Compositional Verification of CCS Processes

Mads Dam and Dilian Gurov

Dept. of Teleinformatics, Royal Institute of Technology (KTH), Stockholm, and SICS

Abstract. We present a proof system for verifying CCS processes in the modal μ-calculus. Its novelty lies in the generality of the proof judgements allowing parametric, and in this way, through a rule for subterm cut, also compositional reasoning, in the complex setting of a logic with recursion. Another advantage of the proof system is complete separation of the rules of the proof system concerning the logic from the rules encoding the opartional semantics of CCS, which makes the proof system easily adaptable to other languages with a clean transitional semantics.

1 Introduction

In several recent papers [1, 2, 4, 5, 7] proof-theoretical frameworks for compositional verification have been put forward based on Gentzen-style sequents of the shape $\Gamma \vdash \Delta$, where the components of Γ and Δ are correctness assertions $P : \phi$. Several programming or modelling languages have been considered, including CCS [4], the π-calculus [2], CHOCS [1], general GSOS-definable languages [7], and even a significant core fragment of the [real] programming language Erlang [5]. An important precursor to the above papers is [8] which used ternary sequents to build compositional proof systems for CCS and SCCS vs. Hennessy-Milner logic [6].

The key idea behind using this general form of sequents is that it allows correctness properties $P : \phi$ to be stated and proved in a *parametric* fashion, i.e., relative to correctness properties of constituents of P, represented by free variables (parameters). A general rule of subterm cut

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

allows such subterm assumptions to be introduced and used for compositional verification.

It is, however, difficult to find a way of supporting temporal properties within such a framework, especially when expressed in a logic like the modal μ-calculus. In [4] the first author showed one way of doing this, and built, for the first time, a compositional proof system capable of handling general CCS terms, including those that create new processes dynamically. In [5] we used a similar, though considerably improved, approach to address Erlang.

In this paper we improve upon previous approaches in two ways. First, following an idea by Simpson [7] we fully separate the issue of incorporating the

transitional semantics for P from the general handling of the logic by employing process variables and transition assertions of the shape $P \stackrel{\alpha}{\rightarrow} Q$. These assertions provide a semantically explicit bridge between the transitions of P and the one-step modalities of the logic. Second, to handle the fixed point formulas of the logic in a simple and yet transparent (semantically explicit?) manner we employ fixed point approximations using ordinal variables, and ordinal constraints of the shape $\kappa_1 < \kappa_2$. This allows the unfoldings of fixed point formulas in different places of a sequent to be related in a manner which reduces proofs to well-founded (ordinal?) induction arguments. The latter take here the form of global discharge rules.

The paper is organised as follows. ...

2 Logic

Formulas ϕ are generated by the following grammar, where κ ranges over a set of *ordinal variables*, α over a set of *actions*, and X over a set of *propositional variables*.

$$\phi ::= \phi \vee \phi \mid \neg\phi \mid \langle\alpha\rangle\phi \mid X \mid \mu X.\phi \mid (\mu X.\phi)^{\kappa}$$

We assume that the sets of actions, ordinal variables, and propositional variables are countably infinite and mutually disjoint. An occurrence of a subformula ψ in ϕ is *positive*, if ψ appears in the scope of an even number of negation symbols. Otherwise the occurrence is negative. The formation of least fixed point formulas of one of the shapes $\mu X.\phi$ or $(\mu X.\phi)^{\kappa}$ is subject to the usual formal monotonicity condition that occurrences of X in ϕ are positive. We use the symbols U and V to range over (unindexed) fixed point formulas $\mu X.\phi$.

Definition 1.

1. *The formula ϕ is* propositionally closed *if ϕ does not have free ocurrences of propositional variables.*
2. *The formula ϕ is* pure *if ϕ does not have subformulas of the form U^{κ}.*

Observe that standard abbreviations apply, such as

$$\begin{aligned} \mathit{false} &= \mu X.X, \\ \phi \wedge \psi &= \neg(\neg\phi \vee \neg\psi), \\ [\alpha]\phi &= \neg\langle\alpha\rangle\neg\phi, \\ \nu X.\phi &= \neg\mu X.\neg(\phi[\neg X/X]). \end{aligned}$$

The semantics is determined in the usual fashion, with indexed formulas receiving the expected semantics as ordinal approximations. So, we let ρ be an interpretation function (environment), mapping ordinal variables to ordinals, and propositional variables to sets of states:

$$\begin{aligned} \|\phi \vee \psi\|\rho &= \|\phi\|\rho \cup \|\psi\|\rho \\ \|\neg\phi\|\rho &= \mathcal{S}\backslash\|\phi\|\rho \end{aligned}$$

$$\|\langle\alpha\rangle\phi\|\rho = \{P \mid \exists Q \in \|\phi\|\rho . P \xrightarrow{\alpha} Q\}$$
$$\|X\|\rho = \rho(X)$$
$$\|\mu X.\phi\|\rho = \bigcap\{S \mid S \subseteq \|\phi\|\rho[S/X]\}$$
$$\|(\mu X.\phi)^\kappa\|\rho = \begin{cases} \emptyset & \text{if } \rho(\kappa) = 0 \\ \|\phi\|\rho[\|(\mu X.\phi)^\kappa\|\rho/X, \beta/\kappa] & \text{if } \rho(\kappa) = \beta + 1 \\ \bigcup\{\|(\mu X.\phi)^\kappa\|\rho[\beta/\kappa] \mid \beta < \rho(\kappa)\} & \text{if } \rho(\kappa) \text{ is a limit ordinal} \end{cases}$$

The following easy consequence of the definition is used in the proof system to follow:

Theorem 1 (KnasterTarski).

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

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

Observe how this casts the properties U and U^κ as existential properties: This is useful to motivate the proof rules for fixed point formulas given below.

As models are countable, quantification over countable ordinals in Theorem 1 suffices.

Definition 2 (Assertions, Judgements).

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

An assertion of the shape $E : A$ is called a *property assertion*, an assertion of the shape $\kappa < \kappa'$ is called an *ordinal constraint*, and an assertion of the shape $E \xrightarrow{\alpha} F$ is called a *transition assertion.*

3 Proof System: Logical Entailment

We consider the general problem of proving validity of sequents. As a first stop we consider the subproblem of logical entailment, casting this as the problem of proving validity of sequents $\Gamma \vdash \Delta$ where all process terms are variables.

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

$$\textsc{Ax}\ \frac{\cdot}{\Gamma, A \vdash A, \Delta}$$

$$\textsc{Cut}\ \frac{\Gamma \vdash A, \Delta \quad \Gamma, A \vdash \Delta}{\Gamma \vdash \Delta}$$

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

In fact (as in [7]) in the axiom rule A needs only be instantiated to transition assertions, and then Δ can be assumed to be empty. Since Γ and Δ are sets, structural rules like permutation and contraction are vacuous. We conjecture that both cut and the weakening rules are admissible.

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

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

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

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

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

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

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

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

The side condition $\mathit{fresh}(x)$ ($\mathit{fresh}(\kappa)$) is intended to mean that x (κ) does not appear freely in the conclusion of the rule.

The rules for indexed fixed point formulas are directly motivated by the KnasterTarski theorem. Similarly, the rules for unindexed fixed point formulas are directly motivated by the Knaster-Tarski Theorem. The lack of symmetry between the latter two rules is not accidental; their symmetric counterparts are in fact admissable.

Ordinal Constraints Finally, we need to provide rules for reasoning about ordinal constraints. The following ordinal transitivity rule is sufficient:

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

provided ordinal variables and constraints are only being introduced during the proof, but do not appear in the root sequent.

4 Well-founded (Ordinal?) Induction

Processes and formulas can be recursive, allowing for proof trees to grow unboundedly. Intuitively, one would like to terminate an open branch whenever a sequent has been reached which is an instance, up to some substitution σ, of some of its ancestor nodes. A proof structure, all leaf nodes of which are either axioms or such repeats, serves as the basis for well-founded ordinal induction arguments. A *global discharge condition* is a sufficient condition for such an argument to go through. The use of ordinal variables and constraints allows such conditions to be phrased in a clear and semantically transparent way. The most general view of discharge is presented in game-theoretic terms elsewhere [**?**]. In essence, global discharge guarantees, that if one "unfolds" a proof structure, then for every infinite branch there is an ordinal variable for which the "depth" of the constraints being accumulated along this branch grows unboundedly [can one phrase this in a nice way?]. Here we present a discharge condition which is easy to understand and apply, and sufficiently powerful to handle the Example.

Two repeat nodes are called *related* if there is a path connecting these in the graph obtained by identifying these nodes with the respective nodes of which they are instances.

Definition 3. *A node labelled $\Gamma \vdash \Delta$ can be* discharged *with U^κ and substitution σ against an ancestor node labelled $\Gamma' \vdash \Delta'$ if:*

(i) U^κ occurs as subformula in Γ' or Δ';

(ii) $\phi\sigma \in \Gamma$ whenever $\phi \in \Gamma'$, and $\phi\sigma \in \Delta$ whenever $\phi \in \Delta'$;

(iii) $\Gamma \vdash \kappa\sigma < \kappa$ is derivable;

(iv) assuming the related discharge nodes labelled $\Gamma_1 \vdash \Delta_1 \ldots \Gamma_n \vdash \Delta_n$ have been discharged with $U_1^{\kappa_1} \ldots U_n^{\kappa_n}$ and $\sigma_1 \ldots \sigma_n$, there is a linear ordering $\prec$ on these nodes including the present node, such that whenever $i \prec j$: (a) $U_i^{\kappa_i}$ occurs as subformula in Γ_j' or Δ_j', and (b) either $\kappa_i\sigma_j = \kappa_i$, or $\Gamma_j \vdash \kappa_i\sigma_j < \kappa_i$ is derivable.

The full version of the paper will explain the discharge mehanism in greater detail.

5 Proof System: Operational Semantics

Calculus of Communicating Systems CCS processes E are generated by the following grammar, where l ranges over a given set of *labels*, L over subsets of

this set of labels, α over *actions* of the shape τ, l or $\overline{l}$, and x over a set of *process variables.*

$$E ::= 0 \mid \alpha.E \mid E + E \mid E|E \mid E\backslash L \mid \mathit{fix}\, x.E$$

Decomposing Properties

$$\textsc{SubtermCut}\text{-R}\ \frac{\Gamma \vdash F : \psi, \Delta \quad \Gamma, x : \psi \vdash E : \phi, \Delta}{\Gamma \vdash E[F/x] : \phi, \Delta}\ \mathit{fresh}(x)$$

The symmetric rule SUBTERMCUT-L is derivable.

Embedding the Operational Semantics

$$0\text{-L}\ \frac{\cdot}{\Gamma, 0 \xrightarrow{\alpha} x \vdash \Delta}$$

$$\alpha\text{-L-1}\ \frac{\Gamma[E/x] \vdash \Delta[E/x]}{\Gamma, \alpha.E \xrightarrow{\alpha} x \vdash \Delta} \qquad \alpha\text{-R}\ \frac{\cdot}{\Gamma \vdash \alpha.E \xrightarrow{\alpha} E, \Delta}$$

$$\alpha\text{-L-2}\ \frac{\cdot}{\Gamma, \alpha.E \xrightarrow{\beta} x \vdash \Delta}\ \alpha \neq \beta$$

$$+\text{-L}\ \frac{\Gamma[y/x], E \xrightarrow{\alpha} y \vdash \Delta[y/x] \quad \Gamma[z/x], F \xrightarrow{\alpha} z \vdash \Delta[z/x]}{\Gamma, E + F \xrightarrow{\alpha} x \vdash \Delta}$$

$$+\text{-R}\ \frac{\Gamma \vdash E \xrightarrow{\alpha} E', \Delta}{\Gamma \vdash E + F \xrightarrow{\alpha} E', \Delta}$$

$$|\text{-R-1}\ \frac{\Gamma \vdash E \xrightarrow{\alpha} E', \Delta}{\Gamma \vdash E|F \xrightarrow{\alpha} E'|F, \Delta}$$

$$|\text{-R-2}\ \frac{\Gamma \vdash E \xrightarrow{l} E' \quad \Gamma \vdash F \xrightarrow{\overline{l}} F', \Delta}{\Gamma \vdash E|F \xrightarrow{\tau} E'|F', \Delta}$$

$$|\text{-L-1}\ \frac{\Gamma[y|F/x], E \xrightarrow{l} y \vdash \Delta[y|F/x] \quad \Gamma[E|z/x], F \xrightarrow{l} z \vdash \Delta[E|z/x]}{\Gamma, E|F \xrightarrow{l} x \vdash \Delta}$$

$$|\text{-L-2}\ \frac{\begin{array}{c}\Gamma[y_1|F/x], E \xrightarrow{\tau} y_1 \vdash \Delta[y_1|F/x] \\ \Gamma[E|y_2/x], F \xrightarrow{\tau} y_2 \vdash \Delta[E|y_2/x] \\ \Gamma[z_1|z_2/x], l_1 = \overline{l_2}, E \xrightarrow{l_1} z_1, F \xrightarrow{l_2} z_2 \vdash \Delta[z_1|z_2/x]\end{array}}{\Gamma, E|F \xrightarrow{\tau} x \vdash \Delta}$$

$$\backslash\text{-L}\ \frac{\Gamma[y\backslash L/x], E \xrightarrow{\alpha} y \vdash \Delta[y\backslash L/x]}{\Gamma, E\backslash L \vdash \Delta}\ \alpha \notin L$$

$$\backslash\text{-R}\ \frac{\Gamma \vdash E \xrightarrow{\alpha} E', \Delta \quad \Gamma \vdash \alpha \notin L, \Delta}{\Gamma \vdash E\backslash L \xrightarrow{\alpha} E'\backslash L, \Delta}$$

$$\mathit{fix}\text{-L}\ \frac{\Gamma[y/x], E[\mathit{fix}\, x.E/x] \xrightarrow{\alpha} y \vdash \Delta[y/x]}{\Gamma, \mathit{fix}\, x.E \xrightarrow{\alpha} x \vdash \Delta}$$

$$\mathit{fix}\text{-R}\ \frac{\Gamma \vdash E[\mathit{fix}\, x.E/x] \xrightarrow{\alpha} E', \Delta}{\Gamma \vdash \mathit{fix}\, x.E \xrightarrow{\alpha} E', \Delta}$$

Symmetric versions of +-R and |-R-1.

6 Example

Consider a process

$$Counter = fix\, x.\, up.\, (x \,|\, down.x)$$

which can alternatingly engage in *up* and *down* actions, generating a new copy of itself after each *up* action. Clearly, in any point in time, regardless how many counters have already been generated, this system can engage in finite sequences of *down* actions only. This propery can be formalised as the negation of the following formula:

$$\phi = \mu X.\, \neg\, \mu Y.\, \neg\, (\langle up \rangle X \vee \langle down \rangle \neg\, Y)$$

So, we want to prove validity of the sequent

$$\vdash Counter : \neg\, \phi$$

We perform the proof backwards, from this goal sequent towards the axioms, guided by the shape of the formulas and process terms involved. After eliminating the negation and approximating ϕ one obtains

$$Counter : \phi^{\kappa} \vdash \tag{1}$$

Continuing in the same straightforward manner we soon arrive at the following two sequents:

$$\kappa' < \kappa, up.\, (Counter \,|\, down.Counter) \xrightarrow{down} x \vdash x : \psi$$

$$\kappa' < \kappa, Counter \,|\, down.Counter : \phi^{\kappa'} \vdash$$

the first of which is an axiom. The second sequent is similar to sequent (1), with the important difference of a new $down.Counter$ component having appeared. This is the point where one would like to perform an inductive argument on the system structure, and this can be done using subterm cut. The most important question is what the property of the component being cut is that yields the

overall system property being veryfied. A convenient case is when it is the same property, i.e., when the property being veryfied composes nicely. This is the case in our example, partly because there is no communication between the components. So, after two applications of subterm cut we obtain the following three sequents:

$$\kappa' < \kappa, Counter : \phi^{\kappa'} \vdash$$
$$\kappa' < \kappa, down.Counter : \phi^{\kappa'} \vdash x : \phi^{\kappa'}$$
$$\kappa' < \kappa, x \,|\, y : \phi^{\kappa'} \vdash x : \phi^{\kappa'}, x : \phi^{\kappa'}$$

the first of which can be directly discharged against (1). The second sequent is easily reduced to an axiom and a discharge node. Handling the remaining sequent is only slightly more involved (it will be considered in more detail in the full version of the paper).

7 Conclusion

We presented a proof system for verifying CCS processes in the modal μ-calculus. Its novelty lies in the generality of the proof judgements allowing parametric, and hence through subterm cut also compositional reasoning, in the complex setting of a logic with recursion. Another advantage of the proof system is complete separation of the rules of the proof system concerning the logic from the rules encoding the opartional semantics of the process language chosen (here CCS), which makes the proof system easily adaptable to other languages with a clean transitional semantics.

References

1. R. Amadio and M. Dam. Reasoning about higher-order processes. In *Proc. CAAP'95,* Lecture Notes in Computer Science, 915:202–217, 1995.
2. R. Amadio and M. Dam. A modal theory of types for the π-calculus. In *Proc. FTRTFT'96,* Lecture Notes in Computer Science, 1135:347–365, 1996.
3. J. Armstrong, R. Virding, C. Wikström, and M. Williams. *Concurrent Programming in Erlang (Second Edition).* Prentice-Hall International (UK) Ltd., 1996.
4. M. Dam. Proving properties of dynamic process networks. *Information and Computation*, 140:95–114, 1998.
5. M. Dam, L.-å. Fredlund, and D. Gurov. Toward parametric verification of open distributed systems. To appear in *Compositionality: the Significant Difference,* H. Langmaack, A. Pnueli and W.-P. de Roever (eds.), Springer-Verlag, 1998.
6. M. Hennessy and R. Milner. Algebraic laws for nondeterminism and concurrency. *Journal of the ACM*, **32**:137–162, 1985.
7. A. Simpson. Compositionality via cut-elimination: Hennessy-Milner logic for an arbitrary GSOS. In *Proceedings, Tenth Annual IEEE Symposium on Logic in Computer Science*, pages 420–430, San Diego, California, 26–29 1995. IEEE Computer Society Press.
8. C. Stirling. Modal logics for communicating systems. *Theoretical Computer Science*, 49:311–347, 1987.
9. C. Stirling and D. Walker. Local model checking in the modal mu-calculus. *Theoretical Computer Science*, 89:161–177, 1991.