μ-Calculus with Explicit Points and Approximations

MADS DAM, *Department of Microelectronics and Information Technology (IMIT), Royal Institute of Technology (KTH), Electrum 229, SE-164 40 Kista, Sweden.*
E-mail: mfd@it.kth.se

DILIAN GUROV, *Swedish Institute of Computer Science (SICS), Box 1263, SE-164 29 Kista, Sweden.*
E-mail: dilian@sics.se

Abstract

We present a Gentzen-style sequent calculus for program verification which accommodates both model checking-like verification based on global state space exploration, and compositional reasoning. To handle the complexities arising from the presence of fixed-point formulas, programs with dynamically evolving architecture, and cut rules we use transition assertions, and introduce fixed-point approximants explicitly into the assertion language. We address, in a game-based manner, the semantical basis of this approach, as it applies to the entailment subproblem. Soundness and completeness results are obtained, and examples are shown illustrating some of the concepts.

Keywords: μ-calculus, sequent calculus, program verification, compositionality.

1 Introduction

In this paper we study program verification in terms of provability of general sequents of the shape

$$\Gamma \vdash \Delta, \tag{1.1}$$

where the components of Γ and Δ can be temporal correctness assertions $P : \phi$. Since program terms P can involve free program variables, and since assumptions can be stated concerning these (as assertions in Γ), this provides a very general and powerful setting for program verification which accommodates both model checking-like verification based on global state space exploration, and compositional reasoning. This sort of approach has been examined in a number of recent papers, including [1, 2, 5, 6, 12], and traces back to work by Stirling [13] and Winskel [17]. Several programming or modelling languages have been considered, including CCS [5], the π-calculus [2], CHOCS [1], general GSOS-definable languages [12]. In [6, 8] we presented an approach to handling a core fragment of the distributed functional programming language Erlang [3] including features such as data types, a fairly rich sequential structure, asynchronous buffered communication, and dynamic process creation. We have implemented an advanced verification tool EVT [9], and demonstrated the viability of our approach through a number of case studies [4, 8] exhibiting intricate dynamic behaviour.

The key idea is that the general form of sequent (1.1) allows correctness properties $P : \phi$ to be stated and proved in a *modular* fashion, under the assumption of correctness properties

J. Logic Computat., Vol. 12 No. 2, pp. 255–269 2002

of constituents of P, represented by free variables. A general rule of *subterm cut* of the shape

$$\frac{\Gamma \vdash Q : \psi, \Delta \quad \Gamma, x : \psi \vdash P : \phi, \Delta}{\Gamma \vdash P[Q/x] : \phi, \Delta}$$

allows such subterm assumptions to be introduced and used.

The difficulty with this (as with any other approach to modular verification) is to find a way of supporting temporal properties. In [5] we showed one way of doing this, and built, for the first time, a compositional proof system capable of handling general CCS terms, including those that create new processes dynamically (the only source of infiniteness in CCS). In [6] we used a similar (though considerably improved) approach to address Erlang.

Applied to a real programming language such as Erlang, the verification problem is evidently undecidable. Even so, it is of interest to examine subproblems for which completeness results should be obtainable at least in principle. In [5], weak completeness results were obtained for sequents of the restricted shape $\vdash P : \phi$ where ϕ is a modal μ-calculus formula, and where P is a finite-state CCS process. But such a result is not really adequate to illustrate the power of our approach, as it does not improve upon results obtainable using more standard and well-understood model checking techniques based on global state space exploration (such as [15, 18]).

On another track, and addressing Hennessy-Milner logic only (so: no temporal properties), Simpson [12] showed how a proof-theoretical setting like that of (1.1) could be used to produce compositional proof systems from arbitrary GSOS operational semantics definitions in a very systematic way. Moreover, Simpson was able to obtain strong completeness results which were outside the scope of our approach. The main idea of [12] was to introduce transition assertions of the shape $P \overset{\alpha}{\rightarrow} Q$ as components of sequents on a par with correctness assertions like $P : \phi$. Whereas the presence of transition assertions may be argued (wrongly, in our opinion) to go against the 'spirit of compositionality', the facts remains that

1. the approach is powerful enough to derive the sorts of compositional rules used in the previous approaches referred to above, and
2. the very direct and simple embedding of the operational semantics relates structure and behaviour in a far more direct and comprehensive way than the other approaches allow. This is reflected by the strong completeness results obtained in [12].

In this paper we take a first step towards a merge of Simpson's operational semantics embedding with the sort of treatment of fixed points which we proposed in [5], in particular to obtain strong completeness results that apply to sequents of the shape (1.1) *even* in the presence of temporal connectives.

Towards the realization of this one can clearly identify a number of subproblems, including at least the following:

- Model checking: μ-calculus, restricted sequents. In this case, like in [15, 18], sequents (1.1) are required to contain *exactly one* correctness assertion.
- Entailment: μ-calculus, restricted program terms. In this case, which is addressed here, the only program terms P allowed in correctness assertions $P : \phi$ are variables (so: no program, or process constructors, making the fragment just as expressive as Kozen's axiomatization [10]).
- Modal logic. This was the case considered by Simpson [12].

- General case: μ-calculus, general sequents. For what classes of structured program terms describing infinite-state behaviours can one achieve completeness?

In the last case there are several difficulties in realizing our program, over and beyond what must be faced when studying the simpler problems of model checking, entailment, or modal logic. First, undecidability entails that some sort of restriction upon the general sequent format must be imposed. But on a slightly deeper level, a central and very important difficulty is due to the subterm cut rule. While this rule is eliminable (or does not apply) in some of the simpler settings, in the presence of structured program terms the rule is essential and causes severe difficulties for the handling of recursive formulas.

Essentially, recursive formulas are handled using some form of well-founded induction on approximation ordinals. In the absence of the subterm cut rule (or other rules with similar effect, such as the classical cut) approximation ordinals can be guaranteed to occur only in contravariant positions. In the presence of cut this can, however, no longer be guaranteed. In our earlier work [5] this caused us to rely on a handling of fixed points which was extremely syntactical, hedged with side conditions, and also unnecessarily restrictive.

The contribution of the present paper is to show that in fact a far simpler and much more semantical approach is possible, by introducing approximation ordinal variables explicitly into the proof system. This has not only theoretical implications: the construction introduced here underpins our treatment of μ-calculus in the EVT tool [9].

In a previous paper [7] we instantiated our approach to CCS and illustrated the workings of the proof system by means of examples. In this paper we address the semantical basis, as it applies to the entailment subproblem. After briefly introducing the logic and proof system we present, in Section 4, a *refutation game* providing a semantical characterization of validity for cyclic proof structures. We prove the derived notion of refutation-game provability sound, and in Section 5, we prove completeness by reduction to Kozen's axiomatization [10]. For practical proof search the game-based characterization is unsatisfactory — it does not permit loop closure to be determined effectively. For this reason we introduce a rule of *assumption discharge* in Section 6, and show it sound and complete as well. To illustrate the workings of the proof system we exhibit two examples, of a sequent which is provable and of another sequent which is not.

2 Logic

The standard syntax of the modal μ-calculus is augmented by adding a form of fixed point formula approximation, using ordinal variables. Formulas ϕ are generated by the following grammar, where κ ranges over a set of *ordinal variables*, α over a set of *actions*, and X over a set of *propositional variables*.

$$\phi ::= \phi \vee \phi \bigm| \neg\phi \bigm| \langle\alpha\rangle\phi \bigm| X \bigm| \mu X.\phi \bigm| (\mu X.\phi)^\kappa \ .$$

An occurrence of a subformula ψ in ϕ is *positive*, if ψ appears in the scope of an even number of negation symbols. Otherwise the occurrence is negative. The formation of least fixed point formulas of one of the shapes $\mu X.\phi$ or $(\mu X.\phi)^\kappa$ is subject to the usual formal monotonicity condition that occurrences of X in ϕ are positive. We use the symbols U and V to range over (unindexed) fixed point formulas $\mu X.\phi$. A formula ϕ is *propositionally closed* if ϕ does not have free ocurrences of propositional variables. A formula is *pure* if it does not have subformulas of the form U^κ. Standard abbreviations apply, such as $\nu X.\phi =$

$\neg\mu X.\neg(\phi[\neg X/X])$. We assume the standard modal μ-calculus semantics [10], augmented by the clause:

$$\|(\mu X.\phi)^\kappa\|\rho = \begin{cases} \emptyset & \text{if } \rho(\kappa) = 0 \\ \|\phi\|\rho[\|(\mu X.\phi)^\kappa\|\rho[\beta/\kappa]/X] & \text{if } \rho(\kappa) = \beta + 1 \\ \bigcup\{\|(\mu X.\phi)^\kappa\|\rho[\beta/\kappa] \mid \beta < \rho(\kappa)\} & \text{if } \rho(\kappa) \text{ is a limit ordinal} \end{cases}$$

where ρ is an interpretation function (environment), mapping ordinal variables to ordinals, and propositional variables to sets of states. The use of ordinal approximation hinges on the following results (of which (1) is the well-known Knaster-Tarski fixed point theorem).

THEOREM 2.1

1. $\|\mu X.\phi\|\rho = \bigcup_\beta \|(\mu X.\phi)^\kappa\|\rho[\beta/\kappa]$
2. $\|(\mu X.\phi)^\kappa\|\rho = \bigcup_{\beta<\rho(\kappa)} \|\phi\|\rho[\|(\mu X.\phi)^\kappa\|\rho/X, \beta/\kappa]$

Observe how this casts the properties U and U^κ as existential properties: This is useful to motivate the proof rules for fixed point formulas given below. Observe also that, for countable models, quantification over countable ordinals in Theorem 2.1 suffices.

We implicitly assume a process language for forming process terms E, F involving process variables x, y, and assume a transitional semantics specifying the valid action-labelled transitions $s \overset{\alpha}{\to} s'$ between closed process terms called *states*. We extend environments ρ to map process variables to states.

Sequents, or judgements, mention satisfaction assertions $E : \phi$, transition assertions $E \overset{\alpha}{\to} F$, and ordinal variable constraints $\kappa < \kappa'$.

DEFINITION 2.2 (Assertions, Sequents)

1. An *assertion* is an expression of one of the forms $E : \phi$, $E \overset{\alpha}{\to} F$, or $\kappa < \kappa'$, where ϕ is a propositionally closed formula.
2. The assertion $E : \phi$ is *valid for an interpretation function* ρ, if $E\rho \in \|\phi\|\rho$. The assertion $\kappa < \kappa'$ is valid for ρ, if $\rho(\kappa) < \rho(\kappa')$. The assertion $E \overset{\alpha}{\to} F$ is valid for ρ, if $E\rho \overset{\alpha}{\to} F\rho$ is a valid transition.
3. A *sequent* is an expression of the form $\Gamma \vdash \Delta$, where Γ and Δ are sets of assertions. A sequent is termed *pure* if it contains only satisfaction assertions of the shape $x : \phi$.
4. The sequent $\Gamma \vdash \Delta$ is *valid*, if for all interpretation functions ρ, all assertions in Γ are valid for ρ only if some assertion in Δ is valid for ρ as well.

3 A proof system for logical entailment

Structural rules

We assume the axiom rule, the rule of cut, and weakening:

$$\text{AX}\ \frac{\cdot}{\Gamma, A \vdash A, \Delta} \qquad \text{CUT}\ \frac{\Gamma \vdash A, \Delta \quad \Gamma, A \vdash \Delta}{\Gamma \vdash \Delta}$$

$$\text{W-L}\ \frac{\Gamma \vdash \Delta}{\Gamma, A \vdash \Delta} \qquad \text{W-R}\ \frac{\Gamma \vdash \Delta}{\Gamma \vdash A, \Delta}$$

Since Γ and Δ are sets, structural rules like permutation and contraction are vacuous.

Logical rules

In the following listing we assume that $U = \mu X.\phi$:

$$\neg\text{-L}\quad \frac{\Gamma \vdash E : \phi, \Delta}{\Gamma, E : \neg\phi \vdash \Delta} \qquad\qquad \neg\text{-R}\quad \frac{\Gamma, E : \phi \vdash \Delta}{\Gamma \vdash E : \neg\phi, \Delta}$$

$$\vee\text{-L}\quad \frac{\Gamma, E : \phi \vdash \Delta \quad \Gamma, E : \psi \vdash \Delta}{\Gamma, E : \phi \vee \psi \vdash \Delta} \qquad\qquad \vee\text{-R}\quad \frac{\Gamma \vdash E : \phi, E : \psi, \Delta}{\Gamma \vdash E : \phi \vee \psi, \Delta}$$

$$\langle\alpha\rangle\text{-L}\quad \frac{\Gamma, E \xrightarrow{\alpha} x, x : \phi \vdash \Delta}{\Gamma, E : \langle\alpha\rangle\phi \vdash \Delta} \quad \textit{fresh}(x)$$

$$\langle\alpha\rangle\text{-R}\quad \frac{\Gamma \vdash E \xrightarrow{\alpha} E', \Delta \quad \Gamma \vdash E' : \phi, \Delta}{\Gamma \vdash E : \langle\alpha\rangle\phi, \Delta}$$

$$U\text{-L}\quad \frac{\Gamma, E : U^{\kappa} \vdash \Delta}{\Gamma, E : U \vdash \Delta} \quad \textit{fresh}(\kappa) \quad U\text{-R}\quad \frac{\Gamma \vdash E : \phi[U/X], \Delta}{\Gamma \vdash E : U, \Delta}$$

$$U^{\kappa}\text{-L}\quad \frac{\Gamma, \kappa' < \kappa, E : \phi[U^{\kappa'}/X] \vdash \Delta}{\Gamma, E : U^{\kappa} \vdash \Delta} \quad \textit{fresh}(\kappa')$$

$$U^{\kappa}\text{-R}\quad \frac{\Gamma \vdash \kappa' < \kappa, \Delta \quad \Gamma \vdash E : \phi[U^{\kappa'}/X], \Delta}{\Gamma \vdash E : U^{\kappa}, \Delta}$$

The side condition *fresh*(x) (*fresh*(κ)) is intended to mean that x (κ) does not appear freely in the conclusion of the rule.

The rules for unindexed and indexed fixed point formulas are directly motivated by Theorem 2.1.

Ordinal constraints

Finally, we provide a rule for reasoning about ordinal constraints.

$$\textsc{OrdTr}\quad \frac{\Gamma, \kappa' < \kappa \vdash \kappa'' < \kappa', \Delta}{\Gamma, \kappa' < \kappa \vdash \kappa'' < \kappa, \Delta}$$

THEOREM 3.1 (Local soundness)
All rules for logical entailment are individually sound: the conclusion of each rule is valid whenever its premisses are valid.

Observe that the proof system does not include the subterm cut rule. This rule is not needed, since the only program terms considered in this paper are program variables.

Having transition assertions allows the transitional semantics of a process language to be embedded directly into the proof system as a separate set of proof rules. This can be done in a straightforward manner for any GSOS-definable language [12], where the operational semantics of a process language is given as a closure relation on processes through a set of transition rules: the transitions that a closed process term (state) can perform are exactly those derivable by these rules. Hence, the transition rules can be included directly as right introduction rules into our proof system, while the left introduction rules (stating, in some sense, what transitions are *not* possible) come from the closure assumption. See [7] for an instantiation of our approach to CCS.

4 The refutation game

For practical purposes only finite proof trees can be considered as proofs. By themselves the above proof rules are insufficient, as there is no bound on the number of times fixed point formulas can be unfolded. We therefore require some mechanism for determining loops, or repeating nodes, and for determining when proof construction can safely be terminated. We devise a simple one-player game to account for this, implicitly building in well-founded ordinal induction.

First we need some concepts concerning proof trees. A *proof structure* is a finite tree constructed according to the proof rules set out above. Nodes in proof structures are ranged over by N, and the notation $N(\Gamma \vdash \Delta)$ indicates that the node N is labelled by the sequent $\Gamma \vdash \Delta$. Write $N' < N$ if N' appears on the path from the root to N, but not vice versa.

DEFINITION 4.1 (Repeating node, Arena)

1. Suppose $N'(\Gamma' \vdash \Delta') < N(\Gamma \vdash \Delta)$. Then N is a *repeat of* N' up to the substitution σ, if
   (a) $A\sigma \in \Gamma$ whenever $A \in \Gamma'$, and
   (b) $A\sigma \in \Delta$ whenever $A \in \Delta'$.
2. An *arena*, $\mathcal{A}$, is a proof structure for which each leaf node N is either an axiom instance or else to N is associated some node N' and substitution σ such that N is a repeat of N' up to σ.

As usual, substitutions σ map variables to terms of the same type; in the present case we have process variables and ordinal variables only.

Definition 4.1 is motivated in the following manner. Say that ρ is a *falsifying interpretation* for the sequent $\Gamma \vdash \Delta$ if all $A \in \Gamma$ are valid for ρ and all $A \in \Delta$ are invalid for ρ.

PROPOSITION 4.2
Suppose $N(\Gamma \vdash \Delta)$ is a repeat of $N'(\Gamma' \vdash \Delta')$ up to the substitution σ. Suppose that ρ is a falsifying interpretation for $\Gamma \vdash \Delta$. Then $\rho \circ \sigma$ is a falsifying interpretation for $\Gamma' \vdash \Delta'$.

Proof games are often presented as two-player games (cf. [14]), played between a player, player I, whose task it is, roughly, to choose a proof rule to apply at some given game configuration, and an opponent, player II, whose task it is to show that no 'good' choices can be made. In the present setting an arena represents a set of choices for player I, made in a history-free manner. Since all choices for player I are made statically there is no point in distinguishing winning game runs from winning strategies. This is similar in spirit to the two-player game of [11], but is played on finite structures.

The game $\mathcal{G}(\mathcal{A})$ is played by a 'refuter' R on an arena $\mathcal{A}$. The task of R is to refute the claim that the arena represents a valid proof of the root sequent. The refuter R shows that an arena does not represent a valid proof by showing that its repeating nodes introduce recursion into the proof tree in a manner which is ill-founded. This is achieved by tracing an infinite path through the game arena, using only sequents which are non-trivial in the following sense:

DEFINITION 4.3 (Non-trivial sequent)
The sequent $\Gamma \vdash \Delta$ is *non-trivial for the interpretation function* ρ, if whenever $E \xrightarrow{\alpha} F \in \Gamma$ then $E\rho \xrightarrow{\alpha} F\rho$ is a valid transition, and whenever $\kappa' < \kappa \in \Gamma$ then $\rho(\kappa') < \rho(\kappa)$. The pair $(\Gamma \vdash \Delta, \rho)$ is *non-trivial*, if $\Gamma \vdash \Delta$ is non-trivial for ρ.

Notice that $\Gamma \vdash \Delta$ is non-trivial for ρ whenever ρ is a falsifying interpretation for $\Gamma \vdash \Delta$. Intuitively, non-triviality guarantees, due to well-foundedness of ordinals, that the value of no ordinal variable is being decremented infinitely often along this path, thus invalidating any ordinal induction argument on this arena.

Let an arena $\mathcal{A}$ be given, rooted in $N_0(\Gamma_0 \vdash \Delta_0)$. Initially R picks an interpretation ρ_0 for which $\Gamma_0 \vdash \Delta_0$ is non-trivial. R's claim is that ρ_0 is a falsifying interpretation for $\Gamma_0 \vdash \Delta_0$. So the *initial configuration* of the game has the shape (N_0, ρ_0). Suppose the game has reached the configuration (N_i, ρ_i). Then R can chose (N_{i+1}, ρ_{i+1}) as a *possible next configuration* if (N_{i+1}, ρ_{i+1}) is non-trivial and either:

1. N_{i+1} is a child node of N_i in $\mathcal{A}$ and ρ_{i+1} agrees with ρ_i on all common free variables, or:
2. N_i is a repeat of N' up to some substitution σ in $\mathcal{A}$, and then $N_{i+1} = N'$ and $\rho_{i+1} = \rho_i \circ \sigma$.

A game *run*, Π, is a finite or infinite sequence $(N_0, \rho_0), \ldots, (N_i, \rho_i), \ldots$ such that for each $j : 0 \leq j < i$, $\Pi(j+1) = (N_{j+1}, \rho_{j+1})$ is a possible next configuration for $\Pi(j)$.

DEFINITION 4.4 (Winning run, Proof)

1. The refuter R *wins* a game run just in case it is infinite.
2. A *proof* is an arena on which R has no winning run.
3. The sequent $\Gamma \vdash \Delta$ is refutation-game provable, denoted $\Gamma \vdash_r \Delta$, if there is a proof with root $\Gamma \vdash \Delta$.

The chosen game-theoretic setting should be:

- *sound*, in the sense that all refutation-game provable sequents are valid; and, conversely, it should preferably be
- *complete*, in the sense that all valid sequents are refutation-game provable.

Soundness is established next, while completeness is considered in the following section.

THEOREM 4.5 (Soundness)
The sequent $\Gamma \vdash \Delta$ is valid if $\Gamma \vdash_r \Delta$.

PROOF. Assume $\Gamma \vdash_r \Delta$. Suppose, for a contradiction, that $\Gamma \vdash \Delta$ is invalid, and that an arena $\mathcal{A}$ is given, rooted in a node N_0 labelled $\Gamma \vdash \Delta$. We find a falsifying interpretation ρ_0 for $\Gamma \vdash \Delta$. We use this to build a winning run Π for R, containing falsifying interpretations only, thus arriving at a contradiction. Suppose we have already constructed an initial segment $\Pi(0), \ldots, \Pi(i)$ of Π, where $\Pi(k)$ denotes $(N_k(\Gamma_k \vdash \Delta_k), \rho_k)$ and ρ_k falsifies $\Gamma_k \vdash \Delta_k$ for all $0 \leq k \leq i$. There are two possibilities.

(a) Suppose that N_i is a leaf node. Since N_i cannot be an axiom instance we find a node $N'(\Gamma' \vdash \Delta')$ and a substitution σ such that $N' < N_i$, and such that N_i is a repeat of N' up to σ. We pick

$$\Pi(i+1) = (N_{i+1}(\Gamma' \vdash \Delta'), \rho_i \circ \sigma)$$

as a possible next configuration for $\Pi(i)$. Clearly $\rho_i \circ \sigma$ falsifies $\Gamma' \vdash \Delta'$ by virtue of Proposition 4.2.

(b) So assume instead that N_i is an internal node. We pick (N_{i+1}, ρ_{i+1}) according to the proof rule applied to conclude $\Gamma_i \vdash \Delta_i$. Note that in the construction below ρ_{i+1} differs from ρ_i only in the values it assigns to variables freshly introduced by the rule applied, and that ρ_{i+1} is chosen so as to falsify $\Gamma_{i+1} \vdash \Delta_{i+1}$.

Ax Impossible, since $\Gamma_i \vdash \Delta_i$ is invalid.

$\vee$-L Suppose $\Gamma_i = \Gamma'_i, E : \phi \vee \psi$. Since Γ_i is validated by ρ_i, so is either $\Gamma'_i, E : \phi$ or $\Gamma'_i, E : \psi$. As N_{i+1} we choose whichever of these two that applies, and let $\rho_{i+1} = \rho_i$.

$\vee$-R Suppose $\Delta_i = E : \phi \vee \psi, \Delta'_i$. Let N_{i+1} be the predecessor of N_i, and $\rho_{i+1} = \rho_i$. Since Δ_i is invalidated by ρ, so is $E : \phi, E : \psi, \Delta'_i$.

$\neg$-L, $\neg$-R Trivial.

$\langle\alpha\rangle$-L Suppose $\Gamma_i = \Gamma'_i, E : \langle\alpha\rangle\phi$ and that x is fresh. Since $E : \langle\alpha\rangle\phi$ is validated by ρ_i we find some Q such that $E\rho \overset{\alpha}{\rightarrow} Q$ and $Q : \phi$ is validated by ρ_i. So, we let $\rho_{i+1} = \rho_i[Q/x]$ and let N_{i+1} be the predecessor of N_i labelled by $\Gamma'_i, E \overset{\alpha}{\rightarrow} x, x : \phi \vdash \Delta_i$.

$\langle\alpha\rangle$-R Suppose $\Delta_i = E : \langle\alpha\rangle\phi, \Delta'_i$. The two predecessors of N_i are labelled $\Gamma_i \vdash E \overset{\alpha}{\rightarrow} E', \Delta'_i$ and $\Gamma_i \vdash E' : \phi, \Delta'_i$, respectively. Let $\rho_{i+1} = \rho_i$. Either $E \overset{\alpha}{\rightarrow} E'$ or $E' : \phi$ is invalidated by ρ_{i+1}, as $E : \langle\alpha\rangle\phi$ is invalidated by ρ_i. So pick as N_{i+1} whichever one applies.

U-L Suppose $\Gamma_i = \Gamma'_i, E : U$, and let κ be fresh. Since $E : U$ is valid for ρ_i we find some β such that $E : U^\kappa$ is valid for $\rho_i[\beta/\kappa]$. Choose then as N_{i+1} the unique predecessor of N_i, and let $\rho_{i+1} = \rho_i[\beta/\kappa]$.

U-R Let $\Delta_i = E : \mu X.\phi, \Delta'_i$. Then $E : \phi[\mu X.\phi/X], \Delta'_i$ is invalidated by ρ_i so we let N_{i+1} be the unique predecessor of N_i and $\rho_{i+1} = \rho_i$.

U^κ-L Let $\Gamma_i = \Gamma'_i, E : (\mu X.\phi)^\kappa$, and let κ' be fresh. Since $E : (\mu X.\phi)^\kappa$ is validated by ρ_i, by Theorem 2.1 we find a β such that $\beta < \rho_i(\kappa)$, and $E : \phi[(\mu X.\phi)^{\kappa'}/X]$ is validated by $\rho_{i+1} = \rho_i[\beta/\kappa']$.

U^κ-R Let $\Delta_i = E : (\mu X.\phi)^\kappa, \Delta$. The two predecessors of N_i are labelled $\Gamma_i \vdash \kappa' < \kappa, \Delta'_i$ and $\Gamma_i \vdash E : \phi[(\mu X.\phi)^{\kappa'}/X], \Delta'_i$, respectively. Either $\kappa' < \kappa$ or $E : \phi[(\mu X.\phi)^{\kappa'}/X]$ must be invalidated by ρ_i. Pick as N_{i+1} whichever applies, and let $\rho_{i+1} = \rho_i$.

The limit of this construction is a winning run for R, which contradicts the initial assumption $\Gamma \vdash_r \Delta$. Hence $\Gamma \vdash \Delta$ is valid. ■

5 Completeness

In this section we present a completeness result, by reduction to Kozen's axiomatization [10], for the fragment containing pure sequents containing pure formulas only. This fragment is as expressive as Kozen's system itself. Kozen's axiomatization was shown to be complete by Walukiewicz [16]. The basic judgement in Kozen's proof system is an equational assertion of the shape $\phi = \psi$ where ϕ and ψ are closed pure formulas. The intended meaning of the judgment $\phi = \psi$ is that $||\phi||\rho = ||\psi||\rho$ (for some ρ the choice of which is immaterial, as ϕ and ψ are closed).

The judgment $\phi \leq \psi$ is an abbreviation of $\phi \wedge \psi = \phi$, and a formula ϕ is *Kozen provable*, $\vdash_{\mathit{Koz}} \phi$, if the judgment $\phi = \mathit{true}$ is provable.

The axiomatization has the standard axioms and rules Eq of equational logic, and the rule of substitution:

$$\mathit{Subst} \quad \frac{\phi_1 = \phi_2}{\phi[\phi_1/X] = \phi[\phi_2/X]}$$

Besides this the axiomatization has the following six axioms and rules where we, as above, assume that U has the generic shape $\mu X.\phi$:

$$\text{K1}\quad \frac{\cdot}{\phi = \psi}\quad \phi = \psi \text{ ax. Bool. alg.} \qquad \text{K2}\quad \frac{\cdot}{\langle\alpha\rangle\phi \vee \langle\alpha\rangle\psi = \langle\alpha\rangle(\phi \vee \psi)}$$

$$\text{K3}\quad \frac{\cdot}{\langle\alpha\rangle\phi \wedge [\alpha]\psi \leq \langle\alpha\rangle(\phi \wedge \psi)} \qquad \text{K4}\quad \frac{\cdot}{\langle\alpha\rangle \mathit{false} = \mathit{false}}$$

$$\text{K5}\quad \frac{\cdot}{\phi[U/X] \leq U} \qquad \text{K6}\quad \frac{\phi[\psi/X] \leq \psi}{U \leq \psi}$$

To show completeness using Kozen's axiomatization, by soundness, Theorem 4.5, it suffices to show that $x : \phi \vdash_r x : \psi$ whenever $\vdash_{\mathit{Koz}} \phi \leq \psi$, keeping in mind that this argument applies to pure formulas only.

The proof is to large measure routine. The exception is the rule of substitution which we establish by way of the following lemma:

Lemma 5.1
Let ϕ be any pure formula. Then $x : \phi \vdash x : \phi$ is refutation-game provable, so that whenever Ax is applied to a sequent of the shape $\Gamma, y : \psi \vdash y : \psi, \Delta$ then ψ is a propositional variable.

Proof. We generalize the statement of the lemma somewhat, to judgments of the form $\Gamma, x : \phi \vdash x : \psi$ where Γ is a set of ordinal constraints, such that the following (rather severe) conditions are satisfied:

1. ϕ and ψ are identical up to
   (a) the replacement of positive unindexed occurrences of formulas U in ψ by indexed ones in ϕ,
   (b) the replacement of negative unindexed occurrences of formulas U in ϕ by indexed ones in ψ.
2. Whenever $\kappa < \kappa' \in \Gamma$ then κ' does not appear in ϕ.
3. If U^κ and V^κ are both subformulas of ϕ then $U = V$.

Say that a judgment satisfying these three conditions is *admissible* (cf. admissibility condition of [12]). Assume that $\Gamma_0, x_0 : \phi_0 \vdash x_0 : \psi_0$ is admissible. We build an arena, $\mathcal{A}$, for $\Gamma_0, x_0 : \phi_0 \vdash x_0 : \psi_0$, consisting of admissible sequents only. Assume the construction of $\mathcal{A}$ has reached the node $N(\Gamma, x : \phi \vdash x : \psi)$. If $N(\Gamma, x : \phi \vdash x : \psi)$ is a repeat of some $N_1(\Gamma_1, x_1 : \phi_1 \vdash x_1 : \psi_1)$ in such a manner that $\phi = U^\kappa$ (so that $\phi_1 = U_1^{\kappa_1}$) and $\Gamma \vdash \kappa < \kappa_1$ then N is said to be *terminal by* N_1 and is not developed further. Otherwise the construction proceeds by induction on the structure of ϕ, refining N by turning N into the conclusion of an inference rule instance.

- If $\phi = X$ then $\psi = X$ and N is refined by rule Ax.
- If $\phi = \phi_1 \vee \phi_2$ then ψ has the shape $\psi_1 \vee \psi_2$ and we use $\vee$-L, $\vee$-R, and W-R to reduce to the goals $\Gamma, x : \phi_1 \vdash x : \psi_1$ and $\Gamma, x : \phi_2 \vdash x : \psi_2$.
- If $\phi = \neg\phi_1'$ then $\psi = \neg\psi_1'$ and the refinement uses $\neg$-L and $\neg$-R.
- If $\phi = \langle\alpha\rangle\phi'$ then ψ is of the form $\psi = \langle\alpha\rangle\psi'$, and we use $\langle\alpha\rangle$-L, $\langle\alpha\rangle$-R, W-L, and Ax to reduce to $\Gamma, x' : \phi' \vdash x' : \psi'$ where x' is fresh.
- If $\phi = \mu X.\phi' = U$ then ψ has the shape $\mu X.\psi' = U'$, and we use U-L to refine to $\Gamma, x : U^\kappa \vdash x : U'$.

- If $\phi = (\mu X.\phi')^\kappa = U^\kappa$ then, due to the syntactic monotonicity restriction on formulas, ψ has the shape $\mu X.\psi' = U'$, and we use U^κ-L and U-R to refine to $\Gamma, \kappa' < \kappa, x : \phi'[U^{\kappa'}/X] \vdash x : \psi'[U'/X]$ where κ' is fresh.

Observe that admissibility is preserved by each step in the above construction. The construction of $\mathcal{A}$ terminates, by an argument similar to the termination argument of [15], and hence produces an arena. Conditions (1)–(3) above are used to extend a substitution matching the property assertions, when this is possible, to a substitution matching the entire judgment.

Suppose now, for a contradiction, that the refuter has a winning run in $\mathcal{G}(\mathcal{A})$. That run has the shape $\Pi = (N_0, \rho_0), (N_1, \rho_1), \ldots, (N_i, \rho_i), \ldots$. There must be some arena node N which is minimal with respect to $<$ which appears infinitely often in Π. Suppose for simplicity that $N_0 = N$. Thus N_0 is labelled by a judgment of the shape $\Gamma_0, x : U_0^{\kappa_0} \vdash x : \psi_0$, and N_1 will hence be labelled by $\Gamma_0, \kappa_1 < \kappa_0, x : \phi_0[U_0^{\kappa_1}/X] \vdash x : \psi_0$. Let i be minimal such that N_i is labelled $\Gamma_i, \kappa_1 < \kappa_0, x' : U_0^{\kappa_1} \vdash x' : \psi_i$ and is terminal by N. By the definition of terminal by N, N_i is a repeat of $N = N_0$. Hence there is a substitution σ such that $U_i^{\kappa_1} = (U_0^{\kappa_0})\,\sigma$, implying $\kappa_1 = \kappa_0\,\sigma$. Furthermore, $N_{i+1} = N$ and $\rho_{i+1} = \rho_i \circ \sigma$. Since the interpretations $\rho_0, \rho_1, \ldots, \rho_i$ agree on common free variables we have $\rho_i(\kappa_0) = \rho_0(\kappa_0)$, and since N_i is non-trivial for ρ_i we have $\rho_i(\kappa_1) < \rho_i(\kappa_0)$. And since $\rho_{i+1} = \rho_i \circ \sigma$ we have $\rho_{i+1}(\kappa_0) = \rho_i(\kappa_1) < \rho_i(\kappa_0) = \rho_0(\kappa_0)$, i.e. $\rho_{i+1}(\kappa_0) < \rho_0(\kappa_0)$. But ordinals are well-founded and therefore Π must be finite, thus yielding a contradiction. The proof of Lemma 5.1 is thus complete. ■

COROLLARY 5.2
Let $\phi(X)$ be any pure formula and ψ_1, ψ_2 be formulas. If $x : \psi_1 \vdash_r x : \psi_2$ then $x : \phi[\psi_1/X] \vdash_r x : \phi[\psi_2/X]$.

PROOF. The proof of $x : \phi[\psi_1/X] \vdash x : \phi[\psi_2/X]$ is obtained by a simple composition of the proof of $x : \phi \vdash x : \phi$ obtained by Lemma 5.1 with the assumed proof of $x : \psi_1 \vdash x : \psi_2$. ■

The notion of refutation-game provability extends to equations as follows: $\vdash_r \phi = \psi$ iff $x : \phi \vdash_r x : \psi$ and $x : \psi \vdash_r x : \phi$. As a result, $\vdash_r \phi \leq \psi$ iff $x : \phi \vdash_r x : \psi$.

LEMMA 5.3
If ϕ is pure and $\vdash_{Koz} \phi$ then $\vdash_r x : \phi$.

PROOF. We show that all inference rules of Kozen's axiomatization are *admissible* in our proof system, i.e., that the conclusion to a Kozen-rule is refutation-game provable whenever its premisses are. The result then follows by induction on the size of the Kozen-proof of ϕ.

EQ Use AX and CUT.

SUBST This case is a direct consequence of Corollary 5.2.

K1 It suffices to note that for Boolean formulas our proof system reduces to the standard sequent-style formulation.

K2 To show $x : \langle\alpha\rangle\phi \vee \langle\alpha\rangle\psi \vdash_r x : \langle\alpha\rangle(\phi \vee \psi)$ first use $\vee$-L to reduce $x : \langle\alpha\rangle\phi \vee \langle\alpha\rangle\psi \vdash x : \langle\alpha\rangle(\phi \vee \psi)$ to the two sequents $x : \langle\alpha\rangle\phi \vdash x : \langle\alpha\rangle(\phi \vee \psi)$ and $x : \langle\alpha\rangle\psi \vdash x : \langle\alpha\rangle(\phi \vee \psi)$. These are handled in the same fashion; e.g., use $\langle\alpha\rangle$-L to reduce the first sequent to $x \xrightarrow{\alpha} y, y : \phi \vdash x : \langle\alpha\rangle(\phi \vee \psi)$. By $\langle\alpha\rangle$-R this sequent reduces to $x \xrightarrow{\alpha} y, y : \phi \vdash x \xrightarrow{\alpha} y$ and $x \xrightarrow{\alpha} y, y : \phi \vdash y : \phi \vee \psi$. The result is then obtained by applying AX, and $\vee$-R followed by AX, respectively. $x : \langle\alpha\rangle(\phi \vee \psi) \vdash_r x : \langle\alpha\rangle\phi \vee \langle\alpha\rangle\psi$ can be shown in a similar fashion.

K3, K4 These are proved as easily as K2.

K5 $x : \phi[U/X] \vdash_r x : U$ is shown by applying rule U-R followed by AX.

K6 Assume $x : \phi[\psi/X] \vdash_r x : \psi$. We have to show $x : U \vdash_r x : \psi$, so we start with $x : U \vdash x : \psi$. By rule U-L this sequent reduces to

$$x : U^{\kappa} \vdash x : \psi \tag{5.1}$$

Applying rule CUT with $x : \phi[\psi/X]$, followed by weakening, yields the two sequents $x : \phi[\psi/X] \vdash x : \psi$ and $x : U^{\kappa} \vdash x : \phi[\psi/X]$, the first of which is refutation-game provable by assumption. The latter sequent can be reduced by rule U^{κ}-L to $\kappa' < \kappa, x : \phi[U^{\kappa'}/X] \vdash x : \phi[\psi/X]$. Using the construction of Lemma 5.1, this sequent is reduced to a (possibly empty) set of proof goals of the shape $\Gamma, \kappa' < \kappa, x' : U^{\kappa'} \vdash x' : \psi$, all of which are repeats of sequent (5.1) up to substitution $\sigma = [x \mapsto x', \kappa \mapsto \kappa']$. It is easy to see that the refuter can have no winning run in the arena obtained, and hence the arena proves $x : U \vdash_r x : \psi$. ■

Since Kozen's axiomatization is complete, and hence $\vdash_{Koz} \phi$ whenever the pure formula ϕ is valid, we obtain a completeness result for our proof system through the following theorem. Recall that a sequent is called pure if it only contains satisfaction assertions of the shape $x : \phi$.

THEOREM 5.4 (Completeness)
If the pure sequent $\Gamma \vdash \Delta$ involving pure formulas only is valid, then $\Gamma \vdash_r \Delta$.

PROOF. Let the sequent $\Gamma \vdash \Delta$ involving pure formulas only be pure and valid. The process variables occurring in the sequent induce a partitioning on the satisfaction assertions, and thereby induce a set of sequents $\Gamma_1 \vdash \Delta_1, \ldots, \Gamma_k \vdash \Delta_k$, where $\Gamma_i \subseteq \Gamma$ and $\Delta_i \subseteq \Delta$ for $1 \leq i \leq k$, each sequent mentioning one process variable only. At least one of these sequents must be valid, since otherwise there would be environments $\rho_1, \ldots, \rho_k$ invalidating $\Gamma_1 \vdash \Delta_1$, $\ldots, \Gamma_k \vdash \Delta_k$, respectively, and hence the union ρ of $\rho_1, \ldots, \rho_k$ (well-defined since $\rho_1, \ldots,$ ρ_k assign values to disjoint sets of variables only) would invalidate $\Gamma \vdash \Delta$ thus contradicting the validity assumption.

Let $x : \phi_1, \ldots, x : \phi_m \vdash x : \psi_1, \ldots, x : \psi_n$ be one such valid sequent. Then, by propositional logic, the sequent $\vdash x : \theta$ is valid where θ is the pure formula $\neg\phi_1 \vee \cdots \vee \neg\phi_m \vee \psi_1 \vee \cdots \vee \psi_n$. ¿From completeness of Kozen's system it follows that $\vdash_{Koz} x : \theta$, implying $\vdash_r x : \theta$ by Lemma 5.3. But $\Gamma \vdash \Delta$ is reducible by applying the two weakening rules to $x : \phi_1, \ldots, x : \phi_m \vdash x : \psi_1, \ldots, x : \psi_n$, which is in turn reducible by applying one cut followed by propositional logic rules to $\vdash x : \theta$, and hence $\Gamma \vdash_r \Delta$. ■

6 Discharge conditions

The refutation game described in Section 4 gives an abstract condition for when an arena can be considered a proof. Due to its generality, however, it is not suitable for the practical purposes of proving validity of a sequent. Rather, it can be used for justifying simpler, possibly not even complete, conditions for accepting an arena as a proof.

DEFINITION 6.1 (Discharge condition)
A *discharge condition* is a sufficient condition for an arena to be a proof.

Let us fix an arena $\mathcal{A}$ with non-axiom leaves $N_1, \ldots, N_n$, and associated nodes $N_1', \ldots,$ N_n' and substitutions $\sigma_1, \ldots, \sigma_n$, respectively. Recall that $N_1, \ldots, N_n$ are thus repeats of N_1',

..., N'_n up to σ_1, ..., σ_n. We refer to N_i as *discharge nodes*, and to N'_i as the corresponding *companion nodes*. We call two discharge nodes *related* iff they are members of one and the same strongly connected component C of the directed graph obtained from the arena by identifying the discharge nodes with their respective companions.

DEFINITION 6.2 (Progress, Preservation)
Let $\pi(N_i, N'_i)$ be the path in $\mathcal{A}$ from the discharge node $\Gamma_i \vdash \Delta_i$ denoted N_i to its companion node $\Gamma'_i \vdash \Delta'_i$ denoted N'_i, and let κ be an ordinal variable. We say that:

1. $\pi(N_i, N'_i)$ *progresses* on ordinal variable κ with the substitution σ_i if some approximated fixed-point formula U^κ occurs in N'_i, and $\Gamma_i \vdash \kappa\sigma_i < \kappa$ is derivable;
2. $\pi(N_i, N'_i)$ *preserves* κ with σ_i if some approximated fixed-point formula U^κ occurs in N'_i, and either $\kappa\sigma_i = \kappa$, or else $\Gamma_i \vdash \kappa\sigma_i < \kappa$ is derivable.

We now present one concrete discharge condition. In essence, it guarantees well-foundedness of proofs through well-foundedness of ordinal constraints.

DEFINITION 6.3 (DC)
Arena $\mathcal{A}$ satisfies condition DC iff for every strongly connected component C of $\mathcal{A}$ there is a linearization N_{i_1}, ..., N_{i_m} of the discharge nodes in C, and there are ordinal variables κ_1, ..., κ_m such that for all j, $1 \leq j \leq m$,

1. $\pi(N_{i_j}, N'_{i_j})$ progresses on κ_j with σ_{i_j}, and
2. $\pi(N_{i_j}, N'_{i_j})$ preserves κ_l with σ_{i_j} for all l such that $j < l \leq m$.

We show that DC is indeed a discharge condition in the sense of Definition 6.1.

THEOREM 6.4
If $\mathcal{A}$ is an arena satisfying condition DC then $\mathcal{A}$ is a proof.

PROOF. (Sketch) By contradiction. Suppose that $\mathcal{A}$ is an arena satisfying condition DC, and suppose that there is a winning run Π for refuter R on $\mathcal{G}(\mathcal{A})$. Then Π eventually stabilizes in a strongly connected component C. By condition DC there is a linearization N_{i_1}, ..., N_{i_m} of the discharge nodes appearing in C. Conditions 1 and 2 of Definition 6.3 imply that the value given by the interpretation functions on Π to the tuple $(\kappa_1, \ldots, \kappa_m)$ of ordinal variables lexicographically decreases whenever Π passes from a repeat node to a companion node in C. Hence, this value decreases infinitely often along Π, a contradiction to the well-foundedness of ordinals. ■

THEOREM 6.5 (DC completeness)
If the pure sequent $\Gamma \vdash \Delta$ involving pure formulas only is valid, then it is provable using the rules for logical entailment (cf. Section 3) together with DC taken as a rule of assumption discharge.

PROOF. (Sketch) To obtain the completeness result of Theorem 5.4, we twice made an argument about an arena being a proof: once in the proof of Lemma 5.1, and once in the proof of Lemma 5.3. The result follows from the observation that condition DC applies on both occasions, with linearizations as induced by the syntax-tree of the corresponding formula ϕ. ■

7 Examples

The following two examples illustrate the use of discharge condition DC for establishing validity of a sequent.

EXAMPLE 7.1
Assume an agent has the property that every sequence of a or b actions that it can engage in contains only finitely many a actions. This property can be formalized as $\mu X.\nu Y.[a]X \wedge [b]Y$ or, equivalently, as $\mu X.\neg\mu Y.\langle a\rangle\neg X \vee \langle b\rangle Y$ using the basic connectives only. Then this agent surely has also the property that every sequence of a or b actions that it can engage in contains only finite-length subsequences of a actions; a property formalizable as $\nu Y.\mu X.[a]X \wedge [b]Y$ or, equivalently, as $\neg\mu Y.\neg\mu X.\neg(\langle a\rangle\neg X \vee \langle b\rangle Y)$. To prove that the first property logically entails the second we construct an arena rooted at sequent $x : \phi \vdash x : \neg\psi$, using the following abbreviations:

$$\begin{array}{ll} \phi = \mu X.\neg\mu Y.\langle a\rangle\neg X \vee \langle b\rangle Y & \psi = \mu Y.\neg\mu X.\neg(\langle a\rangle\neg X \vee \langle b\rangle Y) \\ \phi_1 = \mu Y.\langle a\rangle\neg\phi^{\kappa'_\phi} \vee \langle b\rangle Y & \psi_1 = \mu X.\neg(\langle a\rangle\neg X \vee \langle b\rangle\psi^{\kappa'_\psi}) \\ \phi_2 = \mu Y.\langle a\rangle\neg\phi^{\kappa''_\phi} \vee \langle b\rangle Y & \psi_2 = \mu X.\neg(\langle a\rangle\neg X \vee \langle b\rangle\psi^{\kappa''_\psi}). \end{array}$$

The arena is shown in Figure 1, with the companion node, which is common to both discharge nodes, being indicated by $\vdash_*$ in the sequent, and substitutions $\sigma_L = [x \mapsto x', \kappa_\phi \mapsto \kappa'_\phi, \kappa'_\phi \mapsto \kappa''_\phi]$ and $\sigma_R = [x \mapsto x', \kappa_\psi \mapsto \kappa'_\psi, \kappa'_\psi \mapsto \kappa''_\psi]$ for the left and right leaves, respectively. Annotation RS1 stands for $\neg$-R, $\vee$-R, $\vee$-L, W-R, and RS2 for $\langle\alpha\rangle$-L, $\langle\alpha\rangle$-R, AX, W-L.

This arena satisfies condition DC for any choice of linearization of the two discharge nodes, since the left path progresses on κ'_ϕ and preserves κ'_ψ, and the right path progresses on κ'_ψ and preserves κ'_ϕ. The arena is hence a proof of $x : \phi \vdash x : \neg\psi$.

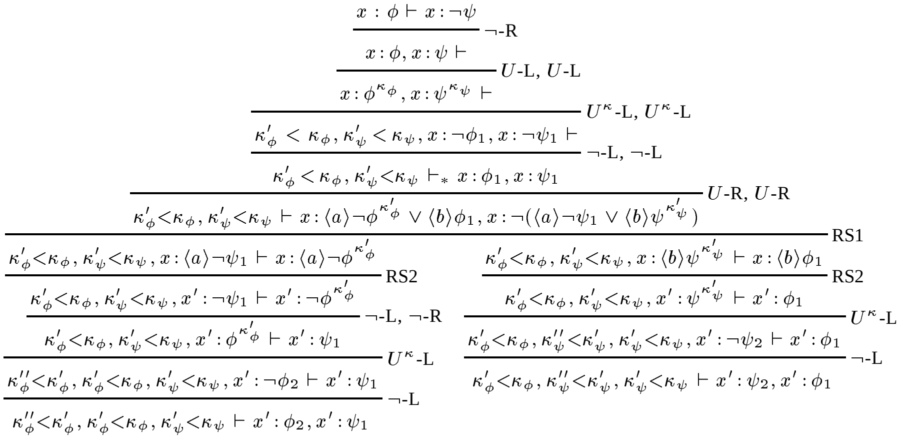
$x : \phi \vdash x : \neg\psi$
$\neg$-R
$x : \phi, x : \psi \vdash$
U-L, U-L
$x : \phi^{\kappa_\phi}, x : \psi^{\kappa_\psi} \vdash$
U^κ-L, U^κ-L
$\kappa'_\phi < \kappa_\phi, \kappa'_\psi < \kappa_\psi, x : \neg\phi_1, x : \neg\psi_1 \vdash$
$\neg$-L, $\neg$-L
$\kappa'_\phi < \kappa_\phi, \kappa'_\psi < \kappa_\psi \vdash_* x : \phi_1, x : \psi_1$
U-R, U-R
$\kappa'_\phi < \kappa_\phi, \kappa'_\psi < \kappa_\psi \vdash x : \langle a\rangle\neg\phi^{\kappa'_\phi} \vee \langle b\rangle\phi_1, x : \neg(\langle a\rangle\neg\psi_1 \vee \langle b\rangle\psi^{\kappa'_\psi})$
RS1
$\kappa'_\phi < \kappa_\phi, \kappa'_\psi < \kappa_\psi, x : \langle a\rangle\neg\psi_1 \vdash x : \langle a\rangle\neg\phi^{\kappa'_\phi}$
$\kappa'_\phi < \kappa_\phi, \kappa'_\psi < \kappa_\psi, x : \langle b\rangle\psi^{\kappa'_\psi} \vdash x : \langle b\rangle\phi_1$
RS2
RS2
$\kappa'_\phi < \kappa_\phi, \kappa'_\psi < \kappa_\psi, x' : \neg\psi_1 \vdash x' : \neg\phi^{\kappa'_\phi}$
$\kappa'_\phi < \kappa_\phi, \kappa'_\psi < \kappa_\psi, x' : \psi^{\kappa'_\psi} \vdash x' : \phi_1$
$\neg$-L, $\neg$-R
U^κ-L
$\kappa'_\phi < \kappa_\phi, \kappa'_\psi < \kappa_\psi, x' : \phi^{\kappa'_\phi} \vdash x' : \psi_1$
$\kappa'_\phi < \kappa_\phi, \kappa''_\psi < \kappa'_\psi, \kappa'_\psi < \kappa_\psi, x' : \neg\psi_2 \vdash x' : \phi_1$
U^κ-L
$\neg$-L
$\kappa''_\phi < \kappa'_\phi, \kappa'_\phi < \kappa_\phi, \kappa'_\psi < \kappa_\psi, x' : \neg\phi_2 \vdash x' : \psi_1$
$\kappa'_\phi < \kappa_\phi, \kappa''_\psi < \kappa'_\psi, \kappa'_\psi < \kappa_\psi \vdash x' : \psi_2, x' : \phi_1$
$\neg$-L
$\kappa''_\phi < \kappa'_\phi, \kappa'_\phi < \kappa_\phi, \kappa'_\psi < \kappa_\psi \vdash x' : \phi_2, x' : \psi_1$

FIGURE 1. Arena for $x : \phi \vdash x : \neg\psi$

EXAMPLE 7.2
Let us now attempt to prove the opposite, namely validity of sequent $x : \neg\psi \vdash x : \phi$. Intuitively this should fail, but let us see how close one can come to a proof. We use the abbreviations:

$$\begin{array}{ll} \phi = \mu X.\neg\mu Y.\langle a\rangle\neg X \vee \langle b\rangle Y & \psi = \mu Y.\neg\mu X.\neg(\langle a\rangle\neg X \vee \langle b\rangle Y) \\ \phi_1 = \mu Y.\langle a\rangle\neg\phi \vee \langle b\rangle Y & \psi_1 = \mu X.\neg(\langle a\rangle\neg X \vee \langle b\rangle\psi). \end{array}$$

The arena is shown in Figure 2. The two substitutions are $\sigma_L = [x \mapsto x', \kappa_\phi \mapsto \kappa''_\phi, \kappa_\psi \mapsto \kappa'_\psi]$ and $\sigma_R = [x \mapsto x', \kappa_\phi \mapsto \kappa'_\phi, \kappa_\psi \mapsto \kappa''_\psi]$. Annotation RS1 stands here for $\neg$-L, $\vee$-R, $\vee$-L, W-R, and RS2 is as in the previous example.

This arena does not satisfy condition DC for any choice of linearization of the two discharge nodes, since the left path progresses on κ_ψ but does not preserve κ_ϕ, and the right path progresses on κ_ϕ but does not preserve κ_ψ.

$$\begin{array}{cl}
x:\neg\psi \vdash x:\phi & \\ \hline
\vdash x:\phi, x:\psi & \neg\text{-L} \\ \hline
\vdash x:\neg\phi_1 , x:\neg\psi_1 & U\text{-R, } U\text{-R} \\ \hline
x:\phi_1 , x:\psi_1 \vdash & \neg\text{-R, } \neg\text{-R} \\ \hline
x:\phi_1^{\kappa_\phi} , x:\psi_1^{\kappa_\psi} \vdash_* & U\text{-L, } U\text{-L} \\ \hline
\kappa'_\phi{<}\kappa_\phi , \kappa'_\psi{<}\kappa_\psi \, x:\langle a\rangle\neg\phi \vee \langle b\rangle\phi_1^{\kappa'_\phi} , x:\neg(\langle a\rangle\neg\psi_1^{\kappa'_\psi} \vee \langle b\rangle\psi) \vdash & U^\kappa\text{-L, } U^\kappa\text{-L} \\ \hline
\begin{array}{cl}
\kappa'_\phi{<}\kappa_\phi , \kappa'_\psi{<}\kappa_\psi , x:\langle a\rangle\neg\phi \vdash x:\langle a\rangle\neg\psi_1^{\kappa'_\psi} & \\ \hline
\kappa'_\phi{<}\kappa_\phi , \kappa'_\psi{<}\kappa_\psi , x':\neg\phi \vdash x':\neg\psi_1^{\kappa'_\psi} & \text{RS2} \\ \hline
\kappa'_\phi{<}\kappa_\phi , \kappa'_\psi{<}\kappa_\psi , x':\psi_1^{\kappa'_\psi} \vdash x':\phi & \neg\text{-L, } \neg\text{-R} \\ \hline
\kappa'_\phi{<}\kappa_\phi , \kappa'_\psi{<}\kappa_\psi , x':\psi_1^{\kappa'_\psi} \vdash x':\neg\phi_1 & U\text{-R} \\ \hline
\kappa'_\phi{<}\kappa_\phi , \kappa'_\psi{<}\kappa_\psi , x':\phi_1 , x':\psi_1^{\kappa'_\psi} \vdash & \neg\text{-R} \\ \hline
\kappa'_\phi{<}\kappa_\phi , \kappa'_\psi{<}\kappa_\psi , x':\phi_1^{\kappa''_\phi} , x':\psi_1^{\kappa'_\psi} \vdash & U\text{-L}
\end{array}
\qquad
\begin{array}{cl}
\kappa'_\phi{<}\kappa_\phi , \kappa'_\psi{<}\kappa_\psi , x:\langle b\rangle\phi_1^{\kappa'_\phi} \vdash x:\langle b\rangle\psi & \\ \hline
\kappa'_\phi{<}\kappa_\phi , \kappa'_\psi{<}\kappa_\psi , x':\phi_1^{\kappa'_\phi} \vdash x':\psi & \text{RS2} \\ \hline
\kappa'_\phi{<}\kappa_\phi , \kappa'_\psi{<}\kappa_\psi , x':\phi_1^{\kappa'_\phi} \vdash x':\neg\psi_1 & U\text{-R} \\ \hline
\kappa'_\phi{<}\kappa_\phi , \kappa'_\psi{<}\kappa_\psi , x':\phi_1^{\kappa'_\phi} , x':\psi_1 \vdash & \neg\text{-R} \\ \hline
\kappa'_\phi{<}\kappa_\phi , \kappa'_\psi{<}\kappa_\psi , x':\phi_1^{\kappa'_\phi} , x':\psi_1^{\kappa''_\psi} \vdash & U\text{-L}
\end{array} & \text{RS1}
\end{array}$$

FIGURE 2. Arena for $x:\neg\psi \vdash x:\phi$

8 Conclusion

We presented a Gentzen-style sequent calculus for program verification suitable for both model checking-like verification based on global state space exploration, and compositional reasoning. Its novelty lies in the generality of the proof judgements allowing parametric and compositional reasoning, in the complex setting of the modal μ-calculus. This is achieved, in part, by the use of explicit fixed point ordinal approximations, and in part by a complete separation, following Simpson [12], in the proof system of the rules concerning the logic from the rules encoding the operational semantics of the process language. We addressed, in a game-based manner, the semantical basis of this approach, and presented a soundness and a completeness result as it applies to the entailment subproblem.

The introduction of explicit variables for states and approximation ordinals seems useful and semantically clear from the point of view of practical verification. This has been confirmed by experience with developing a toolset for practical verification of concurrent programs written in the Erlang programming language [6, 4, 8, 9]. On the other hand, it also raises foundational issues not addressed in this paper, such as various completeness and decidability problems. For instance, can explicit state and ordinal variables be helpful in obtaining a direct completeness proof, i.e. without reduction to Kozen's system, of the proof system for logical entailment considered here, which is simpler than the one by Walukiewicz [16]? And how general a completeness result can be obtained for settings involving structured states?

Acknowledgements

The authors are indebted to Christoph Sprenger from EPFL Lausanne and the two anonymous reviewers for many helpful suggestions. Both authors were supported by a Swedish Foundation for Strategic Research Junior Individual Grant. The second author was also supported by the Swedish National Board for Technical and Industrial Development (NUTEK) through the ASTEC competence centre.

References

[1] R. Amadio and M. Dam. Reasoning about higher-order processes. In *Proceedings of CAAP'95,* Lecture Notes in Computer Science, Volume 915, pp. 202–217, Springer Verlag, Berlin, 1995.

[2] R. Amadio and M. Dam. A modal theory of types for the π-calculus. In *Proceedings of FTRTFT'96,* Lecture Notes in Computer Science, Volume 1135, pp. 347–365, Springer Verlag, Berlin, 1996.

[3] J. L. Armstrong, M. C. Williams, C. Wikström, and S. R. Virding. *Concurrent Programming in Erlang*. Prentice Hall, 2nd edition, 1995.

[4] T. Arts and M. Dam. Verifying a distributed database lookup manager written in Erlang. In *Proceedings of Formal Methods Europe'99,* Lecture Notes in Computer Science, Volume 1708, pp. 682–700, Springer Verlag, Berlin, 1999.

[5] M. Dam. Proving properties of dynamic process networks. *Information and Computation*, **140**, 95–114, 1998.

[6] M. Dam, L.-å. Fredlund, and D. Gurov. Toward parametric verification of open distributed systems. In *Compositionality: the Significant Difference,* H. Langmaack, A. Pnueli and W.-P. de Roever, eds., , pp. 150–185. Lecture Notes in Computer Science, Volume 1536, Springer Verlag, Berlin, 1998.

[7] M. Dam and D. Gurov. Compositional verification of CCS processes. In *Proceedings of PSI'99,* Lecture Notes in Computer Science, Volume 1755, pp. 247–256, Springer Verlag, Berlin, 2000.

[8] L.-å. Fredlund and D. Gurov. A framework for formal reasoning about open distributed systems. In *Proceedings of ASIAN'99,* Lecture Notes in Computer Science, Volume 1742, pp. 87–100, Springer Verlag, Berlin, 1999.

[9] L.-å. Fredlund, D. Gurov, T. Noll, M. Dam, T. Arts, and G. Chugunov. A verification tool for Erlang. *Software Tools for Technology Transfer*, 2001. Accepted for publication. EVT is available from `http://www.sics.se/fdt/VeriCode/evt.html`.

[10] D. Kozen. Results on the propositional μ-calculus. *Theoretical Computer Science*, **27**, 333–354, 1983.

[11] D. Niwinski and I. Walukiewicz. Games for the μ-calculus. *Theoretical Computer Science,* **163**, 99–116, 1997.

[12] A. Simpson. Compositionality via cut-elimination: Hennessy–Milner logic for an arbitrary GSOS. In *Proc. LICS*, pp. 420–430, 26–29, 1995.

[13] C. Stirling. Modal logics for communicating systems. *Theoretical Computer Science*, **49**, 311–347, 1987.

[14] C. Stirling. Games and modal mu-calculus. In *Lecture Notes in Computer Science,* Volume 1055, pp. 298–312, Springer Verlag, Berlin, 1996.

[15] C. Stirling and D. Walker. Local model checking in the modal mu-calculus. *Theoretical Computer Science,* **89**, 161–177, 1991.

[16] I. Walukiewicz. Completeness of Kozen's axiomatisation of the propositional mu-calculus. In *Proc. LICS'95,* pp. 14–24, 1995.

[17] G. Winskel. A complete proof system for SCCS with modal assertions. *Fundamentae Informaticae,* **IX**, 401–420, 1986.

[18] G. Winskel. A note on model checking the modal ν-calculus. *Theoretical Computer Science,* **83**, 157–187, 1991.

Received September 2000