Theoretical Computer Science **96** (1992) 157–174

LOCAL MODEL CHECKING FOR INFINITE STATE SPACES

Julian BRADFIELD and Colin STIRLING

Department of Computer Science, University of Edinburgh, The King's Buildings, EDINBURGH, U.K. EH9 3JZ

Abstract. We present a sound and complete tableau proof system for establishing whether a set of elements of an arbitrary transition system model has a property expressed in (a slight extension of) the modal mu-calculus. The proof system, we believe, offers a very general verification method applicable to a wide range of computational systems.

1. Introduction

In the last twenty years many approaches to program verification have been developed. Hoare's partial correctness logic for simple **while** programs gave an early sound and relatively complete proof system. This approach was subsequently extended to total correctness and to richer classes of programs. Dynamic logics offered a more abstract view of Hoare logics, especially in their propositional versions.

Pnueli pioneered the use of propositional temporal logics as more general program logics, capable of describing crucial properties of perpetual concurrent systems. A variety of temporal logics have been studied, particularly branching and linear time. Many useful decidability and expressiveness results (relating logics and automata) have been obtained, as well as sound and complete axiomatizations of validity.

A slightly earlier tradition in the study of correctness was given by the work on program schemes where second order logics were advocated, especially in the form of the mu-calculus due to de Bakker, de Roever and Park. An elegant generalization of propositional dynamic and temporal logics drawing on this tradition is the *propositional modal mu-calculus*, due to Pratt and Kozen. The modal mu-calculus has been shown to include Propositional Dynamic Logic, Process Logic, linear time temporal logic, TL, as well as the branching time computation tree logics CTL and CTL* ([7], [4]). It also generalizes Hennessy–Milner logic, while preserving the characterization of bisimulation equivalence, and thereby provides a natural temporal logic for process theory ([8], [12]). In fact, it is closely related in expressive power to Rabin's SnS ([6]) despite having an elementary decision procedure ([15]). Consequently, the modal mu-calculus can be viewed as a *general purpose* program logic.

A hallmark of modal and temporal logics is that their primary truth definition relates elements of a model (states, runs, or whatever) and formulae. But when these logics are applied to reason about programs it is common to abstract from this relative truth. Both Hoare logic and PDL are based on truth *simpliciter*, that is, truth at every element in the model or structure. Moreover, in temporal logic approaches, as advocated in [9], models are dispensed with by coding them in the logic as theories: verifying that elements of a model have a crucial property reduces to showing that it is formally derivable within the appropriate theory. This abstraction, however, is not adopted by the model checking method, as pioneered by Clarke, Emerson and Sistla. This approach, extended to the modal mu-calculus in [4], hinges on constructing algorithms for computing *all* the states of the finite model which have the relevant property. But as a verification method, it can not (and is not intended to) cope with arbitrary programs whose state space is more likely than not to be infinite.

Logically, underpinning model checking is a tableau method for showing relative truth (rather than for establishing truth or validity as in [5], for instance). [11] advocates this as an appropriate verification technique for process theory where compositional proof systems for modal logic are presented. It was extended to the modal mu-calculus in [13]. Both of these capture a notion of *local* model checking, focussed on establishing whether particular elements of a model have a property. A clear advantage of this technique is that it may avoid that exhaustive traversal of a model inherent in standard model checking. It also naturally permits the use of other techniques that may be specific to the program under consideration (such as, those offered by the algebraic theory of processes either via equations or via bisimulation).

In this paper we extend local model checking to arbitrary infinite models. We present a sound and complete tableau proof system for establishing whether a set of elements of an arbitrary transition system model has a property expressed in (a slight extension of) the modal mu-calculus. The proof system, we believe, offers a very general verification method applicable to a wide range of computational systems. Section 2 contains an account of the syntax and semantics of the mu-calculus. In section 3 we present the tableau proof system. The proof of soundness and completeness is the topic of section 5. Finally, section 4 briefly examines applications of the verification method including examples drawn from Petri nets and process theory. We also outline how standard Hoare logic is subsumed by the tableau proof system.

2. The modal mu-calculus

Many interesting computational systems may be conveniently modelled using labelled transition systems, structures $\mathcal{T} = (\mathcal{S}, \{ \xrightarrow{a} \mid a \in \mathcal{L} \})$ consisting of a possibly infinite set $\mathcal{S}$ (or $\mathcal{S}_\mathcal{T}$) of points together with a family of binary relations on $\mathcal{S}$ indexed by a set

$\mathcal{L}$ (or $\mathcal{L}_{\mathcal{T}}$) of labels, the sort of $\mathcal{T}$. Examples include (concurrent) imperative programs; Petri nets; and CCS (or CSP) processes with value passing. Transition systems are generated from imperative programs when they are accorded structured operational interpretations. The points consist of programs and state information and the labels actions which may, for instance, also indicate an ability to communicate. In the case of Petri nets the points are markings and the labels are (families of) events, while in CCS the points are processes and the labels are actions. Usually, instead of the possible transitions from one point (state, configuration, marking, or process) to another, it is the overall behaviour of the system which is of interest. This behaviour arises from the runs of the system, a run being a maximal path $E_0 \xrightarrow{a_0} E_1 \xrightarrow{a_1} \cdots$ through $\mathcal{T}$, where maximality means that either the path has infinite length, or from its final point no transition is possible.

A very rich and succinct logic for expressing program behaviour within transition systems is a slight extension of the modal mu-calculus, whose syntax is:

$$\Phi \ ::= \ Z \bigm| \neg\Phi \bigm| \Phi_1 \wedge \Phi_2 \bigm| [K]\Phi \bigm| \nu Z.\Phi$$

where Z ranges over propositional variables, and K over subsets of a label set $\mathcal{L}$. In the formula $\nu Z.\Phi$, $\nu Z.$ binds free occurrences of Z in Φ. A syntactic restriction on $\nu Z.\Phi$ is that each free occurrence of Z in Φ lies within the scope of an even number of negations. The modal mu-calculus is due to Kozen [7] and Pratt [10]†. The slight extension here is that sets of labels may appear in the modalities $[K]$ instead of single labels.

Derived operators are defined in the familiar way: $\mathrm{tt} \stackrel{\mathrm{def}}{=} \nu Z.Z$; $\mathrm{ff} \stackrel{\mathrm{def}}{=} \neg\mathrm{tt}$; $\Phi_1 \vee \Phi_2 \stackrel{\mathrm{def}}{=} \neg(\neg\Phi_1 \wedge \neg\Phi_2)$; $\langle K\rangle\Phi \stackrel{\mathrm{def}}{=} \neg[K]\neg\Phi$; and $\mu Z.\Phi \stackrel{\mathrm{def}}{=} \neg\nu Z.\neg\Phi[Z := \neg Z]$ where $\Phi[Z := \neg Z]$ is the result of substituting $\neg Z$ for every free occurrence of Z in Φ. Further useful abbreviations are $[-K]\Phi \stackrel{\mathrm{def}}{=} [\mathcal{L} - K]\Phi$; $[a_1, \ldots, a_n]\Phi \stackrel{\mathrm{def}}{=} [\{a_1, \ldots, a_n\}]\Phi$; $[-]\Phi \stackrel{\mathrm{def}}{=} [\mathcal{L}]\Phi$; and similarly for $\langle K\rangle$.

Formulae of this logic are interpreted on labelled transition systems (of sort $\mathcal{L}$) as follows. A model is a pair $(\mathcal{T}, \mathcal{V})$ where $\mathcal{T}$ is a transition system and $\mathcal{V}$ a valuation assigning sets of points to propositional variables: $\mathcal{V}(Z) \subseteq \mathcal{S}_{\mathcal{T}}$. We assume the customary updating notation: $\mathcal{V}[\mathcal{E}/Z]$ is the valuation $\mathcal{V}'$ which agrees with $\mathcal{V}$ except that $\mathcal{V}'(Z) = \mathcal{E}$. Finally, the set of points of $\mathcal{T}$ having the property Φ in the model $(\mathcal{T}, \mathcal{V})$ is inductively defined as $\|\Phi\|^{\mathcal{T}}_{\mathcal{V}}$ (where for ease of notation we drop the index $\mathcal{T}$ which is assumed to be fixed):

$$\|Z\|_{\mathcal{V}} = \mathcal{V}(Z)$$
$$\|\neg\Phi\|_{\mathcal{V}} = \mathcal{S} - \|\Phi\|_{\mathcal{V}}$$

† Pratt introduces a least root (as in recursive function theory) rather than a least fix-point.

$$\|\Phi_1 \wedge \Phi_2\|_{\mathcal{V}} = \|\Phi_1\|_{\mathcal{V}} \cap \|\Phi_2\|_{\mathcal{V}}$$
$$\|[K]\Phi\|_{\mathcal{V}} = \{\, E \in \mathcal{S} \mid \forall F \in \mathcal{S} \,.\, \forall a \in K \,.\, \text{if } E \xrightarrow{a} F \text{ then } F \in \|\Phi\|_{\mathcal{V}} \,\}$$
$$\|\nu Z.\Phi\|_{\mathcal{V}} = \bigcup \{\, \mathcal{E} \subseteq \mathcal{S} \mid \|\Phi\|_{\mathcal{V}[\mathcal{E}/Z]} \supseteq \mathcal{E} \,\}$$

The expected clause for the derived operator $\mu Z.\Phi$ is:

$$\|\mu Z.\Phi\|_{\mathcal{V}} = \bigcap \{\, \mathcal{E} \subseteq \mathcal{S} \mid \|\Phi\|_{\mathcal{V}[\mathcal{E}/Z]} \subseteq \mathcal{E} \,\}$$

This logic allows the expression of a very wide range of temporal properties. The formula $\nu Z.\neg\Phi \wedge [-]Z$ (assuming that Φ does not contain Z free) expresses that Φ never becomes true: a point E has this property (in a model) provided that there is no run from E containing a point with the property Φ. Similarly, $\nu Z.[K]\mathrm{ff} \wedge [-]Z$ expresses that a K action can never happen. In contrast, $\mu Z.\Phi \vee ([-]Z \wedge \langle - \rangle \mathrm{tt})$ (assuming again that Φ does not contain Z free) expresses the strong eventuality that Φ must eventually become true. And $\mu Z.[-K]Z \wedge \langle - \rangle \mathrm{tt}$ expresses that K actions must eventually *happen*: a point E fails to have it as a property if there is an infinite length run from E all of whose labels are drawn from $-K$.

By using nested fix-points, it is possible to give much more complicated liveness properties, ranging from

$$\nu Y.\mu Z.((\Phi \wedge \langle - \rangle Y) \vee \langle - \rangle Z)$$

which expresses that Φ holds infinitely often on some run of the system, to the complex fairness requirement (from [14])

$$\nu X_1.([-a]X_1) \wedge (\mu Y.[K_1](\nu X.([-a]X) \wedge ([K_2]\nu Y_1.([-a]Y_1) \wedge Y)))$$

which says that the action a eventually happens on any fair run involving K_1 actions and K_2 actions infinitely often, a property not expressible in standard branching time logics such as CTL.

As mentioned earlier, the modal mu-calculus incorporates PDL and CTL. A translation for PDL may be found in [7]; a translation for CTL is:

$$\mathbf{AX}\Phi = [-]\Phi \wedge \langle - \rangle \mathrm{tt}$$
$$\mathbf{EX}\Phi = \langle - \rangle \Phi$$
$$\mathbf{A}[\Phi_1 \,\mathbf{U}\, \Phi_2] = \mu Z.\Phi_2 \vee (\Phi_1 \wedge [-]Z \wedge \langle - \rangle \mathrm{tt})$$
$$\mathbf{E}[\Phi_1 \,\mathbf{U}\, \Phi_2] = \mu Z.\Phi_2 \vee (\Phi_1 \wedge \langle - \rangle Z)$$

Similarly, the mu-calculus subsumes Process Logic, linear time logic, LT, and the full branching and linear time logic CTL* ([4],[3]).

3. The tableau system

Assume a fixed mu-calculus model $(\mathcal{T}, \mathcal{V})$ where $\mathcal{T}$ is a transition system which may contain an infinite number of points. We wish to provide a technique for verifying that a set of points $\mathcal{E}$ has the temporal property Φ, when $\mathcal{E} \subseteq \|\Phi\|^{\mathcal{T}}_{\mathcal{V}}$. Below a tableau proof system for verification is presented, built on sequents of the form $\mathcal{E} \vdash_\Delta \Phi$ where Δ is a *definition list* which keeps track of unrolling of fix-point formulae. A definition has the form $U = \Psi$ where U is a propositional constant and Ψ a fix-point formula. A finite list of definitions $(U_1 = \Psi_1, \ldots, U_n = \Psi_n)$ has the two properties: first, that each U_i is distinct and second, that each Ψ_i only mentions propositional constants belonging to the set $\{U_1, \ldots, U_{i-1}\}$. Lists of definitions can be extended: if U is not declared in Δ and Ψ only mentions constants declared in Δ then $\Delta \cdot U = \Psi$ is the definition list that results from appending $U = \Psi$ to Δ. When Δ is a definition list and U is declared to be Ψ in Δ then we let $\Delta(U) = \Psi$.

We assume the interpretation of formulae relative to definition lists as in [13] by, in effect, treating constants as variables: if Δ is $(U_1 = \Psi_1, \ldots, U_n = \Psi_n)$ then $\|\Phi_\Delta\|^{\mathcal{T}}_{\mathcal{V}}$ is defined as $\|\Phi\|^{\mathcal{T}}_{\mathcal{V}_n}$ where $\mathcal{V}_0 = \mathcal{V}$ and $\mathcal{V}_{i+1} = \mathcal{V}_i[\|\Psi_{i+1}\|^{\mathcal{T}}_{\mathcal{V}_i}/U_{i+1}]$. This interpretation accords with the expected meaning of Φ_Δ in terms of syntactic substitution: a routine induction on Φ establishes that $\|\Phi_{\Delta \cdot U = \Psi}\|^{\mathcal{T}}_{\mathcal{V}} = \|\Phi[U := \Psi]_\Delta\|^{\mathcal{T}}_{\mathcal{V}}$. (Semantically, restricting $\Delta(U)$ to be a fix-point formula is unnecessary—in section 5 we permit the declaration of other formulae.)

The rules of the tableau system below are inverse natural deduction style rules. The premise sequent is the goal to be achieved (that $\mathcal{E} \subseteq \|\Phi_\Delta\|^{\mathcal{T}}_{\mathcal{V}}$) while the consequents are the subgoals. The rules are presented only for formulae in positive form (where all negations are moved inwards by using the dual operators $\vee$, $\langle K \rangle$ and $\mu Z.$). We assume that σ ranges over $\{\mu, \nu\}$. In the rule for $\langle K \rangle$ we assume that f is a function from the set $\mathcal{E}$ to the set $f(\mathcal{E})$ such that $\forall E \in \mathcal{E} \,.\, \exists a \in K \,.\, E \xrightarrow{a} f(E)$.

$\wedge$	$\dfrac{\mathcal{E} \vdash_\Delta \Phi_1 \wedge \Phi_2}{\mathcal{E} \vdash_\Delta \Phi_1 \qquad \mathcal{E} \vdash_\Delta \Phi_2}$	
$\vee$	$\dfrac{\mathcal{E} \vdash_\Delta \Phi_1 \vee \Phi_2}{\mathcal{E}_1 \vdash_\Delta \Phi_1 \qquad \mathcal{E}_2 \vdash_\Delta \Phi_2}$	$\mathcal{E} = \mathcal{E}_1 \cup \mathcal{E}_2$
$[K]$	$\dfrac{\mathcal{E} \vdash_\Delta [K]\Phi}{\mathcal{E}' \vdash_\Delta \Phi}$	$\mathcal{E}' = \{\, E \mid \exists F \in \mathcal{E} \,.\, \exists a \in K \,.\, F \xrightarrow{a} E \,\}$
$\langle K \rangle$	$\dfrac{\mathcal{E} \vdash_\Delta \langle K \rangle\Phi}{f(\mathcal{E}) \vdash_\Delta \Phi}$	
$\sigma Z.$	$\dfrac{\mathcal{E} \vdash_\Delta \sigma Z.\Phi}{\mathcal{E} \vdash_{\Delta'} U}$	$\Delta' = \Delta \cdot (U = \sigma Z.\Phi)$
Un	$\dfrac{\mathcal{E} \vdash_\Delta U}{\mathcal{E} \vdash_\Delta \Phi[Z := U]}$	$\Delta(U) = \sigma Z.\Phi$
Thin	$\dfrac{\mathcal{E} \vdash_\Delta \Phi}{\mathcal{E}' \vdash_\Delta \Phi}$	$\mathcal{E}' \supseteq \mathcal{E}$

To test if every point in $\mathcal{E}$ has the property Φ (relative to Δ) one has to achieve the goal $\mathcal{E} \vdash_\Delta \Phi$ by building a tableau, a proof tree whose root is labelled with this initial sequent. Sequents labelling the immediate successors of a node are determined by an application of one of the rules. The boolean and modal rules are straightforward. New constants are introduced when fix-point formulae are met, and then these are unfolded by rule Un. The rule Thin allows the set of points to be enlarged (and need only be applied to sequents whose formula is a constant—see the completeness proof and definition of canonical tableaux).

An essential missing ingredient is when a node in a proof-tree counts as a leaf. We assume that the rules above only apply to nodes that are not terminal. A node $\mathbf{n}$ labelled by the sequent $\mathcal{F} \vdash_\Delta \Psi$ is terminal if one of the following conditions holds:

(i) $\mathcal{F} = \emptyset$

(ii) $\Psi = Z$ or $\Psi = \neg Z$

(iii) $\Psi = \langle K \rangle \Phi$ and $\exists F \in \mathcal{F} . \forall a \in K . \operatorname{not}(F \xrightarrow{a})$

(iv) $\Psi = U$ and $\Delta(U) = \sigma Z.\Phi$ and there is a node above $\mathbf{n}$ in the proof tree labelled $\mathcal{E} \vdash_{\Delta'} U$ with $\mathcal{E} \supseteq \mathcal{F}$

A node fulfilling condition (iv) is called a *σ-terminal*. A node fulfilling conditions (i) or (iv) when $\sigma = \nu$ is said to be a *successful* terminal, whereas a node fulfilling (iii) is unsuccessful. In the case of (ii) success depends on the valuation $\mathcal{V}$ of the model: if $\mathcal{F} \subseteq \mathcal{V}(Z)$ when Ψ is Z, or $\mathcal{F} \subseteq \mathcal{S} - \mathcal{V}(Z)$ when Ψ is $\neg Z$, then it is successful; otherwise it is unsuccessful. The definition of a successful μ-terminal, a σ-terminal when $\sigma = \mu$, is intricate and requires some notation.

Suppose a node $\mathbf{n}$ is labelled by $\mathcal{E} \vdash_\Delta \Phi$ and $\mathbf{n}'$ labelled $\mathcal{E}' \vdash_{\Delta'} \Phi'$ is an immediate successor of $\mathbf{n}$. We say that $E' \in \mathcal{E}'$ at $\mathbf{n}'$ is a *dependant* of $E \in \mathcal{E}$ at $\mathbf{n}$ if

- the rule applied to $\mathbf{n}$ is $\vee$, $\wedge$, $\sigma Z.$, Un or Thin and $E = E'$, or
- the rule is $[K]$ and $E \xrightarrow{a} E'$ for some $a \in K$, or
- the rule is $\langle K \rangle$ and $E' = f(E)$.

Assume that the *companion* of a σ-terminal is the lowest node above it which makes it a terminal†. Next we define a *trail* to F at a σ-terminal $\mathbf{m}$ from E at its companion $\mathbf{n}$ to be a sequence of pairs of nodes and points $(\mathbf{n}_1, E_1), \ldots, (\mathbf{n}_k, E_k)$ with $(\mathbf{n}_1, E_1) = (\mathbf{n}, E)$ and $(\mathbf{n}_k, E_k) = (\mathbf{m}, F)$ such that for all i with $1 \leq i < k$ either

(i) E_{i+1} at $\mathbf{n}_{i+1}$ is a dependant of E_i at $\mathbf{n}_i$, or

(ii) $\mathbf{n}_i$ is the immediate predecessor of a σ-terminal node $\mathbf{n}'$ (where $\mathbf{n}' \neq \mathbf{m}$) whose companion is $\mathbf{n}_j$ for some $j \leq i$, and $\mathbf{n}_{i+1} = \mathbf{n}_j$ and E_{i+1} is a point at $\mathbf{n}'$ (and so at $\mathbf{n}_{i+1}$) which is a dependant of E_i at $\mathbf{n}_i$.

† As can be seen from the proof later, it is not necessary to choose the lowest such node—any suffices. However, the lower the companion the simpler the termination condition, so we decree that the companion is the lowest.

Then each companion node $\mathbf{n}$ of a σ-terminal node induces an ordering $\sqsubset_{\mathbf{n}}$ on its point set $\mathcal{E}$ by $F \sqsubset_{\mathbf{n}} E$ if there is a trail to F at a node $\mathbf{m}$ from E at its companion $\mathbf{n}$. (Notice that a node may be a companion of various σ-terminal nodes, and that $F \in \mathcal{E}$ by definition of companion). A μ-terminal node is *successful* if the ordering induced by its companion node $\mathbf{n}$ is well-founded: that is, if there is no infinite descending chain

$$\cdots \sqsubset_{\mathbf{n}} E_2 \sqsubset_{\mathbf{n}} E_1 \sqsubset_{\mathbf{n}} E_0.$$

To illustrate this definition, consider the following tableau (for a system $(\mathcal{T}, \mathcal{V})$ where $\mathcal{T} = (\{A, B, C\}, \{\xrightarrow{a}\})$ and $A \xrightarrow{a} A$, $A \xrightarrow{a} B \xrightarrow{a} C$ and $C \xrightarrow{a} C$, and $\mathcal{V}(P) = \{A, C\}$):

$$\dfrac{\{A,B,C\} \vdash_{()} \mu Y.\nu Z.(P \wedge [-]Z) \vee [-]Y}{{}_1\,\{A,B,C\} \vdash_{\Delta} U} \quad \Delta = (U = \mu Y. \ldots)$$

$$\dfrac{{}_1\,\{A,B,C\} \vdash_{\Delta} U}{\dfrac{\{A,B,C\} \vdash_{\Delta} \nu Z.(P \wedge [-]Z) \vee [-]U}{{}_2\,\{A,B,C\} \vdash_{\Delta'} V}} \quad \Delta' = \Delta \cdot (V = \nu Z. \ldots)$$

$$\dfrac{{}_2\,\{A,B,C\} \vdash_{\Delta'} V}{\dfrac{\{A,B,C\} \vdash_{\Delta'} (P \wedge [-]V) \vee [-]U}{\dfrac{\{A,C\} \vdash_{\Delta'} (P \wedge [-]V)}{\{A,C\} \vdash_{\Delta'} P \qquad {}_3\,\dfrac{\{A,C\} \vdash_{\Delta'} [-]V}{{}_4\,\{B,C\} \vdash_{\Delta'} V}} \qquad \dfrac{\{B\} \vdash_{\Delta'} [-]U}{{}_5\,\{C\} \vdash_{\Delta'} U}}}$$

We need to consider $\sqsupset_1$. Now $B \sqsupset_1 C$ because of the simple trail marked in solid lines; but also $A \sqsupset_1 C$ because of the trail marked in dashed lines, in which we go from $(1, A)$ to $(3, A)$ and then invoke (ii) to jump up to $(2, B)$ and then go down to $(5, C)$ along the solid trail.

Finally, we say that a tableau is *successful* if it is finite and all its leaves are successful terminals. The following theorem states that the tableau technique is both *sound* and *complete* for arbitrary (infinite) transition systems. The proof is presented in section 5.

Theorem 3.1. *$\mathcal{E} \vdash_{\Delta} \Phi$ is the root of a successful tableau on $(\mathcal{T}, \mathcal{V})$ iff $\mathcal{E} \subseteq \|\Phi_{\Delta}\|^{\mathcal{T}}_{\mathcal{V}}$.* $\square$

The tableau rules presented above are in fact more general than is needed for Theorem 3.1 to hold. As mentioned earlier, the Thin rule, while essential for completeness, is only required for thinning constants, and this could be incorporated into the rules by omitting Thin and replacing Un by the three-stage rule

$$\text{Un} \quad \dfrac{\dfrac{\mathcal{E} \vdash_{\Delta} U}{\mathcal{E}' \vdash_{\Delta} U} \;\; \mathcal{E}' \supseteq \mathcal{E}}{\mathcal{E}' \vdash_{\Delta} \Phi[Z := U]} \quad \Delta(U) = \sigma Z.\Phi$$

Another restriction arises from the desire to ensure that tableaux are finite: while the system as presented is intended to allow the user flexibility in developing proofs, it does not guarantee that the tableau construction terminates. However, it is the case that an infinite tableau can only arise from infinitely many unfoldings of some constant, so one can consider restricting to k, say, the number of applications of Un to any one constant, which can be included in the rules by adding

(v) $\Psi = U$ and there are k applications above $\mathbf{n}$ of the Un rule to nodes labelled by sequents of the form $\mathcal{E}' \vdash_{\Delta'} U$

to the conditions for $\mathcal{F} \vdash_\Delta \Psi$ to be terminal.

Such a tableau is said to be of *degree* k, and since the proof of Theorem 3.1 in section 5 gives completeness by constructing a *canonical* tableau of degree 1, we have

Theorem 3.2. *The system restricted to degree k is sound and complete, and its tableaux are finite.* $\square$

Restricting k to 1 gives a system in which constants can be dispensed with (see [1] for such an account); but, of course, any complete system is undecidable: canonical tableaux are in general neither unique nor effectively constructible—hence the use of reasoning outside the system proper to show success.

A different restriction allows us to recapture the model checker of [13] (for finite state systems) by omitting the rule Thin, limiting the sets of elements $\mathcal{E}$ of sequents to singleton sets, and then replacing the $[K]$ rule with

$$[K]' \quad \frac{\{E\} \vdash_\Delta [K]\Phi}{\{E_1\} \vdash_\Delta \Phi \quad \cdots \quad \{E_n\} \vdash_\Delta \Phi} \quad \{E_1, \ldots, E_n\} = \{\, F \mid \exists a \in K \,.\, E \xrightarrow{a} F \,\}$$

Note that now all μ-terminals are unsuccessful as there must be a trail from E at the companion $\mathbf{n}$ to itself at a terminal, giving a cyclic $\sqsubset_{\mathbf{n}}$.

4. Applications

In this section we illustrate the generality of the tableau proof system by applying it to disparate examples. The first example, given in more detail in [2], is a CCS representation of a slot-machine, for which we shall prove that it is possible to win a million pounds infinitely often. The slot machine is the process SM_0, defined by the following equations:

$$
\begin{aligned}
IO &\stackrel{\text{def}}{=} slot.\overline{bank}.(lost.\overline{loss}.IO + release(y).\overline{win}(y).IO) \\
B_n &\stackrel{\text{def}}{=} bank.\overline{max}(n+1).left(y).B_y \\
D &\stackrel{\text{def}}{=} max(z).\Big(\overline{lost}.\overline{left}(z).D + \sum_{1\le y\le z} \overline{release}(y).\overline{left}(z-y).D\Big) \\
SM_n &\stackrel{\text{def}}{=} (IO|B_n|D)\backslash K
\end{aligned}
$$

where $K = \{\, bank(v), max(v), left(v), release(v) \mid v \in \mathbb{N}\,\}$. The parameter n represents the amount of money in the machine's bank, and the three components IO, B_n, D handle respectively taking in and paying out money, the bank's holdings, and deciding the payout.

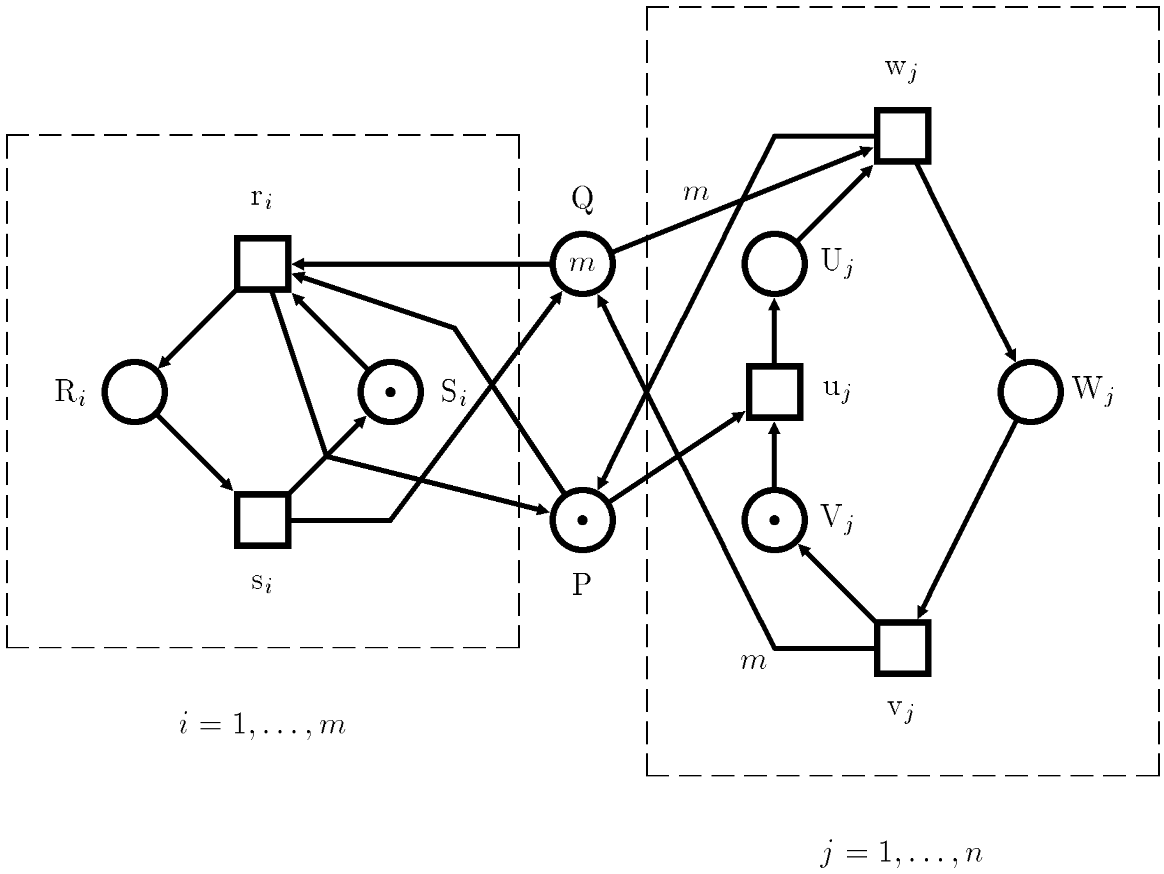
The property of SM_0 we wish to prove is

$$\nu Y.\mu Z.(\langle\overline{win}(10^6)\rangle Y) \vee \langle -\rangle Z$$

A successful canonical tableau for this is presented in figure 1 where () is the empty definition list. Here, $\mathcal{E}$ is the set of derivatives of $\{\, SM_n \mid n \ge 0\,\}$; the vital rules in this tableau are the disjunction at node 1, where $\mathcal{E}_1$ is exactly those processes capable of performing a $\overline{win}(10^6)$ action, and $\mathcal{E}_2$ is the remainder; and the $\langle K\rangle$ rule at node 3, where f is defined to ensure that $\mathcal{E}_1$ is eventually reached: for processes with less than 10^6 in the bank, f chooses events leading towards loss, so as to increase the amount in the bank; and for processes with more than 10^6, f chooses to $\overline{release}(10^6)$. The formal proof requires partitioning $\mathcal{E}_2$ into several classes, each parametrized by an integer n, and showing that while $n < 10^6$, n is strictly increasing over a cycle through the classes; then when $n = 10^6$, f selects a successor that is not in $\mathcal{E}_2$, and so a chain $E_0 \xrightarrow{a_0} \cdots$ through nodes 1, 2, 3, 4 terminates, and therefore $\sqsubset_4$ is well-founded.

$$
\dfrac{\{SM_0\} \vdash_{()} \nu Y.\mu Z.\langle\overline{win}(10^6)\rangle Y \vee \langle -\rangle Z}
{\dfrac{\{\, SM_n \mid n \ge 0\,\} \vdash_{()} \nu Y.\mu Z.\langle\overline{win}(10^6)\rangle Y \vee \langle -\rangle Z}
{\dfrac{\mathcal{E} \vdash_{()} \nu Y.\mu Z.\langle\overline{win}(10^6)\rangle Y \vee \langle -\rangle Z}
{\dfrac{\mathcal{E} \vdash_{\Delta} U}
{\dfrac{\mathcal{E} \vdash_{\Delta} \mu Z.\langle\overline{win}(10^6)\rangle U \vee \langle -\rangle Z}
{\dfrac{{}_1\,\mathcal{E} \vdash_{\Delta'} V}
{\dfrac{{}_2\,\mathcal{E} \vdash_{\Delta'} \langle\overline{win}(10^6)\rangle U \vee \langle -\rangle V}
{\dfrac{\mathcal{E}_1 \vdash_{\Delta'} \langle\overline{win}(10^6)\rangle U}{\mathcal{E}_1' \vdash_{\Delta'} U} \qquad \dfrac{{}_3\,\mathcal{E}_2 \vdash_{\Delta'} \langle -\rangle V}{{}_4\,\mathcal{E}_2' \vdash_{\Delta'} V}}}
\;\Delta'=\Delta\cdot(V=\mu Z.\langle\overline{win}(10^6)\rangle U \vee \langle -\rangle Z)}}
\;\Delta=()\cdot(U=\nu Y.\mu Z.\langle\overline{win}(10^6)\rangle Y \vee \langle -\rangle Z)}}
$$

Figure 1



Legend

For m reader processes		and n writer processes	
S$_i$	process not reading	V$_j$	process not writing
r$_i$	process starts reading	w$_j$	process starts writing
R$_i$	process reading	W$_j$	process writing
s$_i$	process stops reading	v$_j$	process stops writing
		u$_j$	process requests write access
		U$_j$	process waiting to write
Q	the resource		
P	an interlock		

Figure 2

$$
\begin{array}{c}
\{M_0\} \vdash_{()} \nu Y.[-]Y \land [\mathrm{u}_{j_0}](\mu Z.(\mathrm{W}_{j_0} = 1) \lor [-]Z) \\ \hline
\{\,\Phi\,\} \vdash_{()} \nu Y.[-]Y \land [\mathrm{u}_{j_0}](\mu Z.(\mathrm{W}_{j_0} = 1) \lor [-]Z) \\ \hline
\{\,\Phi\,\} \vdash_{\Delta} U \qquad \Delta=(U=\nu Y.[-]Y \land [\mathrm{u}_{j_0}](\mu Z.(\mathrm{W}_{j_0}=1)\lor[-]Z)) \\ \hline
\{\,\Phi\,\} \vdash_{\Delta} [-]U \land [\mathrm{u}_{j_0}](\mu Z.(\mathrm{W}_{j_0} = 1) \lor [-]Z) \\ \hline
\begin{array}{c}
\{\,\Phi\,\} \vdash_{\Delta} [-]U \\ \hline
{}_1\,\{\,\Phi \land \Phi'\,\} \vdash_{\Delta} U
\end{array}
\qquad
\begin{array}{c}
\{\,\Phi\,\} \vdash_{\Delta} [\mathrm{u}_{j_0}](\mu Z.(\mathrm{W}_{j_0} = 1) \lor [-]Z) \\ \hline
\{\,\Phi \land \Psi\,\} \vdash_{\Delta} \mu Z.(\mathrm{W}_{j_0} = 1) \lor [-]Z \\ \hline
\{\,\Phi \land \Psi\,\} \vdash_{\Delta'} V \qquad \Delta'=\Delta\cdot(U=\mu Z.(\mathrm{W}_{j_0}=1)\lor[-]Z) \\ \hline
{}_2\,\{\,\Phi \land \Psi'\,\} \vdash_{\Delta'} V \\ \hline
\{\,\Phi \land \Psi'\,\} \vdash_{\Delta'} (\mathrm{W}_{j_0} = 1) \lor [-]V
\end{array} \\ \hline
\{\,\Phi \land \neg\Psi \land (\mathrm{W}_{j_0} = 1)\,\} \vdash_{\Delta'} (\mathrm{W}_{j_0} = 1)
\qquad
\begin{array}{c}
{}_3\,\{\,\Phi \land \Psi \land (\mathrm{W}_{j_0} = 0)\,\} \vdash_{\Delta'} [-]V \\ \hline
{}_4\,\{\,\Phi \land (\Psi' \lor (\mathrm{W}_{j_0} = 1))\,\} \vdash_{\Delta'} V
\end{array}
\end{array}
$$

Figure 3

The second example is taken from [1], and is a Petri net system implementing multiple-read–single-write interlock. The net is shown in figure 2. We prove a strong liveness property, that any process that wants to write, eventually will write. Regarding the net as a transition system whose points are markings M, and whose transitions are generated by the firing of a (single) net-transition, this property is

$$\nu Y.[-]Y \land [\mathrm{u}_{j_0}](\mu Z.(\mathrm{W}_{j_0} = 1) \lor ([-]Z \land \langle - \rangle \mathrm{tt})$$

for a particular writer j_0, where $(\mathrm{W}_{j_0} = 1)$ is an atomic formula which is true at any marking in which the place W_{j_0} holds one token, and similarly for other such formulae. In figure 3 we give a successful tableau to show that this property is true at the initial marking M_0 shown in figure 2; the notation $\{\Phi\}$ means the set of markings satisfying Φ. In this tableau, Φ is an invariant of the net, given by

$$
\begin{aligned}
\Phi \stackrel{\mathrm{def}}{=} & \bigwedge_j (\mathrm{U}_j + \mathrm{V}_j + \mathrm{V}_j = 1) \\
\land & \bigwedge_i (\mathrm{R}_i + \mathrm{S}_i = 1) \\
\land & \, (\mathrm{Q} + m \sum_j \mathrm{W}_j + \sum_i \mathrm{R}_i = m) \\
\land & \, (\mathrm{P} + \sum_j \mathrm{U}_j = 1)
\end{aligned}
$$

and $\Psi \stackrel{\mathrm{def}}{=} (\mathrm{U}_{j_0} = 1) \land (\mathrm{P} = 0)$ and $\Psi' \stackrel{\mathrm{def}}{=} \Psi \lor (\mathrm{W}_{j_0} = 1)$.

The success of the tableau requires the inclusion condition to hold at nodes 1 and 4, which it does, and the relation $\sqsubset_2$ to be well-founded: for this, suppose that $M \sqsupset_2 M' \sqsupset_2 M''$. Then we have that $\mathrm{P} = 0$ and $\mathrm{U}_{j_0} = 1$ and $\mathrm{W}_{j_0} = 1$ for both M and M' (from node 3, since M goes to M' via node 3). Combining these conditions with the invariants in Φ tells us that the only transitions that may fire to give M' from M are v_j and s_i, for any i, j; now if we assign a non-negative measure δ to markings by $\delta(M) = \sum_i \mathrm{R}_i + m \sum_j \mathrm{V}_j$, we see that $\delta(M') < \delta(M)$, and so $\sqsubset_2$ is well-founded.

Finally, we sketch how Floyd–Hoare methods can be translated into the tableau system. Consider a simple **while** language, with atomic programs a, and the constructors $c_1;\, c_2$, **if** b **then** c_1 **else** c_2, and **while** b **do** c. Taking memories as points and transitions to be labelled by atomic programs, we can translate Hoare assertions of the form $c\{\Psi\}$ (by which we mean the weakest precondition for c wrt Ψ) into the mu-formulae $\mathrm{Tr}(c\{\Psi\})$ thus:

$$
\begin{aligned}
\mathrm{Tr}(a\{\Psi\}) &= [a]\Psi \\
\mathrm{Tr}(c_1;\, c_2\{\Psi\}) &= \mathrm{Tr}(c_1\{\mathrm{Tr}(c_2\{\Psi\})\}) \\
\mathrm{Tr}(\textbf{if } b \textbf{ then } c_1 \textbf{ else } c_2\{\Psi\}) &= (b \wedge \mathrm{Tr}(c_1\{\Psi\})) \vee (\neg b \wedge \mathrm{Tr}(c_2\{\Psi\})) \\
\mathrm{Tr}(\textbf{while } b \textbf{ do } c\{\Psi\}) &= \nu Z.(\neg b \wedge \Psi) \vee (b \wedge \mathrm{Tr}(c\{Z\}))
\end{aligned}
$$

Proving a partial correctness assertion $\{\Phi\}c\{\Psi\}$ then amounts to finding a tableau for $\{\,\Phi\,\} \vdash_\Delta \mathrm{Tr}(c\{\Psi\})$ (where $\{\,\Phi\,\}$ as before means the set of points satisfying Φ). The Hoare rules can then be viewed as rules for generating tableaux. As an illustration, consider the Hoare rule for **while** loops:

$$
\frac{\{\mathrm{I} \wedge b\}c\{\mathrm{I}\}}{\{\mathrm{I}\}\textbf{while } b \textbf{ do } c\{\mathrm{I} \wedge \neg b\}}
$$

Associated with it is a successful tableau of the form:

$$
\dfrac{\dfrac{\dfrac{\{\,\mathrm{I}\,\} \vdash_{()} \nu Z.(\neg b \wedge \mathrm{I}) \vee (b \wedge \mathrm{Tr}(c\{Z\}))}{\{\,\mathrm{I}\,\} \vdash_\Delta U}\ \Delta{=}(U{=}\nu Z.(\neg b \wedge \mathrm{I}) \vee (b \wedge \mathrm{Tr}(c\{Z\})))}{\{\,\mathrm{I}\,\} \vdash_\Delta (\neg b \wedge \mathrm{I}) \vee (b \wedge \mathrm{Tr}(c\{U\}))}}{\dfrac{\{\,\mathrm{I} \wedge \neg b\,\} \vdash_\Delta \neg b \wedge \mathrm{I}}{\vdots} \qquad \dfrac{\{\,\mathrm{I} \wedge b\,\} \vdash_\Delta b \wedge \mathrm{Tr}(c\{U\})}{\{\,\mathrm{I} \wedge b\,\} \vdash_\Delta b \qquad \dfrac{\{\,\mathrm{I} \wedge b\,\} \vdash_\Delta \mathrm{Tr}(c\{U\})}{\dfrac{\cdot^{\cdot^{\cdot}}}{\{\,\mathrm{I}\,\} \vdash_{\Delta_1} U} \quad \begin{matrix}\vdots \\ \cdots\end{matrix} \quad \dfrac{\ddots}{\{\,\mathrm{I}\,\} \vdash_{\Delta_n} U}}}}
$$

The soundness of the Hoare rules is then reflected by the fact that this procedure builds successful tableaux. This extends to total correctness: then the translation for

while uses μ rather than ν, and Floyd's method of well-founded sets is exactly a method for showing well-foundedness of the $\sqsubset$ relations in the resulting tableaux.

In the case of partial correctness, Cook completeness of the Hoare rules follows from the requirement that there is always a formula expressing $\mathrm{Tr}(c\{\Psi\})$ for any c and Ψ. This raises the very intricate question of completeness of the tableau proof system relative to particular presentations of point sets, a topic which is currently being examined for the case of Petri nets, and which has scope for much further investigation. Other important topics for future research are the incorporation of modular reasoning techniques, and the use of system-specific techniques such as algebraic process theory.

5. Proof of soundness and completeness

In this section we present a proof of the main result of the paper. First, we recall the standard notion of ordinal approximants of fixed points. Let α range over ordinals, and λ over limit ordinals. We add infinitary disjunction and conjunction to the language and define the formula $\sigma^\alpha Z.\Phi$ as follows, :

$$\mu^0 Z.\Phi \stackrel{\text{def}}{=} \mathrm{ff} \qquad\qquad \nu^0 Z.\Phi \stackrel{\text{def}}{=} \mathrm{tt}$$
$$\sigma^{\alpha+1}.\Phi \stackrel{\text{def}}{=} \Phi[Z := \sigma^\alpha Z.\Phi]$$
$$\mu^\lambda.\Phi \stackrel{\text{def}}{=} \bigvee\{\, \mu^\alpha Z.\Phi \mid \alpha < \lambda \,\} \qquad \nu^\lambda.\Phi \stackrel{\text{def}}{=} \bigwedge\{\, \nu^\alpha Z.\Phi \mid \alpha < \lambda \,\}$$

The following proposition is a consequence of the Knaster–Tarski fixed point theorem, where $\mathrm{Ord}_{\mathcal{T}}$ is the set of ordinals whose cardinality is less than, or equal to, the cardinality of $\mathcal{S}_{\mathcal{T}}$, and where $\bigvee$ is interpreted as union $\bigcup$ and $\bigwedge$ as $\bigcap$ in the usual way:

Proposition 5.1.
(i) $\|\mu Z.\Phi_\Delta\|^{\mathcal{T}}_{\mathcal{V}} = \| \bigvee\{\, \mu^\alpha Z.\Phi_\Delta \mid \alpha \in \mathrm{Ord}_{\mathcal{T}} \,\}\|^{\mathcal{T}}_{\mathcal{V}}$
(ii) $\|\nu Z.\Phi_\Delta\|^{\mathcal{T}}_{\mathcal{V}} = \| \bigwedge\{\, \nu^\alpha Z.\Phi_\Delta \mid \alpha \in \mathrm{Ord}_{\mathcal{T}} \,\}\|^{\mathcal{T}}_{\mathcal{V}}$ $\square$

Next we introduce some terminology and a little notation. A constant U is *active* in Φ_Δ if either U occurs in Φ or if V occurs in Φ and U is active in $\Delta(V)$ for some constant V. The definition of a list Δ guarantees that this is non-circular. Moreover, there is at most a finite number of active constants in Φ_Δ, for any Φ and Δ. Suppose $U_1, \ldots, U_n$ with $n \geq 0$ are all the active ν-constants (μ-constants) in Φ_Δ in order of declaration (so U_i is declared before U_j in Δ when $i < j$). Then let $\Phi(U_1^{\alpha_1} \ldots U_n^{\alpha_n})_\Delta$ denote $\Phi_{\Delta'}$ where $\Delta'(V) = \Delta(V)$ when $V \notin \{U_1, \ldots, U_n\}$ and if $\Delta(U_i) = \sigma Z.\Psi$ then $\Delta'(U_i) = \sigma^{\alpha_i} Z.\Psi$. We say that $\alpha_1 \ldots \alpha_n$ is then a *ν-signature* (*μ-signature*) for Φ_Δ. We assume that $<$ is the lexicographical ordering on signatures. The following is an easy consequence of Proposition 5.1:

Proposition 5.2.

(i) if $E \notin \|\Phi_\Delta\|^{\mathcal{T}}_{\mathcal{V}}$ *then there exists a* ν*-signature* $\alpha_1 \dots \alpha_n$ *such that* $E \notin \|\Phi(U_1^{\alpha_1} \dots U_n^{\alpha_n})_\Delta\|^{\mathcal{T}}_{\mathcal{V}}$ *and for all* $\beta_1 \dots \beta_n < \alpha_1 \dots \alpha_n$, $E \in \|\Phi(U_1^{\beta_1} \dots U_n^{\beta_n})_\Delta\|^{\mathcal{T}}_{\mathcal{V}}$

(ii) if $E \in \|\Phi_\Delta\|^{\mathcal{T}}_{\mathcal{V}}$ *then there exists a* μ*-signature* $\alpha_1 \dots \alpha_m$ *such that* $E \in \|\Phi(U_1^{\alpha_1} \dots U_m^{\alpha_m})_\Delta\|^{\mathcal{T}}_{\mathcal{V}}$ *and for all* $\beta_1 \dots \beta_m < \alpha_1 \dots \alpha_m$, $E \notin \|\Phi(U_1^{\beta_1} \dots U_m^{\beta_m})_\Delta\|^{\mathcal{T}}_{\mathcal{V}}$ □

Notice that for both parts of this proposition, no α_i in the least signature $\alpha_1 \dots \alpha_n$ is a limit ordinal. In the sequel we abbreviate $\Phi(U_1^{\alpha_1} \dots U_n^{\alpha_n})_\Delta$ to $\Phi(\boldsymbol{U}^{\boldsymbol{\alpha}})_\Delta$ and $\alpha_1 \dots \alpha_n$ to $\boldsymbol{\alpha}$.

Given these preliminary results, we prove soundness:

Theorem 5.3. *If* $\mathcal{E} \vdash_\Delta \Phi$ *has a successful tableau on* $(\mathcal{T}, \mathcal{V})$ *then* $\mathcal{E} \subseteq \|\Phi_\Delta\|^{\mathcal{T}}_{\mathcal{V}}$.

Proof. From now on we drop the indices $\mathcal{T}$ and $\mathcal{V}$ from $\|\Phi_\Delta\|^{\mathcal{T}}_{\mathcal{V}}$ which we assume to be fixed. Suppose that $\mathcal{E} \vdash_\Delta \Phi$ has a successful tableau τ on $(\mathcal{T}, \mathcal{V})$ but that $\mathcal{E} \not\subseteq \|\Phi_\Delta\|$. We show that, therefore, there is an infinite sequence of the form:

$$\gamma = (\mathbf{n}_0, E_0, \Phi_0(\boldsymbol{U}_0^{\boldsymbol{\alpha}_0})_{\Delta_0}), \dots, (\mathbf{n}_k, E_k, \Phi_k(\boldsymbol{U}_k^{\boldsymbol{\alpha}_k})_{\Delta_k}), \dots$$

with the properties:

(i) each $\mathbf{n}_j$ is a node of τ and $\mathbf{n}_0$ is the root

(ii) $\mathbf{n}_{2j} = \mathbf{n}_{2j+1}$ and $\mathbf{n}_{2j+2}$ is an immediate successor of $\mathbf{n}_{2j}$ in τ

(iii) if $\mathbf{n}_j$ is labelled by $\mathcal{F} \vdash_\Sigma \Psi$ in τ then $E_j \in \mathcal{F}$, $\Sigma = \Delta_j$, and $\Psi = \Phi_j$

(iv) $E_j \notin \|\Phi_j(\boldsymbol{U}_j^{\boldsymbol{\alpha}_j})_{\Delta_j}\|$ where $\boldsymbol{\alpha}_j$ is a ν-signature

(v) each $\mathbf{n}_j$ is not a terminal in τ

But this is impossible as τ has finite depth. We now construct γ from τ, letting γ_i denote the ith member of γ.

First, we define γ_0. As $\mathcal{E} \not\subseteq \|\Phi_\Delta\|$ there is an $E_0 \in \mathcal{E}$ such that $E_0 \notin \|\Phi_\Delta\|$. By Proposition 5.2 there is a least ν-signature $\boldsymbol{\alpha}$ such that $E_0 \notin \|\Phi(\boldsymbol{U}^{\boldsymbol{\alpha}})_\Delta\|$. So $\gamma_0 = (\mathbf{n}_0, E_0, \Phi(\boldsymbol{U}^{\boldsymbol{\alpha}})_\Delta)$. Clearly conditions (i)–(iv) hold. Moreover, (v) must hold too. Otherwise $\mathcal{E} \vdash_\Delta \Phi$ is a successful terminal of τ and so $\Phi = Z$ or $\neg Z$ or $[K]\Psi$ which would contradict that $\mathcal{E} \not\subseteq \|\Phi_\Delta\|$.

Suppose γ_k has been constructed for $k \leq 2j$. γ_{2j+1} is defined as follows. First, if $\mathbf{n}_{2j}$ is *not* a companion in τ then $\gamma_{2j+1} = \gamma_{2j}$. Otherwise, $\Phi_{2j} = U$ and $\mathbf{n}_{2j}$ is a companion. Let $E^*_{2j} = \{ F \mid E_{2j} \sqsupset^*_{\mathbf{n}_{2j}} F \}$. If U is a ν-constant then let E_{2j+1} be any member of E^*_{2j} such that $E_{2j+1} \notin \|\Phi_{2j}(\boldsymbol{U}_{2j}^{\boldsymbol{\beta}})_{\Delta_{2j}}\|$ with least ν-signature $\boldsymbol{\beta}$. So γ_{2j+1} is the triple $(\mathbf{n}_{2j}, E_{2j+1}, \Phi_{2j}(\boldsymbol{U}_{2j}^{\boldsymbol{\beta}})_{\Delta_{2j}})$. Notice that $\boldsymbol{\beta} \leq \boldsymbol{\alpha}_{2j}$. If U is a μ-constant then let E_{2j+1} be any member of E^*_{2j} which is least with respect to $\sqsubset_{\mathbf{n}_{2j}}$ and which has the feature that $E_{2j+1} \notin \|\Phi_{2j}(\boldsymbol{U}_{2j}^{\boldsymbol{\alpha}_{2j}})_{\Delta_{2j}}\|$. There are such members as $\sqsubset_{\mathbf{n}_{2j}}$ is well-founded and

$E_{2j} \notin \|\Phi_{2j}(\boldsymbol{U}_{2j}^{\boldsymbol{\alpha}_{2j}})_{\Delta_{2j}}\|$. Let $\gamma_{2j+1} = (\mathbf{n}_{2j}, E_{2j+1}, \Phi_{2j}(\boldsymbol{U}_{2j}^{\boldsymbol{\alpha}_{2j}})_{\Delta_{2j}})$. In both these cases γ_{2j+1} fulfills properties (i)–(v).

Next we examine the construction for γ_{2j}, $j > 0$, given γ_k has been built when $k < 2j$, which is determined by the rule applied to $\mathbf{n}_{2j-1}$ in τ. If the rule is $\vee$ then $\Phi_{2j-1} = \Psi_1 \vee \Psi_2$. Hence $E_{2j-1} \notin \|\Psi_1(\boldsymbol{U}_1^{\boldsymbol{\alpha}_1})_{\Delta_{2j-1}}\|$ and $E_{2j-1} \notin \|\Psi_2(\boldsymbol{U}_2^{\boldsymbol{\alpha}_2})_{\Delta_{2j-1}}\|$ where $\boldsymbol{\alpha}_1$ ($\boldsymbol{\alpha}_2$) is the signature $\boldsymbol{\alpha}_{2j-1}$ restricted to the active ν-constants in Ψ_1 (Ψ_2). Consider any immediate successor of $\mathbf{n}_{2j-1}$ in τ, say $\mathbf{n}_{2j}$, labelled by $\mathcal{F} \vdash_{\Delta_{2j}} \Psi_i$ with $E_{2j-1} \in \mathcal{F}$: there is at least one of the two successors with this feature. Let $\gamma_{2j} = (\mathbf{n}_{2j}, E_{2j-1}, \Psi_i(\boldsymbol{U}_i^{\boldsymbol{\alpha}_i})_{\Delta_{2j}})$. It is clear that conditions (i)–(iv) hold (condition (v) is considered below). If the rule is $\wedge$ then $\Phi_{2j-1} = \Psi_1 \wedge \Psi_2$. Hence $E_{2j-1} \notin \|\Psi_1(\boldsymbol{U}_1^{\boldsymbol{\alpha}_1})_{\Delta_{2j-1}}\|$ or $E_{2j-1} \notin \|\Psi_2(\boldsymbol{U}_2^{\boldsymbol{\alpha}_2})_{\Delta_{2j-1}}\|$ where as above $\boldsymbol{\alpha}_1$ ($\boldsymbol{\alpha}_2$) is the signature $\boldsymbol{\alpha}_{2j-1}$ restricted to the active ν-constants in Ψ_1 (Ψ_2). Consider any immediate successor of $\mathbf{n}_{2j-1}$ in τ, say $\mathbf{n}_{2j}$, labelled by $\mathcal{F} \vdash_{\Delta_{2j}} \Psi_i$ such that $E_{2j-1} \notin \|\Psi_i(\boldsymbol{U}_i^{\boldsymbol{\alpha}_i})_{\Delta_{2j-1}}\|$. Let $\gamma_{2j} = (\mathbf{n}_{2j}, E_{2j-1}, \Psi_i(\boldsymbol{U}_i^{\boldsymbol{\alpha}_i})_{\Delta_{2j}})$. Again γ_{2j} fulfills conditions (i)–(iv). If the rule is $[K]$ then $\Phi_{2j-1} = [K]\Psi$. As $E_{2j-1} \notin \|[K]\Psi(\boldsymbol{U}_{2j-1}^{\boldsymbol{\alpha}_{2j-1}})_{\Delta_{2j-1}}\|$ there is an E_{2j} and an $a \in K$ such that $E_{2j-1} \xrightarrow{a} E_{2j}$ and $E_{2j} \notin \|\Psi(\boldsymbol{U}_{2j-1}^{\boldsymbol{\alpha}_{2j-1}})_{\Delta_{2j-1}}\|$. Consider the immediate successor of $\mathbf{n}_{2j-1}$ in τ, say $\mathbf{n}_{2j}$ labelled by $\mathcal{F} \vdash_{\Delta_{2j}} \Psi'$. Then clearly, $E_{2j} \in \mathcal{F}$, and $\Psi' = \Psi$. So $\gamma_{2j} = (\mathbf{n}_{2j}, E_{2j}, \Psi(\boldsymbol{U}_{2j-1}^{\boldsymbol{\alpha}_{2j-1}})_{\Delta_{2j}})$. If the rule is $\langle K\rangle$ then $\Phi_{2j-1} = \langle K\rangle\Psi$. As $E_{2j-1} \notin \|\langle K\rangle\Psi(\boldsymbol{U}_{2j-1}^{\boldsymbol{\alpha}_{2j-1}})_{\Delta_{2j-1}}\|$ we know that $f(E_{2j-1}) \notin \|\Psi(\boldsymbol{U}_{2j-1}^{\boldsymbol{\alpha}_{2j-1}})_{\Delta_{2j-1}}\|$ where f is the function given by the $\langle K\rangle$ rule. The immediate successor of $\mathbf{n}_{2j-1}$ in τ is $\mathbf{n}_{2j}$ labelled $\mathcal{F} \vdash_{\Delta_{2j}} \Psi$ and $f(E_{2j-1}) \in \mathcal{F}$. So $\gamma_{2j} = (\mathbf{n}_{2j}, f(E_{2j-1}), \Psi(\boldsymbol{U}_{2j-1}^{\boldsymbol{\alpha}_{2j-1}})_{\Delta_{2j}})$. If the rule applied is σZ. then $\Phi_{2j-1} = \sigma Z.\Psi$ and the immediate successor $\mathbf{n}_{2j}$ in τ is labelled $\mathcal{F} \vdash_{\Delta_{2j}} U$. If $\sigma = \mu$ then $\gamma_{2j} = (\mathbf{n}_{2j}, E_{2j-1}, U(\boldsymbol{U}_{2j-1}^{\boldsymbol{\alpha}_{2j-1}})_{\Delta_{2j}})$. If $\sigma = \nu$ then instead $\gamma_{2j} = (\mathbf{n}_{2j}, E_{2j-1}, U(\boldsymbol{U}_{2j-1}^{\boldsymbol{\alpha}_{2j-1}} U^{\alpha})_{\Delta_{2j}})$ where α is the least ordinal such that $E_{2j-1} \notin \|U(\boldsymbol{U}_{2j-1}^{\boldsymbol{\alpha}_{2j-1}} U^{\alpha})_{\Delta_{2j}}\|$. By Proposition 5.1 there is such a least α. If the rule applied is Un then $\Phi_{2j-1} = U$. So $\Delta_{2j-1}(U) = \sigma Z.\Psi$. Let $\mathbf{n}_{2j}$ be the immediate successor of $\mathbf{n}_{2j-1}$ in τ. If $\sigma = \mu$ then $\gamma_{2j} = (\mathbf{n}_{2j}, E_{2j-1}, \Psi[Z := U](\boldsymbol{U}_{2j-1}^{\boldsymbol{\alpha}_{2j-1}})_{\Delta_{2j}})$. Clearly, $E_{2j-1} \notin \|\Psi[Z := U](\boldsymbol{U}_{2j-1}^{\boldsymbol{\alpha}_{2j-1}})_{\Delta_{2j}}\|$. If $\sigma = \nu$ and $\boldsymbol{U}_{2j-1}^{\boldsymbol{\alpha}_{2j-1}} = U_1^{\alpha_1} \ldots U_n^{\alpha_n}$ then $U = U_n$. Let $\gamma_{2j} = (\mathbf{n}_{2j}, E_{2j-1}, \Psi[Z := U](U_1^{\alpha_1} \ldots U_n^{\alpha_n - 1})_{\Delta_{2j}})$: clearly, as α_n is not a limit ordinal, and is greater than 0, then $E_{2j-1} \notin \|U(U_1^{\alpha_1} \ldots U_n^{\alpha_n})_{\Delta_{2j}}\|$ iff $E_{2j-1} \notin \|\Psi[Z := U](U_1^{\alpha_1} \ldots U_n^{\alpha_n - 1})_{\Delta_{2j}}\|$. Finally, if the rule applied is Thin then $\gamma_{2j} = \gamma_{2j-1}$ except that $\mathbf{n}_{2j-1}$ is replaced by its immediate successor $\mathbf{n}_{2j}$.

The only outstanding property to show is that $\mathbf{n}_{2j}$ is not a terminal in τ. Clearly, it can not label a sequent $\mathcal{F} \vdash_{\Delta_{2j}} \Psi$ in τ which is successful because $\Psi = Z$ or $\Psi = \neg Z$ or $\Psi = [K]\Psi'$, as this contradicts condition (iv) of γ. Moreover, by construction $\mathbf{n}_{2j}$ can not label a σ-terminal. For suppose it does, then its companion is $\mathbf{n}_{2k}$ in γ_{2k}, $k < j$. But consider the construction of γ_{2k+1}. First, by definition of a trail $E_{2k+1} \sqsupset_{\mathbf{n}_{2k}} E_{2j}$. If $\mathbf{n}_{2j}$ labels a μ-terminal then the signature at γ_{2k+1} is less than or equal to the signature at γ_{2j} (since it never increases for the same active constant sequence). But this contradicts γ_{2k+1} as $E_{2j} \in E^*_{2k}$, and $E_{2j} \notin \|\Phi_{2k+1}(\boldsymbol{U}^{\boldsymbol{\alpha}_{2k+1}}_{2k+1})_{\Delta_{2k+1}}\|$. Similarly, if $\mathbf{n}_{2j}$ labels a ν-terminal then the signature at γ_{2k+1} is strictly less than the signature at γ_{2j} (as the constant is then unfolded at node $\mathbf{n}_{2k+1}$). And this also contradicts the construction of γ_{2k+1} as $E_{2j} \in E^*_{2k}$. $\square$

Next we prove completeness in a stronger version utilizing *canonical* tableaux. Moreover, the proof also appeals to the notation $\Phi(\boldsymbol{U}^{\boldsymbol{\alpha}})_\Delta$ where now $\boldsymbol{\alpha}$ is a μ-signature for Φ_Δ.

Theorem 5.4. *If $\mathcal{E} \subseteq \|\Phi_\Delta\|^{\mathcal{T}}_{\mathcal{V}}$ then $\mathcal{E} \vdash_\Delta \Phi$ has a successful canonical tableau on $(\mathcal{T}, \mathcal{V})$.*

Proof. We construct a canonical tableau τ for $\mathcal{E} \vdash_\Delta \Phi$ on $(\mathcal{T}, \mathcal{V})$ with the property that if node $\mathbf{n}_i$ of τ is labelled with $\mathcal{E}_i \vdash_{\Delta_i} \Phi_i$ then $\mathcal{E}_i \subseteq \|\Phi_{i\Delta_i}\|^{\mathcal{T}}_{\mathcal{V}}$. As before we drop the indices ${}^{\mathcal{T}}_{\mathcal{V}}$. The root $\mathbf{n}_0$ of τ is labelled with $\mathcal{E} \vdash_\Delta \Phi$. Assume that nodes $\mathbf{n}_0, \ldots, \mathbf{n}_j$ have been constructed and $\mathbf{n}_k$, $k \leq j$, is a frontier node that is not a terminal, labelled $\mathcal{F} \vdash_\Sigma \Psi$. We proceed by defining the immediate successors of $\mathbf{n}_k$, depending on the form of Ψ. If $\Psi = \Psi_1 \wedge \Psi_2$ then two successors $\mathbf{n}_1$ and $\mathbf{n}_2$ are introduced labelled with the sequents $\mathcal{F} \vdash_\Sigma \Psi_1$ and $\mathcal{F} \vdash_\Sigma \Psi_2$. Clearly, $\mathcal{F} \subseteq \|\Psi_{i\Sigma}\|$. If $\Psi = \Psi_1 \vee \Psi_2$ then again two immediate successors $\mathbf{n}_1$ and $\mathbf{n}_2$ are introduced labelled with the sequents $\mathcal{F}_1 \vdash_\Sigma \Psi_1$ and $\mathcal{F}_2 \vdash_\Sigma \Psi_2$ where $\mathcal{F}_1 \cup \mathcal{F}_2 = \mathcal{F}$. We define when $F \in \mathcal{F}_j$, $j = 1, 2$. By Proposition 5.2 (ii), when $F \in \mathcal{F}$ there is a least μ-signature $\boldsymbol{\alpha}$ such that $F \in \|\Psi(\boldsymbol{U}^{\boldsymbol{\alpha}})_\Sigma\|$. Hence $F \in \|\Psi_1(\boldsymbol{U}_1^{\boldsymbol{\alpha}_1})_\Sigma\|$ or $F \in \|\Psi_2(\boldsymbol{U}_2^{\boldsymbol{\alpha}_2})_\Sigma\|$, where $\boldsymbol{\alpha}_j$ is the signature $\boldsymbol{\alpha}$ restricted to the active constants in Ψ_j: if $F \in \|\Psi_j(\boldsymbol{U}_j^{\boldsymbol{\alpha}_j})_\Sigma\|$ then $F \in \mathcal{F}_j$. If $\Psi = [K]\Psi_1$ then one immediate successor $\mathbf{n}_1$ is introduced labelled with the sequent $\mathcal{F}' \vdash_\Sigma \Psi_1$ where $\mathcal{F}' = \{\, F' \mid \exists\, F \in \mathcal{F} \,.\, \exists\, a \in K \,.\, F \xrightarrow{a} F' \,\}$. If $\Psi = \langle K \rangle \Psi_1$ then again one immediate successor $\mathbf{n}_1$ is introduced labelled with the sequent $f(\mathcal{F}) \vdash_\Sigma \Psi_1$. The function f is defined as follows: for each $F \in \mathcal{F}$ consider the least μ-signature $\boldsymbol{\alpha}$ such that $F \in \|\Psi(\boldsymbol{U}^{\boldsymbol{\alpha}})_\Sigma\|$. Hence $\exists\, F' \,.\, \exists\, a \in K \,.\, F \xrightarrow{a} F'$ and $F' \in \|\Psi_1(\boldsymbol{U}^{\boldsymbol{\alpha}})_\Sigma\|$: choose any such F' to be $f(F)$. If $\Psi = \sigma Z.\Psi_1$ then one immediate successor is introduced labelled $\mathcal{F} \vdash_{\Sigma'} U$ where $\Sigma' = \Sigma \cdot U = (\sigma Z.\Psi_1)$. Clearly, $\mathcal{F} \subseteq \|U_{\Sigma'}\|$. Finally, if $\Psi = U$ then two nodes $\mathbf{n}_1$ and $\mathbf{n}_2$ are introduced where $\mathbf{n}_1$ is the immediate successor of $\mathbf{n}_k$ and is labelled $\mathcal{F}' \vdash_\Sigma U$ and $\mathbf{n}_2$ is the immediate successor of $\mathbf{n}_1$ and is labelled $\mathcal{F}' \vdash_\Sigma \Psi_1[Z := U]$ given that $\Sigma(U) = \sigma Z.\Psi_1$. The set $\mathcal{F}'$ is defined as follows. For each $F \in \mathcal{F}$ let $\mathcal{R}(F) = \{\, F' \mid \exists\, n \geq 0 \,.\, \exists\, a_1 \ldots a_n \in \mathcal{L}^*_{\mathcal{T}} \,.\, F \xrightarrow{a_1 \ldots a_n} F' \,\}$. So $\mathcal{R}(F)$ is the set of reachable points from F (including F) in $\mathcal{T}$. Now let $\mathcal{F}' = \bigcup\{\, \mathcal{R}(F) \mid F \in \mathcal{F} \,\} \cap \|U_\Sigma\|$: a more subtle account could be given by defining the reachable points relative to the formula U. By definition $\mathcal{F}' \subseteq \|U_\Sigma\|$ and as $\mathcal{F} \subseteq \|U_\Sigma\|$ it follows that $\mathcal{F} \subseteq \mathcal{F}'$.

The remainder of the proof establishes that this construction yields a successful canonical tableau τ. First, τ is both finite and canonical as there can only be one unfolding of any constant U: a node labelled $\mathcal{F} \vdash_\Delta U$ is terminal if there are nodes $\mathbf{n}_j$ and $\mathbf{n}_{j1}$ (the immediate successor of $\mathbf{n}_j$) above it labelled $\mathcal{F}_1 \vdash_{\Delta'} U$ and $\mathcal{F}_2 \vdash_{\Delta'} U$ as $\mathcal{F} \subseteq \mathcal{F}_2$ (since by construction $\mathcal{F}_2$ is all points reachable from $\mathcal{F}_1$ and satisfying U). Consequently, the only possibility of failure of τ as a successful tableau is if $\sqsubset_{\mathbf{n}}$ for some companion $\mathbf{n}$ of a μ-terminal is not well-founded: suppose $E_0 \sqsupset_{\mathbf{n}} E_1 \sqsupset_{\mathbf{n}} \cdots \sqsupset_{\mathbf{n}} E_m \sqsupset_{\mathbf{n}} \cdots$, and that $\mathbf{n}$ is labelled by the sequent $\mathcal{F} \vdash_\Sigma U$. Let $\boldsymbol{\alpha}_j$ be the least μ-signature associated with E_j at $\mathbf{n}$: so $E_j \in \|U(\boldsymbol{U}_j^{\boldsymbol{\alpha}_j})_\Sigma\|$ and for all $\boldsymbol{\beta} < \boldsymbol{\alpha}_j$, $E_j \notin \|U(\boldsymbol{U}_j^{\boldsymbol{\beta}})_\Sigma\|$. Clearly, by the construction of τ (as the immediate successor of $\mathbf{n}$ is determined by an unfolding of U) $\boldsymbol{\alpha}_j > \boldsymbol{\alpha}_k$ when $j < k$. But this is impossible. $\square$

References

[1] J. C. Bradfield, Proving temporal properties of Petri nets, to appear in: *Advances in Petri Nets 1990*, Lecture Notes in Computer Science **???** (Springer–Verlag, Berlin, 1991).

[2] J. C. Bradfield and C. P. Stirling, Verifying temporal properties of processes, in: J. C. M. Baeten and J. W. Klop (eds.) *Proc. CONCUR '90* Lecture Notes in Computer Science **458** (Springer–Verlag, Berlin, 1990).

[3] M. F. Dam, Translating CTL* into the modal μ-calculus, Technical Report ECS–LFCS–90–123, LFCS, Dept. of Computer Science, University of Edinburgh, Edinburgh, U.K. (1990).

[4] E. A. Emerson and C.-L. Lei, Efficient model checking in fragments of the propositional mu-calculus, in: *Proc. First IEEE Symp. on Logic in Computer Science* (1986) (267–278).

[5] M. Fitting, *Proof Methods for Modal and Intuitionistic Logics* (Reidel, 1983).

[6] Hüt90 H. Hüttel, *SnS* can be modally characterized. *Theoret. Comput. Sci.* **74** 239–248 (1990).

[7] D. Kozen, Results on the propositional mu-calculus. *Theoret. Comput. Sci.* **27** (1983) 333–354.

[8] K. Larsen, Proof systems for satisfiability in Hennessy–Milner logic with recursion, *Theoret. Comput. Sci.* **72** (1990) 265–288.

[9] Z. Manna and A. Pnueli, The anchored version of the temporal framework, in: J. de Bakker, W.-P. de Roever and G. Rozenberg, *Proc. Workshop on Linear Time, Branching Time and Partial Order in Logics for Concurrency* Lecture Notes in Computer Science **354** (Springer–Verlag, Berlin, 1989).

[10] V. Pratt, A decidable mu-calculus, in: *Proc. 22nd Ann. ACM Symp. on Foundation of Computer Science* (1981) (421–427).

[11] C. P. Stirling, Modal logics for communicating systems, *Theoret. Comput. Sci.* **49** (1987) (311–347).

[12] C. P. Stirling, Temporal logics for CCS, in: J. de Bakker, W.-P. de Roever and G. Rozenberg, *Proc. Workshop on Linear Time, Branching Time and Partial Order in Logics for Concurrency* Lecture Notes in Computer Science **354** (Springer–Verlag, Berlin, 1989).

[13] C. P. Stirling and D. J. Walker, Local model checking in the modal mu-calculus. in: *Proc. International Joint Conference on Theory and Practice of Software Development* Lecture Notes in Computer Science **351** (Springer–Verlag, Berlin, 1989) (369–382).

[14] C. P. Stirling and D. J. Walker, A general tableau technique for verifying temporal properties of concurrent programs, *Proc. Int. BCS–FACS Workshop on Semantics for Concurrency* Workshops in Computing (Springer–Verlag, Berlin, 1990) (1–15).

[15] R. S. Streett and E. A. Emerson, An automata theoretic decision procedure for the propositional mu-calculus, *Information and Computation* **81** (1989) (249–264).