Winskel

Formal Methods in System Design, 1: 323–354 (1992)

Compositional Checking of Satisfaction

HENRIK REIF ANDERSEN
Department of Computer Science, Aarhus University, Ny Munkegade 116, DIC-8000 Aarhus C, Denmark

GLYNN WINSKEL
Department of Computer Science, Aarhus University, Ny Munkegade 116, DIC-8000 Aarhus C, Denmark

Abstract. We present a compositional method for deciding whether a process satisfies an assertion. Assertions are formulas in a modal ν-calculus, and processes are drawn from a very general process algebra inspired by CCS and CSP. Well-known operators from CCS, CSP, and other process algebras appear as derived operators. The method is *compositional in the structure of processes* and works purely on the syntax of processes. It consists of applying a sequence of *reductions*, each of which only takes into account the top-level operator of the process. A reduction transforms a satisfaction problem for a composite process into equivalent satisfaction problems for the immediate subcomponents. Using process variables, systems with undefined subcomponents can be defined, and given an overall requirement to the system, *necessary and sufficient conditions* on these subcomponents can be found. Hence the process variables make it possible to specify and reason about what are often referred to as *contexts, environments,* and *partial implementations.* Since reductions are algorithms that work on syntax, they can be considered as forming a bridge between traditional noncompositional model checking and compositional proof systems.

Keywords: process calculi, modal μ-calculus, model checking, compositionality

1. Introduction

In this article we present a compositional method for deciding whether a finite-state process satisfies a specification. Processes will be described in a very general and rich process algebra, which includes common operators from process algebras such as CCS and CSP. This algebra contains primitive operators to reflect sequentiality (by the well-known operation of prefixing), nondeterministic choice, asynchronous and synchronous parallel composition, recursion, relabeling, and restriction. Specifications will be drawn from a modal ν-calculus with negation, in which a variety of properties can be specified. These include the usual *liveness, safety,* and *fairness* properties, as well as all operators from ordinary linear and branching-time temporal logics (see, e.g., [1] and [2]).

The method we advocate is *compositional in the structure of processes* and works purely on the syntactical level without any explicit references to the underlying transition system. Compositionality is important for at least the following two reasons. Firstly, it makes the verification *modular*, so that when changing a part of a system only the part of the verification concerning that particular component must be redone. Secondly, when designing a system or *synthesizing* a process, the compositionality makes it possible to have undefined parts of a process and

still be able to reason about it. For instance, it might be possible to reveal inconsistencies in the specification or prove that with the choices already taken in the design, no component supplied for the missing parts will ever be able to make the overall system satisfy the original specification.

This approach is unlike traditional model checking, where a transition system model of a process is built and the specification formula is checked by applying some algorithm to the transition system. There are several versions of this basic idea in the literature, e.g., Emerson and Lei [3], Clarke et al. [4], Stirling and Walker [5], Larsen [6], Winskel [7], Cleaveland [8], and Arnold and Crubille [9]. Recently there have been attempts to extend some of these methods based on transition systems to compositional methods by Clarke, Long, and McMillan [10] and Larsen and Xinxin [11], but none of these are compositional in the structure of processes.

Our method consists of applying a sequence of *reductions*, each of which removes the top-most operator of the process, i.e., a reduction transforms a satisfaction problem for a composite process to satisfaction problems for the immediate subcomponents of the process—without inspecting these. Starting with a process term, one can repeatedly use the reductions until a trivial process (for which satisfaction is easily decided) or a variable remains.

2. The languages

2.1. Syntax

Assume given a set of *state names Nam*, and a finite set of *actions Act*. Processes are denoted by syntactic terms t constructed from the following grammar:

$$t ::= nil \mid at \mid t_0 + t_1 \mid t_0 \times t_1 \mid t \restriction \Lambda \mid t\{\Xi\} \mid rec\ P.t \mid P,$$

where P is an element in *Nam*, i.e., a state identifier. The usual notion of free and bound will apply to state identifiers P, so that P will be bound in *rec* $P.t$ but free in $P + nil$.

Nil is the inactive process, and at is the usual prefix and $t_0 + t_1$ the usual sum operations known from CCS. The product term $t_0 \times t_1$ denotes a very general kind of parallel composition that allows the components t_0 and t_1 to proceed both synchronously and asynchronously. The exact semantics is defined below.

A state identifier P in the body of *rec* $P.t$ works as a *recursion point*, and in effect will behave as the normal recursion in CCS: a term *rec* $P.t$ has the same behavior as the *unfolded* term $t[rec\ P.t/P]$ (the result of substituting *rec* $P.t$ for all free occurrences of P in t). We impose the syntactic restriction on recursive terms, that no product must appear in the body, which ensures that all definable processes are finite state, and for technical reasons we also require every occurrence of P in *rec* $P.t$ to be *strongly guarded*, i.e., appear immediately under a prefix.

In the prefix *at*, a denotes an action in *Act*. For a given set of actions *Act*, we define a set of *composite actions*. Let $*$ be a distinguished symbol not contained in *Act*. The symbol $*$ is called the *idling action* and is interpreted as "no action" or "inaction." Define Act_* to be the least set including $Act \cup \{*\}$ and such that $\alpha, \beta \in Act_*$ implies $\alpha \times \beta \in Act_*$ taking $* \times * = *$. Now $\Xi : Act_* \rightharpoonup Act$, is a *relabeling* that is a partial function, with finite domain, mapping nonidling actions to nonidling actions. This relabeling can be extended to a total function on Act_* by taking it to behave as the identity outside the domain. The term $t \upharpoonright \Lambda$ is a *restriction* where Λ is a finite subset of Act_*.

Properties of processes are denoted by assertions A from a modal ν-calculus:

$$A ::= \neg A \mid A_0 \vee A_1 \mid \langle \alpha \rangle A \mid X \mid \nu X.A \mid (t : A),$$

where X ranges over a set of assertion variables. In the maximal fixed-point formula $\nu X.A$, any free occurrence of X must be within an even number of negations in order to guarantee the existence of a unique maximal fixed point. The action name α belongs to the set of composite actions Act_*. The *correctness assertion* $(t : A)$ denotes true if t satisfies A and false otherwise. An assertion is said to be *pure* if it does not contain any correctness assertions.

Many derived operators can easily be defined in terms of the core language and will be used throughout this article:

$$\begin{array}{lllllll} [\alpha]A & = & \neg\langle\alpha\rangle\neg A, & \mu X.A & = & \neg\nu X.\neg A[\neg X/X], \\ T & = & \nu X.X, & A \rightarrow B & = & \neg A \vee B, \\ F & = & \neg T, & A \leftrightarrow B & = & (A \rightarrow B) \wedge (B \rightarrow A). \end{array}$$

Here we have used the notation $A[B/X]$, which denotes the assertion resulting from substituting B for all free occurrences of X in A. We will say that an assertion A is *closed* if it contains no free variables. Furthermore, for a finite set $K \subseteq Act_*$ we define $\langle K \rangle A = \bigvee_{\kappa \in K} \langle \kappa \rangle A$ where disjunction over an empty set gives false (F).

The correctness assertions $(t : A)$ are atoms in a propositional logic that will be used to express reductions. A grammar for the logic is

$$L ::= T \mid \neg L \mid L_0 \vee L_1 \mid (t : A).$$

In the logical language L, we are able to express complex relationships between properties of different processes. For example,

$$(p + q : \langle \alpha \rangle A) \leftrightarrow (p : \langle \alpha \rangle A) \vee (q : \langle \alpha \rangle A),$$

expresses a very simple example of a reduction. It states that the process $p + q$ can do an α and get into a state that satisfies A if and only if p or q can do an α and get into a state that satisfies A. It is a reduction because the formula is valid for all p's and q's, and the validity of $(p + q : \langle \alpha \rangle A)$ is reduced to validity

of correctness assertions over the subterms p and q. Although this reduction is almost trivial, in general, it might be quite difficult to get reductions. Consider, for example, the problem of choosing a B such that

$$(rec\ P.t : \nu X.A) \leftrightarrow (t : B)$$

holds. The aim of this article is to describe a method for supplying such a B and analogous assertions for all the other operators.

2.2. *Semantics*

In order to define the semantics, we first recall some well-known definitions of transition systems.

Definition 1. A *transition system* T is a triple $(S, L, \rightarrow)$ where S is a set of states, L a set of labels, and $\rightarrow \subseteq S \times L \times S$ a transition relation. The *set of reachable states* R_p from a state $p \in S$ is defined as the least subset of S containing p and closed under $\xrightarrow{L}$, where $\xrightarrow{L} = \bigcup_{l \in L} \xrightarrow{l}$. A *pointed transition system* T is a quadruple $(S, L, \rightarrow, i)$ where $(S, L, \rightarrow)$ is a transition system, $i \in S$ is an initial state, and all states in S must be reachable from i, i.e., S must equal R_i.

Given a pointed transition system $T = (S, L, \rightarrow, i)$, the *rooting of* T is a pointed transition system $\underline{T} = (S \cup \{\underline{i}\}, L, \rightarrow', \underline{i})$, where $\underline{i}$ is a new state assumed not to be in S, and the transition relation $\rightarrow' \subseteq (S \cup \{\underline{i}\}) \times L \times (S \cup \{\underline{i}\})$ is defined by

$$\rightarrow' = \rightarrow \cup \{(\underline{i}, \alpha, q) \mid i \xrightarrow{\alpha} q\}.$$

Pictorially, the rooting of a pointed transition system is constructed by adjoining a new initial state with the same outgoing transitions as the old initial state.

The rooting of a transition system T is just as good as T with respect to satisfaction in our logic. This claim is made precise by the rooting lemma below.

The semantics of process terms is given by the transition system $\mathcal{T} = (\mathcal{S}, Act_*, \rightarrow)$, where $\mathcal{S}$ is the set of process terms (including terms with free state identifiers), Act_* the set of composite actions, and $\rightarrow \subseteq \mathcal{S} \times Act_* \times \mathcal{S}$ is the transition relation given as the least relation satisfying the following rules:

$$p \xrightarrow{*} p, \qquad ap \xrightarrow{a} p,$$

$$\frac{p \xrightarrow{\alpha} p'}{p + q \xrightarrow{\alpha} p'} \quad \alpha \neq *, \qquad \frac{q \xrightarrow{\alpha} q'}{p + q \xrightarrow{\alpha} q'} \quad \alpha \neq *,$$

$$\frac{p \xrightarrow{\alpha} p' \quad q \xrightarrow{\beta} q'}{p \times q \xrightarrow{\alpha \times \beta} p' \times q'}, \qquad \frac{t[rec\ P.t/P] \xrightarrow{\alpha} t'}{rec\ P.t \xrightarrow{\alpha} t'} \quad \alpha \neq *,$$

$$\frac{p \xrightarrow{\alpha} p'}{p\{\Xi\} \xrightarrow{\beta} p'\{\Xi\}} \quad \Xi(\alpha) = \beta, \qquad \frac{p \xrightarrow{\alpha} p'}{p \upharpoonright \Lambda \xrightarrow{\alpha} p' \upharpoonright \Lambda} \quad \alpha \in \Lambda.$$

Note in particular the rule for product. One of the components in the product may *idle* by means of the idling action $*$, allowing the other component to proceed independently, as in the transition

$$p \xrightarrow{\alpha \times *} p',$$

where the left component of p performs an α-action and the right component idles.

For a transition system $T = (S, L, \rightarrow)$, an assertion A denotes a *property* of T, which we take to be a subset of S. Hence, the set of all properties of T is the powerset $\mathcal{P}(S)$. Since assertions may contain free variables, we introduce the notion of an environment that describes the interpretation of the variables. An *environment of assertions* for T is a map

$$\phi : \mathrm{Var}_A \to \mathcal{P}(S),$$

which assigns properties to assertion variables. The environment $\phi[U/X]$ is like ϕ except that the variable X is mapped to U.

Formally, relative to the transition system $T = (S, L, \rightarrow)$, the assertion A denotes the property $[\![A]\!]_T\phi$ defined inductively on the structure of A:

$$\begin{aligned}
[\![\neg A]\!]_T\phi &= S \backslash [\![A]\!]_T\phi \\
[\![A_0 \vee A_1]\!]_T\phi &= [\![A_0]\!]_T\phi \cup [\![A_1]\!]_T\phi \\
[\![\langle\alpha\rangle A]\!]_T\phi &= \{s \in S \mid \exists s' \in S.\, s \xrightarrow{\alpha} s' \ \&\ s' \in [\![A]\!]_T\phi\} \\
[\![X]\!]_T\phi &= \phi(X) \\
[\![\nu X.A]\!]_T\phi &= \nu U \subseteq S.\psi(U) \\
&\qquad \text{where } \psi : U \mapsto [\![A]\!]_T\phi[U/X] \\
[\![(t : A)]\!]_T\phi &= \begin{cases} S & \text{if } t \in [\![A]\!]_T\phi \\ \emptyset & \text{otherwise} \end{cases}
\end{aligned}$$

The powerset $\mathcal{P}(S)$ ordered by inclusion is a complete lattice, and since we require all variables to appear under an even number of negations, the map ψ will always be monotonic. Consequently, by Tarski's lemma [2], ψ will have a maximum fixed point (the *largest postfixed point*), which we denote by $\nu U \subseteq S.\psi(U)$.

Define $[\![A]\!]\phi = [\![A]\!]_T\phi$. This gives the standard *global interpretation* of assertions over all states S.

For a transition system $T = (S, L, \rightarrow)$, and for a subset Q of S we have the induced transition system

$$T_Q = (Q, L, \rightarrow \cap (Q \times L \times Q)),$$

which is T restricted to the set of states Q. Writing $[\![A]\!]_Q\phi$ for $[\![A]\!]_{T_Q}\phi$, we get a *local interpretation* of A. For particular choices of the subset Q, the local and

global interpretations coincide, as is captured by the locality lemma below. Let ϕ_Q denote the environment that on the variable X gives $\phi(X) \cap Q$.

Lemma 1 (locality lemma). Let $T = (S, L, \rightarrow)$ be a transition system. Given an assertion A, an environment ϕ and a subset Q of S, suppose Q satisfies the closedness criterion: Q is closed under $\xrightarrow{K}$, where K is the set of actions appearing inside diamonds in A. Then the following equality holds:

$$[\![A]\!]_{T_Q}\phi_Q = [\![A]\!]_T\phi \cap Q.$$

Proof. The proof is straightforward using structural induction on A. □

With the transition system $\mathcal{T}$, one particularly interesting choice of Q is the set of reachable states R_p from a state p that by definition satisfies the closedness criterion of the locality lemma. Suppose we wanted to check whether a particular state p belongs to the set of states denoted by an assertion A. Then by the locality lemma we obtain:

$$\begin{aligned} p \in [\![A]\!]\phi \quad &\text{iff} \quad p \in [\![A]\!]\phi \cap R_p \\ &\text{iff} \quad p \in [\![A]\!]_{R_p}\phi_{R_p}. \end{aligned}$$

As mentioned previously, the rooting of a transition system T is "just as good as" T with respect to satisfaction in our logic—which is the intuitive content of the following lemma.

Lemma 2 (rooting lemma). Given a pointed transition system, $T = (S_T, L_T, \rightarrow_T, i_T)$, with the rooting $\underline{T}$, let $r : \mathcal{P}(S_T) \rightarrow \mathcal{P}(S_T)$ be the map on properties that take the initial state of T to the two copies of it in $\underline{T}$ and that take all other states to their obvious counterparts. Let $\phi : \mathrm{Var}_A \rightarrow \mathcal{P}(S_T)$ be an environment of assertions. Assume S_T is countable and A pure. Then

$$r([\![A]\!]_T\phi) = [\![A]\!]_{\underline{T}}(r \circ \phi).$$

Proof. See appendix 1. □

The connection given by the rooting lemma between pointed transition systems T and their rootings $\underline{T}$ is very useful: the set of states satisfying an assertion will be the same in both interpretations up to application of the map r. In particular the initial state of T will satisfy A if and only if the initial state of $\underline{T}$ satisfies A—an observation central to our development of reductions in section 3.

There is another technical lemma that states a close relationship between syntactic and semantic substitution on assertions and that will be used frequently in the proofs.

Lemma 3 (substitution lemma). For B a closed assertion, X a variable, A an arbitrary, pure assertion, and ϕ an environment for T, we have

$$[\![A[B/X]]\!]_T\phi = [\![A]\!]_T\phi[[\![B]\!]_T\phi/X].$$

Proof. The proof is straightforward using structural induction on A. □

For the propositional logic, we define the satisfaction predicate $\models_\phi$ relative to an environment ϕ:

$\models_\phi T$ always

$\models_\phi \neg L$ iff not $\models_\phi L$

$\models_\phi L_0 \vee L_1$ iff $\models_\phi L_0$ or $\models_\phi L_1$

$\models_\phi t : A$ iff $t \in [\![A]\!]\phi$

Furthermore, we define the derived predicate $\models$ as

$\models L$ iff for all ϕ $\models_\phi L$.

Taking $\bullet$ to be the trivial transition system with one state (denoted $\bullet$) and no transitions, we observe that the set of assertions built from correctness assertions, negations, and conjunctions when interpreted over $\bullet$ is essentially a copy of the logic L, i.e., for such an assertion A we have $[\![A]\!]_\bullet\phi = \{\bullet\}$ if and only if $\models_\phi A$, where A is interpreted as a formula in the propositional logic.

3. Reductions

Our method for compositional checking of satisfaction is based on the notion of a *reduction*, which we explain in terms of the prefix operator.

Given a pure and closed assertion A and a prefix *at*, we would like to find a propositional expression B over atoms $(t : B_i)$ such that the following holds:

$$\models (at : A) \leftrightarrow B.$$

Having found such a B, the validity of $(at : A)$ has been *reduced* to validity of a propositional expression containing only atoms on the subterm t. In other words, B is a *necessary* and *sufficient* condition on the subterm t ensuring that *at* satisfies A. By the word *reduction* we will henceforth understand *an algorithmic description of how to find B given A and at.*

It is not obvious that such a B exists. Although we can easily express the set of processes that will make the correctness assertion valid as

$$\{t \in \mathcal{S} \mid \models at : A\},$$

it is not necessarily the case that this set can be expressed *within the logic* L as an assertion B over atoms $(t : B_i)$ such that

$$\{t \in \mathcal{S} \mid \models B\} = \{t \in \mathcal{S} \mid \models at : A\}.$$

In general, the ability to do so will depend on the expressive power of the logic and on the kind of operation for which we are trying to find a reduction. We will show that for our modal logic and all operators of our process algebra, such a B does indeed exist, and furthermore we give for each operator an algorithm that computes one particular choice of B.

In providing this B, the most difficult part concerns—not surprisingly—the fixed points. The single most important property of fixed points around which all the reductions are centered is expressed by the reduction lemma. Recall that a map on a complete meet semilattice is ω-anticontinuous if it preserves meets of all decreasing ω-chains.

Lemma 4 (reduction lemma). Suppose D and E are powersets over countable sets, and in: $D \to E$ an ω-anticontinuous function with $in(\top_D) = \top_E$. Suppose $\psi : E \to E$ and $\theta : D \to D$ are both monotonic and have the property

$$\psi \circ in = in \circ \theta.$$

We can then conclude that

$$\nu\psi = in(\nu\theta).$$

Proof. See appendix 2. □

To understand the role of the reduction lemma, take E to be the lattice of properties of a compound process and D to be a lattice built from properties of immediate subprocesses. The lemma allows us to express a fixed-point property of the original compound process in terms of fixed points of functions over properties of its immediate subcomponents via the transformation *in*.

For example, the properties of a process at can be identified with certain subsets of the states $S_{\underline{at}}$ in the rooting of the transition system pointed by at, and the properties of t can be identified with subsets of the states S_t of the transition system pointed by t. Now we take the transformation to be

$$in : \mathcal{P}(S_t) \times \mathcal{P}(\{\bullet\}) \to \mathcal{P}(S_{\underline{at}}),$$

where $in(V_0, V_1) = V_0 \cup \{\underline{at} \mid \bullet \in V_1\}$. The role of the extra product component is to record whether or not the property holds at the initial state $\underline{at}$ of $S_{\underline{at}}$ (The rooting is required to ensure that the initial state at is not confused with later occurrences.)[1]

An assertion with a free variable occurring positively essentially denotes a monotonic function $\psi : \mathcal{P}(S_{\underline{at}}) \to \mathcal{P}(S_{\underline{at}})$. The definition of the reduction is

given by structural induction on assertions ensuring that assertions denoting such functions ψ and their reductions denoting monotonic functions $\theta : \mathcal{P}(S_t) \times \mathcal{P}(\{\bullet\}) \to \mathcal{P}(S_t) \times \mathcal{P}(\{\bullet\})$, are related by *in* in the manner demanded by the reduction lemma. The lemma then allows the reduction to proceed for fixed points. As this case of prefixing makes clear, reductions of fixed points can be simultaneous fixed points. However, the use of Bekić's theorem [13] replaces the simultaneous fixed points by fixed points in the individual components. In the case where these individual components lie in powersets of singletons, they end up being replaced by Boolean values for closed assertions.

In the course of this definition by structural induction, we will be faced with the problem of giving a reduction for assertion variables. One solution to this problem can be found by introducing a syntactic counterpart of *in* called *IN* and defining a *change of variables* σ to be a map taking all variables X to $IN(X_0, X_1)$. An application of such a substitution to an assertion A has to satisfy certain technical requirements: it should be *fresh*, i.e., for an assertion A when 1) for all variables X at which σ is defined, the free variables in $\sigma(X)$ are disjoint from those in A, and 2) for distinct variables X and X', at which σ is defined, the free variables in $\sigma(X)$ and $\sigma(X')$ are disjoint. We will use the notation $A[\sigma]$ to denote the assertion resulting from performing the substitution σ, and we use $\sigma\backslash X$ to denote the substitution that is like σ except that X is left unchanged. The meaning of *IN* can be summarized by the equation

$$[\![IN(X_0, X_1)]\!]_{\underline{at}}\phi = in(\phi(X_0), \phi(X_1)),$$

justifying that *IN* is the "syntactic counterpart of *in*." It is emphasized that while the syntactic counterparts *IN* of the transformations play the important part in reductions of expressing relationships between variables, they do *not* appear in the reductions themselves.

Reductions for all operators can be established along the lines sketched. Each operator involves a judicious choice of *in*, which *IN* is to denote. In the following sections we present this choice and the accompanying reductions.

3.1. *Prefix*

The reduction for prefix is defined inductively on the structure of assertions and is shown in figure 1. Note that $\mathrm{red}^0(at : A; \sigma)$ just renames the variables of A from X to X_0. The transformation *in* was explained in the previous section.[2]

The reduction is constructed in such a way that the two components are related to A through *in* by

$$[\![A[\sigma]]\!]_{\underline{at}}\phi = in([\![\mathrm{red}^0(at : A; \sigma)]\!]_t\phi, [\![\mathrm{red}^1(at : A; \sigma)]\!]_\bullet\phi), \tag{1}$$

where σ is a change of variables for A. From the rooting lemma, we know that

$$at \in [\![A]\!]_{at}\phi \text{ iff } \underline{at} \in [\![A]\!]_{\underline{at}}\phi$$

$$
\begin{array}{lcl}
\mathrm{red}^0(at : X;\sigma) & = & X_0 \\
 & & \text{where } \sigma(X) = IN(X_0, X_1) \\
\mathrm{red}^0(at : \nu X.A;\sigma) & = & \nu X_0.\mathrm{red}^0(at : A;\sigma) \\
 & & \text{where } \sigma(X) = IN(X_0, X_1) \\
\mathrm{red}^0(at : \langle\alpha\rangle A;\sigma) & = & \langle\alpha\rangle\mathrm{red}^0(at : A;\sigma) \\
\mathrm{red}^0(at : \neg A;\sigma) & = & \neg\mathrm{red}^0(at : A;\sigma) \\
\mathrm{red}^0(at : A \vee B;\sigma) & = & \mathrm{red}^0(at : A;\sigma) \vee \mathrm{red}^0(at : B;\sigma) \\
\mathrm{red}^1(at : X;\sigma) & = & X_1 \\
 & & \text{where } \sigma(X) = IN(X_0, X_1) \\
\mathrm{red}^1(at : \nu X.A;\sigma) & = & \mathrm{red}^1(at : A;\sigma)[\mathrm{red}^0(at : \nu X.A;\sigma)/X_0][T/X_1] \\
 & & \text{where } \sigma(X) = IN(X_0, X_1) \\
\mathrm{red}^1(at : \langle\alpha\rangle A;\sigma) & = & \begin{cases} t : \mathrm{red}^0(at : A;\sigma) & \text{if } \alpha = a \\ F & \text{if } \alpha \neq a \end{cases} \\
\mathrm{red}^1(at : \neg A;\sigma) & = & \neg\mathrm{red}^1(at : A;\sigma) \\
\mathrm{red}^1(at : A \vee B;\sigma) & = & \mathrm{red}^1(at : A;\sigma) \vee \mathrm{red}^1(at : B;\sigma)
\end{array}
$$

Figure 1. Reduction for prefix defined inductively on the structure of assertions.

and from the definition of *in* and equation (1), we get

$$at \in [\![A]\!]_{at}\phi \text{ iff } \bullet \in [\![\mathrm{red}^1(at : A;\sigma)]\!]_\bullet\phi.$$

As $\mathrm{red}^1(at : A;\sigma)$ consists of correctness assertions, negations, and conjunctions only, we can consider it to be a formula in our propositional logic, yielding our reduction

$$\models (at : A) \leftrightarrow \mathrm{red}^1(at : A;\sigma).$$

Theorem 1 (reduction for prefix). Given a closed, pure assertion A, a change of variables σ that is fresh for A, and an arbitrary process term t, then

$$\models (at : A) \leftrightarrow \mathrm{red}^1(at : A;\sigma).$$

Proof. See appendix 3. □

3.2. *Nil*

The reduction for nil is defined inductively on the structure of assertions and is

$$
\begin{array}{lcl}
\mathrm{red}(nil : X; \sigma) & = & Y \text{ where } \sigma(X) = IN(Y) \\
\mathrm{red}(nil : \nu X.A; \sigma) & = & \mathrm{red}(nil : A; \sigma)[T/Y] \text{ where } \sigma(X) = IN(Y) \\
\mathrm{red}(nil : \langle\alpha\rangle A; \sigma) & = & F
\end{array}
$$

Figure 2. Reduction for nil.

$$
\begin{array}{lcl}
\mathrm{red}^0(t_0 + t_1 : X; \sigma) & = & X_0 \\
 & & \text{where } \sigma(X) = IN(X_0, X_1) \\
\mathrm{red}^0(t_0 + t_1 : \nu X.A; \sigma) & = & \nu X_0.\mathrm{red}^0(t_0 + t_1 : A; \sigma) \\
 & & \text{where } \sigma(X) = IN(X_0, X_1) \\
\mathrm{red}^0(t_0 + t_1 : \langle\alpha\rangle A; \sigma) & = & \langle\alpha\rangle \mathrm{red}^0(t_0 + t_1 : A; \sigma) \\
 & & \\
\mathrm{red}^1(t_0 + t_1 : X; \sigma) & = & X_1 \\
 & & \text{where } \sigma(X) = IN(X_0, X_1) \\
\mathrm{red}^1(t_0 + t_1 : \nu X.A; \sigma) & = & \mathrm{red}^1(t_0 + t_1 : A; \sigma)[\mathrm{red}^0(t_0 + t_1 : \nu X.A; \sigma)/X_0][T/X_1] \\
 & & \text{where } \sigma(X) = IN(X_0, X_1) \\
\mathrm{red}^1(t_0 + t_1 : \langle\alpha\rangle A; \sigma) & = & (t_0 : \langle\alpha\rangle A^0) \vee (t_1 : \langle\alpha\rangle A^0) \\
 & & \text{where } A^0 = \mathrm{red}^0(t_0 + t_1 : A; \sigma)
\end{array}
$$

Figure 3. Reduction for sum.

shown in figure 2. The definitions of $\neg$ and $\vee$ are similar to the definitions for prefix and are therefore omitted. The transformation *in:* $\mathcal{P}(\{\bullet\}) \to \mathcal{P}(\{nil\})$ is just the direct image of the obvious isomorphism between $\{\bullet\}$ and $\{nil\}$. Note that the reduction for *nil* is quite trivial and just gives true (T) or false (F).

Theorem 2 (reduction for nil). Given a closed, pure assertion A and a change of variables σ that is fresh for A, then $\models (nil : A) \leftrightarrow \mathrm{red}(nil : A; \sigma)$.

Proof. See appendix 3. □

3.3. Sum

The reduction for sum is presented in figure 3. The definitions for $\neg$ and $\vee$ are omitted since they are similar to the definitions for prefix.

$$
\begin{array}{lcl}
\mathrm{red}(t\{\Xi\} : X; \sigma) & = & Y \text{ where } \sigma(X) = IN(Y) \\
\mathrm{red}(t\{\Xi\} : \nu X.A; \sigma) & = & \nu Y.\mathrm{red}(t\{\Xi\} : A; \sigma) \text{ where } \sigma(X) = IN(Y) \\
\mathrm{red}(t\{\Xi\} : \langle\alpha\rangle A; \sigma) & = & \langle\Xi^{-1}(\alpha)\rangle \mathrm{red}(t\{\Xi\} : A; \sigma)
\end{array}
$$

Figure 4. Reduction for relabeling.

To understand the transformation, first note that we have a map $j : S_{t_0} + S_{t_1} \to S_{\underline{t_0+t_1}}$ taking the initial states of t_0 and t_1 to the state $t_0 + t_1$ in $S_{\underline{t_0+t_1}}$ and taking all other states to their obvious counterparts.

We take the transformation to be

$$in : \mathcal{P}(S_{t_0} + S_{t_1}) \times \mathcal{P}(\{\bullet\}) \to \mathcal{P}(S_{\underline{t_0+t_1}}),$$

where $in(V_0, V_1) = \{j(s) \mid s \in V_0\} \cup \{\underline{t_0 + t_1} \mid \bullet \in V_1\}$.

Theorem 3 (reduction for sum). Given a closed, pure assertion A, a change of variables σ that is fresh for A, and arbitrary process terms t_0 and t_1 then

$$\models (t_0 + t_1 : A) \leftrightarrow \mathrm{red}^1(t_0 + t_1 : A; \sigma).$$

Proof. The proof is very similar to the proof of correctness for the reduction of prefix (appendix 3). □

3.4. *Relabeling*

For relabeling we take the transformation to be $in : \mathcal{P}(S_t) \to \mathcal{P}(S_{t\{\Xi\}})$, where $in(V) = \{p\{\Xi\} \mid p \in V\}$. The reduction is given in figure 4.

Theorem 4 (reduction for relabeling). Assume A closed and pure, a change of variables σ that is fresh for A, and an arbitrary process term t; then

$$\models (t\{\Xi\} : A) \leftrightarrow (t : \mathrm{red}(t\{\Xi\} : A; \sigma)).$$

Proof. The proof is like that for restriction; see appendix 4. □

3.5. *Restriction*

For restriction, we take the transformation to be $in : \mathcal{P}(S_t) \to \mathcal{P}(S_{t\restriction A})$, where $in(V) = \{p \restriction A \mid p \in V\} \cap S_{t\restriction A}$. The reduction is given in figure 5.

$$
\begin{array}{lcl}
\mathrm{red}(t \upharpoonright \Lambda : X; \sigma) & = & Y \ \text{ where } \sigma(X) = IN(Y) \\
\mathrm{red}(t \upharpoonright \Lambda : \nu X.A; \sigma) & = & \nu Y.\mathrm{red}(t \upharpoonright \Lambda : A; \sigma) \ \text{ where } \sigma(X) = IN(Y) \\
\mathrm{red}(t \upharpoonright \Lambda : \langle\alpha\rangle A; \sigma) & = & \begin{cases} \langle\alpha\rangle \mathrm{red}(t \upharpoonright \Lambda : A; \sigma) & \text{if } \alpha \in \Lambda \\ F & \text{if } \alpha \notin \Lambda \end{cases}
\end{array}
$$

Figure 5. Reduction for restriction.

$$
\begin{array}{lcl}
\mathrm{red}(rec\ P.t : X; \sigma) & = & Y \ \text{ where } \sigma(X) = IN(Y) \\
\mathrm{red}(rec\ P.t : \nu X.A; \sigma) & = & \nu Y.\mathrm{red}(rec\ P.t : A; \sigma) \ \text{ where } \sigma(X) = IN(Y) \\
\mathrm{red}(rec\ P.t : \langle\alpha\rangle A; \sigma) & = & \langle\alpha\rangle A' \vee (\hat{P} \wedge (t : \langle\alpha\rangle A')) \\
 & & \text{where } A' = \mathrm{red}(rec\ P.t : A; \sigma)
\end{array}
$$

Figure 6. Reduction for recursion. The definitions for $\neg$ and $\vee$ are omitted, since they again are similar to the definitions for prefix.

Theorem 5 (reduction for restriction). Assume A closed and pure, a change of variables σ that is fresh for A, and an arbitrary process term t; then

$$\models (t \upharpoonright A : A) \leftrightarrow (t : \mathrm{red}(t \upharpoonright A : A; \sigma)).$$

Proof. See appendix 4. □

3.6. *Recursion*

In order to define the reduction for recursion (see figure 6), we will need to extend our assertion language with an assertion $\hat{P}$ to identify recursion points. The semantics of $\hat{P}$ is simply[3]

$$[\![\hat{P}]\!]_T \phi = \{P\} \cap S_T.$$

It can be verified that the locality and the rooting lemma still hold. All the reductions mentioned in the previous sections should be extended to take care of the assertions $\hat{P}$, and this is easily done—they should all give F. Furthermore, we add a reduction for P, and this is like the one for *nil*, except that it gives T on $\hat{P}$.

For the first time we will need to put in extra correctness assertions in our reductions, which furthermore might contain free assertion variables. These correctness assertions can, however, be closed by a *closure lemma* and then

"pulled out" by a *purifying lemma*, yielding an expression that belongs to the propositional language without any correctness assertions appearing inside other assertions, and hence being applicable for further reductions.

Theorem 6 (purifying lemma). Let A be an assertion with all correctness assertions closed and let t be a process term. Then there exists an expression B over unnested correctness assertions such that $\models (t : A) \leftrightarrow B$.

Proof. See appendix 5. □

Moreover, the proof of the lemma gives an algorithm for computing such a B. The closure lemma can be found in [14].

Take $j : S_{\underline{t}} \to S_{rec\,P.t}$ to be the map that takes $\underline{t}$ to *rec* $P.t$ and all other states s to $s[rec\ P.t/P]$. The transformation for recursion $in : \mathcal{P}(S_{\underline{t}}) \to \mathcal{P}(S_{rec\ P.t})$ is defined to be the direct image of j.

Theorem 7 (reduction for recursion). Given a closed, pure assertion A, a change of variables σ that is fresh for A, and an arbitrary process term t, then

$$\models (rec\ P.t : A) \leftrightarrow (t : \mathrm{red}(rec\ P.t : A; \sigma)).$$

Proof. See appendix 6. □

3.7. Product

A reduction for a product $q \times p$ should be an assertion B over atoms $(q : B_i)$ and $(p : C_j)$ such that

$$\models q \times p : A \text{ iff } \models B.$$

Unfortunately, if we insist on finding such a B without inspecting either p or q, we can get a very complex expression, which in the case of fixed points will even become infinite unless assumptions on the possible sizes of p and q are made (cf. the remarks at the end of [14]). In [14] it is shown how a very reasonably sized B can be found, when the assertion language is restricted rather severely, excluding disjunctions, negations, minimum fixed points, and general box formulas, but still having maximum fixed points, diamond formulas, a strong version of box formulas, and conjunctions.

Here we present another approach. We give a reduction when p is a process term without restrictions and relabelings, i.e., we find a B (depending on p) such that

$$\models q \times p : A \text{ iff } \models q : B.$$

$$
\begin{array}{lcl}
\neg A/p & = & \neg(A/p) \\
A_0 \vee A_1/p & = & (A_0/p) \vee (A_1/p) \\
X/p & = & X_p \\
\nu X.A/p & = & C_k(\nu(X_{p_1}, \ldots, X_{p_n}).(A/p_1, \ldots, A/p_n)) \\
 & & \text{where } \{p_i\}_i \text{ denotes the set of reachable} \\
 & & \text{states from } p \text{ with } p = p_k. \\
A/q \times r & = & (A/r)/q \\
 & & \text{with the actions in the modalities of } A \text{ reassociated} \\
\langle\alpha\times\beta\rangle A/nil & = & \begin{cases} \langle\alpha\rangle(A/nil) & \text{if } \beta = * \\ F & \text{if } \beta \neq * \end{cases} \\
\langle\alpha\times\beta\rangle A/\gamma q & = & \begin{cases} \langle\alpha\rangle(A/\gamma q) & \text{if } \beta = * \\ \langle\alpha\rangle(A/q) & \text{if } \beta = \gamma \\ F & \text{otherwise} \end{cases} \\
\langle\alpha\times\beta\rangle A/q + r & = & (\langle\alpha\times\beta\rangle A/q) \vee (\langle\alpha\times\beta\rangle A/r) \\
\langle\alpha\times\beta\rangle A/rec\ P.t & = & \langle\alpha\times\beta\rangle A/t[rec\ P.t/P]
\end{array}
$$

Figure 7. Reduction for product.[4] $C_k(\nu\underline{X}.\underline{A})$ denotes the kth component of the n-ary fixed point $\nu\underline{X}.\underline{A}$, closed by repeated application of Bekić's theorem.

Let $R_p = \{p_1, \ldots, p_n\}$ be the finite set of reachable states of p in some fixed enumeration. We define the map $in : \underbrace{\mathcal{P}(R_q) \times \ldots \times \mathcal{P}(R_q)}_{n} \to \mathcal{P}(R_{q\times p})$ as

$$in(U_{p_1}, \ldots, U_{p_n}) = (U_{p_1} \times p_1) \cup \ldots \cup (U_{p_n} \times p_n),$$

where $U \times p = \{u \times p \mid u \in U\}$. As usual, we have a change of variables σ with $\sigma(X) = IN(X_{p_1}, \ldots, X_{p_n})$. As a notational convenience, we write A/p for $\mathrm{red}(q \times p : A; \sigma)$ omitting the σ, which is always assumed to map an X into $X_{p_1}, \ldots, X_{p_n}$. The reduction is shown in figure 7.

Theorem 8 (reduction for product). Assume given a pure and closed assertion A, a change of variables σ, and a term p with no restrictions and relabelings. We then have for an arbitrary term q:

$$\models (q \times p : A) \leftrightarrow (q : \mathrm{red}(q \times p : A; \sigma)).$$

Proof. See appendix 7. □

The case of the maximal fixed point is established by repeated application of Bekić's theorem, and the resulting assertion might become rather complex,

since in the worst case a fixed point will appear for each reachable state of p, and on top of this, Bekić's theorem might increase the size of the assertion considerably. We are currently investigating methods to control the potential blowup in general. We present in the next section an example that indicates that in practice this need not be the case.

4. Examples

It is an important property of all our reductions (except product) that they only depend on the top-most operator of the process term. Hence, we can leave part of a process unspecified and still apply the reductions. Technically this can be done by adding *process variables* to our language of processes. Given an assertion and a process with variables, we can then compute a propositional expression with correctness assertions over the variables, expressing what relationship there should be between them in order to make the process satisfy the assertion. In this way the reductions compute what corresponds to weakest preconditions in Hoare logic.

As pointed out in the previous section, the reductions for product have the potential of becoming rather complex. In this section we show by a small example that, in practice, the reductions need not turn out to be too complex.

First we define a binary parallel operator $\|_{K,L}$ which allows its left and right components to independently perform the actions indicated by the sets K and L, except that they are required to synchronize on common actions of K and L. The precise definition is

$$p\|_{K,L}q \stackrel{\text{def}}{=} (p \times q) \upharpoonright \Lambda\{\Xi\},$$

where $\Lambda = \{a \times a \mid a \in K \cap L\} \cup \{a \times * \mid a \in K \backslash L\} \cup \{* \times a \mid a \in L \backslash K\}$ and

$$\begin{aligned} \Xi(a \times a) &= a, \text{ for all } a \in K \cap L \\ \Xi(a \times *) &= a, \text{ for all } a \in K \backslash L \\ \Xi(* \times a) &= a, \text{ for all } a \in L \backslash K \\ \Xi(\alpha) &\quad \text{undefined otherwise.} \end{aligned}$$

Now assume that we want to construct a small system consisting of a coffee vending machine and a researcher. The coffee machine should be able to accept money and then supply a cup of coffee. The researcher should be able to pay out money, drink coffee, and publish papers. Suppose we know how the researcher behaves, specified by a process term r, but would like to find out what kind of coffee machine x to put into the system, such that eventually the researcher has no other choice than to publish a paper.

In general a property of the form "eventually only the action α can happen" can be expressed by the assertion

$$\mu X.\langle - \rangle T \wedge [-\alpha]X$$

where

$$\langle - \rangle A = \langle Act \rangle A \quad [-K]A = [Act \backslash K]A.$$

Our problem can now be restated.

> Assume the actions to be p for publish, c for taking/giving coffee, and m for taking/giving money, and define $K = \{m, c\}, L = \{m, c, p\}$. Which values of x make the following correctness assertion valid?
>
> $$x \|_{K,L} r : \mu X.\langle - \rangle T \wedge [-p]X$$

Suppose the researcher r behaves as *rec* $P.m.c.(m.c.P + p.P)$. Then expanding the definition of $\|_{K,L}$ and applying the reduction for restriction and relabeling, we get the equivalent correctness assertion

$$x \times r : \mu X.\langle m \times m, c \times c, * \times p \rangle T \wedge [m \times m, c \times c]X,$$

and then, by applying the reduction for product, the equivalent

$$x : \mu X.\langle m \rangle T \wedge [m](\langle c \rangle T \wedge [c][m](\langle c \rangle T \wedge [c]X)). \tag{2}$$

One can now use equation (2) to verify different proposals for coffee machines, without redoing the first two steps. This might be done by our method, or for closed terms by other model-checking algorithms.

An interesting point to note about the assertion in equation (2) is that, although the researcher r had *four* reachable states, and then potentially four fixed points could appear, only *one* fixed point appears in the resulting assertion.

Returning to the example, we can verify that a successful choice of x is $m.c.nil$, i.e., a coffee machine that accepts money and give coffee once, and then breaks down, whereas *rec* $P.m.c.P$ is an unsuccessful choice. Reading the assertion in equation (2) carefully, we can express the requirement to the machine as "after having offered a finite and odd number of m's followed by c's no m should be offered."

Changing the behavior of the researcher slightly and taking $r =$ *rec* $P.m.c.P + m.c.p.P$ and performing the reductions for restriction, relabeling, and product, we arrive at the correctness assertion $x : F$, i.e., there are no coffee machines that will make the system fulfill the requirement.

Acknowledgments

This work is supported by the ESPRIT Basic Research Actions CEDISYS and CLICS, and for the first author also by the Danish Natural Science Research Council. Thanks are due to an anonymous referee for useful comments.

Appendix 1. Proof of rooting lemma

The proof is by structural induction on A.

$A \equiv X$. By the definition of $[\![X]\!]_T\phi$, we immediately get

$$r([\![X]\!]_T\phi) = r(\phi(X)) = r \circ \phi(X) = [\![X]\!]_{\underline{T}}(r \circ \phi).$$

$A \equiv \nu X.B$. By definition, we have

$$r([\![\nu X.B]\!]_T\phi) = r(\nu\theta), \tag{A1}$$

where $\theta : \mathcal{P}(S_T) \to \mathcal{P}(S_T)$ is defined by $\theta(U) = [\![B]\!]_T\phi[U/X]$. Taking $\psi(V) = [\![B]\!]_{\underline{T}}(r \circ \phi)[V/X]$, we obtain

$$\begin{aligned} r \circ \theta(U) &= r([\![B]\!]_T\phi[U/X]) \\ &= [\![B]\!]_{\underline{T}}(r \circ \phi)[r(U)/X] \\ &\qquad \text{by the induction hypothesis} \\ &= \psi(r(U)) = \psi \circ r(U) \end{aligned}$$

Furthermore, r is easily seen to be $\top$-strict and ω-anticontinuous, and since we assume S_T to be countable, the reduction lemma yields

$$r(\nu\theta) = \nu\psi,$$

which by expanding ψ and equation (A1) gives the result

$$\begin{aligned} r([\![\nu X.B]\!]_T\phi) &= \nu V \subseteq S_{\underline{T}}.[\![B]\!]_{\underline{T}}(r \circ \phi)[V/X] \\ &= [\![\nu X.B]\!]_{\underline{T}}(r \circ \phi). \end{aligned}$$

$A \equiv \langle\alpha\rangle B, \alpha \neq *$. We proceed by rewriting the left-hand side:

$$\begin{aligned} r([\![\langle\alpha\rangle B]\!]_T\phi) &= r(\{s \in S_T \mid \exists s' \in S_T.s \xrightarrow{\alpha} s' \ \&\ s' \in [\![B]\!]_T\phi\}) \\ &\qquad \text{by definition} \\ &= r(\{s \in S_T \mid \exists s' \in S_T.s \xrightarrow{\alpha} s' \ \&\ s' \in r([\![B]\!]_T\phi)\}) \\ &\qquad \text{by definition of rooting} \\ &= r(\{s \in S_T \mid \exists s' \in S_T \cup \{\underline{i}\}.s \xrightarrow{\alpha} s' \ \&\ s' \in r([\![B]\!]_T\phi)\}) \\ &\qquad \text{since no transitions enter } \underline{i} \\ &= r(\{s \in S_T \mid \exists s' \in S_T \cup \{\underline{i}\}.s \xrightarrow{\alpha} s' \ \&\ s' \in [\![B]\!]_{\underline{T}}(r \circ \phi)\}) \\ &\qquad \text{by the induction hypothesis} \\ &= \{i, \underline{i} \mid \exists s' \in S_{\underline{T}}.i \xrightarrow{\alpha} s' \ \&\ s' \in [\![B]\!]_{\underline{T}}(r \circ \phi)\} \\ &\quad \cup \{s \in S_{\underline{T}} \backslash \{i, \underline{i}\} \mid \exists s' \in S_{\underline{T}}.s \xrightarrow{\alpha} s' \ \&\ s' \in [\![B]\!]_{\underline{T}}(r \circ \phi)\} \\ &\qquad \text{by applying } r \\ &= [\![\langle\alpha\rangle B]\!]_{\underline{T}}(r \circ \phi). \end{aligned}$$

$A \equiv A_0 \vee A_1$ and $A \equiv \neg B$. The proof is immediate, since r distributes over disjunction and negation. □

Appendix 2. Proof of reduction lemma

We prove the reduction lemma as a corollary of a more general result, for which we need the notion of *height* of a partial order.

Definition 2. Define the *height* of a partial order $(D, \leq)$ to be the smallest cardinal κ such that for any $T \subseteq D$, where $(T, \leq \cap T \times T)$ is totally ordered, $|T| \leq \kappa$. Say a partial order has *countable height* if its height is countable.

Note that if X is a countable set, then the partial orders $(\mathcal{P}(X), \subseteq)$ and $(\mathcal{P}(X), \supseteq)$ have countable height.

We carry out the proof for minimum fixed points, and then derive the result for maximum fixed points by duality from which the reduction lemma directly follows. We will use $\perp$ and $\top$ as names for the bottom and top elements of lattices, respectively.

Lemma 5. Let D, E be complete lattices of countable height. Let $in : D \to E$ be an ω-continuous function such that $in(\perp_D) = \perp_E$. Suppose $\varphi : E \to E$ and $\theta; D \to D$ are monotonic functions such that

$$in \circ \theta = \varphi \circ in.$$

Then

$$in(\mu\theta) = \mu\varphi.$$

Proof. The following facts are well known (see, e.g., [15]): For a monotonic function $\theta : D \to D$,

1. $\mu\theta = \bigvee_{\alpha \in On} \theta^\alpha(\perp_D)$,
   where

   $\theta^0(x) =_{def} x$,

   $\theta^{\alpha+1}(x) =_{def} \theta(\theta^\alpha(x))$, and

   $\theta^\lambda(x) =_{def} \bigvee_{\alpha<\lambda} \theta^\alpha(x)$ for λ a limit-ordinal,

   are such that $\alpha \leq \alpha' \Rightarrow \theta^\alpha(\perp_D) \leq \theta^{\alpha'}(\perp_D)$.

2. In addition, there is a least ordinal β (the closure ordinal) such that $\theta^\beta(\perp_D) = \theta^{\beta+1}(\perp_D)$. Then $\mu\theta = \theta^\beta(\perp_D)$.

Further,

3. If D is of height ω, when β is a countable ordinal: the function $\alpha \longmapsto \theta^\alpha(\perp_D)$ for $\alpha \in \beta$ is 1-1 and has range a total order in D; hence, because D has height ω, the ordinal β is countable. It follows that when D has height ω, then

$$\mu\theta = \bigvee_{\alpha \in Con} \theta^\alpha(\perp_D), \tag{A2}$$

where Con is the set of countable ordinals.

Now, we proceed to the main proof. Under the assumptions stated in the lemma, we see

$$\begin{aligned} in(\mu\theta) &= in\left(\bigvee_{\alpha \in Con} \theta^\alpha(\perp_D)\right) \\ &= in(\theta^\beta(\perp_D)), \\ &\qquad \text{where } \beta \text{ is the closure ordinal as in 2 above} \\ &= \bigvee_{\alpha \in Con} in(\theta^\alpha(\perp_D)) \text{ by } \omega\text{-continuity of } in. \end{aligned} \tag{A3}$$

By ordinal induction, we show

$$in(\theta^\alpha(\perp_D)) = \varphi^\alpha(in(\perp_D)) \tag{A4}$$

for all $\alpha \in Con$:

When $\alpha = 0$, then $in(\theta^\alpha(\perp_D)) = in(\perp_D)) = \varphi^\alpha(in(\perp_D))$.

For a successor ordinal,

$$\begin{aligned} in(\theta^{\alpha+1}(\perp_D)) &= in(\theta(\theta^\alpha(\perp_D))) && \text{by definition} \\ &= \varphi(in(\theta^\alpha(\perp_D))) && \text{as } in \circ \theta = \varphi \circ in \\ &= \varphi(\varphi^\alpha(in(\perp_D))) && \text{by induction} \\ &= \varphi^{\alpha+1}(in(\perp_D)). \end{aligned}$$

Assume λ is a countable limit ordinal. Then λ is cofinal with ω in the sense that there is an ω-sequence of elements of λ

$$\beta_0, \beta_1, \ldots, \beta_n, \ldots$$

such that for all $\alpha \in \lambda$ there is some $n \in \omega$ such that $\alpha \leq \beta_n$: with respect to $\alpha_0, \alpha_1, \ldots, \alpha_n, \ldots$, a countable enumeration of elements of λ, take $\beta_0 = \alpha_0$ and inductively take β_{n+1} to be the maximum of β_n and α_{n+1}.

Now we argue:

$$
\begin{aligned}
in(\theta^{\lambda}(\bot_D)) &= in\,(\bigvee_{\alpha<\lambda}\theta^{\alpha}(\bot_D)) \\
&= in\,(\bigvee_{n\in\omega}\theta^{\beta_n}(\bot_D)) && \text{by cofinality} \\
&= \bigvee_{n\in\omega} in(\theta^{\beta_n}(\bot_D)) && \text{by } \omega\text{-continuity of } in \\
&= \bigvee_{n\in\omega}\varphi^{\beta_n}(in\bot_D)) && \text{by induction.} \\
&= \bigvee_{\alpha<\lambda}\varphi^{\alpha}(in(\bot_D)) && \text{by cofinality} \\
&= \varphi^{\lambda}(in(\bot_D)).
\end{aligned}
$$

This completes the inductive proof of equation (A4).

Recalling $in(\bot_D) = \bot_E$ we conclude:

$$
\begin{aligned}
in(\mu\theta) &= \bigvee_{\alpha\in Con} in(\theta^{\alpha}(\bot_D)) && \text{by (A3)} \\
&= \bigvee_{\alpha\in Con}\varphi^{\alpha}(in(\bot_D)) && \text{by (A4)} \\
&= \bigvee_{\alpha\in Con}\varphi^{\alpha}(\bot_E) \\
&= \mu\varphi && \text{by (A2).}
\end{aligned}
$$

□

By duality we obtain the following result for maximum fixed points in which the assumptions of ω-continuity and $\bot$-strictness are replaced by the dual conditions of ω-anticontinuity and $\top$-strictness.

Corollary. Let D, E be complete lattices of countable height. Let $in : D \to E$ be an ω-anticontinuous function such that $in(\top_D) = \top_E$. Suppose $\varphi : E \to E$ and $\theta : D \to D$ are monotonic functions such that

$$in \circ \theta = \varphi \circ in.$$

Then

$$in(\nu\theta) = \nu\theta.$$

We remark that the ω-continuity of in is necessary, as the following example shows.

Example. Let E consist of $\bot < \top$ and D be the ordinal $\omega + 1$ ordered by the usual ordering on ordinals. Let $in : D \to E$ be the monotonic (but not continuous) function such that

$$
\begin{aligned}
&in(n) = \bot \text{ for } n \in \omega, \\
&in(\omega) = \top.
\end{aligned}
$$

Take $\varphi : E \to E$ to be the identity on E, and $\theta : D \to D$ to act so that

$$\theta(n) = n + 1, \text{ for } n \in \omega,$$
$$\theta(\omega) = \omega.$$

Then $\mu\varphi = \bot$ and $\mu\theta = \omega$. Hence in this case where *in* is monotonic and not continuous, we have $\mu\varphi = \bot$ and $in(\mu\theta) = \top$, so $\mu\varphi \neq in(\mu\theta)$. (Monotonicity of *in* guarantees $\mu\varphi \leq in(\mu\theta)$.) □

If D, E are powersets of countable sets, then they are complete lattices of height ω and so meet the conditions required by the reduction lemma and its dual, yielding the special case, lemma 4, used in this article.

Appendix 3. Proof of reduction for prefix

We prove by structural induction on A that for a change of variables σ that is fresh for A, we have for all environments ϕ:

$$[\![A[\sigma]]\!]_{\underline{at}}\phi = in([\![\text{red}^0(at : A; \sigma)]\!]_t\phi, [\![\text{red}^1(at : A; \sigma)]\!]_\bullet\phi). \tag{A5}$$

The result then follows from the discussion preceding theorem 1.

$A \equiv X$. Assuming that $\sigma(X) = IN(X_0, X_1)$, we get

$$\begin{aligned}
[\![X[\sigma]]\!]_{\underline{at}}\phi &= in(\phi(X_0), \phi(X_1)) \\
&= in([\![X_0]\!]_t\phi, [\![X_1]\!]_\bullet\phi) \\
&= in([\![\text{red}^0(at : X; \sigma)]\!]_t\phi, [\![\text{red}^1(at : X; \sigma)]\!]_\bullet\phi).
\end{aligned}$$

$A \equiv \nu X.B$. By definition, we have

$$[\![(\nu X.B)[\sigma]]\!]_{\underline{at}}\phi = \nu\psi,$$

where ψ is defined by

$$\psi(U) = [\![B[\sigma\backslash X]]\!]_{\underline{at}}\phi[U/X].$$

Taking as abbreviations $B^0 = red^0(at : B; \sigma)$ and $B^1 = red^1(at : B; \sigma)$ and defining

$$\theta(V_0, V_1) = ([\![B^0]\!]_t\phi[V_0/X_0, V_1/X_1], [\![B^1]\!]_\bullet\phi[V_0/X_0, V_1/X_1]),$$

we can show that θ and ψ are related as required by the reduction lemma:

$$\begin{aligned}
in \circ \theta(V_0, V_1) &= in([\![B^0]\!]_t\phi[V_0/X_0, V_1/X_1], [\![B^1]\!]_\bullet\phi[V_0/X_0, V_1/X_1]) \\
&= [\![B[\sigma]]\!]_{\underline{at}}\phi[V_0/X_0, V_1/X_1] \\
&\qquad \text{by the induction hypothesis} \\
&= [\![B[\sigma\backslash X]]\!]_{\underline{at}}\phi[in(V_0, V_1)/X] \\
&\qquad \text{as } \sigma(X) = IN(X_0, X_1) \text{ and } \sigma \text{ is fresh for } \nu X.B \\
&= \psi \circ in(V_0, V_1) \\
&\qquad \text{by definition of } \psi
\end{aligned}$$

It is easy to see that in is $\top$-strict and ω-anticontinuous. Hence, the reduction lemma yields

$$in(\nu\theta) = \nu\psi.$$

Writing out $\nu\theta$ in full detail, we can proceed by applying Bekić's theorem:

$$\begin{aligned}
\nu\theta &= \nu(V_0, V_1).(\llbracket B^0 \rrbracket_t \phi[V_0/X_0, V_1/X_1], \llbracket B^1 \rrbracket_\bullet \phi[V_0/X_0, V_1/X_1]) \\
&= (\nu V_0.\llbracket B^0 \rrbracket_t \phi[V_0/X_0], \nu V_1.\llbracket B^1 \rrbracket_\bullet \phi[(\nu V_0.\llbracket B^0 \rrbracket_t \phi[V_0/X_0])/X_0, V_1/X_1]) \\
&\quad \text{by Bekić's theorem and the observation that } X_1 \text{ is not free in } B^0 \\
&= (\llbracket \nu X_0.B^0 \rrbracket_t \phi, \nu V_1.\llbracket B^1 \rrbracket_\bullet \phi[\llbracket \nu X_0.B^0 \rrbracket_t \phi/X_0, V_1/X_1]) \\
&\quad \text{by definition} \\
&= (\llbracket \nu X_0.B^0 \rrbracket_t \phi, \nu V_1.\llbracket B^1[\nu X_0.B^0/X_0] \rrbracket_\bullet \phi[V_1/X_1]) \\
&\quad \text{by the substitution lemma} \\
&= (\llbracket \nu X_0.B^0 \rrbracket_t \phi, \llbracket B^1[\nu X_0.B^0/X_0] \rrbracket_\bullet \phi[\{\bullet\}/X_1]) \\
&\quad \text{since } \mathcal{P}(\{\bullet\}) \text{ is just a two-point lattice with top element } \{\bullet\} \\
&= (\llbracket \nu X_0.B^0 \rrbracket_t \phi, \llbracket B^1[\nu X_0.B^0/X_0][T/X_1] \rrbracket_\bullet \phi) \\
&\quad \text{by the substitution lemma} \\
&= (\llbracket \mathrm{red}^0(at : \nu X.B; \sigma) \rrbracket_t \phi, \llbracket \mathrm{red}^1(at : \nu X.B; \sigma) \rrbracket_\bullet \phi).
\end{aligned}$$

We have established that

$$\llbracket (\nu X.B)[\sigma] \rrbracket_{\underline{at}} \phi = in(\llbracket \mathrm{red}^0(at : \nu X.B; \sigma) \rrbracket_t \phi, \llbracket \mathrm{red}^1(at : \nu X.B; \sigma) \rrbracket_\bullet \phi)$$

as required.

$A \equiv \langle\alpha\rangle B, \alpha \neq *$. We rewrite from the definition:

$$\begin{aligned}
\llbracket \langle\alpha\rangle B[\sigma] \rrbracket_{\underline{at}} \phi &= \{s \in S_{\underline{at}} \mid \exists s' \in S_{\underline{at}}.s \xrightarrow{\alpha} s' \ \&\ s' \in \llbracket B[\sigma] \rrbracket_{\underline{at}} \phi\} \\
&= \{s \in S_{\underline{at}} \mid \exists s' \in S_{\underline{at}}.s \xrightarrow{\alpha} s' \ \&\ s' \in in(\llbracket B^0 \rrbracket_t \phi, \llbracket B^1 \rrbracket_\bullet \phi)\} \\
&\quad \text{by the induction hypothesis where} \\
&\quad B^0 \text{ abbreviates } \mathrm{red}^0(at : B; \sigma) \\
&\quad \text{and } B^1 \text{ abbreviates } \mathrm{red}^1(at : B; \sigma) \\
&= \{s \in S_{\underline{at}} \mid \exists s' \in S_{\underline{at}} \backslash \{\underline{at}\}.s \xrightarrow{\alpha} s' \\
&\quad \ \&\ s' \in in(\llbracket B^0 \rrbracket_t \phi, \llbracket B^1 \rrbracket_\bullet \phi)\} \\
&\quad \text{since no transitions enter } \underline{at} \\
&= \{s \in S_{\underline{at}} \mid \exists s' \in S_{\underline{at}} \backslash \{\underline{at}\}.s \xrightarrow{\alpha} s' \ \&\ s' \in \llbracket B^0 \rrbracket_t \phi\} \\
&\quad \text{by definition of } in \\
&= \{\underline{at} \mid \exists s' \in S_{\underline{at}} \backslash \{\underline{at}\}.\underline{at} \xrightarrow{\alpha} s' \ \&\ s' \in \llbracket B^0 \rrbracket_t \phi\} \\
&\quad \cup \{s \in S_{\underline{at}} \backslash \{\underline{at}\} \mid \exists s' \in S_{\underline{at}} \backslash \{\underline{at}\}.s \xrightarrow{\alpha} s' \ \&\ s' \in \llbracket B^0 \rrbracket_t \phi\}
\end{aligned}$$

by simple splitting

$$= \{\underline{at} | \alpha = a \ \&\ t \in [\![B^0]\!]_t\phi\}$$

since the only transition from $\underline{at}$ is $\underline{at} \xrightarrow{a} t$ and by observing $S_{\underline{at}} \backslash \{\underline{at}\} = S_t$

$$= \begin{cases} [\![\langle\alpha\rangle B^0]\!]_t\phi & \text{if } \alpha \neq a \\ \{\underline{at} | t \in [\![B^0]\!]_t\phi\} \cup [\![\langle\alpha\rangle B^0]\!]_t\phi & \text{if } \alpha = a \end{cases}$$

from the definition of $[\![\langle\alpha\rangle B^0]\!]_t\phi$

$$= in([\![\mathrm{red}^0(at : \langle\alpha\rangle B; \sigma)]\!]_t\phi, [\![\mathrm{red}^1(at : \langle\alpha\rangle B; \sigma)]\!]_*\phi)$$

by definition of red^0, red^1, and *in*.

$A \equiv A_0 \vee A_1$ and $A \equiv \neg B$. The proof is straightforward. □

Appendix 4. Proof of reduction for restriction

We show by structural induction on A that for a change of variables σ that is fresh for A, we have for all ϕ:

$$[\![A[\sigma]]\!]_{t \restriction \Lambda}\phi = in([\![\mathrm{red}(t \restriction \Lambda : A; \sigma)]\!]_t\phi). \tag{A5}$$

From equation (A5) and the definition of *in*, it follows that

$$t \restriction \Lambda \in [\![A[\sigma]]\!]_{t \restriction \Lambda}\phi \text{ iff } t \in [\![\mathrm{red}(t \restriction \Lambda : A; \sigma)]\!]_t\phi.$$

Hence, by the locality lemma,

$$\models (t \restriction \Lambda : A) \leftrightarrow (t : \mathrm{red}(t : \mathrm{red}(t \restriction \Lambda : A; \sigma)),$$

as required.

$A \equiv X$. Assuming that $\sigma(X) = IN(Y)$, we get:

$$\begin{aligned} [\![X[\sigma]]\!]_{t \restriction \Lambda}\phi &= in(\phi(Y)) \\ &= in([\![Y]\!]_t\phi) \\ &= in([\![\mathrm{red}(t \restriction \Lambda : X; \sigma)]\!]_t\phi). \end{aligned}$$

$A \equiv \nu X.B$. By definition, we have

$$[\![(\nu X.B)[\sigma]]\!]_{t \restriction \Lambda}\phi = \nu\psi,$$

where $\psi : \mathcal{P}(S_{t \restriction \Lambda}) \to \mathcal{P}(S_{t \restriction \Lambda})$ is defined by

$$\psi(U) = [\![B[\sigma \backslash X]]\!]_{t \restriction \Lambda}\phi[U/X].$$

Defining $\theta : \mathcal{P}(S_t) \to \mathcal{P}(S_t)$ by

$$\theta(V) = [\![\mathrm{red}(t \restriction \Lambda : B; \sigma)]\!]_t\phi[V/Y],$$

we show that θ and ψ are related as required by the reduction lemma:

$$\begin{aligned} in \circ \theta(V) &= in([\![\mathrm{red}(t \restriction \Lambda : B; \sigma)]\!]_t \phi[V/Y]) \\ &= [\![B[\sigma]]\!]_t[V/Y] \\ &\quad \text{by the induction hypothesis} \\ &= [\![B[\sigma \backslash X]]\!]_t \phi[in(V)/X] \\ &\quad \text{since } \sigma(X) = IN(Y) \text{ and } \sigma \text{ is fresh for } \nu X.B \\ &= \psi \circ in(V) \\ &\quad \text{by definition of } \psi. \end{aligned}$$

It is easy to see that *in* is $\top$-strict and ω-anticontinuous. Hence, the reduction lemma applies, yielding

$$\nu\psi = in(\nu\theta).$$

Therefore,

$$\begin{aligned} [\![(\nu X.B)[\sigma]]\!]_{t \restriction \Lambda} \phi &= in(\nu V \subseteq S_{t \restriction \Lambda}. [\![\mathrm{red}(t \restriction \Lambda : B; \sigma)]\!]_t \phi[V/Y]) \\ &= in([\![\nu Y.\mathrm{red}(t \restriction \Lambda : B; \sigma)]\!]_t \phi) \\ &\quad \text{by definition of the } \nu\text{-operator} \\ &= in([\![\mathrm{red}(t \restriction \Lambda : \nu Y.B; \sigma)]\!]_t \phi) \\ &\quad \text{by definition of } \mathrm{red}(t \restriction \Lambda : \nu Y.B; \sigma). \end{aligned}$$

$A \equiv \langle\alpha\rangle B, \alpha \neq *$. We rewrite the left-hand side:

$$\begin{aligned} &[\![\langle\alpha\rangle B[\sigma]]\!]_{t \restriction \Lambda} \phi \\ &= \left\{ s \in S_{t \restriction \Lambda} \mid \exists s' \in S_{t \restriction \Lambda}. s \xrightarrow{\alpha} s' \ \&\ s' \in [\![B[\sigma]]\!]_{t \restriction \Lambda} \phi \right\} \\ &\quad \text{by definition} \\ &= \left\{ s \in S_{t \restriction \Lambda} \mid \exists s' \in S_{t \restriction \Lambda}. s \xrightarrow{\alpha} s' \ \&\ s' \in in([\![\mathrm{red}(t \restriction \Lambda : B; \sigma)]\!]_t \phi) \right\} \\ &\quad \text{by the induction hypothesis (equation (A5))} \\ &= \begin{cases} \{ s \restriction \Lambda \mid s \in S_t \ \&\ \exists s' \in S_t. s \xrightarrow{\alpha} s' \\ \qquad \&\ s' \in [\![\mathrm{red}(t \restriction \Lambda : B; \sigma)]\!]_t \phi \} \cap S_{t \restriction \Lambda} & \text{if } \alpha \in \Lambda \\ \emptyset & \text{if } \alpha \notin \Lambda \end{cases} \\ &\quad \text{by definition of } in \text{ and the restriction operator} \\ &= \begin{cases} in([\![\langle\alpha\rangle \mathrm{red}(t \restriction \Lambda : B; \sigma)]\!]_t \phi) & \text{if } \alpha \in \Lambda \\ in(\emptyset) & \text{if } \alpha \notin \Lambda \end{cases} \\ &\quad \text{by definition of } in \text{ and } \langle\alpha\rangle \\ &= in([\![\mathrm{red}(t \restriction \Lambda : \langle\alpha\rangle B; \sigma)]\!]_t \phi) \\ &\quad \text{by definition of } \mathrm{red}(t \restriction \Lambda : \langle\alpha\rangle B; \sigma). \end{aligned}$$

$A \equiv A_0 \vee A_1$ and $A \equiv \neg B$. The proof is straightforward. □

Appendix 5. Proof of purifying lemma

For an assertion A, let $s(A)$ denote the number of correctness assertions appearing in A. We show by mathematical induction that for all n, the theorem holds for all assertions A with $s(A) = n$.

For $n = 0$, the proof is trivial: take $B \equiv (t : A)$. For $n > 0$, assume A is a closed assertion in which all correctness assertions are closed. Pick a correctness assertion (e.g., the leftmost) in A, $(t' : A')$, say, (writing $A[(t' : A')]$ to identify the occurrence). Define

$$C \equiv ((t' : A') \wedge A[T]) \vee (\neg(t' : A') \wedge A[F]),$$

where $A[T]$ denotes the resulting of replacing T for $(t' : A')$ in A, and similarly for $A[F]$. Obviously, $\models (t : A) \leftrightarrow (t : C)$. Now, since $s(A[T]) < n$ and $s(A[F]) < n$, we have by the induction hypothesis that there exist B_0 and B_1 with no nested correctness assertions, such that

$$\models (t : A[T]) \leftrightarrow B_0 \text{ and } \models (t : A[F]) \leftrightarrow B_1.$$

Since

$$\models (t : C) \leftrightarrow ((t' : A') \wedge (t : A[T])) \vee (\neg(t' : A') \wedge (t : A[F])),$$

we get

$$\models (t : C) \leftrightarrow ((t' : A') \wedge B_0) \vee (\neg(t' : A') \wedge B_1),$$

which proves the result by taking $B \equiv ((t' : A') \wedge B_0) \vee (\neg(t' : A') \wedge B_1)$. □

Appendix 6. Proof of reduction for recursion

In order to show the correctness of the reduction for recursion, we will need a small lemma that describes a useful relationship between transitions in $\underline{t}$ and *rec P.t.*

Lemma 6. Let j be the function described in the main text. Then for all $s, s' \in S_{\underline{t}}$ and $\alpha \neq *$, we have

$$j(s) \xrightarrow{\alpha} j(s')$$

if and only if

$$\exists s'' \in S_{\underline{t}}. j(s'') = j(s') \;\&\; ((s = P \;\&\; t \xrightarrow{\alpha} s'') \text{ or } (s \neq P \;\&\; s \xrightarrow{\alpha} s'')).$$

Proof. Suppose $s = P$. Then $j(s) = rec\ P.t$, and

$$rec\ P.t \xrightarrow{\alpha} j(s') \quad \text{iff} \quad t[rec\ P.t/P] \xrightarrow{\alpha} j(s')$$

since only the "unfolding rule" applies when $\alpha \neq *$

$$\text{iff} \quad \exists s'' \in S_{\underline{t}}.t \xrightarrow{\alpha} s'' \ \&\ s''[rec\ P.t/P] = j(s')$$

since P is strongly guarded

$$\text{iff} \quad \exists s'' \in S_{\underline{t}}.t \xrightarrow{\alpha} s'' \ \&\ j(s'') = j(s')$$

by definition of j.

Now suppose $s \neq P$. We first consider the case where $j(s) \neq rec\ P.t$, i.e., $s \notin \{\underline{t}, rec\ P.t\}$. Then, since P is strongly guarded, the first transition from $j(s)$ is independent of whether $rec\ P.t$ is substituted for P or not:

$$j(s) \xrightarrow{\alpha} j(s') \text{ iff } \exists s'' \in S_{\underline{t}}.s \xrightarrow{\alpha} s'' \ \&\ j(s'') = j(s').$$

When $s = \underline{t}$ we get by the same arguments as in the case of $s = P$, that

$$j(s) \xrightarrow{\alpha} j(s') \text{ iff } \exists s'' \in S_{\underline{t}}.t \xrightarrow{\alpha} s'' \ \&\ j(s'') = j(s'),$$

which by definition of rooting is equivalent to

$$\exists s'' \in S_{\underline{t}}.s \xrightarrow{\alpha} s'' \ \&\ j(s'') = j(s').$$

For $s = rec\ P.t$ the result is trivial since $j(s) = s$. □

In the inductive proof of correctness it turns out that we will need a stronger induction hypothesis than for the other reductions. We will introduce a notion of "balanced subset," in the sense that if a state $s \in S_{\underline{t}}$ belongs to the subset, then every other state, which under j maps to the same state in $s_{rec\ P.t}$ belongs to the subset. Formally, a subset $U \subseteq S_{\underline{t}}$ is said to be *balanced* if $j^{-1} \circ in(U) = U$. Note that if j is injective, all subsets are trivially balanced. An environment ϕ is said to be balanced if $\phi(X)$ is balanced for all variables X. It is easily seen that $D = \{U \subseteq S_{\underline{t}} | U \text{ is balanced}\}$ is a complete sublattice of $\mathcal{P}(S_{\underline{t}})$.

We are now able to prove theorem 7, (reduction for recursion).

Proof. By structural induction on A, we show that $P(A)$ holds for all A, where P is defined by:

$$P(A) \Leftrightarrow_{def} \text{ for all balanced } \phi.$$

$$[\![A[\sigma]]\!]_{rec\ P.t}\phi = in([\![\text{red}(rec\ P.t : A; \sigma)]\!]_{\underline{t}}\phi)$$

$$\text{and } [\![\text{red}(rec\ P.t : A; \sigma)]\!]_{\underline{t}}\phi \in D \qquad \text{(A6)}$$

From this it follows that

$$\models (rec\ P.t : A) \leftrightarrow (t : \mathrm{red}(rec\ P.t : A; \sigma))$$

for all closed, pure A.
$A \equiv X$. By definition, we have

$$[\![X[\sigma]]\!]_{rec\ P.t}\phi = in(\phi(Y)) = in([\![Y]\!]_{\underline{t}}\phi),$$

assuming that $\sigma(X) = IN(Y)$. From the assumption that ϕ is balanced, we immediately get $[\![Y]\!]_{\underline{t}}\phi \in D$.
$A \equiv \nu X.B$. By definition, we have

$$[\![(\nu X.B)[\sigma]]\!]_{rec\ P.t}\phi = \nu\psi,$$

where $\psi : \mathcal{P}(S_{rec\ P.t}) \to \mathcal{P}(S_{rec\ P.t})$ is defined by

$$\psi(U) = [\![B[\sigma\backslash X]]\!]_{rec\ P.t}\phi[U/X].$$

Defining $\theta : \mathcal{P}(S_{\underline{t}}) \to \mathcal{P}(S_{\underline{t}})$ by

$$\theta(V) = [\![\mathrm{red}(rec\ P.t : B; \sigma)]\!]_{\underline{t}}\phi[V/Y],$$

we show that ψ and θ are related as required by the reduction lemma:

$$\begin{aligned} in \circ \theta(V) &= in([\![\mathrm{red}(rec\ P.t : B; \sigma)]\!]_{\underline{t}}\phi[V/Y]) \\ &= [\![B[\sigma]]\!]_{rec\ P.t}\phi[V/Y] \\ &\qquad \text{by the induction hypothesis (equation (A6))} \\ &= [\![B[\sigma\backslash X]]\!]_{rec\ P.t}\phi[in(V)/X] \\ &\qquad \text{since } \sigma(X) = IN(Y) \text{ and } \sigma \text{ is fresh for } \nu X.B \\ &= \psi \circ in(V) \\ &\qquad \text{by definition of } \psi. \end{aligned}$$

It is easy to see that in is $\top$-strict and ω-anticontinuous. Hence, the reduction lemma yields

$$in(\nu\theta) = \nu\psi.$$

Writing out θ and ψ and using the definition of the ν-operator, we get

$$[\![(\nu X.B)[\sigma]]\!]_{rec\ P.t}\phi = in([\![\mathrm{red}(rec\ P.t : \nu Y.B; \sigma)]\!]_{\underline{t}}\phi).$$

Moreover, θ restricts to a function θ' on D, as can be seen from the induction hypothesis: for a balanced environment ϕ, $P(B)$ states that $\theta(V)$ is balanced for all balanced V, i.e., θ maps balanced sets to balanced sets. Hence, letting in' be the embedding of D into $\mathcal{P}(S_{\underline{t}})$—easily seen to be $\top$-strict and ω-anticontinuous—we have that

$$\theta \circ in' = in' \circ \theta',$$

which by the reduction lemma gives $\nu\theta = in'(\nu\theta')$. In other words, $\nu\theta \in D$.
$A \equiv \langle\alpha\rangle B, \alpha \neq *$. We rewrite from the left-hand side:

$$[\![\langle\alpha\rangle B[\sigma]]\!]_{rec\ P.t}\phi$$

$$= \{s \in S_{rec\ P.t} | \exists s' \in S_{rec\ P.t}\, s \xrightarrow{\alpha} s' \ \&\ s' \in [\![B[\sigma]]\!]_{rec\ P.t}\phi\}$$

by definition

$$= \{s \in S_{rec\ P.t} | \exists s' \in S_{rec\ P.t}\, s \xrightarrow{\alpha} s' \ \&\ s' \in in\ ([\![B']\!]_{\underline{t}}\phi)\}$$

by the induction hypothesis, where $B' = \mathrm{red}(rec\ P.t : B; \sigma)$

$$= in(\{s \in S_{\underline{t}} | \exists s' \in S_{\underline{t}}\, j(s) \xrightarrow{\alpha} j(s') \ \&\ in\ j(s') \in ([\![B']\!]_{\underline{t}}\phi)\})$$

by the fact that j is surjective (in is $\top$-strict)

$$= in(\{s \in S_{\underline{t}} | \exists s' \in S_{\underline{t}}.\, j(s') \in in([\![B']\!]_{\underline{t}}\phi)$$
$$\&\ ((s = P \text{ and } t \xrightarrow{\alpha} s' \text{ or } s \xrightarrow{\alpha} s')\})$$

by lemma 6.

$$= in(\{s \in S_{\underline{t}} | \exists s' \in S_{\underline{t}}.\, s' \in [\![B']\!]_{\underline{t}}\phi$$
$$\&\ ((s = P \ \&\ t \xrightarrow{\alpha} s') \text{ or } s \xrightarrow{\alpha} s')\})$$

by the second part of the induction hypothesis

$$= in([\![(\hat{P} \wedge (t : \langle\alpha\rangle B')) \vee \langle\alpha\rangle B']\!]_{\underline{t}}\phi)$$

by definition of $[\![\_]\!]_{\underline{t}}\phi$

$$= in([\![\mathrm{red}(rec\ P.t : \langle\alpha\rangle B; \sigma)]\!]_{\underline{t}}\phi)$$

by definition.

Let $in(U)$ be the right-hand side of the third equality (marked $*$). It is easy to observe that $j^{-1}(in(U)) = U$, since the predicate determining whether $s \in U$ only depends on the value of $j(s)$. Moreover, notice that the last five equalities hold without in; hence $[\![\mathrm{red}(rec\ P.t : \langle\alpha\rangle B; \sigma)]\!]_{\underline{t}}\phi = U$ and is therefore balanced (this property actually dictated the construction of the reduction for $\langle\alpha\rangle$).
$A \equiv A_0 \vee A_1$ and $A \equiv \neg B$. The proof is simple. □

Appendix 7. Proof of reduction for product

We will prove that

$$[\![A[\sigma]]\!]_{q\times p}\phi = in([\![A/p_1]\!]_q\phi, \ldots, [\![A/p_n]\!]_q\phi) \tag{A7}$$

for all environments ϕ. Assuming without loss of generality that $p = p_1$, it follows that

$\models q \times p : A \leftrightarrow q : (A/p_1)$.

Let $slice_i : \mathcal{P}(S_{q\times p}) \to \mathcal{P}(S_q)$ be the function that projects onto the ith component, i.e.,

$$slice_i(U) = \{s \in S_q | s \times p_i \in U\}.$$

From the definition of in, it is easy to see that equation (A7) is equivalent to the following:

$$\forall 1 \leq i \leq n.\ slice_i([\![A[\sigma]]\!]_{q\times p}\phi) = [\![A/p_i]\!]_q\phi, \tag{A8}$$

which we will take as our induction hypothesis (but apply equation (A7) when most appropriate).

$A \equiv X$. By definition, we immediately have

$$slice_i([\![X[\sigma]]\!]_{q\times p}\phi) = \phi(X_{p_i}) = [\![X/p_i]\!]_q\phi.$$

$A \equiv \mu X.B$. Let $\theta : \mathcal{P}(S_q)^n \to \mathcal{P}(S_q)^n$ be defined by

$$\theta(V_1, \ldots, V_n) = ([\![B/p_1]\!]_q\phi', \ldots, [\![B/p_n]\!]_q\phi'),$$

where

$$\phi' = \phi[V_1/X_{p_1}, \ldots, V_n/X_{p_n}].$$

Let $\psi : \mathcal{P}(S_{q\times p}) \to \mathcal{P}(S_{q\times p})$ be defined by

$$\psi(U) = [\![B[\sigma\backslash X]]\!]_{q\times p}\phi[U/X].$$

We show that θ and ψ are related as required by the reduction lemma:

$$\begin{aligned} in \circ \theta(V_1, \ldots, V_n) &= [\![B[\sigma]]\!]_{q\times p}\phi[V_1/X_{p_1}, \ldots, V_n/X_{p_n}] \\ &\quad \text{by the induction hypothesis} \\ &= [\![B[\sigma\backslash X]]\!]_{q\times p}\phi[in(V_1, \ldots, V_n)/X] \\ &\quad \text{since } \sigma(X) = IN(X_{p_1}, \ldots, X_{p_n}) \text{ and } \sigma \text{ is fresh} \\ &= \psi \circ in(V_1, \ldots, V_n) \\ &\quad \text{by definition of } \psi. \end{aligned}$$

From the reduction lemma, we now conclude:

$$in(\nu\theta) = \nu\psi,$$

which, by writing out θ and ψ, yields

$$in(\nu \underline{V}.([\![B/p_1]\!]_q\phi[\underline{V}/\underline{X}], \ldots, [\![B/p_n]\!]_q\phi[\underline{V}/\underline{X}]) = [\![\nu X.B]\!]_{q\times p}\phi.$$

By repeated application of Bekić's theorem, the simultaneous fixed point on the left-hand side can be converted into a unary fixed point, yielding the claimed reduction.

$A \equiv \neg B$ and $A \equiv A_0 \vee A_1$. This is immediate by definition.

$p_i \equiv r \times s$. We rewrite from the left-hand side:

$$\begin{aligned} slice_i([\![A[\sigma]]\!]_{q\times p}\phi) &= \{u \in S_q | u \times (r \times s) \in [\![A[\sigma]]\!]_{q\times p}\phi\} \\ &\quad \text{by definition of } slice_i \\ &= \{u \in S_q | (u \times r) \times s \in [\![\widetilde{A}[\sigma]]\!]_{q\times p}\phi\} \\ &\quad \text{by reassociating modalities in } A \\ &= [\![(\widetilde{A}/s)/r]\!]_{q\times p}\phi \\ &\quad \text{by definition.} \end{aligned}$$

$A \equiv \langle \alpha \times \beta \rangle$ and $p_i \equiv nil, \alpha \times \beta \neq *$. We immediately get

$$\begin{aligned} &slice_i([\![\langle \alpha \times \beta \rangle B]\!]_{q\times p}\phi) \\ &= \begin{cases} \{u \in S_q | \exists u' \in S_q . u \xrightarrow{\alpha} u' \text{ and } u' \times p_i \in [\![B[\sigma]]\!]_{q\times p}\phi\} & \text{if } \beta = * \\ \emptyset & \text{if } \beta \neq * \end{cases} \\ &\quad \text{by definition} \\ &= \begin{cases} \{[\![\langle \alpha \rangle (B/p_i)]\!]_q \phi\} & \text{if } \beta = * \\ \emptyset & \text{if } \beta \neq * \end{cases} \\ &\quad \text{by the induction hypothesis} \\ &= [\![A/p_i]\!]_q \phi \\ &\quad \text{by definition.} \end{aligned}$$

The missing cases are all similar to the last case considered. □

Notes

1. Because of the isomorphism $\mathcal{P}(A_0) \times \cdots \times \mathcal{P}(A_n) \times \cdots \cong \mathcal{P}(A_0 + \cdots + A_n + \cdots)$, we can still meet the conditions of the reduction lemma when D is a countable product of powersets of countable sets.
2. For this and the following reductions, we have that $\text{red}(at : \langle * \rangle; \sigma) = \text{red}(at : A; \sigma)$, and henceforth we will omit these trivial cases from the presentation.
3. The general semantics should be $[\![\widehat{P}]\!]_T \phi = \{P, \underline{P}\} \cap S_T$, but due to our requirement of guardedness, we will never be involved with rooting a state identifier, so the stated semantics is sufficient.
4. Termination is ensured by the well-founded order consisting of the number of products in the process term combined lexicographically with the structure

of assertions again combined lexicographically with the maximal depth to a prefix in the process term.

References

[1] Colin Stirling. Modal and temporal logics. In *Handbook of Logic in Computer Science*, S. Abramsky, D. Gabbay, and T. Maibaum, (eds.). Oxford University Press, Oxford, 1991.

[2] Mads Dam. Translating CTL* into the modal μ-calculus. Technical Report ECS-LFCS-90-123, Laboratory for Foundations of Computer Science, University of Edinburgh, November 1990.

[3] E. Allen Emerson and Chin-Luang Lei. Efficient model checking in fragments of the propositional mu-calculus. In *Symposium on Logic in Computer Science, Proceedings*. IEEE, 1986, pp. 267–278.

[4] E.M. Clarke, E.A. Emerson, and A.P. Sistla. Automatic verification of finite-state concurrent systems using temporal logic specifications. *ACM Transactions on Programming Languages and Systems*, 8(2): 244–263, 1986.

[5] Colin Stirling and David Walker. Local model checking in the modal mu-calculus. In *Proceedings of TAPSOFT*, Barcelona, Spain, March 1989, LNCS 351: 369–383, 1989.

[6] Kim G. Larsen. Proof systems for Hennessy-Milner logic with recursion. In *Proceedings of CAAP*, 1988.

[7] Glynn Winskel. A note on model checking the modal ν-calculus. In Ausiello, Dezani-Ciancaglini, and Rocca (eds.). *Proceedings of ICALP*, Ausiello, *Lecture Notes in Computer Science*, 372: 761–772, 1989.

[8] Rance Cleaveland. Tableau-based model checking in the propositional mu-calculus. *Acta Informatica*, 27: 725–747, 1990.

[9] André Arnold and Paul Crubille. A linear algorithm to solve fixed-point equations on transitions systems. *Information Processing Letters*, 29: 57–66, 1988.

[10] E.M. Clarke, D.E. Long, and K.L. McMillan. Compositional model checking. In *Proceedings of 4th Annual Symposium on Logic in Computer Science*, Pacific Grove, CA. IEEE, 353–362, 1989.

[11] Kim G. Larsen and Liu Xinxin. Compositionality through an operational semantics of contexts. In *Proceedings of ICALP*, M.S. Paterson (ed.). *Lecture Notes in Computer Science*, 443, 526–539, 1990.

[12] A. Tarski. A lattice-theoretical fixpoint theorem and its applications. *Pacific Journal of Mathematics*, 5: 285–309, 1955.

[13] H. Bekić. Definable operations in general algebras, and the theory of automata and flow charts. *Lecture Notes in Computer Science*, 177: 30–55, 1984.

[14] Glynn Winskel. On the compositional checking of validity. In *Proceedings of CONCUR '90*, J.C.M. Baeten and J.W. Klop (eds.). *Lecture Notes in Computer Science*, 458: 481–501, 1990.

[15] P. Aczel. An introduction to inductive definitions. In *Handbook of Mathematical Logic*, Jon Barwise (ed.). North-Holland, Amsterdam, 1983.