interpretation of this is to say that we have actually proved $\forall x. p\ a\ x$. While this kind of quantified tabled entries is harmless in Strategy I and II, it is less clear whether it is sound for Strategy III and IV. We shall leave this as future work.

We have concentrated on strong bisimulation as an application in this paper, but the framework we established here should apply to weak bisimulation as well, at least as far as the cyclic structure of proofs is concerned. The theory of weak-bisimulation up-to is a lot of more complex than the strong bisimulation up-to and less uniform, e.g., some obvious up-to functions (e.g., up-to weak bisimilarity) is unsound [12]. In terms of formalization in our framework, however, this complexity is mostly isolated in the theory part, i.e., in establishing $Snd(\mathcal{T})$. We plan to investigate weak-bisimilarity in immediate future work.

References

1. D. Baelde, A. Gacek, D. Miller, G. Nadathur, and A. Tiu. The Bedwyr system for model checking over syntactic expressions. In F. Pfenning, editor, *21th Conf. on Automated Deduction (CADE)*, LNAI 4603, pp. 391–397.
2. F. Bonchi and D. Pous. Checking NFA equivalence with bisimulations up to congruence. In *Proceedings of the 40th annual ACM SIGPLAN-SIGACT symposium on Principles of Programming Languages*, pp. 457–468. ACM, 2013.
3. J. Brotherston, N. Gorogiannis, and R. L. Petersen. A generic cyclic theorem prover. In *APLAS*, LNCS 7705, pp. 350–367. Springer, 2012.
4. J. Brotherston and A. Simpson. Complete sequent calculi for induction and infinite descent. In *22th Symp. on Logic in Computer Science*, pp. 51–62, 2007.
5. J. Jaffar, A. E. Santosa, and R. Voicu. A CLP proof method for timed automata. In *RTSS*, pp. 175–186. IEEE Computer Society, 2004.
6. R. McDowell, D. Miller, and C. Palamidessi. Encoding transition systems in sequent calculus. *Theoretical Computer Science*, 294(3):411–437, 2003.
7. D. Miller and G. Nadathur. *Programming with Higher-Order Logic.* Cambridge University Press, June 2012.
8. D. Miller and V. Nigam. Incorporating tables into proofs. In J. Duparc and T. A. Henzinger, editors, *CSL 2007: Computer Science Logic*, LNCS 4646, pp. 466–480.
9. D. Miller and A. Tiu. Extracting proofs from tabled proof search: Extended version. Technical report, INRIA, 2013.
10. D. Pous and D. Sangiorgi. Enhancements of the bisimulation proof method. In D. Sangiorgi and J. Rutten, editors, *Advanced Topics in Bisimulation and Coinduction*, pp. 233–289. Cambridge University Press, 2011.
11. D. Sangiorgi. On the bisimulation proof method. *Mathematical Structures in Computer Science*, 8(5):447–479, 1998.
12. D. Sangiorgi and R. Milner. The problem of "weak bisimulation up to". In *CONCUR*, LNCS 630, pp. 32–46. Springer, 1992.
13. L. Simon, A. Mallya, A. Bansal, and G. Gupta. Coinductive logic programming. In *ICLP*, LNCS 4079, pp. 330–345. Springer, 2006.
14. C. Sprenger and M. Dam. On global induction mechanisms in a μ-calculus with explicit approximations. *ITA*, 37(4):365–391, 2003.
15. A. Tiu. *A Logical Framework for Reasoning about Logical Specifications.* PhD thesis, Pennsylvania State University, May 2004.
16. A. Tiu and D. Miller. Proof search specifications of bisimulation and modal logics for the π-calculus. *ACM Trans. on Computational Logic*, 11(2), 2010.
17. A. Tiu and A. Momigliano. Cut elimination for a logic with induction and coinduction. *Journal of Applied Logic*, 2012.
18. I. Walukiewicz. Completeness of Kozen's axiomatisation of the propositional μ-calculus. *Inf. Comput.*, 157(1-2):142–182, 2000.

A An example of tabled proofs

We consider a fragment of CCS with replication but without the restriction and the choice operators. Process expressions are generated by the grammar:

$$P ::= 0 \mid \mu.P \mid (P \mid P) \mid\ !P$$

where μ is either a name a, a co-name $\bar{a}$ or τ. We use α to denote either a name or a co-name, and we remove the trailing 0 in $a.0$ to simplify presentation. A process context is a process with a hole $[\ ]$. We omit the transition rules and their encodings as definitions; these can be found in, e.g., [6].

We now illustrate how one can prove $!(a \mid b) \sim\ !a \mid !b$ using the tabled proof system. We shall assume a set of up-to theories $\mathcal{T}$ for up-to bisimilarity and up-to context, as presented in Example 2. We shall also assume a set of lemmas, denoted by $\mathcal{U}$, which consists of the following clauses:

$$\begin{gathered}\forall P, Q.P \mid Q \sim Q \mid P, \quad \forall P, Q, R.P \mid (Q \mid R) \sim (P \mid Q) \mid R, \quad \forall P.P \mid 0 \sim P,\\ \forall P.P \sim P, \qquad \forall P.!P \sim !P \mid P, \qquad \forall P, Q.P \sim Q \supset C[P] \sim C[Q]\end{gathered}$$

for a selection of context $C[\ ]$. All these clauses are valid for CCS.

To prove $!(a \mid b) \sim !a \mid !b$ (reading the rules upwards), we start with

$$\cdot; \mathcal{U}, \mathcal{T}; \cdot \longrightarrow !(a \mid b) \sim (!a \mid !b); \cdot$$

The only choice here is to apply ν_R rule to unfold the definition of $\sim$, thus one ends up with a big sequent of the shape:

$$!(a|b) \sim^\circ (!a|!b); \mathcal{U}, \mathcal{T}; \cdot \longrightarrow [\forall A, P.!(a|b) \xrightarrow{A} P \supset \exists Q.!a|!b \xrightarrow{A} Q \wedge P \sim^* Q] \wedge \cdots$$

We show informally how the first conjunct of this formula can be proved. It should be straightforward to construct a formal sequent proof from our informal discussion. After applying the right rules for introducing $\forall$ and $\supset$ and then the unfolding rules for the least fixed point on the left, we end up with a case analysis on $!(a \mid b) \xrightarrow{A} P$, which which yields the two cases where $A = a$ and $P = (!(a \mid b) \mid (0 \mid b))$ and where $A = b$ and $P = (!(a \mid b) \mid (a \mid 0))$. We show here the former. The only choice for Q is $(!a \mid 0) \mid !b$, so it remains to prove

$$!(a \mid b) \sim^\circ (!a \mid !b); \mathcal{U}, \mathcal{T}; \cdot \longrightarrow !(a \mid b) \mid (a \mid 0) \sim^* (!a \mid 0) \mid !b. \tag{1}$$

We claim that $!(a \mid b) \mid (a \mid 0) \sim^* (!a \mid 0) \mid !b$ is deducible from $!(a \mid b) \sim^\circ (!a \mid !b)$ and $\mathcal{U}$ and $\mathcal{T}$. Viewing $\mathcal{U}$ and $\mathcal{T}$ as inference rules, one can derive:

$$\frac{\begin{array}{ll} (i) & !(a \mid b) \mid (0 \mid b) \sim\ !(a \mid b) \mid (0 \mid b) \\ (ii) & !(a \mid b) \mid (0 \mid b) \sim^* (!a \mid !b) \mid (0 \mid b) \\ (iii) & (!a \mid !b) \mid (0 \mid b) \sim\ (!a \mid 0) \mid !b \end{array}}{!(a \mid b) \mid (0 \mid b) \sim^* (!a \mid 0) \mid !b}\ bs$$

Formulas (i) and (iii) follow from $\mathcal{U}$. Formula (ii) is reduced as follows:

$$\cfrac{\cfrac{!(a \mid b) \sim^{\circ} (!a \mid !b)}{!(a \mid b) \sim^{*} (!a \mid !b)}\ b}{!(a \mid b) \mid (0 \mid b) \sim^{*} (!a \mid !b) \mid (0 \mid b)}\ cng$$

Thus, sequent (1) above can be proved using the *init* rule.

B Soundness of strategy I and II

We now show that strategies I and II are sound by translating tables into proofs (with cuts) in Linc.

Theorem 6. *If $\mathcal{P};\mathcal{T};\Gamma \longrightarrow B;\mathcal{P}'$ is derivable in $\mathcal{TD}_1$, then there is a derivation of $\mathcal{P}^\bullet, \Gamma \longrightarrow B$ in* Linc*, and for every $C \in (\mathcal{P}')^\bullet$, there is a derivation in* Linc *of the sequent $\mathcal{P}^\bullet \longrightarrow C$.*

Proof. The proof is by induction on the height of derivation Π of $\mathcal{P};\mathcal{T};\Gamma \longrightarrow B;\mathcal{P}'$. The only non-trivial steps in this proof occurs when Π ends with a branching rule, e.g.,

$$\frac{\mathcal{P};\mathcal{T};\cdot \longrightarrow B;\mathcal{P}_1 \quad \mathcal{P},\mathcal{P}_1;\mathcal{T};\cdot \longrightarrow B_2;\mathcal{P}'}{\mathcal{P};\mathcal{T};\cdot \longrightarrow B_1 \wedge B_2;\mathcal{P}',\mathcal{P}_1}\ \wedge\mathcal{R}$$

By the induction hypothesis, the sequents $\mathcal{P}^\bullet \longrightarrow B_1$, $\mathcal{P}^\bullet \longrightarrow C$ and $\mathcal{P}^\bullet, (\mathcal{P}')^\bullet \longrightarrow B_2$ are all provable in Linc, for every $C \in (\mathcal{P}')^\bullet$. We need only to show that $\mathcal{P}^\bullet \longrightarrow B_1 \wedge B_2$ are provable:

$$\cfrac{\mathcal{P}^\bullet \longrightarrow B_1 \quad \cfrac{\{\mathcal{P}^\bullet \longrightarrow C \mid C \in (\mathcal{P}')^\bullet\} \quad \mathcal{P}^\bullet, (\mathcal{P}')^\bullet \longrightarrow B_2}{\mathcal{P}^\bullet \longrightarrow B_2}\ cut^*}{\mathcal{P}^\bullet \longrightarrow B_1 \wedge B_2}$$

□

We say that a formula B is derivable in $\mathcal{TD}_i$ if the sequent $\cdot;\cdot \longrightarrow B;\mathcal{P}$ is derivable in $\mathcal{TD}_i$ for some $\mathcal{P}$.

Corollary 2. *If B is derivable in $\mathcal{TD}_1$ then B is also derivable in* Linc.

Adding history-free theories theories $\mathcal{T}$ does not change much the structure of the soundness proof in Theorem 6.

Theorem 7. *Let $\mathcal{T}$ be a history-free theory. If $\mathcal{P};\mathcal{T};\Gamma \longrightarrow B;\mathcal{P}'$ is derivable in $\mathcal{TD}_2$, then there is a derivation of $\mathcal{P}^\bullet, \mathcal{T}, \Gamma \longrightarrow B$ in* Linc*, and for every $C \in (\mathcal{P}')^\bullet$, there is a derivation in* Linc *of the sequent $\mathcal{P}^\bullet, \mathcal{T} \longrightarrow C$.*

Corollary 3. *If formula B is derivable in $\mathcal{TD}_2$ then B is also derivable in* Linc.

C Soundness of Strategy III

Theorem 8 (Soundness of $\mathcal{TD}_3$). *Suppose $\mathcal{P};\mathcal{T};\Gamma \longrightarrow C;\mathcal{P}'$ is derivable in $\mathcal{TD}_3$, where Γ is history-free and C contains no negative occurrences of history atoms. Let $\{p_1,\dots,p_n\}$ be the set of co-inductive predicates occurring in $\mathcal{P}, C$ and $\mathcal{P}'$. Then there exist invariants $S_1,\dots,S_n$ such that*

- *the sequent $(\mathcal{P}^\bullet, \Gamma \longrightarrow C[S_1/p_1^*,\dots,S_n/p_n^*])$ is derivable in* Linc*,*
- *for each $B \in (\mathcal{P}')^\bullet$, the sequent $(\mathcal{P}^\bullet \longrightarrow B)$ is derivable in* Linc*, and*
- *for each $p_i^\circ\vec{t} \in (\mathcal{P}')^\circ$, where $p_i\,\vec{x} \overset{\nu}{=} D_i\,p_i\,\vec{x}$, the sequent $(\mathcal{P}^\bullet \longrightarrow D_i\,S_i\,\vec{t})$ is derivable in* Linc.

Proof. Suppose Π is a derivation of $\mathcal{P};\mathcal{T};\Gamma \longrightarrow C;\mathcal{P}'$. We prove this theorem by induction on the height of Π.

For a given sequent $\mathcal{P};\mathcal{T};\Gamma \longrightarrow C;\mathcal{P}'$, construct S_i as follows:

$$S_i = \lambda\vec{x}.\bigwedge(\mathcal{P})^\bullet \wedge \bigvee\{(\vec{x}=\vec{t}) \mid p_i^\circ\,\vec{t} \in \mathcal{P}\cup\mathcal{P}'\},$$

where $p_i\vec{x} \overset{\nu}{=} D_i\,p_i\,\vec{x}$.

We look at some interesting cases:

- Π ends with *init*: the only interesting case is when the atom on the right is $p_i^*\,\vec{t}$ for some $\vec{t}$.
$$\frac{\mathcal{P},\mathcal{L} \vdash_I p_i^*\vec{t}}{\mathcal{P};\mathcal{T};\cdot \longrightarrow p_i^*\,\vec{t};\cdot}\ \mathit{init}$$
The only way $\mathcal{P} \vdash_I p_i^\circ\vec{t}$ could have been proved is via a backchaining using the clause $\forall\vec{x}.p_i^\circ \supset p_i^\circ\vec{x}$ in $\mathcal{L}$, followed by the identity rule, i.e., when $p_i^\circ\vec{t} \in \mathcal{P}$. Obviously in this case we have that $\mathcal{P} \longrightarrow S_i\vec{t}$ is also provable, since the equations $\vec{t}=\vec{t}$ are among the disjunctions in $S_i\vec{t}$.
- Π ends with $\wedge\mathcal{R}$:
$$\frac{\mathcal{P};\mathcal{T};\cdot \longrightarrow B;\mathcal{P}_1 \quad \mathcal{P},\mathcal{P}_1;\mathcal{T};\cdot \longrightarrow C;\mathcal{P}_2}{\mathcal{P};\mathcal{T};\cdot \longrightarrow B\wedge C;\mathcal{P}_1,\mathcal{P}_2}\ \wedge\mathcal{R}$$
where $\mathcal{P}' = \mathcal{P}_1\cup\mathcal{P}_2$. By induction, we have invariants $S_1',\dots,S_n'$ and $S_1'',\dots,S_n''$ s.t.
  1. $S_i' = \lambda\vec{x}.\bigwedge(\mathcal{P})^\bullet \wedge \bigvee\{(\vec{x}=\vec{t}) \mid p_i^\circ\,\vec{t} \in \mathcal{P}\cup\mathcal{P}_1\}$,
  2. $S_i'' = \lambda\vec{x}.\bigwedge(\mathcal{P},\mathcal{P}_1)^\bullet \wedge \bigvee\{(\vec{x}=\vec{t}) \mid p_i^\circ\,\vec{t} \in \mathcal{P}\cup\mathcal{P}_1\cup\mathcal{P}_2\}$,
  3. a Linc derivation Π_1 of $\mathcal{P}^\bullet \longrightarrow B[S_1'/p_1^*,\dots,S_n'/p_n^*]$,
  4. a Linc derivation Π_1^E of $\mathcal{P}^\bullet \longrightarrow E$, for each E in $\mathcal{P}_1^\bullet$,
  5. a Linc derivation $\Pi_1^i(\vec{t})$ of $\mathcal{P}^\bullet \longrightarrow D_iS_i'\,\vec{t}$, for each $p_i^\circ\vec{t} \in \mathcal{P}_1^\circ$, where $p_i\vec{x} \overset{\nu}{=} D_ip_i\vec{x}$,
  6. a Linc derivation Π_2 of $\mathcal{P}^\bullet,\mathcal{P}_1^\bullet \longrightarrow C[S_1''/p_1^*,\dots,S_n''/p_n^*]$,
  7. a Linc derivation Π_2^E of $\mathcal{P}^\bullet,\mathcal{P}_1^\bullet \longrightarrow E$, for each E in $\mathcal{P}_2^\bullet$, and
  8. a Linc derivation $\Pi_2^i(\vec{t})$ of $\mathcal{P}^\bullet,\mathcal{P}_1^\bullet \longrightarrow D_i\,S_i''\vec{t}$, for each $p_i^\circ\vec{t} \in \mathcal{P}_2^\circ$, where $p_i\vec{x} \overset{\nu}{=} D_ip_i\vec{x}$.

From the definition of S_i, we have

$$S_i = \lambda \vec{x}. \bigwedge(\mathcal{P})^\bullet \wedge \bigvee\{(\vec{x} = \vec{t}) \mid p_i^\circ\, \vec{t} \in \mathcal{P} \cup \mathcal{P}_1 \cup \mathcal{P}_2\}.$$

Then it is easy to show that, from $\mathit{\Pi}_1$ and $\mathit{\Pi}_2$, we have a derivation $\mathit{\Psi}_1$ of

$$\mathcal{P}^\bullet \longrightarrow B[S_1/p_1^*, \ldots, S_n/p_n^*]$$

and a derivation $\mathit{\Psi}_2$ of

$$\mathcal{P}^\bullet, \mathcal{P}_1^\bullet \longrightarrow C[S_1/p_1^*, \ldots, S_n/p_n^*].$$

We then need to prove the following:

- There is a Linc derivation of

  $$\mathcal{P}^\bullet \longrightarrow B \wedge C[S_1/p_1^*, \ldots, S_n/p_n^*].$$

  This derived as follows:

  $$\dfrac{\begin{array}{c}\mathit{\Psi}_1\\ \mathcal{P} \longrightarrow B[\cdots]\end{array} \qquad \dfrac{\left\{\begin{array}{c}\mathit{\Pi}_1^E\\ \mathcal{P}^\bullet \longrightarrow E\end{array}\right\}_{E\in\mathcal{P}_1^\bullet} \quad \begin{array}{c}\mathit{\Psi}_2\\ \mathcal{P}^\bullet, \mathcal{P}_1^\bullet \longrightarrow C[\cdots]\end{array}}{\mathcal{P} \longrightarrow C[\cdots]}\ \mathit{cut}^*}{\mathcal{P}^\bullet \longrightarrow B \wedge C[S_1/p_1^*, \ldots, S_n/p_n^*]}\ \wedge\mathcal{R}$$

- For each $E \in \mathcal{P}_1^\bullet \cup \mathcal{P}_2^\bullet$, there is a Linc derivation of $\mathcal{P}^\bullet \longrightarrow E$. This follows from (4) and (7) above.
- For each $p_i^\circ \vec{t} \in \mathcal{P}_1^\circ \cup \mathcal{P}_2^\circ$, there is a Linc derivation of $\mathcal{P}^\bullet \longrightarrow D_i\, S_i\, \vec{t}$. This follows from (5) and (8).

– $\mathit{\Pi}$ ends with ν_R^*:

$$\dfrac{\mathcal{P} \not\vdash_I p_k^* \vec{t} \quad \mathcal{P}, p_k^\circ \vec{t}; \mathcal{T}; \cdot \longrightarrow D_k\, p_k^* \vec{t}; \mathcal{P}_1}{\mathcal{P}; \mathcal{T}; \cdot \longrightarrow p_k^* \vec{t}; \mathcal{P}_1, p_k^\circ \vec{t}}\ \nu_R^*$$

By the induction hypothesis, we have a Linc-derivation $\mathit{\Psi}$ of

$$\mathcal{P}^\bullet \longrightarrow D_k S_k \vec{t},$$

a Linc-derivation $\mathit{\Pi}_E$ for each $E \in \mathcal{P}_1^\bullet$ and a Linc-derivation $\mathit{\Pi}_i(\vec{s})$ of $\mathcal{P}^\bullet \longrightarrow D_i S_i \vec{s}$ for each $p_i \vec{s} \in \mathcal{P}_1^\circ$. It remains to show that both $\mathcal{P}^\bullet \longrightarrow S_k \vec{t}$ and $\mathcal{P}^\bullet \longrightarrow D_k S_k \vec{t}$ are derivable in Linc. The latter follows from the induction hypothesis, whereas the former is trivial. This is because S_k takes the form:

$$\lambda \vec{x}. \bigwedge(\mathcal{P})^\bullet \wedge \bigvee(\cdots \vee \vec{x} = \vec{t} \vee \cdots)$$

so $\mathcal{P} \longrightarrow S_k \vec{t}$ is proved (reading the derivation upwards) via a series of $\wedge\mathcal{R}$ followed by $\vee\mathcal{R}$, selecting an appropriate equation, $\vec{t} = \vec{t}$.

- Π ends with ν_R:

$$\dfrac{\mathbb{S}(\mathcal{P}\setminus p_k^\circ,\mathcal{T})\not\vdash_I p_k\,\vec t \qquad (\mathcal{P}\setminus p_k^\circ), p_k^\circ\vec t;\mathcal{T};\cdot\longrightarrow D_k\,p_k^*\,\vec t;\mathcal{P}'}{\mathcal{P};\mathcal{T};\cdot\longrightarrow p_k\,\vec t;\mathcal{P}'[p_k/p_k^\circ],p_k\,\vec t}\ \nu_R$$

Note that $(\mathcal{P}\setminus p_k^\circ)^\bullet=\mathcal{P}^\bullet$. Notice that the history-predicate p_k° is removed in the conclusion, substituted by p_k. The difficult part in this case is to show that the sequent $\mathcal{P}^\bullet\longrightarrow p_k\vec s$, for every $p_k^\circ\vec s\in\mathcal{P}'\cup\{p_k^\circ\vec t\}$, is actually derivable in Linc. To do this we will need to use the CI$\mathcal{R}$ rule, and this requires us to construct a post fixed point of p_k. By the induction hypothesis, we have

$$S_k=\lambda\vec x.\bigwedge\mathcal{P}^\bullet\wedge\bigvee\{\vec x=\vec s\mid p_k^\circ\vec s\in\mathcal{P}'\cup\{p_k^\circ\vec t\}\}$$

and for each $p_k^\circ\vec s\in\mathcal{P}'\cup\{p_k^\circ\vec t\}$, we have a derivation $\Pi(\vec s)$ of $\mathcal{P}^\bullet\longrightarrow D_kS_k\vec s$. We claim that S_k is indeed a post-fixed of p_k. Since for each $p_k^\circ\vec s$ in $\mathcal{P}'\cup\{p_k^\circ\vec t\}$, the equation $\vec x=\vec s$ is among the equations in S_k, we can trivially construct a derivation $\Psi(\vec s)$ of $\mathcal{P}^\bullet\longrightarrow S_k\vec t$ in Linc. The derivation of $\mathcal{P}^\bullet\longrightarrow p_k\vec s$ is then constructed as follows:

$$\dfrac{\begin{array}{c}\Psi(\vec s)\\ \mathcal{P}^\bullet\longrightarrow S_k\vec s\end{array}\qquad \dfrac{\dfrac{\left\{\begin{array}{c}\Pi(\vec u)\\ \mathcal{P}^\bullet\longrightarrow D_kS_k\vec u\end{array}\right\}_{p_k^\circ\vec u\in\mathcal{P}'\cup\{p_k^\circ\vec t\}}}{\mathcal{P}^\bullet,\bigvee\{\vec x=\vec u\mid p_k^\circ\vec u\in\mathcal{P}'\cup\{p_k^\circ\vec t\}\}\longrightarrow D_kS_k\vec x}\ \vee\mathcal{L},\mathrm{eq}\mathcal{L}}{S_k\vec x\longrightarrow D_kS_k\vec x}\ \wedge\mathcal{L}}{\mathcal{P}^\bullet\longrightarrow p_k\vec s}\ \mathrm{CI}\mathcal{R}$$

□

Lemma 4. *$\mathcal{F}_{\mathcal{T}_1}\circ\mathcal{F}_{\mathcal{T}_2}$ and $\mathcal{F}_{\mathcal{T}_1\circ\mathcal{T}_2}$ define the same function.*

Proof. This follows directly from Definition 2 and Definition 4. Suppose $\mathcal{T}_1=\{\forall\vec z.p^\circ\vec y\supset p^*\vec z\}\cup\mathcal{T}_1'$. By simply expanding the definition of $\mathcal{F}_{\mathcal{T}_1\circ\mathcal{T}_2}\mathcal{R}\vec x$ we get:

$$\begin{array}{l}\mathcal{F}_{\mathcal{T}_1\circ\mathcal{T}_2}\mathcal{R}\vec x\\ =\ \forall q'.\bigwedge\mathcal{T}_1[q'/p^*]\wedge(\forall\vec z.(\forall q.\bigwedge\mathcal{T}_2[q/p^*,\mathcal{R}/p^\circ]\supset q\,\vec z)\supset q'\,\vec z)\ \supset\ q'\vec x\\ =\ \forall q'.\bigwedge\mathcal{T}_1[q'/p^*]\wedge(\forall\vec z.(\mathcal{F}_{\mathcal{T}_2}\mathcal{R}\vec z\supset q'\,\vec z))\supset q'\vec x\\ =\ \mathcal{F}_{\mathcal{T}_1}(\mathcal{F}_{\mathcal{T}_2}\mathcal{R})\,\vec x\\ =\ (\mathcal{F}_{\mathcal{T}_1}\circ\mathcal{F}_{\mathcal{T}_2})\mathcal{R}\,\vec x\end{array}$$

□

D Soundness of a simple up-to theory

Let us look at a simple example of a sound up-to function and how its soundness can be proved within Linc. To be more concrete, we consider the co-inductive definition of bisimulation for CCS, defined via the predicate *bisim* mentioned earlier. Suppose that *bisim* is defined by $bisim\ x\ y\overset{\nu}{=}D\ bisim\ x\ y$ where D is an abstraction.

Consider the following simple up-to function:

$$\mathcal{F}(\mathcal{R}) = \mathcal{R} \cup \sim$$

where $\sim$ denotes the largest bisimulation. This up-to function can be encoded as the following theory $\mathcal{T}$:

$$\forall x, y. bisim\ x\ y \supset bisim^*\ x\ y \qquad \forall x, y. bisim\ x\ y^\circ \supset bisim^*\ x\ y.$$

The formula $Snd(\mathcal{T})$ is thus

$$\forall \mathcal{R}.(\forall x, y.(\mathcal{R}\ x\ y \supset D\ (\mathcal{F}_\mathcal{T}\mathcal{R})\ x\ y)) \supset (\forall x, y.\mathcal{R}\ x\ y \supset bisim\ x\ y).$$

This can be proved as follows: Let P be the formula $\forall x, y.(\mathcal{R}\ x\ y \supset D\ (\mathcal{F}_\mathcal{T}\mathcal{R})\ x\ y)$ and let S be the abstraction:

$$\lambda x \lambda y. P \wedge (\mathcal{R}\ x\ y \vee bisim\ x\ y).$$

Using S as the co-inductive invariant, we construct a partial derivation as follows:

$$\dfrac{\dfrac{P, \mathcal{R}\ x\ y \longrightarrow S\ x\ y \quad S\ x\ y \longrightarrow D\ S\ x\ y}{\forall x, y.(\mathcal{R}\ x\ y \supset D\ (\mathcal{F}_\mathcal{T}\mathcal{R})\ x\ y)), \mathcal{R}\ x\ y \longrightarrow bisim\ x\ y}\ \mathrm{CI}\mathcal{R}}{\longrightarrow \forall \mathcal{R}.(\forall x, y.(\mathcal{R}\ x\ y \supset D\ (\mathcal{F}_\mathcal{T}\mathcal{R})\ x\ y)) \supset (\forall x, y.\mathcal{R}\ x\ y \supset bisim\ x\ y)}\ \forall\mathcal{R}, \supset\mathcal{R}$$

The left premise is straightforward to prove, so we show here only a derivation of the second premise, which establishes that S is a post fixed point of *bisim*.

$$\dfrac{\dfrac{\dfrac{}{\mathcal{R}xy \longrightarrow \mathcal{R}xy} \quad \dfrac{\begin{array}{c}\{P, \mathcal{F}_\mathcal{T}\mathcal{R}st \longrightarrow Sst\}\\ \vdots\end{array}}{P, D(\mathcal{F}_\mathcal{T}\mathcal{R})xy \longrightarrow DSxy}}{P, \mathcal{R}xy \longrightarrow DSxy} \quad \dfrac{\dfrac{\begin{array}{c}\{P, bisimuv \longrightarrow Suv\}\\ \vdots\end{array}}{P, Dbisimxy \longrightarrow DSxy}}{P, bisimxy \longrightarrow DSxy}\ \mathrm{CI}\mathcal{L}}{\dfrac{P, \mathcal{R}xy \vee bisimxy \longrightarrow DSxy}{Sxy \longrightarrow DSxy}\ \wedge\mathcal{L}}\ \vee\mathcal{L}$$

The dotted parts consist of straightforward applications of left and right logical rules. As *bisim* occurs only positively in $D\ bisim$, these rule applications leave us with open leaves of the form $(P, \mathcal{F}_\mathcal{T}\ s\ t \longrightarrow S\ s\ t)$ and $(P, bisim\ u\ v \longrightarrow S\ u\ v)$. The latter is easily derivable so we show here the former. Expanding the definition of $\mathcal{F}_\mathcal{T}\mathcal{R}$, we get:

$$\dfrac{\dfrac{\begin{array}{c}\vdots\\ P \longrightarrow \forall x, y.\mathcal{R}xy \supset Sxy\end{array} \quad \begin{array}{c}\vdots\\ P \longrightarrow \forall x, y.bisimxy \supset Sxy\end{array} \quad \dfrac{}{P, Sst \longrightarrow Sst}}{P, ((\forall x, y.\mathcal{R}xy \supset Sxy) \wedge (\forall x, y.bisimxy \supset Sxy)) \supset Sst \longrightarrow Sst}}{P, \forall q.((\forall x, y.\mathcal{R}xy \supset qxy) \wedge (\forall x, y.bisimxy \supset qxy)) \supset qst \longrightarrow Sst}\ \forall\mathcal{L}$$

Here the dotted parts consist of applications of $\forall\mathcal{R}$, $\supset\mathcal{R}$, $\wedge\mathcal{R}$ and $\vee\mathcal{R}$.

E Soundness of strategy IV

Theorem 9. *Suppose $\mathcal{P};\mathcal{T};\Gamma \longrightarrow C;\mathcal{P}'$ is derivable in $\mathcal{TD}_4$., Then there exists an invariant S such that*

- *the sequent $(Snd(\mathcal{T}), \mathcal{P}^\bullet, \Gamma \longrightarrow C[\mathcal{F}_\mathcal{T} S/p^*])$ is derivable in* Linc,
- *for each $B \in (\mathcal{P}')^\bullet$, the sequent $(\mathcal{P}^\bullet \longrightarrow B)$ is derivable in* Linc, *and*
- *for each $p\vec{t} \in (\mathcal{P}')^\circ$, the sequent $\mathcal{P}^\bullet \longrightarrow D\,(\mathcal{F}_\mathcal{T} S)\,\vec{t})$ is derivable in* Linc.

Proof. Given $\mathcal{P};\mathcal{T};\Gamma \longrightarrow C;\mathcal{P}'$, define S as

$$S = \lambda\vec{x}. \bigvee\{(\vec{x} = \vec{u}) \mid p^\circ\vec{u} \in \mathcal{P} \cup \mathcal{P}'\}.$$

Let Π be the proof of $\mathcal{P};\mathcal{T};\Gamma \longrightarrow C;\mathcal{P}'$. We prove this theorem by induction on the height of Π. We look at a couple of interesting cases: Suppose $\mathcal{T} = \{\forall\vec{y}.p^\circ\vec{y} \supset p^*\vec{y}\} \cup \mathcal{T}'$. Then

$$\bigwedge \mathcal{T}[q/p^\circ, S/p^*] = (\forall\vec{y}.S\,\vec{y} \supset q\,\vec{y}) \wedge \bigwedge \mathcal{T}'[q/p^*].$$

- Π ends with *init*:

$$\frac{\mathcal{P}, \forall\vec{y}.p^\circ\vec{y} \supset p^*\vec{y}, \mathcal{T}' \vdash_I p^*\vec{t}}{\mathcal{P}; \forall\vec{y}.p^\circ\vec{y} \supset p^*\vec{y}, \mathcal{T}'; \cdot \longrightarrow p^*\vec{t}; \cdot}\ init$$

In this case, $S = \mathcal{P}^\circ \cup \{p^\circ\vec{t}\}$. We only need to show that $Snd(\mathcal{T}), \mathcal{P}^\bullet \longrightarrow \mathcal{F}_\mathcal{T} S\,\vec{t}$.

$$\frac{Snd(\mathcal{T}), \mathcal{P}^\bullet, \forall\vec{y}.S\,\vec{y} \supset q\,\vec{y}, \mathcal{T}'[q/p^*] \longrightarrow q\,\vec{t}}{Snd(\mathcal{T}), \mathcal{P}^\bullet \longrightarrow \forall q.\,[(\forall\vec{y}.S\,\vec{y} \supset q\,\vec{y}) \wedge \bigwedge \mathcal{T}'[q/p^*] \supset q\,\vec{t}]}\ \forall\mathcal{R}, \supset\mathcal{R}, \wedge\mathcal{L}$$

The premise of this partial derivation can be can be constructed from the derivation of $\mathcal{P}, \mathcal{T} \vdash_I p^\circ\,\vec{t}$, by substituting q for p^*, where the identity instances are replaced as follows:

$$\frac{p^\circ\,\vec{s} \in \mathcal{P}}{\mathcal{P}, \ldots \longrightarrow p^\circ\,\vec{s}}\ init \quad \rightsquigarrow \quad \dfrac{\dfrac{\dfrac{}{\cdots \longrightarrow \vec{s} = \vec{s}}\ \wedge\mathcal{R}, \mathrm{eq}\mathcal{R}}{\cdots \longrightarrow S\,\vec{s}}\ \vee\mathcal{R} \qquad \dfrac{}{\cdots, q\,\vec{s} \longrightarrow q\,\vec{s}}\ init}{\mathcal{P}^\bullet, \forall\vec{y}.S\,\vec{y} \supset q\,\vec{y}, \ldots \vdash_I q\,\vec{s}}\ \forall\mathcal{L}, \supset\mathcal{L}$$

By the definition of S, the equation $\vec{s} = \vec{s}$ is in $S\,\vec{s}$, so it is trivially provable.

- Suppose Π ends with ν_R:

$$\frac{\mathcal{P} \setminus p^\circ, \mathcal{T} \not\vdash_I p\,\vec{t} \quad \mathcal{P} \setminus p^\circ; \mathcal{T}; \cdot \longrightarrow D\,p^*\,\vec{t}; \mathcal{P}'}{\mathcal{P}; \mathcal{T}; \cdot \longrightarrow p\,\vec{t}; \mathcal{P}'[p^\circ/p], p\,\vec{t}}\ \nu_R$$

Note that $(\mathcal{P} \setminus p^\circ)$ is in fact the same as $\mathcal{P}^\bullet$ because all our history atoms are of the form $p^\circ\vec{s}$. As in the proof of Theorem 1, the interesting part of the proof here is to show that $Snd(\mathcal{T}), \mathcal{P}^\bullet \longrightarrow p\,\vec{u}$ for every $p^\circ\vec{u} \in (\mathcal{P}' \cup \{p^\circ\vec{t}\})^\circ$.

By the induction hypothesis, for each $p^\circ \vec{u}$, we have a Linc-derivation $\Pi(\vec{u})$ of $\mathcal{P}^\bullet \longrightarrow D\ (\mathcal{F}_\mathcal{T} S)\ \vec{u}$.

$$
\dfrac{
  \dfrac{
    \dfrac{
      \left\{ \begin{array}{c} \Pi(\vec{s}) \\ \mathcal{P}^\bullet \longrightarrow D\ (\mathcal{F}_\mathcal{T} S)\ \vec{s} \end{array} \right\}_{p^\circ \vec{s} \in (\mathcal{P}' \cup \{p^\circ \vec{t}\})^\circ}
    }{
      \mathcal{P}^\bullet, S\,\vec{x} \longrightarrow D\ (\mathcal{F}_\mathcal{T} S)\ \vec{x}
    }\ \vee\mathcal{L}, \mathrm{eq}\mathcal{L}
  }{
    \mathcal{P}^\bullet \longrightarrow \forall \vec{x}.(S\,\vec{x} \supset D\ (\mathcal{F}_\mathcal{T} S)\ \vec{x})
  }\ \forall\mathcal{R}, \supset\mathcal{R}
  \qquad
  \begin{array}{c} \vdots \\ (\forall \vec{y}.S\,\vec{y} \supset p\ \vec{y}), \mathcal{P}^\bullet \longrightarrow p\,\vec{u} \end{array}
}{
  \dfrac{
    (\forall \vec{x}.(S\,\vec{x} \supset D\ (\mathcal{F}_\mathcal{T} S)\ \vec{x})) \supset (\forall \vec{y}.S\,\vec{y} \supset p\ \vec{y}), \mathcal{P}^\bullet \longrightarrow p\,\vec{u}
  }{
    \forall \mathcal{R}.(\forall \vec{x}.(\mathcal{R}\vec{x} \supset D\ (\mathcal{F}_\mathcal{T} \mathcal{R})\ \vec{x})) \supset (\forall \vec{y}.\mathcal{R}\vec{y} \supset p\ \vec{y}), \mathcal{P}^\bullet \longrightarrow p\,\vec{u}
  }\ \forall\mathcal{L}
}\ \supset\mathcal{L}
$$

The right-premise of the instance of $\supset\mathcal{L}$ above can be proved by instantiating $\vec{y}$ with $\vec{u}$ and use $\supset\mathcal{L}$ on $S\,\vec{u} \supset p\,\vec{u}$. This is because by definition, S contains an equation of the form $\vec{y} = \vec{u}$.

□

Lemma 5. *Let $\mathcal{T}_1$ and $\mathcal{T}_2$ be two Horn up-to theories such that $\mathcal{D}(\mathcal{T}_2)$ permutes over $\mathcal{D}(\mathcal{T}_1)$. Then $(\mathcal{P}, \mathcal{T}_1, \mathcal{T}_2 \vdash_I p^* \vec{t})$ iff $(\mathcal{P}, \mathcal{T}_1 \circ \mathcal{T}_2 \vdash_I p^* \vec{t})$, for every set of $*$-free atoms $\mathcal{P}$ and every $\vec{t}$.*

Proof. (*Outline*) We make use of the fact that goal-directed proofs involving hH theories are complete; see [7]. One direction, from $\mathcal{T}_1 \circ \mathcal{T}_2$ to $\mathcal{T}_1 \cup \mathcal{T}_2$ is easy, as the structure of $\mathcal{T}_1 \circ \mathcal{T}_2$ forces one to apply $\mathcal{T}_1$ first before applying $\mathcal{T}_2$ in a goal-directed proof. This proof is easily mimicked in $\mathcal{T}_1 \cup \mathcal{T}_2$. The other direction follows from the fact that we can permute $\mathcal{D}(\mathcal{T}_2)$ over $\mathcal{D}(\mathcal{T}_1)$, so when there exists a derivation of $p^* \vec{t}$ in $\mathcal{D}(\mathcal{T}_1) \cup \mathcal{D}(\mathcal{T}_2)$, there also exists another derivation in which rules of $\mathcal{D}(\mathcal{T}_1)$ are applied first, before rules of $\mathcal{D}(\mathcal{T}_2)$. This can then be translated to a derivation using $\mathcal{T}_1 \circ \mathcal{T}_2$. □