Compositional Proof Systems for Model Checking Infinite State Processes

Mads Dam*
Swedish Institute of Computer Science

February 28, 1995

Abstract

We present the first compositional proof system for checking processes against formulas in the modal μ-calculus which is capable of handling general infinite-state processes. The proof system is obtained in a systematic way from the operational semantics of the underlying process algebra. A non-trivial proof example is given, and the proof system is shown to be sound in general, and complete for finite-state processes.

1 Introduction

In this paper we address the problem of verifying modal μ-calculus properties of general infinite-state processes, and we present what we believe to be the first genuinely compositional solution to this problem.

The value of compositionality in program logics has been well understood ever since the advent of Hoare logic. Compositionality allows better structuring and decomposition of the verification task, it allows proof reuse, and it allows reasoning about partially instantiated programs, thus supporting program synthesis. Even more fundamentally it allows, at least in principle, verification exercises to be undertaken which are beyond the scope of more global approaches because the set of reachable global states grows in an unbounded manner. The problem of how to build compositional proof systems for concurrent systems, however, has long been recognised as a very difficult one. Many techniques have been suggested in the literature, such as rely-guarantee pairs, history variables, quotienting, reduction, phantom moves, simulations, edge propositions, quiescent traces, to name but a few. These techniques, however, give only partial and ad-hoc solutions in that they work only for particular concurrency primitives, static process networks and, most often, linear time logic only.

Much recent research in the area has focused on process algebra and the modal μ-calculus. A large number of algorithms, tableau systems, and proof systems for verifying processes against modal μ-calculus specifications by some form of global state space exploration have been given (c.f. [6, 7, 8, 10, 15, 22] and many others). Compositional accounts have been developed based on some form of quotienting, or reduction (c.f. [16, 3, 2]). These approaches, however, are only applicable for finite-state processes, or at least when the holding of a property depends only on a finite portion of a potentially infinite-state process.

Finite-state processes, however, are inadequate as modelling tools in many practical situations. Value- or channel passing, for instance, can cause even the simplest processes to

*Work partially supported by ESPRIT BRA project 8130 LOMAPS. Author address: SICS, Box 1263, S-164 28 Kista, Sweden. Email: mfd@sics.se. Phone: +46 8 752 1500. Fax: +46 8 751 7230

become infinite state. While some decidability results can be obtained in the absence of process spawning (c.f. [9]), in general the model checking problem becomes undecidable, even in very sparse fragments of, e.g., CCS [11]. Process spawning, however, is needed in many applications: Unbounded buffers, dynamic resource or process creation/forking, data types and higher order features in the π-calculus (c.f.[18]). In fact it is hard to conceive of useful program logics for distributed functional languages such as CML [20], Facile [24], Erlang [4], or PICT [19] that can not deal with process spawning, and indeed the development of such logics is one aim of the research reported here.

Because of undecidability, and because the modal μ-calculus is closed under negation, finitary proof systems for model checking general infinite state processes will necessarily be incomplete. This, however, does not make the model checking problem go away! The currently prevailing finite-state approaches (whether they are based on iterative or local techniques) provide little assistance: They are inadequate for even rather simple infinite state problems such as the "counter" example considered below. Here we explore instead a compositional approach. Our aim is to obtain a compositional proof system which is (1) sound, (2) practically useful, (3) powerful enough to prove the kinds of infinite state problems we would hope to be able to address, and (4) complete for the finite state fragment. For (1) and (4) we have positive answers, while more work is needed to answer (2) and (3).

Compositionality is addressed by taking a more general view of model checking. Instead of focusing on closed assertions like $\models P : \phi$ we look at sequents of the form $x_1 : \phi_1, ..., x_n : \phi_n \models P(x_1, ..., x_n) : \phi$. That is, properties of the open process term $P(x_1, ..., x_n)$ are relativised to properties of its free variables $x_1, ..., x_n$. This provides a more general proof-theoretical setting which can be used to give a structural account of recursive properties. This is a fairly easy task for those connectives like $\wedge$, $\vee$, or the modal operators, that depend only on "local" behaviour. For the fixed point operators the problem is much more difficult. Here we offer an approach based on loop detection, generalising approaches to local model checking of finite state processes such as that of Stirling and Walker [22]. To guide us towards a general solution we offer in this paper a formal proof to show that the CCS process $Counter = up.(Counter \mid down.\mathbf{0})$ after any sequence of consecutive *up* transitions can only perform a finite sequence of consecutive *down* transitions.

An important feature of our approach is that, in contrast to other existing compositional accounts, the sequent style proof system we obtain is constructed from the operational semantics in quite a general and systematic manner. The proof system contains four separate elements: Structural rules, including a cut-rule, to account for sequent structure; logical rules that deal with boolean connectives and recursive formulas; dynamical rules that deal with the modal operators; and finally a single rule of discharge that is responsible for detecting "safe" recurrences of sequents. Only the dynamical rules are dependent upon the specific process algebra under consideration. Moreover the dynamical rules are constructed in a way that one can easily foresee being automated for a range of process algebras.

2 CCS

Our use of CCS follows [17] closely. An *action*, α, is either the invisible action τ or a label l. A *label* is either a (port- or channel-) *name* a, b, say, or a *co-name* $\overline{a}$, $\overline{b}$. Generally $\overline{\overline{a}}$ and a are identified. We assume that the set of labels is finite and ranged over by $l_0, \ldots, l_m$. Sets of labels are ranged over by L, K. *Agent expressions*, E, F, are given as follows:

$$E ::= \mathbf{0} \mid \alpha.E \mid E + E \mid E \mid E \mid E \setminus L \mid x \mid \mathrm{fix}x.E$$

where x (and y) range over agent variables. An agent expression is an *agent* if it contains no free agent variables. Agents are ranged over by P, Q. Note that we do not consider the CCS renaming operator here since it adds little of interest to the present account. We refer to [17] for the operational semantics rules.

3 The Modal μ-Calculus

Formulas in the modal μ-calculus involve boolean conjunction and disjunction, modal operators indexed by actions, and least and greatest fixed points. In addition equality and inequality of actions are useful (though not required), primarily to give a reasonable account of the τ-indexed box operator. Formulas, ranged over by ϕ, ψ, γ, are given by

$$\phi ::= \alpha = \beta \mid \neg\phi \mid \phi \wedge \phi \mid [\alpha]\phi \mid X \mid \nu X.\phi$$

where X (Y, Z) ranges over propositional variables. To form fixed point formulas $\nu X.\phi$ we need to require the formal monotonicity condition that all occurrences of X in ϕ are within the scope of a even number of negation symbols $\neg$. We introduce some abbreviations, most of which are familiar:

$$\begin{aligned}
\alpha \neq \beta &\triangleq \neg(\alpha = \beta) && (1)\\
ff &\triangleq \alpha \neq \alpha\\
\phi \vee \psi &\triangleq \neg(\neg\phi \wedge \neg\psi) && (2)\\
\phi \supset \psi &\triangleq \neg\phi \vee \psi\\
\langle\alpha\rangle\phi &\triangleq \neg[\alpha]\neg\phi && (3)\\
\mu X.\phi &\triangleq \neg\nu X.\neg\phi[\neg X/X] && (4)\\
\forall\alpha.\phi &\triangleq \bigwedge\{\phi(\alpha) \mid \alpha \text{ an action}\}
\end{aligned}$$

where in the last case $\phi(\alpha) = \phi[\alpha/a]$ substitutes α for a name a in ϕ, and it is required that ϕ does not have free occurrences of the action $\overline{a}$ so that action terms like $\overline{\tau}$ are avoided.

The semantics of formulas is determined by the set of agents $\|\phi\|\mathcal{V}$ where $\mathcal{V}$ is a valuation assigning sets of agents to propositional variables. The semantics is given as follows:

$$\|\alpha = \beta\|\mathcal{V} = \begin{cases} \mathcal{A} & \text{if } \alpha = \beta \\ \emptyset & \text{otherwise} \end{cases}$$

$$\|\neg\phi\|\mathcal{V} = \{P \mid P \notin \|\phi\|\mathcal{V}\}$$

$$\|\phi \wedge \psi\|\mathcal{V} = \|\phi\|\mathcal{V} \cap \|\psi\|\mathcal{V}$$

$$\|[\alpha]\phi\|\mathcal{V} = \{P \mid \forall P'.\ \text{if } P \xrightarrow{\alpha} P' \text{ then } P' \in \|\phi\|\mathcal{V}\}$$

$$\|X\|\mathcal{V} = \mathcal{V}(X)$$

$$\|\nu X.\phi\|\mathcal{V} = \bigcup\{A \mid A \subseteq \|\phi\|\mathcal{V}[X \mapsto A]\}.$$

Instead of $P \in \|\phi\|\mathcal{V}$ we sometimes write $\models_{\mathcal{V}} P : \phi$, or $\models P : \phi$ if ϕ is closed. Any closed formula can be rewritten, while preserving semantics, into *positive form*, using negation only as needed by use of the derived operators given by (1)–(4) above. The proof system below uses positive forms extensively.

4 Sequents

The basic judgment of the proof system is the sequent.

Definition 4.1 (Sequents, declarations, basic assertions) A *sequent* is an expression of the form $\Gamma \vdash t$ where Γ is a sequence of *declarations* of one of the forms $x : \phi$ or $X = \phi$, and t is a *basic assertion* of one of the forms $X = \phi$ or $E : \phi$.

Sequents are ranged over by s. Declarations of the form $X = \phi$ are called *namings*, and if s contains the naming $X = \phi$ then X is said to *name* ϕ in s. An occurrence of a variable X to the left of the equality sign in a naming $X = \phi$ is regarded as binding. Namings are used as constants in [22], and serve to keep track of the unfoldings of fixed point formula occurrences in the proof system. We use σ as a meta-variable over $\{\nu, \mu\}$. If X names a formula of the form $\sigma Y.\psi$ in s then X is called a *σ-variable*.

Declaration sequences and sequents are subject to an inductively defined *well-formedness constraint*, in order to ensure that (proposition and process) variables are properly declared. This condition states that variables can be declared at most once, and that for a sequent $\Gamma \vdash E : \phi$, if a variable occurs freely in E or in ϕ then it is declared in Γ, and, for a sequent $\Gamma_1, x : \phi_i, \Gamma_2 \vdash E : \phi$, if a variable occurs freely in ϕ_i then it is declared in Γ_1.

Definition 4.2 (Sequent semantics)

1. The sequent $\vdash P : \phi$ is $\mathcal{V}$-true if and only if $P \in \|\phi\|\mathcal{V}$
2. The sequent $\vdash X = \phi$ is $\mathcal{V}$-true if and only if $\mathcal{V}(X) = \|\phi\|\mathcal{V}$
3. The sequent $\Gamma, x : \phi \vdash t$ is $\mathcal{V}$-true if and only if for all agent expressions E, if $\Gamma \vdash E : \phi$ is $\mathcal{V}$-true then so is $\Gamma \vdash t[E/x]$.
4. The sequent $\Gamma, X = \phi \vdash t$ is $\mathcal{V}$-true if and only if $\Gamma \vdash t$ is $\mathcal{V}[X \mapsto \|\phi\|\mathcal{V}]$-true.

If the sequent s is well-formed then the $\mathcal{V}$-truthhood of s is well-defined and independent of $\mathcal{V}$. Notice that the quantification over agent expressions in def. 4.2.3 could equivalently be replaced by a quantification over agents.

5 Local Rules

We are now in a position to present the proof system. It consists of two subsystems, a *local* and a *global* one. We first introduce the local subsystem. The local subsystem, shown on fig. 1, is subdivided into three groups: *Structural rules* governing the use of declarations, *logical rules* responsible for the left and right introduction of logical operators, and finally *dynamical rules* for the modal operators which depend on process structure. The first two sets of rules require little comment, coming, as they do, straight from proof theory. Only noteworthy points are the use of variables to name fixed point formulas, and that symmetric versions of the $\wedge$-LEFT and $\vee$-RIGHT rules have been omitted. Similarly, symmetric versions of the $+$-$\Diamond$, $|$-$\Diamond$, and rules derived from $+$-$\Box$ and the rules for parallel composition, obtained by systematically exchanging the declarations for x and for y, have been omitted.

The rationale behind the dynamical rules is best explained through a little example. Suppose we wish to prove $\vdash P \mid Q : \langle\alpha\rangle\phi$, because we suspect that (1) $P \xrightarrow{\alpha} P'$ and (2) $\models P' \mid Q : \phi$. Our task is to

1. guess a property ϕ_1 of P' and a property ϕ_2 of Q,

$$\textsc{Declaration}\quad \frac{\cdot}{\Gamma_1, t, \Gamma_2 \vdash t} \qquad \textsc{Cut}\quad \frac{\Gamma_1, \Gamma_2 \vdash E : \phi \quad \Gamma_1, x : \phi, \Gamma_2 \vdash F : \psi}{\Gamma_1, \Gamma_2 \vdash F[E/x] : \psi}$$

Logical rules:

$$\neg\neg\text{-}\textsc{Right}\quad \frac{\Gamma \vdash E : \phi}{\Gamma \vdash E : \neg\neg\phi} \qquad \neg\neg\text{-}\textsc{Left}\quad \frac{\Gamma_1, x : \phi, \Gamma_2 \vdash E : \psi}{\Gamma_1, x : \neg\neg\phi, \Gamma_2 \vdash E : \psi}$$

$$=\text{-}\textsc{Right}\quad \frac{\cdot}{\Gamma \vdash E : \alpha = \alpha} \qquad =\text{-}\textsc{Left}\quad \frac{\cdot}{\Gamma_1, x : \alpha = \beta, \Gamma_2 \vdash E : \psi}\quad (\alpha \neq \beta)$$

$$\neq\text{-}\textsc{Right}\quad \frac{\cdot}{\Gamma \vdash E : \alpha \neq \beta}\quad (\alpha \neq \beta) \qquad \neq\text{-}\textsc{Left}\quad \frac{\cdot}{\Gamma_1, x : \alpha \neq \alpha, \Gamma_2 \vdash E : \psi}$$

$$\wedge\text{-}\textsc{Right}\quad \frac{\Gamma \vdash E : \phi \quad \Gamma \vdash E : \psi}{\Gamma \vdash E : \phi \wedge \psi} \qquad \wedge\text{-}\textsc{Left}\quad \frac{\Gamma_1, x : \phi, \Gamma_2 \vdash E : \gamma}{\Gamma_1, x : \phi \wedge \psi, \Gamma_2 \vdash E : \gamma}$$

$$\vee\text{-}\textsc{Right}\quad \frac{\Gamma \vdash E : \phi}{\Gamma \vdash E : \phi \vee \psi} \qquad \vee\text{-}\textsc{Left}\quad \frac{\Gamma_1, x : \phi, \Gamma_2 \vdash E : \gamma \quad \Gamma_1, x : \psi, \Gamma_2 \vdash E : \gamma}{\Gamma_1, x : \phi \vee \psi, \Gamma_2 \vdash E : \gamma}$$

$$\sigma\text{-}\textsc{Right}\quad \frac{\Gamma, Y = \sigma X.\phi \vdash E : Y}{\Gamma \vdash E : \sigma X.\phi} \qquad \sigma\text{-}\textsc{Left}\quad \frac{\Gamma_1, Y = \sigma X.\phi, x : Y, \Gamma_2 \vdash E : \psi}{\Gamma_1, x : \sigma X.\phi, \Gamma_2 \vdash E : \psi}$$

$$Y\text{-}\textsc{Right}\quad \frac{\Gamma \vdash Y = \sigma X.\phi \quad \Gamma \vdash E : \phi[Y/X]}{\Gamma \vdash E : Y}$$

$$Y\text{-}\textsc{Left}\quad \frac{\Gamma_1 \vdash Y = \sigma X.\phi \quad \Gamma_1, x : \phi[Y/X], \Gamma_2 \vdash E : \psi}{\Gamma_1, x : Y, \Gamma_2 \vdash E : \psi}$$

Dynamical rules:

$$\mathbf{0}\text{-}\Box\quad \frac{}{\Gamma \vdash \mathbf{0} : [\alpha]\phi} \qquad \alpha.\text{-}\Diamond\quad \frac{\Gamma_1, x : \phi, \Gamma_2 \vdash x : \psi}{\Gamma_1, x : \phi, \Gamma_2 \vdash \alpha.x : \langle\alpha\rangle\psi}$$

$$\alpha.\text{-}\Box\text{-}1\quad \frac{\Gamma_1, x : \phi, \Gamma_2 \vdash x : \psi}{\Gamma_1, x : \phi, \Gamma_2 \vdash \alpha.x : [\alpha]\psi} \qquad \alpha.\text{-}\Box\text{-}2\quad \frac{\cdot}{\Gamma \vdash \alpha.E : [\beta]\phi}\quad (\alpha \neq \beta)$$

$$+\text{-}\Diamond\quad \frac{\Gamma_1, x : \phi, \Gamma_2 \vdash x : \psi}{\Gamma_1, x : \langle\alpha\rangle\phi, \Gamma_2 \vdash x + F : \langle\alpha\rangle\psi} \qquad +\text{-}\Box\quad \frac{\Gamma_1, x : \phi_1, \Gamma_2, \Gamma_3 \vdash x : \psi \quad \Gamma_1, \Gamma_2, y : \phi_2, \Gamma_3 \vdash y : \psi}{\Gamma_1, x : [\alpha]\phi_1, \Gamma_2, y : [\alpha]\phi_2, \Gamma_3 \vdash x + y : [\alpha]\psi}$$

$$|\text{-}\langle\alpha\rangle\quad \frac{\Gamma_1, x : \phi, \Gamma_2, y : \psi, \Gamma_3 \vdash x \mid y : \gamma}{\Gamma_1, x : \langle\alpha\rangle\phi, \Gamma_2, y : \psi, \Gamma_3 \vdash x \mid y : \langle\alpha\rangle\gamma} \qquad |\text{-}\langle\tau\rangle\quad \frac{\Gamma_1, x : \phi, \Gamma_2, y : \psi, \Gamma_3 \vdash x \mid y : \gamma}{\Gamma_1, x : \langle l\rangle\phi, \Gamma_2, y : \langle\overline{l}\rangle\psi, \Gamma_3 \vdash x \mid y : \langle\tau\rangle\gamma}$$

$$|\text{-}[\alpha]\quad \frac{\Gamma_1, x : \phi_1, \Gamma_2, y : \psi_2, \Gamma_3 \vdash x \mid y : \gamma \quad \Gamma_1, x : \phi_2, \Gamma_2, y : \psi_1, \Gamma_3 \vdash x \mid y : \gamma}{\Gamma_1, x : \phi_1 \wedge [\alpha]\phi_2, \Gamma_2, y : \psi_1 \wedge [\alpha]\psi_2, \Gamma_3 \vdash x \mid y : [\alpha]\gamma}\quad (\alpha \neq \tau)$$

$$|\text{-}[\tau]\quad \frac{\begin{array}{c}\Gamma_1, x : \phi_2(\tau), \Gamma_2, y : \psi_1, \Gamma_3 \vdash x \mid y : \gamma \\ \Gamma_1, x : \phi_1, \Gamma_2, y : \psi_2(\tau), \Gamma_3 \vdash x \mid y : \gamma \\ \Gamma_1, x : \phi_2(l_0), \Gamma_2, y : \psi_2(\overline{l_0}), \Gamma_3 \vdash x \mid y : \gamma \\ \vdots \\ \Gamma_1, x : \phi_2(l_m), \Gamma_2, y : \psi_2(\overline{l_m}), \Gamma_3 \vdash x \mid y : \gamma\end{array}}{\Gamma_1, x : \phi_1 \wedge \forall\alpha.[\alpha]\phi_2(\alpha), \Gamma_2, y : \psi_1 \wedge \forall\beta.[\beta]\psi_2(\beta), \Gamma_3 \vdash x \mid y : [\tau]\gamma}$$

$$\backslash\text{-}\Box\text{-}1\quad \frac{\cdot}{\Gamma \vdash E \setminus K : [\alpha]\psi}\quad (\alpha \in K) \qquad \backslash\text{-}\Box\text{-}2\quad \frac{\Gamma_1, x : \phi, \Gamma_2 \vdash x \setminus K : \psi}{\Gamma_1, x : [\alpha]\phi, \Gamma_2 \vdash x \setminus K : [\alpha]\psi}$$

$$\backslash\text{-}\Diamond\quad \frac{\Gamma_1, x : \phi, \Gamma_2 \vdash x \setminus K : \psi}{\Gamma_1, x : \langle\alpha\rangle\phi, \Gamma_2 \vdash x \setminus K : \langle\alpha\rangle\psi}\quad (\alpha \notin K) \qquad \textsc{Fix}\quad \frac{\Gamma \vdash E[\mathrm{fix}x.E/x] : \phi}{\Gamma \vdash \mathrm{fix}x.E : \phi}$$

Figure 1: Local proof rules

2. prove $\vdash P : \langle\alpha\rangle\phi_1$ and $\vdash Q : \phi_2$,

3. prove $x_1 : \phi_1, x_2 : \phi_2 \vdash x_1 \mid x_2 : \phi$, and finally,

4. put (2) and (3) together using two cuts and $|\text{-}\langle\alpha\rangle$ to conclude $\vdash P \mid Q : \langle\alpha\rangle\phi$.

Comparing with local model checking systems such as Stirling and Walker's [22] this account has sacrificed a subformula property ($\vdash P \mid Q : \langle\alpha\rangle\phi$ is proved in terms of processes having the property ϕ) in favour of a subprocess property ($\vdash P \mid Q : \langle\alpha\rangle\phi$ is proved in terms of properties holding of the processes P and Q). We regard this as quite natural and reflecting closely the *compositional* nature of the proof system. We do not expect that any of the tasks (1)–(3) can be automated, although this is possible in special cases, in particular for the case of finite state processes considered later.

Note that some slight modifications of the proof system are possible. For instance can a rule like $\alpha.\text{-}\Diamond$ be formulated as a nullary rule:

$$\frac{\cdot}{\Gamma_1, x : \phi, \Gamma_2 \vdash \alpha.x : \langle\alpha\rangle\phi}$$

The specific formulation chosen here is chosen to avoid side-conditions which would otherwise be required when we come to consider loop detection. Note also that FIX is just a mild generalisation of two rules, one dealing with $\langle\alpha\rangle$ and the other with $[\alpha]$, that adhere more closely to the format of the other dynamical rules.

6 An Example Proof

In this section we give an example proof to (1) show the local rules at work, and to (2) serve as a setting for discussing termination conditions. The example proves that the infinite state process $Counter = \mathrm{fix}x.up.(x \mid down.\mathbf{0})$ satisfies the property $\phi = \nu X.(\mu Y.[down]Y) \wedge [up]X$, i.e. after any finite consecutive sequence of up's only a finite number of consecutive $down$'s is possible. We use a goal-directed approach. Thus the initial goal is $\vdash Counter : \phi$. We first name ϕ, obtaining the sequent

$$U{=}\phi \vdash Counter : U. \tag{5}$$

We then unfold $Counter$ and U, apply CUT and $\wedge$-RIGHT, and arrive at the two subgoals

$$U{=}\phi \vdash up.(Counter \mid down.\mathbf{0}) : (\mu Y.[down]Y) \tag{6}$$

$$U{=}\phi \vdash up.(Counter \mid down.\mathbf{0}) : [up]U. \tag{7}$$

Subgoal (6) is easily handled by naming the μ-formula, unfolding, and then applying $\alpha.\text{-}\Box\text{-}2$, so we proceed refining subgoal (7). First using $\alpha.\text{-}\Box\text{-}1$ we obtain

$$U{=}\phi \vdash Counter \mid down.\mathbf{0} : U$$

Now let $\psi = [down][down]ff \wedge [up]ff$, and by two applications of CUT refine to the subgoals

$$U{=}\phi \vdash Counter : U \tag{8}$$

$$U{=}\phi \vdash down.\mathbf{0} : \psi \tag{9}$$

$$U{=}\phi, x : U, y : \psi \vdash x \mid y : U. \tag{10}$$

Of these, (9) is eliminated by $\wedge$-RIGHT, α.-$\Box$-1 and α.-$\Box$-2. For (8) our intention is to terminate because (8) has previously been expanded as (5), and U is a ν-variable so termination is safe. This indeed is a rough but basically sound interpretation of the termination conditions, and proof development at the goal (8) *can* in fact be terminated. Thus, (10) is all that remains. Now, by unfolding and $\wedge$-RIGHT we obtain the subgoals

$$U{=}\phi, x : U, y : \psi \vdash x \mid y : \mu Y.[down]Y \tag{11}$$

$$U{=}\phi, x : U, y : \psi \vdash x \mid y : [up]U. \tag{12}$$

We delay consideration of (11) and concentrate on (12). First unfold the left hand occurrence of U to obtain

$$U{=}\phi, x : \mu Y.[down]Y \wedge [up]U, y : \psi \vdash x \mid y : [up]U. \tag{13}$$

Now the rule |-[α] applies to reduce to the four subgoals

$$U{=}\phi, x : \mu Y.[down]Y \wedge [up]U, y : \psi \vdash x : [up]U \tag{14}$$

$$U{=}\phi, x : \mu Y.[down]Y \wedge [up]U, y : \psi \vdash y : [up]ff \tag{15}$$

$$U{=}\phi, x : U, y : \psi \vdash x \mid y : U \tag{16}$$

$$U{=}\phi, x : \mu Y.[down]Y \wedge [up]U, y : ff \vdash x \mid y : U \tag{17}$$

Of these, (14) and (15) are easily proved using $\wedge$-L and some simple boolean reasoning. (17) is proved using $\neq$-LEFT. Finally, (16) is discharged using (10) since U is a ν-variable (!). We then need to consider (11). We first name the right hand μ-formula, letting $\gamma = \mu Y.[down]Y$:

$$U{=}\phi, x : U, y : \psi, V{=}\gamma \vdash x \mid y : V.$$

We next unfold the left hand occurrence of U, and after an application of $\wedge$-LEFT we obtain

$$U{=}\phi, x : \gamma, y : \psi, V{=}\gamma \vdash x \mid y : V.$$

Now the left hand μ-formula is named too:

$$U{=}\phi, W{=}\gamma, x : W, y : \psi, V{=}\gamma \vdash x \mid y : V \tag{18}$$

and V is unfolded:

$$U{=}\phi, W{=}\gamma, x : W, y : \psi, V{=}\gamma \vdash x \mid y : [down]V. \tag{19}$$

Unfolding W to the left gives

$$U{=}\phi, W{=}\gamma, x : [down]W, y : \psi, V{=}\gamma \vdash x \mid y : [down]V. \tag{20}$$

which reduces through CUT, |-[α], and some logical reasoning to the two subgoals

$$U{=}\phi, W{=}\gamma, x : W, y : \psi, V{=}\gamma \vdash x \mid y : V \tag{21}$$

$$U{=}\phi, W{=}\gamma, x : [down]W, y : [down]ff, V{=}\gamma \vdash x \mid y : V. \tag{22}$$

We now arrive at a key point in the proof where we wish to discharge (21) with reference to subgoal (18), because even though the μ-variable V to the right of the turnstyle has been

unfolded from (18) to (21) so has another μ-variable, W, to the left of the turnstyle. Thus, intuitively, if we assume the left hand side to be true this will ensure that in fact W, and hence V, will only be unfolded a finite number of times, and hence termination at the point (21) is safe. As above, this is a rough but basically sound interpretation of the termination conditions which we go on to explain in the next section. Finally the proof is completed, refining (22) by first unfolding V and then using $|\text{-}[\alpha]$ in a very similar way to the way (19) was dealt with. This is left as an exercise for the reader.

7 Side-conditions and Global Rules

We proceed to explain the global rules justifying the discharge of hypotheses at steps (8), (16), and (21) in the previous section.

A *basic proof structure* (b.p.s.) $\mathcal{B}$ is a proof tree constructed according to the local proof rules. Basic proof structures may contain occurrence of *hypotheses*. The global subsystem consists of a single *rule of discharge* that determines which occurrences of hypotheses can be discharged, along the lines suggested in section 6. A *proof*, then, is a basic proof structure for which all occurrences of hypotheses have been discharged.

To arrive at a sound rule of discharge it is necessary to

1. consider the ways formulas are "regenerated" along paths through a basic proof structure, and

2. count the number of unfoldings of ν-variables.

The first problem is familiar from most accounts of local fixed point unfolding in the modal μ-calculus such as Strett and Emerson [23], or Cleaveland [7], where it is handled using a subformula condition, and Stirling and Walker [22] where it is handled using propositional constants. Here the problem is more delicate, due, principally, to the CUT rule which admits a branching of the regeneration relation which is not otherwise possible. The second problem is due to the fact that we are working with left- and right-handed sequents. Let us anticipate the soundness proof a little. We prove soundness by assuming a proof to be given and a sequent in the proof to be false. Let the sequent concerned by of the form $\Gamma \vdash E : X$, X a ν-variable. We can then find a substitution σ validating Γ and making $\sigma(X)$ false when X is annotated by some suitable ordinal. By applications of CUT this annotation may cause occurrences of X to the left of the turnstyle in some "later" sequent $\Gamma' \vdash E' : \phi'$ to be annotated too. Unfolding specific occurrences of X may cause the annotation of that occurrence to be decreased. We need to arrive at a contradiction even when $\Gamma' \vdash E' : \phi'$ is the conclusion of a nullary rule, say DECLARATION. But if the annotation of a X to the left is *less than* the annotation of X to the right then there is no guarantee of a contradiction, and soundness may fail.

7.1 Colouring, Generation, Activity

The basic device we use for handling regeneration is the concept of *colouring*.

Definition 7.1 (Colouring) A *colouring* of a sequent $\Gamma \vdash t$ is an assignment of distinct colours to formula occurrences either as ψ in a declaration $x : \psi$ or as ϕ when t has the form $E : \phi$.

Given a colouring of the conclusion of a local proof rule colourings for the antecedents are derived. A few examples suffice to illustrate the general pattern. Consider e.g. the

rule $\wedge$-RIGHT as stated above, and assume a colouring of $\Gamma \vdash E : \phi \wedge \psi$. The antecedent $\Gamma \vdash E : \phi$ is coloured by keeping the colouring of Γ unchanged and colouring ϕ as $\phi \wedge \psi$. The other antecedent $\Gamma \vdash E : \psi$ is coloured similarly. All other rules except the rule CUT are coloured in a similar fashion. For CUT let a colouring of the conclusion $\Gamma_1, \Gamma_2 \vdash F[E/x] : \psi$ be given. The antecedent $\Gamma_1, \Gamma_2 \vdash E : \phi$ is coloured by keeping the colouring of Γ_1 and Γ_2 unchanged and colouring ϕ as ψ. The antecedent $\Gamma_1, x : \phi, \Gamma_2 \vdash F : \psi$, finally, is coloured by keeping the colourings of Γ_1, Γ_2, and ψ unchanged and choosing a new colour for ϕ. Now, a *coloured basic proof structure* is a b.p.s. $\mathcal{B}$ which is coloured according to the above rules.

Definition 7.2 (Generation) Let a coloured basic proof structure $\mathcal{B}$ and a sequent s_1 in $\mathcal{B}$ be given. Let $\Pi = s_1, \ldots, s_k$ be a path downwards from s_1 to some other sequent s_k in $\mathcal{B}$. Whenever formula occurrences ϕ_1 in s_1 and ϕ_k in s_k exists that are coloured with the same colour then ϕ_k is said to *generate* ϕ_1 *along* Π.

The term "generates" is chosen since we envisage proofs to be constructed in a bottom up fashion from goal to subgoals. Note that the generation relation is independent of choice of colouring. The notion of generation is important since it respects activity of variables in a sense which we go on to explain.

Definition 7.3 (Activity) Let $s = \Gamma \vdash t$ be a well-formed sequent and let ϕ be any formula. Then a variable X is said to be *active in* ϕ (with respect to s) if either X is free in ϕ or else some Y is free in ϕ, Y names some ψ in s, and X is active in ψ.

Note that well-formedness ensures that the "active-in" relation on variables is a preorder. We impose the following side condition on CUT:

Proviso 7.4 Applications of CUT are subject to the condition: For any ν-variable X, if X is active in ϕ then X is also active in ψ.

We can now show the following crucial property concerning "preservation of activity":

Proposition 7.5 *In a b.p.s. $\mathcal{B}$ let a path downwards from s to s' be given. Let ϕ (ϕ') be an occurrence of a formula in s (s'). If ϕ' generates ϕ, X is declared in s', and X is active in ϕ then X is active in ϕ'.* □

7.2 Indexing

We now turn to the counting of unfoldings. An *indexing* is a partial assignment of indices $n \in \omega$ to occurrences of names such that for any sequent $\Gamma \vdash t$, if two occurrences of X in the same formula in Γ or t is given then one is indexed n iff the other one is. Only the rules σ-LEFT, σ-RIGHT, Y-RIGHT, Y-LEFT, DECLARATION, and CUT are affected by indexing. The modifications needed are the following:

1. σ-LEFT and σ-RIGHT: Y is indexed by 0 in both rules.
2. Y-RIGHT and Y-LEFT: The occurrence of Y in the conclusion is indexed by n and occurrences of Y in the antecedent by $n + 1$.
3. DECLARATION: If t has the form $X = \phi$ then there is no change. If t has the form $x : \phi$ then if X is an n-indexed ν-variable in ϕ to the right of the turnstyle then the corresponding occurrence of X in ϕ to the left of the turnstyle is indexed by some $n' \leq n$.
4. CUT: For any ν-variable X, if X is active in ϕ then all occurrences of X in ϕ or ψ are indexed by the same index.

7.3 Regeneration and the Rule of Discharge

We can now state the property of regeneration and the rule of discharge.

Definition 7.6 (Regeneration) In a b.p.s. $\mathcal{B}$ let a path Π downwards from s to s' be given. Suppose that (1) ϕ' generates ϕ along Π, that (2) ϕ and ϕ' are identical up to indexing of variables, that (3) a variable Y is active in ϕ, that (4) Y names the fixed point formula $\sigma X.\psi$ at s', and that (5) ϕ' generates Y along some strict suffix of Π such that Y results from the application of one of the rules Y-RIGHT or Y-LEFT. Then ϕ is *σ-regenerated along* Π *(through Y)*.

Definition 7.7 (Rule of Discharge) Let $\mathrm{FV}(E) = \{x_1, \ldots, x_k\}$ and let

$$s = \Gamma(x_1 : \phi_1, \ldots, x_k : \phi_k) \vdash E : \phi$$

be an occurrence of a hypothesis in a given basic proof structure $\mathcal{B}$. Then s can be discharged provided there below s is, up to indexing, an occurrence of a sequent

$$s' = \Gamma'(x_1 : \phi_1, \ldots, x_k : \phi_k) \vdash E : \phi$$

such that condition (1) below holds along with one of conditions (2) or (3):

1. For all ν-variables X with respect to s', if X is active in ϕ with index n and if X is active in ϕ_i, $1 \leq i \leq k$, with index n' then $n' \leq n$.

2. ϕ is ν-regenerated along the path from s' to s through some X, say. Then it has to be the case that for all $i : 1 \leq i \leq k$, if ϕ_i is ν-regenerated along the path from s' to s through some Y which is active in X then $Y = X$, and if n (n') is the index of X in ϕ in s (s') and if m (m') is the index of Y in ϕ_i in s (s') then $m - m' \leq n - n'$.

3. ϕ is μ-regenerated along the path from s' to s. Then it has to be the case for some $i : 1 \leq i \leq k$, that ϕ_i is μ-regenerated along the path from s' to s too. Moreover, for all $i : 1 \leq i \leq k$, if ϕ_i is ν-regenerated along the path from s' to s through some Y then Y is not active in ϕ.

8 Soundness

We prove that if $\Gamma \vdash E : \phi$ is provable then it is true. The proof uses approximation ordinals. A *partial σ-approximation* ι of a sequent $s = \Gamma \vdash P : \phi$ is a partial annotation of ordinals to free occurrences of variables X in s such that if X occurs in ϕ then X is a ν-variable. It is important to keep apart approximation ordinals and indexing. The latter is a pure book-keeping device designed to keep track of the number of times ν-variables are unfolded as one passes upwards in a proof structure. The semantics of formulas is extended slightly to take variable declarations and approximations into account by the clause

$$\|X\|\Gamma\mathcal{V} = \|\sigma^{(\kappa)}Y.\phi\|\Gamma\mathcal{V} \tag{23}$$

when Γ contains the declaration $X = \sigma Y.\phi$ and X is annotated by κ.

Definition 8.1 (Truth for substitution and partial approximation) The sequent $\Gamma \vdash P : \phi$ is *true for a substitution σ* of agent variables to agents, *and a partial approximation ι* if $P\sigma \in \|\phi\|\Gamma$ provided for all x which are free in P, if $x : \phi_x$ is the declaration of x in Γ then $\sigma(x) \in \|\phi_x\|\Gamma$.

We now embark on the soundness proof proper. Assume that the sequent $s_0 = \Gamma_0 \vdash E_0 : \phi_0$ is false for a substitution σ_0 and partial approximation ι_0. For simplicity we assume that ϕ_0 has no free occurrences of variables. Assume also we have given a proof of $\Gamma_0 \vdash E_0 : \phi_0$. We trace an infinite sequence of the form $\Pi = (s_0, \sigma_0, \iota_0), (s_1, \sigma_1, \iota_1), \ldots$ such that for all i, s_i is false for σ_i and ι_i, and s_i is the conclusion of a proof rule instance for which s_{i+1} is an antecedent. By use of approximation ordinals, and using the fact that infinitely many points along Π must correspond to hypotheses that have been discharged we can then arrive at a contradiction.

Suppose the construction has arrived at the sequent $s_i = \Gamma_i \vdash E_i : \phi_i$. The following properties are maintained invariant:

Property 8.2 *1. Let any two occurrences of a free variable X in ϕ_i be given. If one occurrence is annotated by ι_i they both are, and then the annotations are identical. The same holds for any ψ occurring as part of a declaration $x : \psi$ in Γ_i.*

2. We assume for all ν-variables X that if X is active in both ϕ_i and Γ_i such that X is active in a declaration in Γ_i of a variable which is free in E_i, the active occurrence of X in ϕ_i is indexed by n and approximated by κ, and the active occurrence of X in Γ_i is indexed by n' and approximated by κ', then $\kappa' + n' \geq \kappa + n$.

To motivate the condition 8.2.2 note that $n - n'$ counts how many more times X to the right of the turnstyle has been unfolded than the corresponding occurrence to the left. In some cases, however, unfolding to the left may temporarily outpace unfoldings to the right, violating the invariant temporarily. We postpone discussion of this case until we see it arising.

We show how we can identify $(s_{i+1}, \sigma_{i+1}, \iota_{i+1})$ such that s_{i+1} is false for σ_{i+1} and ι_{i+1} by considering each potential rule in turn.

Structural rules. The only circumstance in which DECLARATION could apply is where some ν-variable occurrence to the left of the turnstyle is annotated by a smaller approximation ordinal than its corresponding occurrence to the right. However, the invariant condition gives $\kappa' + n' \geq \kappa + n$ and $n' \leq n$ (where κ, κ', n and n' are determined as in 8.2.2), hence $\kappa' \geq \kappa$. Suppose then that s_i results from an application of the rule CUT. Consider the instance

$$\textsc{Cut} \quad \frac{\Gamma_1', \Gamma_2' \vdash E' : \phi' \quad \Gamma_1', x' : \phi', \Gamma_2' \vdash F' : \psi'}{\Gamma_1', \Gamma_2' \vdash F'[E'/x'] : \psi'}$$

so that $s_i = \Gamma_1', \Gamma_2' \vdash F'[E'/x'] : \psi'$. Assume that

(i) $\Gamma_1', \Gamma_2' \vdash E' : \phi'$ is true for σ_i and ι_i' where ι_i' annotates variables in Γ_1' or Γ_2' as ι_i, and variables in ϕ' as the corresponding variables in ψ' in s_i.

(ii) $\Gamma_1', x' : \phi', \Gamma_2' \vdash F' : \psi'$ is true for the substitution σ_i' and ι_i' where

- σ_i' is the substitution for which $\sigma_i'(y) = \sigma_i(y)$ whenever $y \neq x'$ and for which $\sigma_i'(x') = E'\sigma_i$.
- ι_i' annotates variable occurrences in Γ_1', Γ_2', and ψ' as the corresponding occurrences in s_i, it annotates no occurrences of μ-variables in ϕ', and it annotates ν-variables in ϕ' as they are annotated in ψ by ι_i.

From (i) and (ii) it follows that s_i must be true for σ_i and ι_i, hence one of them must fail, and we pick as $(s_{i+1}, \sigma_{i+1}, \iota_{i+1})$ whichever combination that does fail. Note that, due to the

side-condition concerning activity for CUT, we ensure that if all ν-variables are annotated in ψ' then the same is true for ϕ'. Note also that the invariants are maintained true by this construction. Note thirdly that it is this step in the construction that requires ν-variables to be approximated (hence also indexed) both to the right and to the left of the turnstyle. This situation does not arise for μ-variables.

Logical rules. The delicate case concerns the rule Y-LEFT, for Y a ν-variable. Let κ' be the annotation of Y. If κ' is a successor ordinal κ' can decremented by 1 without affecting the invariant. However, if κ' is a limit ordinal the invariant may have to be broken. This is the situation, in particular, when $n' \geq n$ — i.e. when the left hand side, due to the application of Y-LEFT is (temporarily) overtaking the right hand side when counting numbers of unfoldings of Y. In this situation we need in obtaining ι_{i+1} to replace κ' by some $\kappa'' < \kappa'$. *Any* such choice of κ'' is in principle possible. Choose some such κ'' at random, and we argue that by using a little backtracking we can eventually reinstate the invariant. The strategy is the following: In the context of any subsequent choice of s_j with $j \geq i+1$ let n refer to the index of Y to the right of the turnstyle, and n' refer to the index of Y in that particular declaration which is generated by the occurrence of Y which is currently being unfolded. Whenever $n \geq n'$ then we can inspect the invariant to see if it is still broken, and if it is then we can backtrack and increment our choice of κ'' to reinstate the invariant at that particular point. We cannot reach a leaf which is not a discharged occurrence of a hypothesis without this situation having arisen. For the same reason neither can we reach a loop sequent — i.e. a sequent which serves as justification for the discharge of a hypothesis. If we reach a discharged occurrence of a hypothesis then, because of conditions 7.7.1–3, we would know that $n \geq n'$ so the invariant will have been reinstated. But then the backtracking argument is completed since we can bound how far into the future we will have to go before we can guarantee that the invariant will have been reinstated. This argument is what motivates condition 7.7.1.

Dynamical rules. These cause no real complications and are left to the reader.

Global rules. Finally we need to consider the case where s_i is a discharged occurrence of a hypothesis. We then find a sequent s_j, $j \leq i$, which is the loop sequent justifying the discharge of s_i. s_i will have the form $s_i = \Gamma_i(x_1 : \phi_{i,1}, \ldots, x_k : \phi_{i,k}) \vdash E_i : \phi_i$ and s_j will have the form $s_j = \Gamma_j(x_1 : \phi_{i,1}, \ldots, x_k : \phi_{i,k}) \vdash E_i : \phi_i$ where $\mathrm{FV}(E) = \{x_1, \ldots, x_k\}$. We know that the invariant holds for s_j (and that no subsequent backtracking will modify the annotations of s_j). In identifying $(s_{i+1}, \sigma_{i+1}, \iota_{i+1})$ we wish to replace s_i by s_j keeping σ_i and ι_i unchanged. We need to check that the invariant is maintained. Let X be any ν-variable which is active in ϕ_i, indexed n_j in s_j, n_i in s_i, and annotated by, say, κ in s_i. Further, let X also be active in one of the $\phi_{i,j}$, indexed n'_j in s_j, n'_i in s_i, and annotated by κ' in s_i. Since the invariant holds for s_i we know that $\kappa' + n'_i \geq \kappa + n_i$. There are two cases: Either X is the ν-variable through which ϕ_i is regenerated, and then $\kappa' + n'_j \geq \kappa' + n'_i - n_i + n_j \geq \kappa + n_j$ as desired. Otherwise ϕ_i is regenerated through some other ν- or μ-variable. In this case we know that, since X is active in ϕ_i that $n_i = n_j$. Moreover by conditions 7.7.2 and 3 we know that, since X is also active in $\phi_{i,j}$, that $n'_i = n'_j$ too. Hence also here $\kappa' + n'_j \geq \kappa + n_j$ and we have shown the invariant to be maintained. Now $(s_{i+1}, \sigma_{i+1}, \iota_{i+1})$ can be derived since one of the local rules apply.

Completing the proof. Having built the infinite sequence Π we find a σ-variable X which is infinitely often regenerated to the right of the turnstyle along Π. If $\sigma = \nu$ a contradiction is obtained since the initial annotation of X is infinitely often decreased along Π. If $\sigma = \mu$ we find a μ-variable which is infinitely often unfolded to the left of the turnstyle along Π and a similar argument applies, completing the soundness proof.

9 Completeness for Finite-state Processes

While we view soundness for general processes as the main contribution of the paper, completeness for finite-state processes is important as a check that no proof power has accidentally been sacrificed.

Theorem 9.1 *If P is a finite-state process and $\models P : \phi$ then $\vdash P : \phi$ is provable.*

PROOF: Theorem 9.1 can be proved by embedding the tableau based model checker of Stirling and Walker [22] into the present setting. Consider the proof system obtained by restricting attention to sequents $\Gamma \vdash P : \phi$ where only namings are allowed in Γ, and where the dynamical rules are replaced by the following two global rules:

$$\langle\alpha\rangle\text{-RIGHT} \quad \frac{\Gamma \vdash P' : \phi}{\Gamma \vdash P : \langle\alpha\rangle\phi} \quad (P \stackrel{\alpha}{\rightarrow} P')$$

$$[\alpha]\text{-RIGHT} \quad \frac{\{\Gamma \vdash P' : \phi \mid P \stackrel{\alpha}{\rightarrow} P'\}}{\Gamma \vdash P : [\alpha]\phi}$$

The rule of discharge is modified by allowing a sequent $s = \Gamma \vdash P : X$ to be discharged whenever X is a ν-variable and there strictly below s is another sequent of the form $\Gamma' \vdash P : X$. Call the proof system ensuing from these changes the *Stirling-Walker system*, and write $\Gamma \vdash_{sw} P : \phi$ for provability in this system. By soundness and completeness [22] we know that $\models P : \phi$ iff $\vdash_{sw} P : \phi$. So assume that P is finite-state and that a proof π of $\vdash_{sw} P : \phi$ is given. Assume for simplicity that $P = P_1 \mid P_2$. We derive by induction in the size of π formulas ϕ_1 and ϕ_2 such that $\vdash_{sw} P_1 : \phi_1$ and $\vdash_{sw} P_2 : \phi_2$ by proofs of size not greater than the size of π, and $x_1 : \phi_1, x_2 : \phi_2 \vdash x_1 \mid x_2 : \phi$ is provable in the compositional system. Once a similar result has been proved for restriction (which is quite simple), it is an easy induction in the size of proof of $\vdash_{sw} P : \phi$ to show that then $\vdash P : \phi$ is provable in the compositional proof system too, establishing the result.

As we traverse π from the root upwards we generate pieces of ϕ_1 and ϕ_2 in a manner which respects the structure of π. The greatest difficulty is to deal with names. Assume that the rule Y-RIGHT is applied in π to a sequent, say, $\Gamma \vdash_{sw} P_1' \mid P_2' : X$, and that $\Gamma \vdash_{sw} X = \nu Y.\psi$, so that the resulting antecedent is $\Gamma \vdash_{sw} P_1' \mid P_2' : \psi[X/Y]$. Pick two fresh variables X_1 and X_2. X_1 will be used in ϕ_1 and X_2 in ϕ_2. These variables are generated whenever we strictly above the sequent occurrence $\Gamma \vdash P_1' \mid P_2' : X$ reach a sequent occurrence of the form $\Gamma' \vdash P_1' \mid P_2' : X$. Let ϕ_1', ϕ_2' be the formulas generated by the sequent occurrence $\Gamma \vdash P_1' \mid P_2' : \psi[X/Y]$. Let then $\phi_i = \nu X_i.\phi_i'$, $i \in \{1,2\}$. By one part of the induction hypothesis, $\vdash_{sw} P_i' : \phi_i'$ can be established from the assumption $\vdash P_i' : X_i$. Thus $\vdash_{sw} P_i' : \phi_i$. From the second part of the induction hypothesis we have a proof in the compositional system of $x_1 : \phi_1', x_2 : \phi_2' \vdash x_1 \mid x_2 : \psi[X/Y]$ from the assumption $x_1 : X_1, x_2 : X_2 \vdash x_1 \mid x_2 : X$, thus obtaining a proof in the compositional system of $x_1 : \phi_1, x_2 : \phi_2 \vdash x_1 \mid x_2 : \nu Y.\psi$ from no assumptions.

This deals with the global part of the construction. For the local part we consider the case where $\Gamma \vdash_{sw} P_1 \mid P_2 : \langle\alpha\rangle\phi$. Suppose, e.g., that $P_1 \stackrel{\alpha}{\rightarrow} P_1'$ and $\Gamma \vdash_{sw} P_1' \mid P_2 : \phi$. By induction we find ϕ_1 and ϕ_2 such that $\vdash_{sw} P_i' : \phi_i$, $i \in \{1,2\}$, $\vdash_{sw} P_2' : \phi_2$, and $x_1 : \phi_1, x_2 : \phi_2 \vdash x_1 \mid x_2 : \phi$. But then $\vdash_{sw} P_i : \langle\alpha\rangle\phi_i$, $i \in \{1,2\}$, and $x_1 : \langle\alpha\rangle\phi_1, x_2 : \phi_2 \vdash \langle\alpha\rangle\phi$ as required. □

Notice that — since model checking in the Stirling-Walker system is decidable for finite-state processes — the proof of Theorem 9.1 gives an efficient strategy for building proofs.

Other strategies can be devised, based on e.g. characteristic formulas. Notice also that the proof makes only limited use of the global rules. Termination is needed for greatest fixed points only, and the side-conditions concerning activity and indexing can be eliminated altogether in favour of the much simpler side-condition for CUT that ϕ is a closed formula.

10 Conclusion

A precursor of the present work is [1] where a proof system for a process passing calculus is presented, though recursive specifications are not addressed.

The main issues left for future work are analyses of the proof power of the general proof system, and of its practical usefulness. The former is likely to be a difficult problem. Usefulness is best determined through experimentation. Constructed, as they are, in a systematic way, the local rules may turn out to be quite natural once practice is built up. Moreover, being compositional the proof system is well suited to support macros and derived rules. The quite complicated side-conditions, on the other hand, may seem disconcerting. The hope is that most of the technicalities concerning indexing and activity can be hidden from the user, in a vein similar to the handling of e.g. ML type inference, or universes in extended versions of the Calculus of Constructions (c.f. [13].

The only completeness criterion we have considered here is weak completeness for finite-state processes. In practice this is too weak, and there are useful rules which we are unable to derive. Examples are $\wedge$-$\vee$ distribution ($x : \phi \wedge (\psi \vee \gamma) \vdash x : (\phi \wedge \psi) \vee (\phi \wedge \gamma)$) and monotonicity under the modal operators (e.g. $x : \phi \vdash x : \psi$ implies $x : \langle\alpha\rangle\phi \vdash x : \langle\alpha\rangle\psi$). Maybe the approach using so-called well-described formulas explored in [1] can be used to obtain stronger completeness results.

An interesting issue is to compare with the proof system of Hungar [14] who succeeds in dealing with non-trivial statically parallel compositions of context-free processes using a global approach. It is quite easy based on the present ideas to develop a sound proof system for context-free processes. Is this proof system completeΓ Another issue is to investigate exactly how general our approach is, for instance by investigating arbitrary process calculi specified in one of the various formats for structured operational semantics (c.f. [5, 12]). Related work in this direction was recently reported by Simpson [21] for the case of Hennessy-Milner logic.

Acknowledgements. Thanks are due to Roberto Amadio for numerous discussions on related topics.

References

[1] R. Amadio and M. Dam. Reasoning about higher-order processes. SICS Research report RR:94–18, 1994. To appear in Proc. CAAP'95.

[2] H. Andersen, C. Stirling, and G. Winskel. A compositional proof system for the modal μ-calculus. In Proc. LICS'94, 1994.

[3] H. Andersen and G. Winskel. Compositional checking of satisfaction. *Formal methods in System Design*, 1(4), 1992.

[4] J. Armstrong, R. Virding, and M. Williams. *Concurrent Programming in Erlang*. Prentice-Hall International (UK) Ltd., 1993.

[5] B. Bloom, S. Istrail, and A. R. Meyer. Bisimulation can't be traced: Preliminary report. In *Proc. 15th POPL*, pages 229–239, 1988.

[6] J. Bradfield and C. Stirling. Local model checking for infinite state spaces. *Theoretical Computer Science*, 96:157–174, 1992.

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

[8] R. Cleaveland, M. Dreimüller, and B. Steffen. Faster model checking for the modal mu-calculus. In Proc. CAV'92, *Lecture Notes in Computer Science*, 663:383–394, 1992.

[9] M. Dam. Model checking mobile processes (full version). SICS report RR 94:1, 1994. Prel. version appeared in Proc. Concur'93, LNCS 715, pp. 22–36.

[10] E. A. Emerson and C. Lei. Efficient model checking in fragments of the propositional mu-calculus. In Proc. LICS'86, pages 267–278, 1986.

[11] J. Esparza. Decidability of model checking for infinite-state concurrent systems. Manuscript, 1995.

[12] J. F. Groote and F. W. Vaandrager. Structured operational semantics and bisimulation as a congruence. *Information and Computation*, **100**:202–260, 1992.

[13] R. Harper and R. Pollack. Type checking with universes. *Theoretical Computer Science*, **89**:107–136, 1991.

[14] H. Hungar. Local model checking for parallel compositions of context-free processes. In Proc. CONCUR'94, *Lecture Notes in Computer Science*, 836:114–128, 1994.

[15] K. G. Larsen. Efficient local correctness checking. In Proc. CAV'92, *Lecture Notes in Computer Science*, 663, 1992.

[16] K. G. Larsen and L. Xinxin. Compositionality through an operational semantics of contexts. *Journal of Logic and Computation*, 1:761–795, 1991.

[17] R. Milner. *Communication and Concurrency*. Prentice Hall International, 1989.

[18] R. Milner, J. Parrow, and D. Walker. A calculus of mobile processes, I and II. *Information and Computation*, 100(1):1–40 and 41–77, 1992.

[19] B. C. Pierce, D. Remy, and D. N. Turner. Pict: An experiment in concurrent language design. Manuscript, available from ftp.dcs.ed.ac.uk/pub/bcp/pict.tar.Z, 1994.

[20] J. H. Reppy. CML: A higher-order concurrent language. In Proc. ACM SIGPLAN'91, 1991.

[21] A. Simpson. Compositionality via cut-elimination: Hennessy-Milner logic for an arbitrary GSOS. Manuscript. To appear in Proc. LICS'95, 1995.

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

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

[24] B. Thomsen, L. Leth, S. Prasad, T.-M. Kuo, A. Kramer, F. Knabe, and A. Giacalone. Facile antigua release programming guide. Tech. rep. ECRC–93–20, 1993.