# Verifying Temporal Properties Using Explicit Approximants: Completeness for Context-free Processes

Ulrich Schöpp and Alex Simpson

LFCS, Division of Informatics, University of Edinburgh
JCMB, King's Buildings, Edinburgh, EH9 3JZ

**Abstract.** We present a sequent calculus for formally verifying modal $\mu$-calculus properties of concurrent processes. Building on work by Dam and Gurov, the proof system contains rules for the explicit manipulation of fixed-point approximants. We develop a new syntax for approximants, incorporating, in particular, modalities for approximant modification. We make essential use of this feature to prove our main result: the sequent calculus is complete for establishing arbitrary $\mu$-calculus properties of context-free processes.

## 1 Introduction

In this paper, we present a proof system for establishing temporal properties, expressed in the modal $\mu$-calculus [14], of concurrent processes. The proof system is a sequent calculus in which sequents have the form $\Gamma \vdash \Delta$, where $\Gamma$ and $\Delta$ are sets of assertions. As usual, a derivation of $\Gamma \vdash \Delta$ will establish that if all the assertions in $\Gamma$ hold then so does at least one assertion in $\Delta$. The principal assertion form is $p : \varphi$, which is the syntactic expression of the relation $p \models \varphi$, stating that process $p$ satisfies $\mu$-calculus property $\varphi$. The sequent-based formalism has several virtues:

1. Ordinary verification goals are expressed by sequents of the form $\vdash p : \varphi$.
2. More generally, by allowing process variables, *parameterized* verification goals can be expressed by sequents of the form

$$x_1 : \psi_1,\ \dots,\ x_n : \psi_n \vdash p(x_1, \dots, x_n) : \varphi. \tag{1}$$

   Such a sequent states that the process $p$ satisfies $\varphi$ whenever its parameters $x_1, \dots, x_n$ are instantiated with processes satisfying $\psi_1, \dots, \psi_n$ respectively.
3. Such parameterized goals can be used to support *compositional* reasoning. Using cut and substitution, one obtains a derived rule:[1]

$$\frac{\vdash p(q_1, \dots, q_n) : \varphi}{\vdash q_1 : \psi_1 \qquad \dots \qquad \vdash q_n : \psi_n \qquad x_1 : \psi_1,\ \dots,\ x_n : \psi_n \vdash p(x_1, \dots, x_n) : \varphi}$$

[1] In this paper, we write all inference rules and derivations in tableau form, i.e. with the goal (conclusion) on top and the subgoals (premises) underneath.

M. Nielsen and U. Engberg (Eds.): Fossacs 2002, LNCS 2303, pp. 372–386, 2002.
© Springer-Verlag Berlin Heidelberg 2002

This rule reduces the goal of establishing a property $\varphi$ of a compound process $p(q_1, \ldots, q_n)$ to the subgoals of establishing properties of its components $q_1, \ldots, q_n$ together with a further subgoal justifying the decomposition.

4. The proof system also supports a direct *structural* form of reasoning. The inference rules decompose logical connectives on the left and right of sequents in the familiar Gentzen style, allowing the construction of a derivation to be guided by the form of the goal sequent.

Such a sequent-based approach to process verification was proposed independently by Dam [4] and the second author [17], as a way of uniformly accounting for many specialist techniques for compositional reasoning that had appeared in the earlier literature, especially [18].

The paper [17] presents a sequent calculus, for establishing properties expressed in Hennessy-Milner logic [12], in which sequents contain a second form of assertion, *transition assertions* $p \xrightarrow{a} q$, expressing that process $p$ evolves to process $q$ under action $a$. This device allows the proof system to be adapted to any process calculus with an operational semantics in GSOS format [1]. The main results of [17] are strong completeness and cut-elimination for the system.

In [4,5,6,7,8,9,11], Dam and his co-workers address the interesting question of how best to incorporate fixed-point reasoning into such sequent-based proof systems. In their more recent research, see, in particular, [9], Dam and Gurov propose dealing with this issue by extending the $\mu$-calculus with *ordinal variables*, $\kappa$, which are semantically interpreted as ordinals, and by introducing new formulae $\mu^\kappa X.\, \varphi$ and $\nu^\kappa X.\, \varphi$ standing for the $\kappa$-th iterations in the chain of approximations to the fixed-points $\mu X.\, \varphi$ and $\nu X.\, \varphi$ respectively. This machinery allows a sound notion of proof to be defined, by identifying certain repeats of sequents in a derivation tree and by imposing a global *discharge condition* on a derivation tree, formulated in terms of ordinal variables.

As the first contribution of the present paper, we provide a new proof system for incorporating fixed-point reasoning into the sequent-calculus approach. Our system is strongly based on Dam and Gurov's idea of using explicit fixed-point approximants. However, we provide an alternative formulation of these, not requiring ordinal variables. Instead, we use ordinary propositional variables $X$ to range over approximants. To properly deal with such variables, we include an extra component on the left of sequents, a context $\mathsf{D}$ of approximant declarations. Such declarations have one of two forms: $X \leqslant \varphi$, which declares $X$ to be an approximant of $\mu X.\, \varphi$; and $X \geqslant \varphi$, which declares $X$ to be an approximant of $\nu X.\, \varphi$, see Sect. 2. Thus far, our approach can be seen as merely a less expressive reformulation of Dam and Gurov's syntax. However, we also extend the syntax of the $\mu$-calculus in two significant ways. First, we allow explicit approximant declarations in formulae, introducing two new formula constructions: $\langle X \leqslant \varphi \rangle\, \psi$, which says that there exists an approximant $X$ of $\mu X.\, \varphi$ such that $\psi$; and $[X \geqslant \varphi]\, \psi$, which says that $\psi$ holds for all approximants $X$ of $\nu X.\, \varphi$. Second, we incorporate modalities for approximant "modification" in formulae. If $X$ is an approximant for $\mu X.\, \varphi$ then the formula $\langle -X \rangle\, \psi$ expresses that there exists another approximant $X'$ of $\mu X.\, \varphi$ with $X' \subset X$ (proper inclusion) such that $\psi[X' / X]$. Dually, if $X$ is an approximant for $\nu X.\, \varphi$ then $[+X]\, \psi$ expresses that, for all approximants $X'$ of $\mu X.\,\varphi$ with $X' \supset X$ (proper containment), it holds that $\psi[\,X'\,/\,X]$.

The full proof system is presented in Sect. 3. The use of approximant variables and modifiers allows a straightforward definition of a global combinatorial condition for a derivation tree to be a proof. The soundness of the proof system is then established as Theorem 1.

It is our belief that the proof system we present provides a powerful and flexible tool for verifying a wide class of processes using a compositional style of reasoning. As the verification problem is, in general, undecidable, the proof system is necessarily incomplete, and so it is impossible to back up such a claim with an all-encompassing completeness theorem. Instead, there are two other avenues open for partially substantiating this belief. One is to demonstrate the effectivity of the system on a range of worked examples. Using proof systems closely related to ours, such an enterprise has already been undertaken by Dam, Gurov et al., who have presented applications to CCS [5,8], the $\pi$-calculus [6] and Erlang [7,11]. The second avenue is to obtain restricted completeness theorems. Once again, Dam, Gurov et al. have obtained such results, establishing completeness for sequents of the form $\vdash x : \varphi$, i.e. completeness with respect to $\mu$-calculus validity [9], and proving completeness for finite-state processes [5].

As the main contribution of the paper, Theorem 2, we present a significant extension of the latter result. We show that our proof system is complete for establishing $\mu$-calculus properties of arbitrary *context-free* processes, see e.g. [2].

Of course, many techniques for verifying context-free processes are already known. The decidability of the problem is a direct consequence of the work of Muller and Schupp, who established that full monadic second-order logic (MSOL) is decidable over the wider class of pushdown transition graphs [15]. The decision problem for MSOL is known to be of non-elementary complexity. However, for the special case of $\mu$-calculus properties, elementary decision algorithms have been given in [20,3]. Also, Hungar and Steffen showed how alternation-free $\mu$-calculus properties of context-free processes can be established by a tableau-style proof system embodying a form of compositional reasoning [13].

We stress, however, that the motivation behind the present paper is not merely to contribute one more method of verifying context-free processes to the literature. Indeed, in spite of their applications to dataflow analysis in languages with stack-based procedure calls [10], context-free processes are of limited relevance to the general problem of verifying concurrent systems. Rather, our motivation is to extend the scope of completeness results for proof systems whose full range of application is potentially much wider. Indeed, as far as we know, ours is the first completeness result for a general purpose proof system (i.e. one not tailored in advance to a restricted class of processes) with respect to any significant class of infinite state processes.

We would like to thank Dilian Gurov and the anonymous referees for their comments. For lack of space, in this conference version of the paper, proofs are either sketched or omitted.

## 2 Modal $\mu$-Calculus and Explicit Approximants

Our treatment of the $\mu$-calculus will be brief. The reader is referred to [19] for further details. We consider the $\mu$-calculus in positive normal form, with formulae defined by the grammar:

$$\varphi \ ::= \ X \mid \mathtt{ff} \mid \ \mathtt{tt} \mid \varphi_1 \vee \varphi_2 \mid \varphi_1 \wedge \varphi_2 \mid \langle a \rangle \varphi \mid [a]\varphi \mid \mu X.\, \varphi \mid \nu X.\, \varphi.$$

Here $a$ ranges over a given set $A$ of action symbols. Free and bound variables are defined as usual, and we identify formulae up to renaming of bound variables. We write $FV(\varphi)$ for the set of free variables of $\varphi$, and we say that $\varphi$ is *closed* if $FV(\varphi) = \emptyset$. The negation of a closed formula can be defined by induction on its structure using De Morgan duals.

Formulae are interpreted over a transition system $(T, \{\xrightarrow{a}\}_{a \in A})$ (here $T$ is a set of states and each $\xrightarrow{a}$ is a binary relation on $T$). A formula $\varphi$ is interpreted relative to an *environment* $V$ mapping $FV(\varphi)$ to subsets of $T$, with its interpretation $\|\, \varphi \,\|_V \subseteq T$ defined as in [19].

Next we introduce approximants. Rather than invoking the set-theoretic machinery of ordinal indices, we give a definition that is directly interpretable in monadic third-order logic.

**Definition 1 ($\mu$- and $\nu$-approximants).** For any least-fixed-point formula $\mu X.\, \varphi$, its family of $\mu$*-approximants* $\mathcal{A}_V^{\mu X.\, \varphi}$, relative to an environment $V$ defined on $FV(\mu X.\, \varphi)$, is the smallest family of subsets of $T$ satisfying:

1. if $\mathcal{A}' \subseteq \mathcal{A}_V^{\mu X.\, \varphi}$ then $\bigcup \mathcal{A}' \in \mathcal{A}_V^{\mu X.\, \varphi}$, and
2. if $S \in \mathcal{A}_V^{\mu X.\, \varphi}$ then $\|\, \varphi \,\|_{V[\,S\,/\,X]} \in \mathcal{A}_V^{\mu X.\, \varphi}$.

For any greatest-fixed-point formula $\nu X.\, \varphi$, its family of $\nu$*-approximants* $\mathcal{A}_V^{\nu X.\, \varphi}$ relative to $V$ is the smallest family of subsets of $T$ satisfying:

1. if $\mathcal{A}' \subseteq \mathcal{A}_V^{\nu X.\, \varphi}$ then $\bigcap \mathcal{A}' \in \mathcal{A}_V^{\nu X.\, \varphi}$, and
2. if $S \in \mathcal{A}_V^{\nu X.\, \varphi}$ then $\|\, \varphi \,\|_{V[\,S\,/\,X]} \in \mathcal{A}_V^{\nu X.\, \varphi}$.

Note that, by taking $\mathcal{A}' = \emptyset$ we have that $\emptyset \in \mathcal{A}_V^{\mu X.\, \varphi}$, and $T \in \mathcal{A}_V^{\nu X.\, \varphi}$ (because $T = \bigcap \emptyset$ when $\emptyset$ is considered as the empty family of subsets of $T$).

As discussed in the introduction, the proof system will use a class of *extended formulae* containing declarations and modifiers for approximant variables:

$$\Phi \ ::= \ \varphi \mid \langle X \leqslant \varphi \rangle\, \Phi \mid [X \geqslant \varphi]\, \Phi \mid \langle -X \rangle\, \Phi \mid [+X\,]\, \Phi$$

In this definition, and henceforth, we use lower case Greek letters $\varphi, \psi, \ldots$ to range over ordinary $\mu$-calculus formulae, and upper case letters $\Phi, \Psi, \ldots$ to range over extended formulae.

The sets of free variables of extended formulae are defined by:

$$\begin{aligned} FV(\langle X \leqslant \varphi \rangle\, \Phi) &= FV([X \geqslant \varphi]\, \Phi) = (FV(\varphi) \cup FV(\Phi)) \backslash \{X\} \\ FV(\langle -X \rangle\, \Phi) \quad &= FV([+X\,]\, \Phi) \quad\ = FV(\Phi) \cup \{X\} \end{aligned}$$

Extended formulae are again identified up to renaming of bound variables.

The semantic interpretation of extended formulae is given relative to a finite set, $\mathsf{D}$, of approximant *declarations*, each of the form $X \leqslant \varphi$ or $X \geqslant \varphi$. The former is a *$\mu$-approximant* declaration, the latter a *$\nu$-approximant* declaration, and in each case the *declared variable* is $X$. We write $DV(\mathsf{D})$ for the set of all variables declared in $\mathsf{D}$.

The *declaration contexts* are produced as follows: (i) the empty set is a declaration context; (ii) if $\mathsf{D}$ is a declaration context, $X$ is a variable not declared in $\mathsf{D}$, and $\varphi$ is a $\mu$-calculus formula with $FV(\varphi) \subseteq DV(\mathsf{D}) \cup \{X\}$ then $\mathsf{D}, X \leqslant \varphi$ and $\mathsf{D}, X \geqslant \varphi$ are both declaration contexts (where we write comma for union). The set of *used variables* in a declaration context is defined by:

$$\begin{array}{ll} UV(X \leqslant \varphi) & = UV(X \geqslant \varphi) = FV(\varphi) \backslash \{X\} \\ UV(\mathsf{D}) & = \bigcup\{\, UV(\delta) \mid \delta \in \mathsf{D}\} \end{array}$$

We next define the notion of an extended formula $\Phi$ being *well-formed* relative to a declaration context $\mathsf{D}$. First, any $\mu$-calculus formula $\varphi$ is well-formed relative to any declaration context $\mathsf{D}$ with $FV(\varphi) \subseteq DV(\mathsf{D})$. Second, the extended formula $\langle X \leqslant \varphi \rangle\, \Phi$ (respectively $[X \geqslant \varphi]\, \Phi$) is well-formed relative to $\mathsf{D}$ if $\mathsf{D}, X \leqslant \varphi$ (respectively $\mathsf{D}, X \geqslant \varphi$) is a declaration context, where $X \notin DV(\mathsf{D})$ (which can be always assumed, by the identification of formulae up to renaming of bound variables), and $\Phi$ is well-formed relative to it. Finally, the extended formula $\langle -X \rangle\, \Phi$ (respectively $[+X]\, \Phi$) is well-formed relative to $\mathsf{D}$ if $\mathsf{D}$ contains a declaration $X \leqslant \varphi$ (respectively $X \geqslant \varphi$), $X \notin UV(\mathsf{D})$ and also $\Phi$ is well-formed relative to $\mathsf{D}$. By this definition, we have that $FV(\Phi) \subseteq DV(\mathsf{D})$ whenever $\Phi$ is well-formed relative to $\mathsf{D}$.

Given a declaration context $\mathsf{D}$, a $\mathsf{D}$-*environment* is a function $V$ mapping $DV(\mathsf{D})$ to subsets of $T$ such that: for each declaration $X \leqslant \varphi$ (respectively $X \geqslant \varphi$) in $\mathsf{D}$, it holds that $V(X) \in \mathcal{A}_V^{\mu X.\, \varphi}$ (respectively $V(X) \in \mathcal{A}_V^{\nu X.\, \varphi}$). To give a semantics to extended formulae, we define subsets $\|\, \Phi \,\|_V^{\mathsf{D}} \subseteq T$ whenever $\mathsf{D}$ is a declaration context, $\Phi$ is well-formed relative to $\mathsf{D}$, and $V$ is a $\mathsf{D}$-environment.

$$\begin{array}{rl} \|\, \varphi \,\|_V^{\mathsf{D}} & = \|\, \varphi \,\|_V \\ \|\, \langle X \leqslant \varphi \rangle\, \Phi \,\|_V^{\mathsf{D}} & = \bigcup\{\|\, \Phi \,\|_{V[S\,/\,X]}^{\mathsf{D}, X \leqslant \varphi} \mid S \in \mathcal{A}_V^{\mu X.\, \varphi}\} \quad \text{where } X \notin DV(\mathsf{D}) \\ \|\, [X \geqslant \varphi]\, \Phi \,\|_V^{\mathsf{D}} & = \bigcap\{\|\, \Phi \,\|_{V[S\,/\,X]}^{\mathsf{D}, X \geqslant \varphi} \mid S \in \mathcal{A}_V^{\nu X.\, \varphi}\} \quad \text{where } X \notin DV(\mathsf{D}) \\ \|\, \langle -X \rangle\, \Phi \,\|_V^{\mathsf{D}} & = \bigcup\{\|\, \Phi \,\|_{V[S\,/\,X]}^{\mathsf{D}} \mid S \subset V(X) \text{ and } S \in \mathcal{A}_V^{\mu X.\, \varphi} \text{ where } X \leqslant \varphi \in \mathsf{D}\} \\ \|\, [+X]\, \Phi \,\|_V^{\mathsf{D}} & = \bigcap\{\|\, \Phi \,\|_{V[S\,/\,X]}^{\mathsf{D}} \mid S \supset V(X) \text{ and } S \in \mathcal{A}_V^{\nu X.\, \varphi} \text{ where } X \geqslant \varphi \in \mathsf{D}\} \end{array}$$

## 3 The Proof System

The proof system we present is general purpose in the sense that, following the approach of [17], it can be easily adapted to give a sound system for reasoning about any process algebra whose operational semantics is given in the GSOS format [1]. However, for brevity of exposition, we present proof rules for the special case of context-free processes only.

**Definition 2 (Context-free system).** A *context-free system* is specified by a finite set of *nonterminals* $\Sigma = \{\mathsf{P}_1, \ldots, \mathsf{P}_\mathsf{k}\}$ together with a finite set $\mathcal{P}$ of *productions*, each of the form $\mathsf{P}_\mathsf{i} \xrightarrow{a} \mathsf{p}$, where $\mathsf{p}$ ranges over $\Sigma^*$ (the set of finite words over $\Sigma$) and $a$ ranges over a finite set of action symbols $A$. The transition system $(T, \{\xrightarrow{a}_T\}_{a \in A})$ determined by the specification is defined as follows.

$$
\begin{aligned}
T \;&=\; \Sigma^* \\
\mathsf{s} \xrightarrow{a}_T \mathsf{t} \ \text{ iff } \ &\mathsf{s} = \mathsf{P}_\mathsf{i}\,\mathsf{q} \text{ and } \mathsf{t} = \mathsf{p}\,\mathsf{q} \text{ for some production } \mathsf{P}_\mathsf{i} \xrightarrow{a} \mathsf{p} \in \mathcal{P}.
\end{aligned}
$$

Here, as usual, a juxtaposition $\mathsf{p}\,\mathsf{q}$ means the concatenation of words $\mathsf{p}$ and $\mathsf{q}$.

*Example 1.* As a running example, consider the system with a single nonterminal $\mathsf{P}$, set of actions $A = \{a, b\}$, and with two productions: $\mathsf{P} \xrightarrow{a} \mathsf{PP}$ and $\mathsf{P} \xrightarrow{b} \varepsilon$, where $\varepsilon$ is the empty word. This has as its transition system:

$$\begin{tikzcd}
\varepsilon & \mathsf{P} \arrow[l, "b"'] \arrow[r, shift left, "a"] & \mathsf{P}^2 \arrow[l, shift left, "b"] \arrow[r, shift left, "a"] & \mathsf{P}^3 \arrow[l, shift left, "b"] \arrow[r, shift left, "a"] & \cdots \arrow[l, shift left, "b"]
\end{tikzcd}$$

This is an infinite-state process in which no two distinct states are bisimilar.

Henceforth in this section we assume that we have a fixed specification of a context-free system, as in Definition 2, and we write $(T, \{\xrightarrow{a}_T\}_{a \in A})$ for the transition system it determines.

The proof system uses process terms containing free process variables $x, y, \ldots$

**Definition 3 (Process term).** A *process term* is a word of one of two forms: either $\mathsf{p}\,x$, where $\mathsf{p} \in \Sigma^*$ and $x$ is a process variable; or $\mathsf{p}$ where $\mathsf{p} \in \Sigma^*$.

We use $p, q, \ldots$ to range over process terms. By a *process substitution* we shall mean a mapping $\theta$ from process variables to process terms. The substituted term $p\,[\,\theta\,]$ is defined in the obvious way.

Process terms are interpreted relative to *process environments* $\rho$ mapping process variables to states in the transition system $T$. We extend $\rho$ to a function (also called $\rho$) from process terms to $T$ by: $\rho(\mathsf{p}\,x) = \mathsf{p}\,\rho(x)$ and $\rho(\mathsf{p}) = \mathsf{p}$.

Sequents will be built from two forms of assertion: *verification assertions* of the form $p\!:\!\Phi$, where $\Phi$ is an extended formula, as in Sect. 2; and *transition assertions* of the form $p \xrightarrow{a} q$. We use $J, K, \ldots$ to range over assertions. Given a declaration context $\mathsf{D}$, an assertion is a $\mathsf{D}$-*assertion* if it is either a verification assertion $p\!:\!\Phi$ with $\Phi$ well-formed relative to $\mathsf{D}$, or a transition assertion.

**Definition 4 (Sequent).** *Sequents* have the form $\mathsf{D}\,;\,\Gamma \;\vdash\; \Delta$ where $\mathsf{D}$ is a declaration context and $\Gamma$ and $\Delta$ are finite sets of $\mathsf{D}$-assertions.

Semantically, assertions and sequents will always be interpreted relative to the transition system $(T, \{\xrightarrow{a}_T\}_{a \in A})$. Given a $\mathsf{D}$-environment $V$ and a process environment $\rho$, the relation $\models_{V\rho} J$, for $\mathsf{D}$-assertions $J$, is defined by:

$$
\begin{aligned}
\models_{V\rho}\ p\!:\!\Phi \ &\text{ iff } \ \rho(p) \in \|\,\Phi\,\|^{\mathsf{D}}_{V} \\
\models_{V\rho}\ p \xrightarrow{a} q \ &\text{ iff } \ \rho(p) \xrightarrow{a}_T \rho(q)
\end{aligned}
$$

**General rules**

$$(\text{Axiom})\ \frac{\mathsf{D}\,;\,\mathit{\Gamma} \vdash \mathit{\Delta}}{\phantom{x}}\ \mathit{\Gamma} \cap \mathit{\Delta} \neq \emptyset \qquad (\text{Weak})\ \frac{\mathsf{D}\,;\,\mathit{\Gamma} \vdash \mathit{\Delta}}{\mathsf{D}'\,;\,\mathit{\Gamma}' \vdash \mathit{\Delta}'}\ \mathsf{D}' \subseteq \mathsf{D},\ \mathit{\Gamma}' \subseteq \mathit{\Gamma},\ \mathit{\Delta}' \subseteq \mathit{\Delta}$$

$$(\text{Cut})\ \frac{\mathsf{D}\,;\,\mathit{\Gamma} \vdash \mathit{\Delta}}{\mathsf{D}\,;\,\mathit{\Gamma} \vdash \mathit{\Delta},\ J \qquad \mathsf{D}\,;\,\mathit{\Gamma},\ J \vdash \mathit{\Delta}} \qquad (\text{Sub})\ \frac{\mathsf{D}\,;\,\mathit{\Gamma}\,[\theta] \vdash \mathit{\Delta}\,[\theta]}{\mathsf{D}\,;\,\mathit{\Gamma} \vdash \mathit{\Delta}}$$

**Logical rules**

$$(\mathtt{ff}\text{L})\ \frac{\mathsf{D}\,;\,\mathit{\Gamma},\ p\!:\!\mathtt{ff} \vdash \mathit{\Delta}}{\phantom{x}} \qquad (\mathtt{tt}\text{R})\ \frac{\mathsf{D}\,;\,\mathit{\Gamma} \vdash \mathit{\Delta},\ p\!:\!\mathtt{tt}}{\phantom{x}}$$

$$(\vee\text{L})\ \frac{\mathsf{D}\,;\,\mathit{\Gamma},\ p\!:\!\varphi_1 \vee \varphi_2 \vdash \mathit{\Delta}}{\mathsf{D}\,;\,\mathit{\Gamma},\ p\!:\!\varphi_1 \vdash \mathit{\Delta} \qquad \mathsf{D}\,;\,\mathit{\Gamma},\ p\!:\!\varphi_2 \vdash \mathit{\Delta}} \qquad (\vee\text{R})\ \frac{\mathsf{D}\,;\,\mathit{\Gamma} \vdash \mathit{\Delta},\ p\!:\!\varphi_1 \vee \varphi_2}{\mathsf{D}\,;\,\mathit{\Gamma} \vdash \mathit{\Delta},\ p\!:\!\varphi_1,\ p\!:\!\varphi_2}$$

$$(\wedge\text{L})\ \frac{\mathsf{D}\,;\,\mathit{\Gamma},\ p\!:\!\varphi_1 \wedge \varphi_2 \vdash \mathit{\Delta}}{\mathsf{D}\,;\,\mathit{\Gamma},\ p\!:\!\varphi_1,\ p\!:\!\varphi_2 \vdash \mathit{\Delta}} \qquad (\wedge\text{R})\ \frac{\mathsf{D}\,;\,\mathit{\Gamma} \vdash \mathit{\Delta},\ p\!:\!\varphi_1 \wedge \varphi_2}{\mathsf{D}\,;\,\mathit{\Gamma} \vdash \mathit{\Delta},\ p\!:\!\varphi_1 \qquad \mathsf{D}\,;\,\mathit{\Gamma} \vdash \mathit{\Delta},\ p\!:\!\varphi_2}$$

**Modal rules**

$$(\langle a\rangle\text{L})^{*}\ \frac{\mathsf{D}\,;\,\mathit{\Gamma},\ p\!:\!\langle a\rangle\varphi \vdash \mathit{\Delta}}{\mathsf{D}\,;\,\mathit{\Gamma},\ p \xrightarrow{a} x,\ x\!:\!\varphi \vdash \mathit{\Delta}} \qquad (\langle a\rangle\text{R})\ \frac{\mathsf{D}\,;\,\mathit{\Gamma} \vdash \mathit{\Delta},\ p\!:\!\langle a\rangle\varphi}{\mathsf{D}\,;\,\mathit{\Gamma} \vdash \mathit{\Delta},\ p \xrightarrow{a} q \qquad \mathsf{D}\,;\,\mathit{\Gamma} \vdash \mathit{\Delta},\ q\!:\!\varphi}$$

$$([a]\text{L})\ \frac{\mathsf{D}\,;\,\mathit{\Gamma},\ p\!:\![a]\varphi \vdash \mathit{\Delta}}{\mathsf{D}\,;\,\mathit{\Gamma} \vdash \mathit{\Delta},\ p \xrightarrow{a} q \qquad \mathsf{D}\,;\,\mathit{\Gamma},\ q\!:\!\varphi \vdash \mathit{\Delta}} \qquad ([a]\text{R})^{*}\ \frac{\mathsf{D}\,;\,\mathit{\Gamma} \vdash \mathit{\Delta},\ p\!:\![a]\varphi}{\mathsf{D}\,;\,\mathit{\Gamma},\ p \xrightarrow{a} x \vdash \mathit{\Delta},\ x\!:\!\varphi}$$

* Restriction on ($\langle a\rangle$L) and ($[a]$R): $x$ must not occur free in the goal.

**Operational rules**

$$(\mathsf{P}_{\mathsf{i}}\text{L})\ \frac{\mathsf{D}\,;\,\mathit{\Gamma},\ \mathsf{P}_{\mathsf{i}}\,q \xrightarrow{a} x \vdash \mathit{\Delta}}{\{\,\mathsf{D}\,;\,\mathit{\Gamma}[\,\mathsf{p}\,q\,/\,x] \vdash \mathit{\Delta}[\,\mathsf{p}\,q\,/\,x]\,\}_{\mathsf{P}_{\mathsf{i}} \xrightarrow{a} \mathsf{p} \in \mathcal{P}}}\ x \text{ does not occur in } q$$

$$(\varepsilon\text{L})\ \frac{\mathsf{D}\,;\,\mathit{\Gamma},\ \varepsilon \xrightarrow{a} x \vdash \mathit{\Delta}}{\phantom{x}} \qquad (\mathsf{P}_{\mathsf{i}}\text{R})\ \frac{\mathsf{D}\,;\,\mathit{\Gamma} \vdash \mathit{\Delta},\ \mathsf{P}_{\mathsf{i}}\,q \xrightarrow{a} \mathsf{p}\,q}{\phantom{x}}\ \mathsf{P}_{\mathsf{i}} \xrightarrow{a} \mathsf{p} \in \mathcal{P}$$

**Fig. 1.** Basic rules

We write $\mathsf{D}\,;\,\mathit{\Gamma} \models_{V\rho} \mathit{\Delta}$ to mean that if $\models_{V\rho} J$, for all $J \in \mathit{\Gamma}$, then there exists $K \in \mathit{\Delta}$ such that $\models_{V\rho} K$. We write $\mathsf{D}\,;\,\mathit{\Gamma} \models \mathit{\Delta}$ to mean that $\mathsf{D}\,;\,\mathit{\Gamma} \models_{V\rho} \mathit{\Delta}$ for all $V$ and $\rho$.

**Fixed-point rules**

$$(\mu\mathrm{L})\;\frac{\mathsf{D}\,;\,\Gamma,\ p\colon\mu X.\,\varphi \vdash \Delta}{\mathsf{D}\,;\,\Gamma,\ p\colon\langle X\leqslant\varphi\rangle\,\varphi \vdash \Delta}\qquad(\mu\mathrm{R})\;\frac{\mathsf{D}\,;\,\Gamma \vdash \Delta,\ p\colon\mu X.\,\varphi}{\mathsf{D}\,;\,\Gamma \vdash \Delta,\ p\colon\langle X\leqslant\varphi\rangle\,\varphi}$$

$$(\leqslant\text{-}\mu\mathrm{L})\;\frac{\mathsf{D}\,;\,\Gamma,\ p\colon\langle X\leqslant\varphi\rangle\,\Phi \vdash \Delta}{\mathsf{D}\,;\,\Gamma,\ p\colon\Phi[\,\mu X.\,\varphi\,/\,X] \vdash \Delta}\qquad(\leqslant\text{-}\mu\mathrm{R})\;\frac{\mathsf{D}\,;\,\Gamma \vdash \Delta,\ p\colon\langle X\leqslant\varphi\rangle\,\Phi}{\mathsf{D}\,;\,\Gamma \vdash \Delta,\ p\colon\Phi[\,\mu X.\,\varphi\,/\,X]}$$

$$(\nu\mathrm{L})\;\frac{\mathsf{D}\,;\,\Gamma,\ p\colon\nu X.\,\varphi \vdash \Delta}{\mathsf{D}\,;\,\Gamma,\ p\colon[X\geqslant\varphi]\,\varphi \vdash \Delta}\qquad(\nu\mathrm{R})\;\frac{\mathsf{D}\,;\,\Gamma \vdash \Delta,\ p\colon\nu X.\,\varphi}{\mathsf{D}\,;\,\Gamma \vdash \Delta,\ p\colon[X\geqslant\varphi]\,\varphi}$$

$$(\geqslant\text{-}\nu\mathrm{L})\;\frac{\mathsf{D}\,;\,\Gamma,\ p\colon[X\geqslant\varphi]\,\Phi \vdash \Delta}{\mathsf{D}\,;\,\Gamma,\ p\colon\Phi[\,\nu X.\,\varphi\,/\,X] \vdash \Delta}\qquad(\geqslant\text{-}\nu\mathrm{R})\;\frac{\mathsf{D}\,;\,\Gamma \vdash \Delta,\ p\colon[X\geqslant\varphi]\,\Phi}{\mathsf{D}\,;\,\Gamma \vdash \Delta,\ p\colon\Phi[\,\nu X.\,\varphi\,/\,X]}$$

**Approximant rules**

$$(\leqslant\text{-}X\mathrm{L})^{*}\;\frac{\mathsf{D}\,;\,\Gamma,\ p\colon\langle X\leqslant\varphi\rangle\,\Phi \vdash \Delta}{\mathsf{D},\ X\leqslant\varphi\,;\,\Gamma,\ p\colon\Phi \vdash \Delta}\qquad(\leqslant\text{-}X\mathrm{R})\;\frac{\mathsf{D}\,;\,\Gamma \vdash \Delta,\ p\colon\langle X\leqslant\varphi\rangle\,\Phi}{\mathsf{D}\,;\,\Gamma \vdash \Delta,\ p\colon\Phi}\,X\leqslant\varphi\in\mathsf{D}$$

$$(X_\mu\mathrm{L})\;\frac{\mathsf{D}\,;\,\Gamma,\ p\colon X \vdash \Delta}{\mathsf{D}\,;\,\Gamma,\ p\colon\langle -X\rangle\,\varphi \vdash \Delta}\,X\leqslant\varphi\in\mathsf{D}\qquad(X_\mu\mathrm{R})\;\frac{\mathsf{D}\,;\,\Gamma \vdash \Delta,\ p\colon X}{\mathsf{D}\,;\,\Gamma \vdash \Delta,\ p\colon\langle -X\rangle\,\varphi}\,X\leqslant\varphi\in\mathsf{D}$$

$$(\langle -X\rangle)\;\frac{\mathsf{D}\,;\,\langle -X\rangle\,\Gamma,\ \Gamma' \vdash \langle -X\rangle\,\Delta,\ \Delta'}{\mathsf{D}\,;\,\Gamma,\ \Gamma' \vdash \Delta,\ \Delta'}\,\Gamma\neq\emptyset,\ X\notin UV(\mathsf{D})\cup FV(\Gamma')$$

$$(\geqslant\text{-}X\mathrm{L})\;\frac{\mathsf{D}\,;\,\Gamma,\ p\colon[X\geqslant\varphi]\,\Phi \vdash \Delta}{\mathsf{D}\,;\,\Gamma,\ p\colon\Phi \vdash \Delta}\,X\geqslant\varphi\in\mathsf{D}\qquad(\geqslant\text{-}X\mathrm{R})^{*}\;\frac{\mathsf{D}\,;\,\Gamma \vdash \Delta,\ p\colon[X\geqslant\varphi]\,\Phi}{\mathsf{D},\ X\geqslant\varphi\,;\,\Gamma \vdash \Delta,\ p\colon\Phi}$$

$$(X_\nu\mathrm{L})\;\frac{\mathsf{D}\,;\,\Gamma,\ p\colon X \vdash \Delta}{\mathsf{D}\,;\,\Gamma,\ p\colon[+X]\,\varphi \vdash \Delta}\,X\geqslant\varphi\in\mathsf{D}\qquad(X_\nu\mathrm{R})\;\frac{\mathsf{D}\,;\,\Gamma \vdash \Delta,\ p\colon X}{\mathsf{D}\,;\,\Gamma \vdash \Delta,\ p\colon[+X]\,\varphi}\,X\geqslant\varphi\in\mathsf{D}$$

$$([+X])\;\frac{\mathsf{D}\,;\,[+X]\,\Gamma,\ \Gamma' \vdash [+X]\,\Delta,\ \Delta'}{\mathsf{D}\,;\,\Gamma,\ \Gamma' \vdash \Delta,\ \Delta'}\,\Delta\neq\emptyset,\ X\notin UV(\mathsf{D})\cup FV(\Delta')$$

$^{*}$ Restriction on ($\leqslant$-$X$L) and ($\geqslant$-$X$R): $X$ must not occur free in the goal.

**Fig. 2.** Fixed-point and approximant rules

The proof system provides a means of verifying sequents $\mathsf{D}\,;\Gamma\ \vdash\ \Delta$ for which $\mathsf{D}\,;\Gamma\ \models\ \Delta$. The rules are presented in Figs. 1 and 2. The rules in Fig. 1 concern the modal fragment of the logic, and are essentially from [17]. Figure 2 presents the crucial rules for fixed points and explicit approximants. We emphasise again that we write the rules in tableau style with the *goal* sequent above the line and its (possibly empty) set of *subgoals* below the line. Rules are applicable only in instances in which the subgoals produced are indeed sequents according to Definition 4. Certain rules have additional side conditions, written on the right. In the rules, we use the abbreviations:

$$\begin{aligned} \Gamma\,[\,\theta\,] &= \{\,p\,[\,\theta\,]\!:\!\Phi \mid p\!:\!\Phi \in \Gamma\} \cup \{\,p\,[\,\theta\,] \xrightarrow{a} q\,[\,\theta\,] \mid p \xrightarrow{a} q \in \Gamma\}, \\ \langle -X\rangle\,\Gamma &= \{\,p\!:\!\langle -X\rangle\,\Phi \mid p\!:\!\Phi \in \Gamma\}, \quad [+X\,]\,\Gamma = \{\,p\!:\![+X\,]\,\Phi \mid p\!:\!\Phi \in \Gamma\}, \end{aligned}$$

where, whenever we write $\langle -X\rangle\,\Gamma$ and $[+X\,]\,\Gamma$, we tacitly assume that $\Gamma$ contains only verification assertions. We briefly explain the rule $(\langle -X\rangle)$ which, along with $([+X\,])$, is probably the most obscure. Suppose we have $V$ and $\rho$ invalidating the goal, i.e. such that $\mathsf{D}\,;\langle -X\rangle\,\Gamma,\ \Gamma'\ \not\models_{V\rho}\ \langle -X\rangle\,\Delta,\ \Delta'$. We show that the subgoal is also invalid. Because the goal is invalidated, we have that $\rho(p_j) \in \|\,\langle -X\rangle\,\Phi_j\,\|^{\mathsf{D}}_{V}$, for each $p_j\!:\!\Phi_j\ \in\ \Gamma\ =\ \{p_1\!:\!\Phi_1,\ldots,p_l\!:\!\Phi_l\}$. So, for each $p_j\!:\!\Phi_j$, there exists $S_j \subset V(X)$ with $S_j \in \mathcal{A}_V^{\mu X.\,\varphi}$ such that $\rho(p_j) \in \|\,\Phi_j\,\|^{\mathsf{D}}_{V[\,S_j\,/\,X]}$. As approximants are linearly ordered and $\Gamma \neq \emptyset$, we can take the largest such $S_k \subset V(X)$, and, by monotonicity considerations, simultaneously satisfy $\rho(p_j) \in \|\,\Phi_j\,\|^{\mathsf{D}}_{V[\,S_k\,/\,X]}$ for all $p_j\!:\!\Phi_j \in \Gamma$. Define $V' = V[\,S_k\,/\,X]$. We claim that $\mathsf{D}\,;\ \Gamma,\ \Gamma'\ \not\models_{V'\rho}\ \Delta,\ \Delta'$. We have seen that the assertions in $\Gamma$ are satisfied. Those in $\Gamma'$ are because $X \notin UV(\mathsf{D}) \cup FV(\Gamma')$. The assertions in $\Delta$ are not satisfied under $V'$ because those in $\langle -X\rangle\,\Delta$ weren't under $V$. Finally, by monotonicity considerations, the assertions in $\Delta'$ are also not satisfied under $V'$, because they weren't under $V$.

The above justification for the $(\langle -X\rangle)$ rule modifies a $\mathsf{D}$-environment on $X$ by mapping it to a strictly smaller $\mu$-approximant. Dually, the $([+X\,])$ rule results in $X$ being mapped to a strictly larger $\nu$-approximant. By well-foundedness considerations, neither event can occur infinitely often. This observation motivates the definitions below, which formulate when a derivation tree constitutes a proof.

By a *leaf* in a derivation tree, we mean a sequent occurrence in the tree such that no rule has been applied with that sequent occurrence as its goal (thus sequents to which a rule with an empty set of subgoals has been applied *do not* count as leaves, even though they have no child sequents).

**Definition 5 (Repeat).** In a derivation tree, a leaf $\mathsf{D}\,;\ \Gamma\ \vdash\ \Delta$ is a *repeat* of another sequent occurrence $\mathsf{D}'\,;\ \Gamma'\ \vdash\ \Delta'$ if $\mathsf{D}' \subseteq \mathsf{D}$ and there exists a process substitution $\theta$ such that $\Gamma'\,[\,\theta\,] \subseteq \Gamma$ and $\Delta'\,[\,\theta\,] \subseteq \Delta$.

**Definition 6 (Pre-proof).** A *pre-proof* is a derivation tree in which, to each leaf $\mathsf{D}\,;\ \Gamma\ \vdash\ \Delta$, there is an assigned sequent occurrence $\mathsf{D}'\,;\ \Gamma'\ \vdash\ \Delta'$ (the *companion* of the leaf) such that $\mathsf{D}\,;\ \Gamma\ \vdash\ \Delta$ is a repeat of $\mathsf{D}'\,;\ \Gamma'\ \vdash\ \Delta'$.

In the above definitions, it is worth noting that the companion is not required to appear on the branch from the root sequent to the leaf.

We consider a pre-proof as a directed graph whose vertices are sequent occurrences in the pre-proof, and with edges of two kinds: (i) edges from the goal of a rule application to each subgoal (if any) of the goal; (ii) an edge from each leaf to its companion. By a (finite or infinite) *path* through a pre-proof, we mean a sequence $(\mathcal{S}_i)_{0 \leq i < n \leq \infty}$ of sequent occurrences forming a directed path through the graph. We say that a rule is *applied along* a path $(\mathcal{S}_i)$ if the path contains two consecutive sequents $\mathcal{S}_i$ and $\mathcal{S}_{i+1}$ with $\mathcal{S}_i$ the goal of the rule and $\mathcal{S}_{i+1}$ one of its subgoals.

**Definition 7 (Preservation).** A path *preserves* an approximant variable $X$ if, for every sequent $\mathsf{D}\,;\, \Gamma \vdash \Delta$ occurring on the path, $X \in DV(\mathsf{D})$.

**Definition 8 (Progress).** A $\mu$-approximant variable $X$ *progresses* on a path if it is preserved by the path and the rule $(\langle -X \rangle)$ is applied along the path. Similarly, a $\nu$-approximant variable $X$ *progresses* if it is preserved and the rule $([+X\,])$ is applied.

We say that $X$ progresses *infinitely often* on an infinite path $(\mathcal{S}_i)_{i \geq 0}$ if, for all $n \in \mathbb{N}$, it holds that $X$ progresses on the tail path $(\mathcal{S}_i)_{i \geq n}$.

**Definition 9 (Proof).** A pre-proof is a *proof* if, for every infinite path $(\mathcal{S}_i)_{i \geq 0}$ through it, there exist an approximant variable $X$ and a tail $(\mathcal{S}_i)_{i \geq n}$ on which $X$ progresses infinitely often.

We remark that this condition is necessarily global, in the sense that it cannot be reformulated as a condition to be satisfied by each repeat individually.

**Proposition 1.** *It is decidable whether a pre-proof is a proof or not.*

**Theorem 1 (Soundness).** *If* $\mathsf{D}\,;\, \Gamma \vdash \Delta$ *has a proof then* $\mathsf{D}\,;\, \Gamma \models \Delta$.

In Fig. 3 we give an example proof in the system, showing that the process $\mathsf{P}$, from Example 1, satisfies the property $\nu X.\, \mu Y.\, [a]X \wedge [b]Y$, stating that action $a$ occurs infinitely often along any infinite path of $a$ and $b$ actions. The identified repeats determine a pre-proof, which is easily seen to be a proof.

## 4 Completeness for Context-Free Processes

We assume a fixed specification of a context-free system, as in Definition 2.

**Theorem 2 (Context-free completeness).** *For any* $\mathsf{p} \in \Sigma^*$ *and closed* $\mu$*-calculus formula* $\varphi$*, if* $\mathsf{p} \in \|\, \varphi \,\|$ *then the sequent* $\vdash \mathsf{p} : \varphi$ *has a proof.*

The proof uses a variant of the property-checking games described in [19]. In a transition system $(T, \{\xrightarrow{a}_T\}_{a \in A})$, the property-checking game $\mathsf{G}(s, \varphi)$, where $s \in T$ and $\varphi$ is a closed $\mu$-calculus formula, is a game played by two players, Verifier and Refuter. Verifier aims to show that $s \in \|\, \varphi \,\|$ whereas Refuter attempts to refute this. We use an asymmetric variant of property-checking games, designed to facilitate translating properties of games into the sequent calculus.

Abbreviations: $V \equiv \nu X.\,\mu Y.\,[a]X \wedge [b]Y$, $\;U \equiv \mu Y.\,[a]X \wedge [b]Y$.

$$\dfrac{\vdash \mathsf{P}:V}{\dfrac{\vdash \varepsilon:[X\geqslant U\,]\,U}{\dfrac{X\geqslant U\,;\ \vdash \varepsilon:U}{\dfrac{X\geqslant U\,;\ \vdash \varepsilon:\langle Y\leqslant[a]X\wedge[b]Y\rangle\,[a]X\wedge[b]Y}{\dfrac{X\geqslant U\,;\ \vdash \varepsilon:[a]X\wedge[b]U}{\dfrac{X\geqslant U\,;\ \vdash \varepsilon:[a]X}{\dfrac{X\geqslant U\,;\ \varepsilon\xrightarrow{a}x\ \vdash\ x:X}{}}\quad \dfrac{X\geqslant U\,;\ \vdash \varepsilon:[b]U}{\dfrac{X\geqslant U\,;\ \varepsilon\xrightarrow{b}x\ \vdash\ x:U}{}}}}}}\qquad \dfrac{\varepsilon:[X\geqslant U\,]\,U\ \vdash\ \mathsf{P}:V}{\dfrac{x:[X\geqslant U\,]\,U\ \vdash\ \mathsf{P}\,x:V}{\dfrac{x:[X\geqslant U\,]\,U\ \vdash\ \mathsf{P}\,x:[X\geqslant U\,]\,U}{\dfrac{X\geqslant U\,;\ x:[X\geqslant U\,]\,U\ \vdash\ \mathsf{P}\,x:U}{\vdots}}}\ \text{(Sub)}}}\ \text{(Cut)}$$

We continue with the right-hand branch.

$$\dfrac{\dfrac{\vdots}{X\geqslant U\,;\ x:U\ \vdash\ \mathsf{P}\,x:U}\quad(\star)}{\dfrac{X\geqslant U\,;\ x:U\ \vdash\ \mathsf{P}\,x:\langle Y\leqslant[a]X\wedge[b]Y\rangle\,[a]X\wedge[b]Y}{\dfrac{X\geqslant U\,;\ x:U\ \vdash\ \mathsf{P}\,x:[a]X\wedge[b]U}{\dfrac{X\geqslant U\,;\ x:U\ \vdash\ \mathsf{P}\,x:[a]X}{\dfrac{X\geqslant U\,;\ x:U,\ \mathsf{P}\,x\xrightarrow{a}y\ \vdash\ y:X}{\vdots}}\quad \dfrac{X\geqslant U\,;\ x:U\ \vdash\ \mathsf{P}\,x:[b]U}{\dfrac{X\geqslant U\,;\ x:U,\ \mathsf{P}\,x\xrightarrow{b}y\ \vdash\ y:U}{\dfrac{X\geqslant U\,;\ x:U\ \vdash\ x:U}{}}}}}}$$

We continue with the left-hand branch.

$$\dfrac{\dfrac{\vdots}{X\geqslant U\,;\ x:U\ \vdash\ \mathsf{PP}\,x:X}}{\dfrac{X\geqslant U\,;\ x:U\ \vdash\ \mathsf{P}\,x:[+X\,]\,U}{X\geqslant U\,;\ x:U\ \vdash\ \mathsf{P}\,x:U}\ ([+X\,])\quad \dfrac{X\geqslant U\,;\ \mathsf{P}\,x:[+X\,]\,U\ \vdash\ \mathsf{PP}\,x:X}{\dfrac{X\geqslant U\,;\ \mathsf{P}\,x:[+X\,]\,U\ \vdash\ \mathsf{PP}\,x:[+X\,]\,U}{X\geqslant U\,;\ \mathsf{P}\,x:U\ \vdash\ \mathsf{PP}\,x:U}\ ([+X\,])}}\ \text{(Cut)}$$

Both leaves are repeats of the sequent $(\star)$.

**Fig. 3.** Example Proof

For technical convenience, we assume representations of formulae in which all bound variables have different names, and we assume that we only encounter fixed-point formulae $\mu X.\,\varphi$, $\nu X.\,\varphi$ with $X \in FV(\varphi)$. We use sequences, $\mathsf{E}$, of greatest-fixed-point definitions called *$\nu$-contexts*, together with their sets of *declared variables* $DV(\mathsf{E})$. These are defined by: (i) the empty sequence () is a $\nu$-context with the empty set of declared variables; (ii) if $\mathsf{E}$ is a $\nu$-context, $X \notin DV(\mathsf{E})$ and $FV(\varphi) \subseteq DV(E) \cup \{X\}$ then $\mathsf{E},\ X = \varphi$ is a $\nu$-context with $DV(\mathsf{E}) \cup \{X\}$ as its set of declared variables. The equality $X = \varphi$ in a $\nu$-context declares $X$ to be the greatest fixed-point $\nu X.\,\varphi$.

**Definition 10 (Position).** A *position* is a triple $(s, \mathsf{E}, \varphi)$ where $s \in T$ is any state, $\mathsf{E}$ is a $\nu$-context and $\varphi$ is a formula such that $FV(\varphi) \subseteq DV(\mathsf{E})$ but, for all proper prefixes $\mathsf{E}'$ of $\mathsf{E}$, $FV(\varphi) \not\subseteq DV(\mathsf{E}')$.

**Definition 11 (Move).** The legitimate *moves* from one position $(s, \mathsf{E}, \varphi)$ to another are defined by case analysis on $\varphi$:

**ff:** It is Verifier's move, but she is stuck.
**tt:** It is Refuter's move, but he is stuck.
**$\psi_1 \vee \psi_2$:** Verifier chooses a disjunct $\psi_j$ where $j \in \{1, 2\}$, and the next position is $(s, \mathsf{E}', \psi_j)$, where $\mathsf{E}'$ is the smallest prefix of $\mathsf{E}$ with $FV(\psi_j) \subseteq DV(\mathsf{E}')$.
**$\psi_1 \wedge \psi_2$:** Refuter chooses a conjunct $\psi_j$ where $j \in \{1, 2\}$, and the next position is $(s, \mathsf{E}', \psi_j)$, where $\mathsf{E}'$ is the smallest prefix of $\mathsf{E}$ with $FV(\psi_j) \subseteq DV(\mathsf{E}')$.
**$\langle a \rangle \psi$:** Verifier chooses a transition $s \xrightarrow{a}_T t$, and the next position is $(t, \mathsf{E}, \psi)$.
**$[a]\psi$:** Refuter chooses a transition $s \xrightarrow{a}_T t$, and the next position is $(t, \mathsf{E}, \psi)$.
**$\mu X.\, \psi$:** Verifier moves to the next position $(s, \mathsf{E}, \psi[\,\mu X.\, \psi \,/\, X])$.
**$\nu X.\, \psi$:** Refuter moves to the next position $(s, \mathsf{E}', \psi)$, where $\mathsf{E}'$ is $\mathsf{E}$, $X = \psi$.
**$X$:** Refuter moves to the next position $(s, \mathsf{E}, \psi)$, where $X = \psi \in \mathsf{E}$.

**Definition 12 (Play).** A *play* is a finite or infinite sequence $(s_i, \mathsf{E}_i, \varphi_i)_i$ of positions where each position $(s_{k+1}, \mathsf{E}_{k+1}, \varphi_{k+1})$ is produced from $(s_k, \mathsf{E}_k, \varphi_k)$ by following one of the moves above.

**Definition 13 (Preservation).** We say that a play $(s_i, \mathsf{E}_i, \varphi_i)_i$ *preserves* a variable $X$ if, for each $\mathsf{E}_i$ in the play, $X \in DV(\mathsf{E}_i)$.

**Definition 14 (Progress).** We say that a fixed-point variable $X$ *progresses* along a play if it is preserved by the play and the play contains a move away from a position $(s, \mathsf{E}, X)$.

**Definition 15 (Winning play).** The Verifier *wins* a play either if the play is finite and its last position is one at which it is Refuter's move, or if the play is infinite and there exist a variable $X$ and a tail of the play such that $X$ progresses infinitely often along the tail.

**Definition 16 (The game $\mathsf{G}(s, \varphi)$).** The *game* $\mathsf{G}(s, \varphi)$, where $\varphi$ is a closed formula, is played on the set of all positions reachable from the *initial position* $(s,\ (),\ \varphi)$. The game is a two player game, played by Verifier and Refuter, with play starting from the initial position.

For ordinary property-checking games, the following result appears in [19, §6.3]. The adaptation to our games is straightforward.

**Proposition 2.** *If $s \in \|\,\varphi\,\|$ then Verifier has a history-free winning strategy for the game $\mathsf{G}(s, \varphi)$.*

We now begin the proof of Theorem 2. Henceforth, suppose that $\mathsf{p}_0 \in \Sigma^*$ is such that $\mathsf{p}_0 \in \| \varphi_0 \|$. We use the game $\mathsf{G}(\mathsf{p}_0, \varphi_0)$ to construct a proof of the sequent $\vdash \mathsf{p}_0 : \varphi_0$.

Henceforth, all plays will be of the game $\mathsf{G}(\mathsf{p}_0, \varphi_0)$. By Proposition 2, Verifier has a history-free winning strategy for this game. We henceforth fix on one such strategy, and we call a play a *V-play* if all Verifier's moves in the play follow the strategy. We write $\mathbf{u}_0$ for the initial position $(\mathsf{p}_0,\, (),\, \varphi_0)$. From now on, we shall only consider those positions that arise in some V-play from $\mathbf{u}_0$. We use $\mathbf{u}, \mathbf{v}, \mathbf{w}, \ldots$ to range over such positions, and $\pi, \tau \ldots$ to range over V-plays starting from any such position. Note that Verifier wins any infinite V-play. We write $\mathbf{u}\pi$ and $\pi\mathbf{v}$ to mean that $\mathbf{u}$ and $\mathbf{v}$ are the first and last positions in $\pi$ respectively. Given two V-plays $\pi_1\mathbf{v}$ and $\mathbf{v}\pi_2$, we write $\pi_1\pi_2$ for the evident concatenation of the two plays.

We give a brief summary of the proof structure. Similar to [13], we consider "canonical" sequents of the restricted form:

$$\mathsf{D}\,;\; x\!:\!\Psi_1,\;\ldots,\;x\!:\!\Psi_k \;\vdash\; \mathsf{P}\,x\!:\!\varphi, \tag{2}$$

where $\mathsf{P}$ is a nonterminal. Each such sequent is constructed with reference to a position of the form $\mathbf{u} = (\mathsf{P}\,\mathsf{q}, \mathsf{E}, \varphi)$, with each assumption $x\!:\!\Psi_i$ being determined by a V-play $\pi$ from $\mathbf{u}$ to some position $\mathbf{v}$ whose state is $\mathsf{q}$. Importantly, the extended formula $\Psi_i$ contains $\nu$-approximant declarations and modifiers that reflect preservation and progress properties of the play $\pi$. We use Verifier's strategy to construct a derivation tree in which individual rule applications can be combined into larger steps between sequents of the form (2). Crucially, only finitely many distinct such sequents occur in the constructed derivation, enabling the derivation tree to terminate in repeats. Moreover, paths in the derivation tree reflect preservation and progress properties of the V-plays used to construct the derivation, which allows the choice of repeats to be made so that the resulting pre-proof is a proof.

To define canonical sequents, we need various auxiliary definitions. Given a play $\pi$ ending in the position $(s, \mathsf{E}, \varphi)$, we define functions $\mathit{dec}_\pi(\mathsf{E}',\, \Phi)$ and $\mathit{pro}_\pi(\mathsf{E}',\, \Phi)$ for prefixes $\mathsf{E}'$ of $\mathsf{E}$ and extended formulae $\Phi$ with $FV(\Phi) \subseteq DV(\mathsf{E}')$.

$$\mathit{pro}_\pi(\mathsf{E}',\, \Phi) = \begin{cases} \mathit{dec}_\pi(\mathsf{E}',\, [+X\,]\,\Phi) & \text{if } \mathsf{E}' \text{ is } \mathsf{E}'', X = \varphi \text{ and } X \text{ progresses on a tail of } \pi \\ \mathit{dec}_\pi(\mathsf{E}',\, \Phi) & \text{otherwise} \end{cases}$$

$$\mathit{dec}_\pi(\mathsf{E}',\, \Phi) = \begin{cases} \mathit{pro}_\pi(\mathsf{E}'',\, [X \geqslant \varphi\,]\,\Phi) & \text{if } \mathsf{E}' \text{ is } \mathsf{E}'', X = \varphi \text{ and } \pi \text{ does not preserve } X \\ \Phi & \text{otherwise} \end{cases}$$

**Definition 17 (Characteristic formula).** For any play $\pi$ ending in $(s, \mathsf{E}, \varphi)$, its *characteristic formula* $\chi(\pi)$ is $\mathit{pro}_\pi(\mathsf{E},\, \varphi)$.

**Definition 18 (Assumption set).** For any position $\mathbf{u} = (\mathsf{p}, \mathsf{E}, \varphi)$, its *assumption set* relative to $\mathsf{q}$ is the set

$$\mathit{AS}(\mathbf{u}, \mathsf{q}) = \{\chi(\pi) \mid \mathbf{u}\pi\mathbf{v} \text{ is a V-play with } \mathbf{v} = (\mathsf{q}, \mathsf{E}', \psi)\}.$$

**Definition 19 (Canonical sequent).** For any position $\mathbf{u} = (\mathsf{p}\,\mathsf{q}, \mathsf{E}, \varphi)$, the *canonical sequent* relative to $\mathsf{p}$ is the sequent

$$\mathcal{S}(\mathbf{u}, \mathsf{p}) \;=\; \mathsf{D}_{\mathsf{E}}\,;\, \{x\!:\!\Psi \mid \Psi \in AS(\mathbf{u}, \mathsf{q})\} \;\vdash\; \mathsf{p}\,x\!:\!\varphi,$$

where $\mathsf{D}_{\mathsf{E}} = \{X \geqslant \psi \mid X = \psi \text{ occurs in } \mathsf{E}\}$.

This is a good definition because the set $AS(\mathbf{u}, \mathsf{q})$ is finite.

The next two lemmas show how to build up derivation trees between canonical sequents of the form $\mathcal{S}(\mathbf{u}, \mathsf{P})$, where $\mathsf{P}$ is nonterminal, providing the "larger steps" between such sequents discussed above.

**Lemma 1.** *Given a position* $\mathbf{u} = (\mathsf{Q}_1 \ldots \mathsf{Q}_\mathsf{k}\,\mathsf{r}, \mathsf{E}, \varphi)$*, where* $\mathsf{Q}_1, \ldots, \mathsf{Q}_\mathsf{k}$ *are nonterminals, the sequent* $\mathcal{S}(\mathbf{u}, \mathsf{Q}_1 \ldots \mathsf{Q}_\mathsf{k})$ *occurs as the root of a derivation tree in which each leaf has the form* $\mathcal{S}_\pi = \mathcal{S}(\mathbf{v}_\pi, \mathsf{Q}_\mathsf{i})$*, where* $\mathbf{u}\pi\mathbf{v}_\pi$ *is a V-play and* $\mathbf{v}_\pi = (\mathsf{Q}_\mathsf{i} \ldots \mathsf{Q}_\mathsf{k}\,\mathsf{r}, \mathsf{E}_\pi, \psi_\pi)$ *for some i. Moreover, if the play* $\pi$ *preserves (respectively progresses on)* $X$ *then so does the unique path from the root to* $\mathcal{S}_\pi$*.*

The proof repeatedly uses (Cut) and (Sub) to remove each nonterminal $\mathsf{Q}_\mathsf{i}$ from the sequence $\mathsf{Q}_\mathsf{i} \ldots \mathsf{Q}_\mathsf{k}$, applying appropriate proof rules to convert the induced subgoal for $\mathsf{Q}_{\mathsf{i}+1} \ldots \mathsf{Q}_\mathsf{k}$ into the required form for the process to be repeated.

**Lemma 2.** *Given a position* $\mathbf{u} = (\mathsf{Q}\,\mathsf{r}, \mathsf{E}, \varphi)$*, where* $\mathsf{Q}$ *is nonterminal, the sequent* $\mathcal{S}(\mathbf{u}, \mathsf{Q})$ *occurs as the root of a derivation tree in which each leaf has the form* $\mathcal{S}_\pi = \mathcal{S}(\mathbf{v}_\pi, \mathsf{Q}_\pi)$*, where* $\mathsf{Q}_\pi$ *is nonterminal,* $\mathbf{v}_\pi = (\mathsf{Q}_\pi\,\mathsf{q}_\pi\,\mathsf{r}, \mathsf{E}_\pi, \psi_\pi)$ *and* $\mathbf{u}\pi\mathbf{v}_\pi$ *is a V-play containing at least one move. If the play* $\pi$ *preserves (respectively progresses on)* $X$ *then so does the unique path to* $\mathcal{S}_\pi$ *in the derivation tree.*

The proof is by case analysis on $\varphi$ using the sequent rules to mimic the possible moves of any $V$-play. In the case of the modal rules, Lemma 1 is used to break down any sequence of nonterminals produced.

Lemma 2 builds derivation trees between canonical sequents and relates preservation and progress properties of paths through the trees to analogous properties of the V-plays used to construct them. As discussed above, these properties allow the derivation trees to be combined into a proof, yielding:

**Lemma 3 (Main lemma).** *For any position* $\mathbf{u} = (\mathsf{p}\,\mathsf{q}, \mathsf{E}, \varphi)$ *the canonical sequent* $\mathcal{S}(\mathbf{u}, \mathsf{p})$ *has a proof.*

The proof of Theorem 2 is now concluded by using (Cut) and (Sub) to combine Lemma 3 above with the relatively straightforward:

**Lemma 4.** *For any position* $\mathbf{u} = (\mathsf{p}, \mathsf{E}, \varphi)$ *and* $\Psi$ *in the assumption set* $AS(\mathbf{u}, \varepsilon)$ *the sequent* $\vdash \varepsilon\!:\!\Psi$ *has a proof.*

## 5 Discussion and Future Work

Our proof of completeness for context-free processes makes essential use of $\nu$-approximant declarations and modifiers. These features can be incorporated into Dam and Gurov's proof system [9], by extending their syntax with ordinal quantifiers $\forall\kappa.\,\varphi$ and $\forall\kappa' \!<\! \kappa.\,\varphi$. Indeed, the completeness proof for context-free processes was originally developed in this context in the first author's MSc dissertation [16]. We do not know whether context-free completeness holds for Dam and Gurov's system without ordinal quantifiers.

It is natural to ask whether the approach in this paper might extend to obtain completeness for richer classes of processes, such as pushdown processes [15,20,3]. In a different direction, it would be very interesting to ascertain to what extent one can obtain completeness results for parameterized verification goals of the form (1), see Sect. 1.

## References

1. B. Bloom, S. Istrail, and A.R. Meyer. Bisimulation can't be traced. *J. Assoc. Comput. Mach.*, 42:232–268, 1995.
2. O. Burkart, D. Caucal, F. Moller, and B. Steffen. Verification over infinite states. In *Handbook of Process Algebra*, pages 545–623. Elsevier, 2001.
3. O. Burkart and B. Steffen. Model checking the full modal mu-calculus for infinite sequential processes. *Theoretical Computer Science*, 221(1–2):251–270, 1999.
4. M. Dam. Compositional proof systems for model checking infinite state processes. In *International Conference on Concurrency Theory*, pages 12–26, 1995.
5. M. Dam. Proving properties of dynamic process networks. *Information and Computation*, 140(2):95–114, 1998.
6. M. Dam. Proof systems for $\pi$-calculus logics. In R. de Queiroz, editor, *Logic for Concurrency and Synchronisation*. OUP, 2001.
7. M. Dam, L. Fredlund, and D. Gurov. Toward parametric verification of open distributed systems. In A. Pnueli H. Langmaack and W.-P. de Roever, editors, *Compositionality: the Significant Difference*. Springer, 1998.
8. M. Dam and D. Gurov. Compositional verification of CCS processes. In *Proceedings of PSI'99*. Springer LNCS 1755, 1999.
9. M. Dam and D. Gurov. $\mu$-calculus with explicit points and approximations. *Journal of Logic and Computation*, to appear, 2001. Abstract in Proceedings of FICS 2000.
10. J. Esparza and J. Knoop. An automata-theoretic approach to interprocedural dataflow analysis. In *Proceedings of FOSSACS'99*. Springer LNCS 1578, 1999.
11. L. Fredlund. A framework for reasoning about Erlang code. PhD Thesis, Swedish Institute of Computer Science, 2001.
12. M. Hennessy and R. Milner. Algebraic laws for nondeterminism and concurrency. *J. Assoc. Comput. Mach.*, 32:137–161, 1985.
13. H. Hungar and B. Steffen. Local model checking for context-free processes. *Nordic Journal of Computing*, 1(3):364–385, Fall 1994.
14. D. Kozen. Results on the propositional $\mu$-calculus. *Theoretical Computer Science*, 27:333–354, 1983.
15. D.E. Muller and P.E. Schupp. The theory of ends, pushdown automata, and second-order logic. *Theoretical Computer Science*, 37:51–75, 1985.
16. U. Schöpp. Formal verification of processes. MSc Dissertation, University of Edinburgh, 2001. Available as `http://www.dcs.ed.ac.uk/home/us/th.ps.gz`.
17. A.K. Simpson. Compositionality via cut-elimination: Hennessy-Milner logic for an arbitrary GSOS. In *Logic in Computer Science*, pages 420–430, 1995.
18. C.P. Stirling. Modal logics for communicating systems. *Theoretical Computer Science*, 49:311–347, 1987.
19. C.P. Stirling. *Modal and temporal properties of processes.* Texts in Computer Science. Springer, 2001.
20. I. Walukiewicz. Pushdown processes: games and model-checking. *Information and Computation*, 164(2):234–263, January 2001.