

On Model Checking Infinite-State Systems

Henrik Reif Andersen*

Abstract

This paper presents a proof method for proving that infinite-state systems satisfy properties expressed in the modal μ -calculus. The method is sound and complete relative to externally proving inclusions of sets of states. It can be seen as a recast of a tableau method due to Bradfield and Stirling following lines used by Winskel for finite-state systems. Contrary to the tableau method, it avoids the use of constants when unfolding fixed points and it replaces the rather involved global success criterion in the tableau method with local success criteria. A proof tree is now merely a means of keeping track of where possible choices are made – and can be changed – and not an essential ingredient in establishing the correctness of a proof: A proof will be correct when all leaves are directly seen to be valid. Therefore, it seems well-suited for implementation as a tool, by, for instance, integration into existing general-purpose theorem provers.

1 Introduction

Verifying dynamic properties of infinite-state systems is a difficult task. As soon as the language for describing systems has Turing strength, almost no interesting properties are decidable. Thus, instead of algorithms, we must rely on supplying proof methods that can verify, by human interaction, that such systems possess their desired properties.

The modal μ -calculus, with labelled transition systems as models, is an example of a logic for which the verification problem is undecidable. In fact, it is not even semi-decidable. Hence, we are forced to rely on proof systems that must be either infinitary or incomplete. It is rather difficult to compare and judge such methods; there is no easy measure like time or space complexity that allows for direct comparison. The best one could hope for in terms of a technical result would be some kind of relative completeness, i.e. up to external reasoning in some “simpler” logic, the rules for the modal logic are complete. But it is still not a good

*Department of Computer Science, Technical University of Denmark, Building 344, 2800 Lyngby, Denmark. E-mail: hra@id.dth.dk. Phone: +45 45933332. Fax: +45 42884530. This work is supported by the Danish Technical Research Council.

ground for comparison. How much of the complexity of the proof are being pushed to the simpler logic? It seems that it must unavoidable become more pragmatic issues like ease of use and well-suitedness for machine support that are of importance for such proof systems.

Facing these difficulties, one might consider giving up finding proof systems at all and always rely on the semantics of the logic, performing all the reasoning within the meta-language. However, the fixed points that gives the modal μ -calculus its high expressiveness are rather difficult to handle semantically. Computing the fixed points by taking limits of approximants, requires the use of ordinals (since not all assertions in the logic are continuous), and having to always to refer to the Knaster-Tarski fixed-point theorem, without any benefit from the structure of the models at all, is rather unpleasant.

We propose in this paper an infinitary proof system, complete up to externally proving inclusion on sets of states, that do not rely on computing fixed points explicitly. It uses a well-founded induction principle for minimum fixed points, and for maximum fixed points it employs a rule that is closely related to Park's induction principle [9] (named *co-induction* by Milner and Tofte [8]).

The proof system can be seen as a recast of a tableau method due to Bradfield and Stirling [5] using an idea of Winskel [13] for handling the maximum fixed points. Contrary to the tableau method, there are no complicated global success criteria on the proof trees, but instead an infinitary proof rule.

2 Fixed Points

Fixed points play a central role in the modal μ -calculus, and we shall start by reviewing a few facts about fixed points that will form the basis of our fixed-point rules.

As it is well-known from the Knaster-Tarski fixed point theorem [12], a monotonic function ψ on a powerset $\mathcal{P}(S)$ has a minimum (resp. maximum) fixed point we denote by $\mu\psi$ (resp. $\nu\psi$). According to the theorem these can be found as the intersection (resp. union) over all prefixed (resp. postfix) points of ψ ; a prefixed (postfix) point being a subset U s.t. $\psi(U) \subseteq U$ ($\psi(U) \supseteq U$). This characterization can be used for proving a property of maximum fixed-points which we, following Winskel [13], call the *reduction lemma*:

Lemma 1 (Reduction lemma, Kozen [6], Winskel [13]) *Let ψ be a monotonic function on the powerset $\mathcal{P}(S)$. For $U \subseteq S$ we have*

$$U \subseteq \nu\psi \Leftrightarrow U \subseteq \psi(\nu V.U \cup \psi(V)).$$

Winskel uses this lemma in the situation where U is a singleton $\{p\}$. He defines a relation which in a precise sense makes the right-hand side smaller, thus simpler to verify, and because he works with finite-state systems, this relation turns out to be well-founded, ensuring

termination of the algorithm. As we consider infinite-state systems, termination will no longer be guaranteed in this manner. Moreover, following Bradfield and Stirling [5] we will try to verify that (possibly infinite) sets of states satisfy an assertion, not only singletons, which seems more appropriate for infinite-state systems. Although initially we might only want to know whether one particular state satisfies an assertion, this state can quickly lead to considering whether an infinite number of states satisfy an assertion (an example of this is provided later). Therefore, we shall apply the lemma in its general form in giving a rule for maximum fixed points.

For the minimum fixed points we shall use a principle of *well-founded induction*. Recall, that a relation \sqsubset on the set U is a *well-founded relation* (abbreviated *w.f.r.*) if all decreasing chains $u_0 \sqsupset u_1 \sqsupset \dots \sqsupset u_n \sqsupset \dots$ are finite. For a subset W of U , let $(\sqsubset W)$ be the set of elements of U less than *all* elements of W , i.e.

$$(\sqsubset W) =_{\text{def}} \{u \in U \mid \forall w \in W. u \sqsubset w\}.$$

Finally, define a *covering* of U to be a collection of sets $\{U_i\}_{i \in I}$ s.t. $\bigcup_{i \in I} U_i = U$.

Lemma 2 (Well-founded induction on minimum fixed-points)

Let ψ be a monotonic function on $\mathcal{P}(S)$. For a set $U \subseteq S$, the following holds:

If there exists a well-founded relation \sqsubset on U and a covering $\{U_i\}_{i \in I}$ of U such that
 $\forall i \in I. U_i \subseteq \psi(\mu V. (\sqsubset U_i) \cup \psi(V))$
then $U \subseteq \mu\psi$

The proof, a rather straightforward application of well-founded induction, will be given in the full paper.

The other direction of the implication holds in a trivial way. Take $I = \{1\}$, $U_1 = U$, and \sqsubset to be the empty relation. Then as $(\sqsubset U) = \emptyset$, the requirement to this trivial covering degenerates to the validity of unfolding of fixed points. However, also more interesting choices of covering and well-founded relation exist, indeed in showing completeness of the method, it is argued that a certain canonical covering and relation can be found such that the minimum fixed point will never be unfolded more than once.

3 Logic

We will use a version of Kozen’s propositional μ -calculus [6], which due to the close relationship to minimal modal logic, we – following Stirling [11] – call the *modal μ -calculus*. The syntax is described by the following grammar:

$$A ::= A_0 \vee A_1 \mid A_0 \wedge A_1 \mid \langle \kappa \rangle A \mid [\kappa] A \mid X \mid \mu X \{U\} A \mid \nu X \{U\} A$$

In the modalities, κ is a (possibly infinite) set of *labels*. We use the abbreviation ‘.’ for all labels. As models we take labelled transition systems $T = (S, L, \rightarrow)$, where S is a set of states, L a set of labels, and $\rightarrow \subseteq S \times L \times S$ a transition relation. Due to the presence of variables in the logic the semantics $\llbracket A \rrbracket_T \rho$ will be given relative to an environment ρ taking variables to sets of states. Hence $\llbracket X \rrbracket_T \rho = \rho(X)$. Conjunction and disjunction are interpreted as intersection and union. The denotation of the modalities are

$$\begin{aligned} \llbracket \langle \kappa \rangle A \rrbracket_T \rho &= \{s \in S \mid \exists s' \in S \exists a \in \kappa. s \xrightarrow{a} s' \ \& \ s' \in \llbracket A \rrbracket_T \rho\} \\ \llbracket [\kappa] A \rrbracket_T \rho &= \{s \in S \mid \forall s' \in S \forall a \in \kappa. s \xrightarrow{a} s' \Rightarrow s' \in \llbracket A \rrbracket_T \rho\} \end{aligned}$$

and for the fixed points, let $\psi : \mathcal{P}(S) \rightarrow \mathcal{P}(S)$ be the function $\psi(V) = U \cup \llbracket A \rrbracket_T \rho[V/X]$ and define

$$\begin{aligned} \llbracket \mu X \{U\} A \rrbracket_T \rho &= \mu \psi, \\ \llbracket \nu X \{U\} A \rrbracket_T \rho &= \nu \psi. \end{aligned}$$

Thus the *tags* on a fixed point is a set of states *assumed* to belong to the fixed point, since we always have $U \subseteq \llbracket \sigma X \{U\} A \rrbracket_T \rho$ where σ is either μ or ν . The usual fixed point $\mu X.A$ is now an abbreviation for $\mu X \{\emptyset\} A$. We define the *satisfaction predicate* \models on *correctness assertions* $U : A$ as follows: For a closed assertion A and a set $U \subseteq S$ let

$$\models_T U : A \Leftrightarrow_{\text{def}} U \subseteq \llbracket A \rrbracket_T \rho,$$

for any environment ρ . Throughout the paper we shall assume T fixed and therefore omit the subscripts.

4 The Model Checking Method

The proof system will consist of rules defining two relations \longrightarrow and \Longrightarrow that rewrite sets of correctness assertions to other, potentially, simpler correctness assertions. This is done in a goal-oriented manner: Given a goal – a correctness assertion – we want to prove correct, generate a set of sufficient subgoals to prove. Thus the proof theoretic counterpart of \models will be the predicate \vdash on correctness assertions defined by

$$\vdash U : A \Leftrightarrow_{\text{def}} \{U : A\} \Longrightarrow^* \emptyset,$$

where \Longrightarrow^* is the transitive, reflexive closure of \Longrightarrow . I.e. we have proven $U : A$ correct if it can be rewritten to an empty set of goals.

Formally, let CorrAssn^{cl} be the set of closed correctness assertions. Then $\longrightarrow \subseteq \text{CorrAssn}^{cl} \times \mathcal{P}(\text{CorrAssn}^{cl})$ is a relation between correctness assertions and sets of correctness assertions. However, as the minimum fixed points can result in infinite sets of correctness assertions – all of the same form – we will describe a relation $\Longrightarrow \subseteq \mathcal{P}(\text{CorrAssn}^{cl}) \times \mathcal{P}(\text{CorrAssn}^{cl})$ which will allow *sets* of correctness assertions to be rewritten. The defining clauses for \longrightarrow are given in figure 1 and for \Longrightarrow in figure 2.

Let us consider some of the rules in more detail:

$(\vee)^*$	$U : A \vee B$	\longrightarrow	$\{(U_0 : A), (U_1 : B)\}$	if $U_0 \cup U_1 = U$
(\wedge)	$U : A \wedge B$	\longrightarrow	$\{(U : A), (U : B)\}$	
$(\langle \rangle)^*$	$U : \langle \kappa \rangle A$	\longrightarrow	$\{U' : A\}$	if $U \subseteq (\overset{\kappa}{\rightarrow} U')$
$(\llbracket \rrbracket)$	$U : \llbracket \kappa \rrbracket A$	\longrightarrow	$\{(U \overset{\kappa}{\rightarrow}) : A\}$	
$(\mu 0)$	$U : \mu X \{V\} A$	\longrightarrow	\emptyset	if $U \subseteq V$
$(\mu 1)^*$	$U : \mu X \{V\} A$	\longrightarrow	$\{U_i : A[\mu X \{V \cup (\sqsubset U_i)\} A / X]\}_{i \in I}$	if $U \not\subseteq V$ $\bigcup U_i = U$ \sqsubset w.f.r. on U
$(\nu 0)$	$U : \nu X \{V\} A$	\longrightarrow	\emptyset	if $U \subseteq V$
$(\nu 1)$	$U : \nu X \{V\} A$	\longrightarrow	$\{U : A[\nu X \{V \cup U\} A / X]\}$	if $U \not\subseteq V$
$(\mathbf{W})^*$	$U : A$	\longrightarrow	$\{U' \cup U : A\}$	
(\mathbf{I})	$U : A$	\longrightarrow	$\{U : A\}$	

Figure 1: The rules. Rules marked with a * involve a choice to be made.

$(\langle \rangle), (\llbracket \rrbracket)$ In rule $(\llbracket \rrbracket)$ the set $(U \overset{\kappa}{\rightarrow})$ denotes states that can be reached through an action in κ from a state in U , i.e. they are “ κ -reachable from U ”:

$$U \overset{\kappa}{\rightarrow} =_{\text{def}} \{s \in S \mid \exists u \in U \exists a \in \kappa. u \xrightarrow{a} s\}.$$

The importance of this operator is that

$$U \subseteq \llbracket \llbracket \kappa \rrbracket A \rrbracket \Leftrightarrow (U \overset{\kappa}{\rightarrow}) \subseteq \llbracket A \rrbracket,$$

which makes it possible to give a deterministic rule.

It is, however, not possible to define a similar operation for the diamond-modality, which inevitably must involve some choices. To see this, consider the simple three-state transition system $(\{p, q, r\}, \{a\}, \rightarrow)$ with $p \xrightarrow{a} q$ and $p \xrightarrow{a} r$. Now, if $\{p\} : \langle a \rangle A$ is to be valid, then *either* $\{q\} : A$ *or* $\{r\} : A$ *or* $\{q, r\} : A$ must be valid, but it is not possible to tell a priori whether we should insist on this being $\{q\}$, $\{r\}$ or perhaps $\{q, r\}$. We have chosen to present this choice in a way which also allows for weakening, hence in rule $(\langle \rangle)$, U' is any set which satisfies $U \subseteq (\overset{\kappa}{\rightarrow} U')$, where $\overset{\kappa}{\rightarrow} U'$ is the set of states “ κ -reaching U' ”:

$$\overset{\kappa}{\rightarrow} U' =_{\text{def}} \{s \in S \mid \exists u \in U' \exists a \in \kappa. s \xrightarrow{a} u\}.$$

$(\mu 0), (\mu 1), (\nu 0), (\nu 1)$ The rules $(\mu 0)$ and $(\nu 0)$ are direct consequences of the semantics of the tagged fixed points; the rule $(\nu 1)$ is the proof theoretic version of the reduction lemma. The rule $(\mu 1)$ is the most complicated rule. It is derived from lemma 2. It

$$\frac{\forall \gamma \in \Gamma. \gamma \longrightarrow \Delta_\gamma}{\Gamma \Longrightarrow \bigcup_{\gamma \in \Gamma} \Delta_\gamma} \quad (\Longrightarrow)$$

Figure 2: The infinitary rule.

potentially introduces an infinite number of correctness assertions making the relation \Longrightarrow necessary. In the examples to follow, we will see that the covering and the w.f.r. needed in the application of the rule, is suggested to us quite directly from the system under consideration. Whether this is the case in general can only be revealed by further experiments.

(W) The weakening rule is essential to the completeness of the system, in particular, for the maximum fixed points.

The infinite sets of correctness assertions generated by the μ -rule, initially all have the same form and we can expect that they to a large extent can be rewritten simultaneously using the same rules. The simultaneous rewriting is formalized by \Longrightarrow . As \longrightarrow by (I) is reflexive, the rule allows one to select some correctness assertions to be rewritten according to \longrightarrow and leave others unchanged.

The rules are sound and complete – up to externally proving inclusion of sets of states:

Theorem 1 (Soundness) *Suppose A is a closed assertion. If $\vdash U : A$ then $\models U : A$.*

Theorem 2 (Completeness) *Suppose A is a closed assertion. If $\models U : A$ then $\vdash U : A$.*

Proofs will be included in the full version. They can be found in [1].

5 Examples

In this section we will show how to apply the method to two small examples. We will use (a subset of) Milner’s CCS with value-passing [7] for expressing transition systems, and suggest a notation for infinite sets of states which seems to be particularly useful for a class of *bounded processes*; processes which do not have arbitrarily, unbounded evolving structure.

First, we recall the syntax. Assume that \mathcal{A} is a set of action names, and assume that \mathbb{V} is a set of values. Process expressions are generated from the syntax

$$E ::= \mathbf{0} \mid \pi.E \mid (\psi)E \mid E + E \mid C(e_1, \dots, e_n),$$

where C denotes a process constant with arity n defined through an equation $C(v_1, \dots, v_n) = E$, where the free value variables of E are among v_1, \dots, v_n . Constant definitions can be mutually recursive. The actions π are either input, output, or silent actions,

$$\pi ::= a?v \mid a!e \mid \tau,$$

where $a \in \mathcal{A}$. Value expressions e are build from a set of operators applied to value variables $v \in \text{var}$, and constants $c \in \text{const}$. Guards (ψ) are boolean expressions over predicates on value expressions. The operational semantics of CCS with value-passing is standard, giving a ‘universal’ labelled transition \mathcal{T} (see Milner [7]). In \mathcal{T} states are identified with closed process expressions, so sets of states are sets of closed process expressions, which we suggest can be described by

$$\vec{t}; \vec{\psi} [\vec{v}],$$

where \vec{t} is a list of process expressions, $\vec{\psi}$ a list of predicates, and \vec{v} a list of free value-variables – implicitly assumed to be universally quantified. That is, tentatively, the semantics of $\vec{t}; \vec{\psi} [\vec{v}]$ is the set

$$\llbracket \vec{t}; \vec{\psi} [\vec{v}] \rrbracket = \{t_1[\vec{c}/\vec{v}], \dots, t_m[\vec{c}/\vec{v}] \mid c_1, \dots, c_n \in \mathbb{V} \text{ and } \vec{\psi}[\vec{c}/\vec{v}] \text{ holds}\}.$$

An example when the values are natural numbers is

$$P, Q(n); n > 0, n \leq 3 [n]$$

i.e. the set $\{P, Q(1), Q(2), Q(3)\}$. We will simply write $\vec{t} [\vec{v}]$ instead of $\vec{t}; \vec{\psi} [\vec{v}]$ when $\vec{\psi}$ is empty. Using this notation, entailments will be on the form $\vdash \vec{t}; \vec{\psi} [\vec{v}] : A$ or $\vdash \vec{t} [\vec{v}] : A$.

Example 1 Define P and $Q(n)$ as follows:

$$\begin{aligned} P &= a?n.Q(n) \\ Q(n) &= (n > 0)\tau.Q(n-1) \end{aligned}$$

Hence P inputs a number n on the channel a , and then proceeds by making n τ 's.¹ We will show that P always terminates, i.e. that all execution sequences are finite. This is expressed in the modal μ -calculus as $\mu X\{.\}[\cdot]X$. We proceed as follows:

$$\begin{aligned} P : \mu X\{.\}[\cdot]X &\xrightarrow{(\mu 1)} P : [\cdot]\mu X\{.\}[\cdot]X \\ &\quad \text{with trivial singleton covering, arbitrary w.f.r.} \\ &\xrightarrow{(\square)} Q(n) [n] : \mu X\{.\}[\cdot]X \\ &\quad \text{since for all } n \in \omega, P \xrightarrow{a?n} Q(n) \end{aligned}$$

¹This simple example has been used to illustrate that on a very simple transition system, $\mu X\{.\}[\cdot]X$ cannot be found as the ω -limit of its approximants; the ordinal $\omega + 1$ is necessary.

$$\begin{aligned}
& \xrightarrow{(\mu 1)} \{Q(n) : [\cdot] \mu X \{\sqsubset Q(n)\} [\cdot] X\}_{n \in \omega} \\
& \quad \text{with covering } \{Q(n)\}_{n \in \omega} \text{ and w.f.r. } Q(m) \sqsubset Q(n) \Leftrightarrow_{\text{def}} m < n. \\
& = \{Q(0) : [\cdot] \mu X \{\sqsubset Q(0)\} [\cdot] X\} \cup \{Q(n) : [\cdot] \mu X \{\sqsubset Q(n)\} [\cdot] X\}_{n > 0} \\
& \xrightarrow{(\parallel)} \{\emptyset : \mu X \{\sqsubset Q(0)\} [\cdot] X\} \cup \{Q(n) : [\cdot] \mu X \{\sqsubset Q(n)\} [\cdot] X\}_{n > 0} \\
& \quad \text{since } Q(0) \not\sqsubset \\
& \xrightarrow{(\mu 0)} \{Q(n) : [\cdot] \mu X \{\sqsubset Q(n)\} [\cdot] X\}_{n > 0} \\
& \xrightarrow{(\parallel)} \{Q(n-1) : \mu X \{\sqsubset Q(n)\} [\cdot] X\}_{n > 0} \\
& \xrightarrow{(\mu 0)} \emptyset \quad \text{as } n-1 < n, \text{ implying } Q(n-1) \sqsubset Q(n).
\end{aligned}$$

Notice, that the splitting of the ω -set of correctness assertions after the third step was strongly suggested to us by the guard $n > 0$ in the definition of $Q(n)$.

It is also worthwhile to observe that although we used a covering with singleton sets here, it is not always necessary to fall back on singletons. If we instead had the definition

$$\begin{aligned}
P &= a?n.b?m.Q(n, m) \\
Q(n, m) &= (n > 0)c!m.b?m.Q(n-1, m),
\end{aligned}$$

we could use the covering

$$\{\{Q(n, m)\}_{m \in \omega}\}_{n \in \omega}$$

and the w.f.r. $Q(n', m') \sqsubset Q(n, m) \Leftrightarrow_{\text{def}} n' < n$. \square

Example 2 This is an example from Bradfield [4, p. 6]. Consider the following definition of a process M :

$$\begin{aligned}
M(A, B, C) &= (A \geq 1)a.M(A, B+1, C) \\
&+ (A \geq 1)b.M(A-1, B, C+1) \\
&+ (B \geq 1 \wedge C \geq 1)c.M(A, B-1, C)
\end{aligned}$$

The process $M(l, m, n)$ describes the firing sequence of a certain *Petri net* with l tokens on the place A , m tokens on the place B , and n tokens on the place C , and the actions a , b , and c are *transitions* of the Petri net.

Using the previously defined notation, sets of states will now be described by

$$M(A, B, C); \vec{\psi} [\vec{v}]$$

For convenience, we will omit $M(A, B, C)$ and just write $\vec{\psi} [\vec{v}]$. The initial marking we consider is $A = 1, B = 0, C = 0$ and we will show that c only happens finitely often, expressed as the assertion $\mu X \{\} \nu Y \{\} [c] X \wedge [a, b] Y$. Intuitively this is obvious: Either a fires

indefinitely, increasing the number of tokens on B , or at some point b fires, and then only c can fire. As there is only a finite number of tokens on B when this happens and c removes one token whenever fired, it must eventually stop.

Formally, we show:

$$\{A = 1, B = 0, C = 0 : \mu X \{ \} \nu Y \{ \} [c] X \wedge [a, b] Y\} \Longrightarrow^* \emptyset.$$

Let $X_0 = \mu X \{ \} \nu Y \{ \} [c] X \wedge [a, b] Y$ and rewrite as follows:

$$\begin{aligned} & \{A = 1, B = 0, C = 0 : X_0\} \\ & \xrightarrow{(\mathbf{W})} \{A + C = 1, B = m \ [m] : X_0\} \\ & \xrightarrow{(\mu 1)} \{A + C = 1, B = m : \nu Y \{ \} [c] X_1 \wedge [a, b] Y\}_{m \in \omega} \\ & \quad \text{where the w.f.r. is } M(l, m, n) \sqsubset M(l', m', n') \Leftrightarrow_{\text{def}} m < m' \\ & \quad \text{and hence } X_1 \text{ is } \mu X \{A + C = 1, B < m\} \nu Y \{ \} [c] X \wedge [a, b] Y \\ & \xrightarrow{(\nu 1)} \{A + C = 1, B = m : [c] X_1 \wedge [a, b] Y_0\}_{m \in \omega} \\ & \quad \text{where } Y_0 = \nu Y \{A + C = 1, B = m\} [c] X_1 \wedge [a, b] Y \\ & \xrightarrow{(\wedge)} \{A + C = 1, B = m : [c] X_1, A + C = 1, B = m : [a, b] Y_0\}_{m \in \omega} \\ & \xrightarrow{(\parallel)} \{A + C = 1, B = m : [c] X_1\}_{m \in \omega}, \{A + C = 1, B = m, (m = 0 \Rightarrow C = 1) : Y_0\}_{m \in \omega} \\ & \xrightarrow{(\nu 0)} \{A + C = 1, B = m : [c] X_1\}_{m \in \omega} \\ & \xrightarrow{(\parallel)} \{A = 0, B = m - 1, C = 1 : X_1\}_{m > 0} \\ & \xrightarrow{(\mu 0)} \emptyset \end{aligned}$$

It is essential to initially apply weakening to make the application of rule $(\mu 1)$ successful. The interested reader is encouraged to compare this with the proof of Bradfield [4]. \square

In the previous two examples, the involved processes were of a particular simple kind: they did not have “evolving structure”. To be precise about this, let $\widehat{}$ be the operation which, by simply ignoring values, maps CCS process expressions with values to CCS process expressions without. I.e. on the action prefixes it behaves as: $\widehat{a?v} = a, \widehat{a!v} = \bar{a}, \widehat{\tau} = \tau$. Now, a CCS process with values, P , is *bounded* if the set $\{\widehat{Q} \mid \exists n. \exists a_1, \dots, a_n. P \xrightarrow{a_1} \xrightarrow{a_2} \dots \xrightarrow{a_n} Q\}$ is finite. The notation we have used above seems to be particularly well-suited for bounded processes, as all the reachable states can be described from a finite number of process expressions, together with a collection of constraints based on the guards of the processes. It is not difficult to see that each particular *state* can actually be described by a process expression and a finite number of such constraints, but whether any *set of states* expressible in the modal μ -calculus can actually be described in this manner, yielding a relative completeness result, is another more difficult issue not addressed here.

6 The Tableau Method of Bradfield and Stirling

As mentioned in the introduction, the present proof system can be seen as a recast of the tableau method of Bradfield and Stirling [5]. It was inspired by trying to understand their method and especially the rather involved success criterion they employ for minimum fixed points. To understand the difference, it is necessary to briefly sketch their method.

The tableau method involves a finitary set of goal-oriented rules for building proof trees and success criteria on such proof trees for determining when they provide valid proofs. They use rules corresponding to rules (\vee) – (\Box) and the weakening rule **(W)**, with the minor modification of decorating the rule for the diamond-modality with a function expliciting for each state the choice of successor. For the fixed points, however, there are some essential differences; in particular, for the minimum fixed points. Firstly, they use the more standard untagged fixed points $\mu X.A$ and $\nu X.A$. Secondly, the use of *assertional constants* and corresponding *lists of definitions* for these play a crucial role. Hence, the correctness assertions of a proof tree are decorated by definition lists; for the root the list is empty. When a fixed point assertion $\sigma X.A$ is met, it is replaced by a *fresh constant* Q (i.e. one which is not in the current definition list) and the definition $Q = \sigma X.A$ is added to the set of definitions and used in descending nodes of the proof tree.

A proof tree is complete when all leaves are *terminal*. A leaf $U : A$ is terminal if either (i) U is empty, (ii) A is a diamond-modality for which one of the states of U has no successor, or (iii) A is a constant Q , and there is an ancestor node $U' : Q$ with $U' \supseteq U$ – this is called a *companion* to $U : Q$. (If a correctness assertion $U : Q$ is not terminal, it can be unfolded according to its definition as a fixed point, using the constant as the substitute in the unfolding.)

A proof tree provides a valid proof if it is finite, all leaves terminal, and all terminals *successful*. A terminal of type (i) or type (iii), when Q is a maximum fixed point constant, is always successful. A terminal of type (ii) is never successful, and finally a minimum fixed point constant of type (iii) is successful provided it satisfies the criterion of *mu-success*.

Mu-success requires well-foundedness of a collection of relations $\sqsubset_{\mathbf{n}}$ defined for each node n that is a companion to some terminal mu-node. These relations are defined in terms of *paths* and *extended paths*. A *path* is a sequence of state-node pairs. Neighbouring state-node pairs $(s, \mathbf{n}), (s', \mathbf{n}')$ in a path must satisfy that s (s') is a state in the set at node \mathbf{n} (\mathbf{n}') and \mathbf{n}' is a child of \mathbf{n} . Moreover, if the rule applied to \mathbf{n} is the box-modality rule, s' must be reachable from s through one of the actions of the modality; if it is the diamond-modality rule, s' must be the one selected as the successor state of s ; otherwise s and s' must be identical. Hence, a path reflects transitional dependencies between states in a proof tree.

There is an *extended path* from (s, \mathbf{n}) to (s', \mathbf{n}') if either

1. there is a *path* from (s, \mathbf{n}) to (s', \mathbf{n}') , or

2. there is a node \mathbf{n}'' and a finite sequence of states s_0, s_1, \dots, s_k and nodes $\mathbf{n}_1, \dots, \mathbf{n}_k$, where each \mathbf{n}_i is a terminal with companion \mathbf{n}'' , such that there is a *path* from (s, \mathbf{n}) to (s_0, \mathbf{n}'') , there is an *extended path* from (s_i, \mathbf{n}'') to $(s_{i+1}, \mathbf{n}_{i+1})$ for $0 \leq i < k$, and there is an *extended path* from (s_k, \mathbf{n}'') to (s', \mathbf{n}')

(It can be argued that this recursive definition is well-defined, see Bradfield [3].)

Finally, for any node \mathbf{n} with state set U , that is a companion to some mu-node \mathbf{n}' , the relation $\sqsubset_{\mathbf{n}}$ on U is defined by: $s \sqsubset_{\mathbf{n}} s''$, if and only if, for any terminal \mathbf{n}'' with companion \mathbf{n} , there is an extended path from (s, \mathbf{n}) to (s'', \mathbf{n}'') . The criterion of *mu-success* is that for all companions, the relation $\sqsubset_{\mathbf{n}}$ should be well-founded.

The major difference between the current approach and the tableau method is the mu-success criterion. We have replaced it by a local condition on the unfolding and tagging – the price being that a higher-order rule (for \implies) is needed. We find their criterion quite involved, requiring detailed examinations of single states along the proof tree. On the other hand in our method we are facing the higher-orderness of the \implies -rule.² However, we believe that finding coverings and w.f.r.'s will be quite natural – as in the two examples; however, we do admit that these are rather subjective claims. Further experiments with building tools and working with examples will show whether these claims hold.

7 Conclusion

One shortcoming of the method presented so far, is the inability to show that $\models U : A$ does *not hold*. The rules are not very appropriate for this; one has to show that all the possible choices lead to false expressions. An obvious attempt to remedy this would be to simply try to show that U satisfies another assertion making $\models U : A$ impossible. If U is a singleton $\{u\}$, this is quite easy as $\not\models \{u\} : A$, iff, $\models \{u\} : \neg A$. This is not the case for general U , but we instead observe that

$$\not\models U : A, \text{ if and only if, } \exists U' (\emptyset \neq U' \subseteq U). \models U' : \neg A.$$

Instead of introducing a new rule for negation – which is just as difficult as to cope with as showing non-validity – we consider $\neg A$ to be simply an operation on assertions that dualizes every operator in A (taking $\langle \kappa \rangle$ to $[\kappa]$, μ to ν etc.), thereby making the method applicable as it is. The only remaining difficulty would be to find a suitable U' .

Interesting future work would be to specialize the method to specific areas of infinite-state systems, e.g. context-free processes and real-time processes; and combine it with techniques exploiting the structure of the processes under consideration (like in for instance, Winskel [14] and Stirling [10]).

²An indication of the difficulties of getting hold of the relations $\sqsubset_{\mathbf{n}}$ are given in [4] where instead a stronger (incomplete) condition, *implying* mu-success is used in the proof assistant.

References

- [1] Henrik Reif Andersen. *Verification of Temporal Properties of Concurrent Systems*. PhD thesis, Department of Computer Science, Aarhus University, Denmark, June 1993. PB-445.
- [2] J.C.M. Baeten and J.W. Klop, editors. *Proceedings of CONCUR '90*, volume 458 of *LNCS*. Springer-Verlag, 1990.
- [3] Julian C. Bradfield. *Verifying Temporal Properties of Systems with Applications to Petri Nets*. PhD thesis, Laboratory for Foundations of Computer Science, University of Edinburgh, July 1991.
- [4] Julian C. Bradfield. A proof assistant for symbolic model-checking. Technical Report ECS-LFCS-92-199, Laboratory for Foundations of Computer Science, University of Edinburgh, March 1992.
- [5] Julian C. Bradfield and Colin P. Stirling. Verifying temporal properties of processes. In Baeten and Klop [2], pages 115–125.
- [6] Dexter Kozen. Results on the propositional mu-calculus. *Theoretical Computer Science*, 27, 1983.
- [7] Robin Milner. *Communication and Concurrency*. Prentice Hall, 1989.
- [8] Robin Milner and Mads Tofte. Co-induction in relational semantics. *Theoretical Computer Science*, 87:209–220, 1991.
- [9] David Park. Fixpoint induction and proofs of program properties. *Machine Intelligence*, 5, 1969.
- [10] Colin Stirling. A complete compositional modal proof system for a subset of CCS. volume 194 of *Lecture Notes in Computer Science*, pages 475–486. Springer-Verlag, 1985.
- [11] Colin Stirling. Modal and Temporal Logics. In S. Abramsky, D. Gabbay, and T. Maibaum, editors, *Handbook of Logic in Computer Science*, volume 2, pages 477–563. Oxford University Press, 1992.
- [12] A. Tarski. A lattice-theoretical fixpoint theorem and its applications. *Pacific Journal of Mathematics*, 5:285–309, 1955.
- [13] Glynn Winskel. A note on model checking the modal ν -calculus. In G. Ausiello, M. Dezani-Ciancaglini, and S. Ronchi Della Rocca, editors, *Proceedings of ICALP*, volume 372 of *LNCS*, pages 761–772. Springer-Verlag, 1989.
- [14] Glynn Winskel. On the compositional checking of validity. In Baeten and Klop [2], pages 481–501.