On the Reachability Problem in Cryptographic Protocols

Roberto M. Amadio, Denis Lugiez

INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

On the reachability problem in cryptographic protocols

Roberto M. Amadio Denis Lugiez

N° 3915

Mars 2000

THÈME 1

ISSN 0249-6399 ISRN INRIA/RR--3915--FR+ENG

On the reachability problem in cryptographic protocols

Roberto M. Amadio Denis Lugiez

Thème 1 — Réseaux et systèmes
Projet MIMOSA

Rapport de recherche n° 3915 — Mars 2000 — 33 pages

Abstract: We study the verification of secrecy and authenticity properties for cryptographic protocols which rely on symmetric shared keys. The verification can be reduced to check whether a certain parallel program which models the protocol and the specification can reach an erroneous state while interacting with the environment. Assuming finite principals, we present a simple decision procedure for the reachability problem which is based on a 'symbolic' reduction system.

Key-words: Cryptographic protocols, verification, symbolic computation.

The authors work at *Centre de Mathémathiques et d'Informatique* (LIM-CNRS), 39 rue Joliot-Curie, F-13453, Marseille, France. e-mail: {amadio,lugiez}@cmi.univ-mrs.fr. The first author is a member of *Action MIMOSA* and is partially supported by the working group CONFER and the RNRT project MARVEL.

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles, B.P. 93, 06902 Sophia Antipolis Cedex (France)
Téléphone : 04 92 38 77 77 - International : +33 4 92 38 77 77 — Fax : 04 92 38 77 65 - International : +33 4 92 38 77 65

Sur le problème d'accessibilité dans les protocoles cryptographiques

Résumé : Nous étudions la vérification de propriétés de sécurité et d'authenticité dans les protocoles cryptographiques à clef symétrique. La vérification consiste à s'assurer que certains programmes parallèles qui modélisent le protocole et la spécification ne peuvent pas accéder à un état erroné tout en interagissant avec l'environnement. En supposant les principaux finis, nous présentons une simple procédure de décision pour le problème d'accessibilité qui est basée sur un système de réduction symbolique.

Mots-clés : Protocoles cryptographiques, vérification, calcul symbolique.

1 Introduction

Cryptographic protocols seem good candidates for formal verification and several frameworks have been proposed for making possible formal and automatable analyses. In these approaches a 'perfect' encryption scheme is assumed: encryption is an injective function and the only way to decrypt an encrypted message is to know the key with which it was encrypted.

One class of approaches involves state exploration using model-checking techniques [Low96, CJM98, MMS97]. Lowe [Low96], Schneider [Sch96] and several others have used CSP to specify authentication protocols, analysing them with the FDR model-checking tool. Other state exploration approaches are based on logic programming techniques [Mea94]. The main benefit of these approaches is their automation and efficacy in uncovering subtle bugs in protocols (*e.g.,* Lowe's 'man-in-the-middle' attack on the Needham-Schroeder symmetric key protocol). Usually, these methods make simplifying hypotheses on the behaviour of the environment which are used to bound the state space and thus allow for the application of traditional *finite state* model-checking techniques.

A second class of approaches relies on general-purpose proof assistant tools. Paulson [Pau97] uses induction on traces to formally prove protocol correctness using Isabelle. Bolignano [Bol96] uses a state-based analysis of the protocols, proving invariant properties, with the proofs subsequently mechanized in Coq. Although these approaches are not automatic, recent work (see, *e.g.*, [Wei99]) suggests that certain authentication protocols can be modelled in decidable fragments of first-order logic.

A more recent trend has been the use of name-passing process calculi for studying cryptographic authentication protocols. Abadi and Gordon have presented the *spi*-calculus [AG97], an extension of the π-calculus with cryptographic primitives. Principals of a protocol are expressed in a π-calculus-like notation, whereas the attacker is represented implicitly by the process calculus notion of 'environment'. Security properties are modelled in terms of contextual equivalences, in contrast to the previous approaches.

In this paper, we follow the classical approach of Dolev and Yao [DY83] where all communications are mediated by an hostile environment and we formulate the verification task as a reachability problem. This is the problem of determining if a certain (finite) parallel program which models the protocol and the specification can reach an erroneous state while interacting with the environment.

In previous work [AP99], we have already observed that several secrecy and authentication properties can be expressed in this framework and that the reachability problem is decidable assuming finite principals and bounds on the sorts of the messages synthesized by the environment.

That work was formulated in a name-passing formalism akin to the *spi*-calculus. We found it difficult to obtain stronger decidability results in that framework and for this reason we move in this paper to a 'first-order' formalization where 'messages' are modelled as ground terms of a given first-order signature. Standard tools for symbolic computation such as syntactic unification and tree automata are then available.

We should point out that this approach is not new. For instance, it is the one followed by Huima in [Hui99] where a decidability result for the the reachability problem is claimed (we will mention in section 2 some technical differences). The proof of this result is actually presented in Huima's master's thesis, it runs for about 45 pages and it is quite involved. Our failure to fully understand this result motivated our quest for a simpler approach.

The contribution of this paper is to present a direct, self-contained, and hopefully illuminating proof of decidability for the reachability problem for finite principals (without assuming bounds on the sorts of the messages synthesized by the environment). On this topic, there is an impressive amount of work in progress which is carried on at academic and industrial research institutions. Here, we should mention at least the one by Boreale [Bor00] who pursues similar goals and with whom the first author has had several interesting discussions at the early stages of our research.

2 Model

In this section, we introduce our model, we explain how specifications are expressed, and we state the reachability problem. We consider terms over an infinite signature:

$$\Sigma = \{C_n^0\}_{n\in\omega} \cup \{E^2, \langle\_,\_\rangle^2\} \ .$$

Thus we have an infinite set of constants and two binary functions E for encoding and $\langle\_,\_\rangle$ for pairing. We set $\Sigma(n) = \{C_0,\ldots,C_n\} \cup \{E^2, \langle\_,\_\rangle^2\}$. We use the following notation: $x, y, \ldots$ for (term) variables; V for a set of variables; $T_\Sigma(V)$ for the collection of finite terms over $\Sigma \cup V$; $t, t', \ldots$ for terms in $T_\Sigma(V)$; $\vec{t}$ for vectors of terms; $[t/x]$ for the substitution of t for x. We denote with $Var(t)$ the variables occurring in the term t.

Names and Messages In our modeling of messages we follow [Pau99, BDNP99]. Thus we distinguish between basic names (agent's names, nonces, keys,...) and composed messages. The set of names $\mathcal{N}$ is defined as $\{C_n^0\}_{n\in\omega}$ and the set $\mathcal{M}$ is defined as the least set that contains $\mathcal{N}$ and such that:

$$\begin{array}{lll} t \in \mathcal{M} \text{ and } t' \in \mathcal{N} & \Rightarrow & E(t,t') \in \mathcal{M} \\ t, t' \in \mathcal{M} & \Rightarrow & \langle t, t'\rangle \in \mathcal{M} \ . \end{array}$$

In a similar way, we define the set $\mathcal{N}(m)$ as the initial sequence $\{C_n^0\}_{n\leq m}$ and the set $\mathcal{M}(m)$ as the least set that contains $\mathcal{N}(m)$ and such that (i) $E(t,t') \in \mathcal{M}(m)$ if $t \in \mathcal{M}(m)$ and $t' \in \mathcal{N}(m)$, and (ii) $\langle t, t'\rangle \in \mathcal{M}(m)$ if $t, t' \in \mathcal{M}(m)$.

Processes Processes are defined as follows:

$$\begin{array}{rl} P ::= & 0 \mid err \mid !t.P \parallel ?x.P \parallel \nu x\, P \parallel P \mid P' \mid \mathsf{let}\ x = t\ \mathsf{in}\ P \mid \\ & \mathsf{case}\ E(x,t) = t'\ \mathsf{in}\ P \parallel \mathsf{case}\ \langle x, y\rangle = t\ \mathsf{in}\ P \mid [t = t']P, P' \ . \end{array}$$

(!)	$(!t.P \mid P', n, T)$	$\rightarrow (P \mid P', n, T \cup \{t\})$ if $t \in \mathcal{M}$
(?)	$(?x.P \mid P', n, T)$	$\rightarrow ([t/x]P \mid P', n, T)$ if $t \in S(A(T))$
(ν)	$(\nu x\, P \mid P', n, T)$	$\rightarrow ([C_{n+1}/x]P \mid P', n+1, T)$
(l)	$(\mathsf{let}\ x = t\ \mathsf{in}\ P \mid P', n, T)$	$\rightarrow ([t/x]P \mid P', n, T)$ if $t \in \mathcal{M}$
(c_1)	$(\mathsf{case}\ E(x,t') = E(t,t')\ \mathsf{in}\ P \mid P', n, T)$	$\rightarrow ([t/x]P \mid P', n, T)$ if $t' \in \mathcal{N}, t \in \mathcal{M}$
(c_2)	$(\mathsf{case}\ \langle x,y\rangle = \langle t,t'\rangle\ \mathsf{in}\ P \mid P', n, T)$	$\rightarrow ([t/x,t'/y]P \mid P', n, T)$ if $t,t' \in \mathcal{M}$
(m_1)	$([t=t]P_1, P_2 \mid P', n, T)$	$\rightarrow (P_1 \mid P', n, T)$ if $t \in \mathcal{M}$
(m_2)	$([t=t']P_1, P_2 \mid P', n, T)$	$\rightarrow (P_2 \mid P', n, T)$ if $t \neq t', t, t' \in \mathcal{M}$

Figure 1: Reduction on configurations

Processes, sometimes called principals, describe the behaviour of the agents participating to the protocol. The informal interpretation is the following: 0 is the process which is terminated in a sound state; *err* is the process which is terminated in an erroneous state; $!t.P$ evaluates t and if t is a message, sends it to the environment and becomes P (otherwise it terminates); $?x.P$ receives a message t from the environment and becomes $[t/x]P$; $\nu x\, P$ creates a fresh name C and becomes $[C/x]P$; $P \mid P'$ is the asynchronous parallel composition of P and P'; $\mathsf{let}\ x = t\ \mathsf{in}\ P$ evaluates t and if t is a message becomes $[t/x]P$ (otherwise it terminates); $\mathsf{case}\ E(x,t) = t'\ \mathsf{in}\ P$ evaluates t' and if t' is a message of the shape $E(t'',t)$ it becomes $[t''/x]P$ (otherwise it terminates); $\mathsf{case}\ \langle x,y\rangle = t\ \mathsf{in}\ P$ evaluates t and if t is a message of the shape $\langle t',t''\rangle$ it becomes $[t'/x,t''/y]P$, (otherwise it terminates); $[t=t']P,P'$ evaluates t and t' and if both are messages then it becomes P if $t \equiv t'$ and P' if $t \neq t'$ (otherwise it terminates). We denote with $FV(P)$ the set of variables occurring free in P.

Configuration Let P be a process and T be a set of possibly open terms. We define $cnst(P)$ and $cnst(T)$ as the set of names that occur in P and T, respectively. A *well-formed configuration* k is a triple (P,n,T) where (i) P is a closed process, (ii) $n \in \omega$, (iii) T is a non-empty finite subset of $\mathcal{M}$, and (iv) $cnst(P) \cup cnst(T) \subseteq \mathcal{N}(n)$. P represents the principals' behaviour, n is a counter used to generate fresh names, and T stands for the knowledge of the environment. We assume T non-empty to avoid the paradoxical situation where an input cannot be fired because the knowledge of the environment is empty.

Reduction In figure 1, we define a reduction relation on well-formed configurations. In these rules, we always reduce the leftmost process with the proviso that parallel composition is associative and commutative. Moreover, we take the liberty of writing P as $P \mid 0$ whenever needed to apply a rewriting rule.

In the operational semantics, we suppose that a thread is stuck whenever it tries (i) to evaluate a term which is not a message or (ii) to decrypt with a name a message which is not encrypted with the very same name, or (iii) to project a message which is not a pair. We write $k \rightarrow_R k'$ if the configuration k reduces to the configuration k' by applying the rule R.

The functions S (synthesis) and A (analysis) are closure operators over the powerset of closed terms defined as follows (similar operators have already been considered in the literature, see, *e.g.*, [Pau99]).

Synthesis: $S(T)$ is the least set that contains T and such that

$$\begin{array}{ll} t_1, t_2 \in S(T) & \Rightarrow \langle t_1, t_2\rangle \in S(T) \\ t_1 \in S(T), t_2 \in T \cap \mathcal{N} & \Rightarrow E(t_1, t_2) \in S(T)\ . \end{array}$$

Analysis: $A(T)$ is the least set that contains T and such that

$$\begin{array}{ll} \langle t_1, t_2\rangle \in A(T) & \Rightarrow t_i \in A(T)\ i = 1, 2 \\ E(t_1, t_2) \in A(T), t_2 \in A(T) & \Rightarrow t_1 \in A(T)\ . \end{array}$$

We remark that well-formed configurations are closed under reduction. We also note that in our modelling, processes can only send messages (not arbitrary terms) to the environment and vice versa the environment can only send messages to the processes. The following properties follow from the inductive definition of the operators S and A.

Proposition 2.1 (1) *If $T \subseteq \mathcal{M}$ then $S(T), A(T) \subseteq \mathcal{M}$.*

(2) *If $T \subseteq \mathcal{M}(m)$ then $S(T), A(T) \subseteq \mathcal{M}(m)$ for any $m \geq 0$.*

(3) *Define a function G as*

$$G(T) = T \cup \{t_1, t_2 \mid \langle t_1, t_2\rangle \in T\} \cup \{t_1 \mid E(t_1, t_2), t_2 \in T\}$$

Then $A(T) = \bigcup_{n\in\omega} G^n(T)$.

Related modelling We point out that Monniaux [Mon99] and Huima [Hui99] consider a signature with constructors such as encryption and pairing and *destructors* such as decryption and projection. In their approach, terms are considered up to the equality induced by a canonical term rewriting system. In our approach, the decryption and projection functions are handled implicitly: the principals can decrypt and project using the case operators and the environment can decrypt and project according to the definition of the analysis operator A. In this way, we can work directly with the free algebra (as in [Bor00]).

A second difference, concerns the use of messages –not just names– as encryption keys. It remains to be seen whether our approach can deal with this more general framework.

Finally, we mention that Monniaux advocates the use of tree automata to represent the set of messages that can be synthesized by the environment and to abstract the possible values that can be taken by, say, an input variable. This approach can be followed in our framework too. In particular, we point out the following property where we say that a language of ground terms is *recognizable* if there is a tree automaton that accepts it.

Proposition 2.2 *If $T \subseteq \mathcal{M}$ is recognizable over the signature $\Sigma(n)$ then the sets $S(T)$, $G(T)$ and $A(T) = \bigcup_{n\in\omega} G^n(T)$ are recognizable over the same signature.*

PROOF. We consider bottom up tree automata following the notation in [CDG$^+$, Chpt. 1]. We suppose that T is accepted by the tree automaton $A = (Q, F, \Delta)$ (Q set of states, F set of accepting states, Δ set of rewriting rules). Moreover we assume that A is reduced (all states are accessible), deterministic, and that there are states q_{C_i} such that $t \xrightarrow{*} q_{C_i}$ iff $t \equiv C_i$, for $i = 0, \ldots, n$. If A is a tree automaton then we denote with $\mathcal{L}(A)$ the language it recognizes.

$S(T)$. To recognize $S(T)$, we define $A_S = (Q \cup \{q_S\}, \{q_S\}, \Delta_S)$ where:

$$\Delta_S = \Delta \cup \{q \to q_S \mid q \in F\} \cup \{\langle q_S, q_S\rangle \to q_S\} \cup \{E(q_S, q_{C_i}) \to q_S \mid q_{C_i} \in F\} \ .$$

We show that $\mathcal{L}(A_S) = S(T)$.

$S(T) \subseteq \mathcal{L}(A_S)$. It is enought to observe that $\mathcal{L}(A_S)$ contains T and it is closed under pairing and encryption.

- If $t \in T$ then $t \to^*_\Delta q \to_{\Delta_S} q_s$. Thus $t \in \mathcal{L}(A_S)$.
- If $t_1, t_2 \in \mathcal{L}(A_S)$ then $t_i \to^*_{\Delta_S} q_S$ for $i = 1, 2$. Then

$$\langle t_1, t_2\rangle \to^*_{\Delta_S} \langle q_S, q_S\rangle \to_{\Delta_S} q_S \ .$$

Thus $\langle t_1, t_2\rangle \in \mathcal{L}(A_S)$.

- If $t \in \mathcal{L}(A_S)$ and $C \in T$ then $t \to^*_{\Delta_S} q_S F$ and $C \to_\Delta q_C \in F$. Then

$$E(t, C) \to^*_{\Delta_S} E(q_S, q_C) \to_{\Delta_S} q_S \ .$$

Thus $E(t, C) \in \mathcal{L}(A_S)$.

$\mathcal{L}(A_S) \subseteq S(T)$. Suppose $t \to^*_{\Delta_S} q_S$. We proceed by induction on the length of the reduction and consider the last rule being applied.

- $t \to^*_{\Delta_S} q \to_{\Delta_S} q_S$ with $q \in F$. Then $t \in T \subseteq S(T)$.
- $t \to^*_{\Delta_S} \langle q_S, q_S\rangle \to_{\Delta_S} q_S$. Then $t \equiv \langle t_1, t_2\rangle$ and $t_i \to^*_{\Delta_S} q_S$ for $i = 1, 2$ in less steps. By inductive hypothesis, $t_i \in S(T)$ and by definition of synthesis, $\langle t_1, t_2\rangle \in S(T)$.
- $t \to^*_{\Delta_S} E(q_S, q_C) \to_{\Delta_S} q_S$ with $q_C \in F$. Then $t \equiv E(t', C)$ and $t' \to^*_{\Delta_S} q_S$ in less steps. By inductive hypothesis, $t' \in S(T)$ and by definition of synthesis $E(t', C) \in S(T)$.

$G(T)$. To recognize $G(T)$, we define $A_G = (Q, F_G, \Delta)$ as:

$$F_G = F \cup \{q_1, q_2 \mid \langle q_1, q_2\rangle \to q \text{ and } q \in F\} \cup \{q_1 \mid E(q_1, q_{C_i}) \to q \text{ and } q_{C_i}, q \in F\} \ .$$

We show that $\mathcal{L}(A_G) = G(T)$ using the hypothesis that A is deterministic and reduced.

$G(T) \subseteq \mathcal{L}(A_G)$. We consider three cases following the definition of $G(T)$.

- $t \in T$ then $t \to^*_\Delta q \in F \subseteq F_G$.

• Suppose $\langle t_1, t_2\rangle \in T$. Then $\langle t_1, t_2\rangle \rightarrow_\Delta^* q \in F$. Since A is deterministic we have that $\langle t_1, t_2\rangle \rightarrow_\Delta^* \langle q_1, q_2\rangle \rightarrow_\Delta q$ and $t_i \rightarrow_\Delta^* q_i$ for $i = 1, 2$. Thus $q_i \in F_G$ and $t_i \in \mathcal{L}(A_G)$ for $i = 1, 2$.

• Suppose $E(t', C) \in T$ and $C \in T$. Then $E(t', C) \rightarrow_\Delta^* q \in F$. Since A is deterministic we have that $E(t', C) \rightarrow_\Delta^* E(q', q_C) \rightarrow_\Delta q \in F$ and $t' \rightarrow_\Delta^* q'$. Thus $q' \in F_G$ and $t' \in \mathcal{L}(A_G)$.

$\mathcal{L}(A_G) \subseteq G(T)$. Suppose $t \rightarrow_\Delta^* q \in F_G$. Following the definition of F_G, we consider three cases.

• If $q \in F$ then $t \in T \subseteq G(T)$.

• If for some q', q'', $\langle q, q'\rangle \rightarrow_\Delta q''$ and $q'' \in F$. Since A is reduced, there is t' such that $t' \rightarrow_\Delta^* q'$. Thus

$$\langle t, t'\rangle \rightarrow_\Delta^* \langle q, q'\rangle \rightarrow_\Delta q'' \in F,$$

$\langle t, t'\rangle \in T$, and $t \in G(T)$. We omit the symmetric case.

• If for some $q', q_C \in F$, $E(q, q_C) \rightarrow_\Delta q'$ then $E(t, C) \rightarrow_\Delta^* q'$, $E(t, C) \in T$, $C \in T$, and $t \in G(T)$.

$A(T)$. Finally, we remark that the construction of A_G only adds final states. Therefore it converges to a fixpoint in a number of iterations which is bound by the cardinality of the set of states, thus providing an automaton recognizing $A(T)$. $\diamond$

We turn next to the definition of the reachability problem.

Definition 2.3 *Let $k \equiv (P, n, T)$ be a configuration. We write $k \downarrow err$ if $P \equiv err \mid P'$ (up to associativity, commutativity, and $P \mid 0 \equiv P'$). We also write $k \downarrow_* err$ if $k \rightarrow^* k'$ and $k \downarrow err$. We then say that k can reach error.*

In this paper, the *reachability problem* is the problem of determining whether a configuration can reach error. The method to specify a particular property is to program an *observer* process that will reach error exactly when the property is violated. For instance, suppose we want to specify that in $\nu x\, P$ the name x will remain secret. Upon creating the name x, we spawn an observer process that challenges the environment to send him the name x. If the environment succeeds, the observer ends in an erroneous state. Thus we compile the process $\nu x\, P$ as $\nu x\, (?y.[x = y]err, 0 \mid P)$. Similar techniques can be applied to the specification of authentication properties. To check that a message received by a principal A is the message previously sent by a principal B, we introduce an observer O such that: (i) before sending the message, B makes sure the message is registered with O, (ii) upon receiving a supposedly authentic message, A queries O for a certification, (iii) O reaches error if it receives a certification request which does not correspond to a previously registered message. We refer to [AP99] for the programming of this little protocol.

Definition 2.4 *We say that two well-formed configurations $k_i \equiv (P_i, n_i, T_i)$, $i = 1, 2$ are* equivalent*, and we write $k_1 \cong k_2$ if there is a bijection $\sigma : cnst(P_1) \cup cnst(T_1) \rightarrow cnst(P_2) \cup cnst(T_2)$ such that $\sigma P_1 = P_2$ and $\sigma T_1 = T_2$.*

An important remark is that all reductions but input are strongly confluent up to equivalence.

Proposition 2.5 (1) *The relation $\cong$ is an equivalence relation.*

(2) *If $k_1 \cong k_2$ then $k_1 \downarrow err$ iff $k_2 \downarrow err$.*

(3) *If $k_1 \cong k_2$ and $k_1 \rightarrow_R k_1'$ then for some k_2', $k_2 \rightarrow_R k_2'$ and $k_1' \cong k_2'$.*

(4) *If $k \rightarrow_{R_1} k_1$ and $k \rightarrow_{R_2} k_2$, where R_1 is not an input rule, then*

$$k_1 \rightarrow_{R_2}^{0,1} k_1',\ k_2 \rightarrow_{R_1}^{0,1} k_2',\ \text{and } k_1' \cong k_2'$$

where $k \rightarrow_R^{0,1} k'$ denotes reduction in 0 or 1 steps.

Proof hint. Properties (1-2) are immediately checked. Property (3) is a direct case analysis. Property (4) is a bit longer but easy to check. Let us mention that when performing two (ν) reductions it may happen that $k_1' \cong k_2'$ but not $k_1' \equiv k_2'$. $\diamond$

3 Basic symbolic reduction

In this section, we introduce a *symbolic* reduction relation, namely a reduction on configurations containing free variables ranging over certain infinite sets of messages. We will show that this reduction is finitely branching, terminating, 'sound', and 'complete'. Our decision procedure for the reachability problem will then amount to explore all the symbolic reductions of a given configuration.

Definition 3.1 *We define the set $\mathcal{M}_V$ of 'open' messages as the least set that contains $\mathcal{N} \cup V$ and such that:*

$$\begin{array}{lcl} t \in \mathcal{M}_V \text{ and } t' \in \mathcal{N} & \Rightarrow & E(t,t') \in \mathcal{M}_V \\ t, t' \in \mathcal{M}_V & \Rightarrow & \langle t, t' \rangle \in \mathcal{M}_V\ . \end{array}$$

We note that (i) if $E(t,t')$ is a subterm of a term in $\mathcal{M}_V$ then $t' \in \mathcal{N}$ and that (ii) if $t \in \mathcal{M}_V$ and σ is a substitution associating variables to elements of $\mathcal{M}_V$ then $\sigma(t) \in \mathcal{M}_V$.

Definition 3.2 *Let $T \subseteq \mathcal{M}_V$ and $K \subseteq_{fin} \mathcal{N}$.*

(1) *Suppose $t \in \mathcal{M}_V$. We say that t' is K-accessible in t iff either $t \equiv t'$ or $t \equiv \langle t_1, t_2 \rangle$ and for some $i \in \{1,2\}$, t' is K-accessible in t_i or $t \equiv E(t_1, C)$, $C \in K$, and t' is K-accessible in t_1.*

(2) *We define $P_K(T)$, the K-accessible parts of T, as the set of terms t' that are K-accessible in a term $t \in T$.*

(3) *We define $I_K(T)$, the K-irreducible parts of T, as*

$$I_K(T) = P_K(T) \cap \{E(t,C) \mid t \in \mathcal{M}_V \text{ and } C \notin K\}\ .$$

(4) *If moreover* $T \subseteq \mathcal{M}$, *we define* $S_K(T)$, *the* K*-synthesis of* T, *as the least set of terms that contains* $T \cup K$ *and is closed under pairing and encryption by a name in* K.

(5) *We define* $\mathcal{T}_K$ *as* $\mathcal{T}_K = \{t \in \mathcal{M} \mid A(\{t\}) \cap \mathcal{N} = K\}$. *These are the terms from which exactly the set of names* K *can be learned by analysis.*

(6) *Finally, we define* $K(T)$ *as the least set such that* $C \in P_{K(T)}(T)$ *implies* $C \in K(T)$.

We remark that, assuming T finite, $K(T)$ can be computed in time proportional to the number of symbols in T. Moreover, we note the following properties which can be easily derived from the definition 3.2.

Lemma 3.3 *Suppose* $T \subseteq \mathcal{M}$, $t, t_1, t_2 \in \mathcal{M}$, $K \subseteq_{fin} \mathcal{N}$, $C \in \mathcal{N}$. *Then:*

(1) *If* $t \in \mathcal{T}_K$ *then* $S(A(\{t\})) = S_K(I_K(\{t\}))$.

(2) $t \in \mathcal{T}_K$ *iff* $K = K(\{t\})$.

(3) $C \in S_K(I_K(T))$ *iff* $C \in K$.

(4) $\langle t_1, t_2\rangle \in S_K(I_K(T))$ *iff* $t_i \in S_K(I_K(T)), i = 1, 2$.

(5) *Suppose* $C \in K$. *Then* $E(t, C) \in S_K(I_K(T))$ *iff* $t \in S_K(I_K(T))$,

(6) *Suppose* $C \notin K$. *Then* $E(t, C) \in S_K(I_K(T))$ *iff* $E(t, C) \in I_K(T)$.

If $T = \{t_1, \ldots, t_n\} \subset \mathcal{M}$ then we abbreviate $\langle t_1, \langle \ldots, \langle t_{n-1}, t_n\rangle \cdots\rangle \in \mathcal{T}_K$ by writing $T \in \mathcal{T}_K$.

We now come to the lemma which is the keystone in the definition of symbolic reduction. Let $\vec{T} \equiv T_1 \subseteq \cdots \subseteq T_n$ be a non decreasing sequence of finite sets of terms in $\mathcal{M}_V$ such that $Var(T_i) \subseteq \{x_1, \ldots, x_{i-1}\}$ and let σ be a *compatible* substitution, where compatibility is defined as:

$$\begin{array}{ll} \sigma(x_i) \in S(A(\sigma(T_i))) & \text{for } i = 1, \ldots, n, \\ \sigma(y) = y & \text{if } y \notin \{x_1, \ldots, x_n\}. \end{array}$$

Lemma 3.4 *Under the hypotheses above:*

(1) $\sigma T_i \in \mathcal{T}_{K(T_i)}$.

(2) $I_{K(T_i)}(\sigma T_i) = \{E(\sigma t, C) \mid E(t, C) \in I_{K(T_i)}(T_i)\}$.

Proof. (1) We prove (1) together with the condition

$$(1') \quad C \in P_{K(T_i)}(\sigma x_j) \Rightarrow C \in K(T_i) \quad \text{if } j < i.$$

We proceed by induction on the pair (i, j) lexicographically ordered.

($i = 1$) In this case, condition ($1'$) holds trivially, since $\nexists j \;\; j < 1$ and (1) holds because $\sigma T_1 = T_1$ and lemma 3.3(2) applies.

($i > 1$) We show first condition ($1'$). We know that $\sigma x_j \in S(A(\sigma T_j))$ by compatibility of σ and that $\sigma T_j \in \mathcal{T}_{K(T_j)}$ by inductive hypothesis. By lemma 3.3(1), it follows that $\sigma x_j \in S_{K(T_j)}(I_{K(T_j)}(\sigma T_j))$. There are two cases:

$C \in K(T_j)$. By definition $K(T_j) \subseteq K(T_i)$.

$C \notin K(T_j)$. Then it must be that $C \in P_{K(T_i)}(\sigma T_j)$. There are two subcases:

$C \in P_{K(T_i)}(T_j)$. Then $P_{K(T_i)}(T_j) \subseteq P_{K(T_i)}(T_i)$ since $T_j \subseteq T_i$ and therefore $C \in K(T_i)$.

$\exists\, l < j \;\; x_l \in P_{K(T_i)}(T_j)$ *and* $C \in P_{K(T_i)}(\sigma x_l)$. Then the inductive hypothesis applies.

Next we prove condition (1). If $C \in P_{K(T_i)}(\sigma T_i)$ then two cases can arise:

$C \in P_{K(T_i)}(T_i)$. Then $C \in K(T_i)$.

$\exists\, j < i \;\; x_j \in P_{K(T_i)}(T_i)$ *and* $C \in P_{K(T_i)}(\sigma x_j)$. Then condition $(1')$ applies.

(2) We prove (2) together with the condition

$$(2') \quad I_{K(T_i)}(\sigma x_j) \subseteq \{E(\sigma t, C) \mid E(t, C) \in I_{K(T_i)}(T_j)\} \quad \text{if } j < i.$$

As above, we proceed by induction on the pair (i, j) lexicographically ordered.

$(i = 1)$ Condition $(2')$ holds trivially because there is no $j < 1$. Condition (2) holds because $\sigma T_1 = T_1$.

$(i > 1)$ We prove condition $(2')$ first. Suppose $E(t', C) \in I_{K(T_i)}(\sigma x_j)$. We know that $C \notin K(T_i)$, $\sigma x_j \in S_{K(T_j)}(I_{K(T_j)}(\sigma T_j))$ and $K(T_j) \subseteq K(T_i)$. Therefore, it must be that $E(t', C) \in I_{K(T_i)}(\sigma T_j)$. There are two cases:

$\exists\, E(t, C) \in I_{K(T_i)}(T_j) \;\; E(t', C) = E(\sigma t, C)$. Then we are done.

$\exists\, l < j \;\; x_l \in P_{K(T_i)}(T_j)$ *and* $E(t', C) \in I_{K(T_i)}(\sigma x_l)$. We apply the inductive hypothesis on (i, l).

Next we prove condition (2). By definition, $I_{K(T_i)}(\sigma T_i) \subseteq \{E(\sigma t, C) \mid E(t, C) \in I_{K(T_i)}(T_i)\}$. On the other hand, suppose $E(t', C) \in I_{K(T_i)}(\sigma T_i)$. There are two cases:

$\exists\, E(t, C) \in I_{K(T_i)}(T_i) \;\; E(t', C) = E(\sigma t, C)$. Then we are done.

$\exists\, j < i \;\; x_j \in P_{K(T_i)}(T_i)$ *and* $E(t', C) \in I_{K(T_i)}(\sigma x_j)$. Condition (2) applies. $\diamond$

Definition 3.5 *A symbolic configuration is a triple (P, T, E) where:*

(1) *P is a process and T is a non-empty finite set of terms in $\mathcal{M}_V$ such that for some set of variables $\{x_1, \ldots, x_n\}$, $FV(P) \cup Var(T) \subseteq \{x_1, \ldots, x_n\}$.*

(2) *E is an* environment, *namely a possibly empty list of the shape $x_1 : T_1, \ldots, x_n : T_n$ where $T_1 \subseteq \cdots \subseteq T_n \subseteq T$, $T_1 \neq \emptyset$, and $Var(T_i) \subseteq \{x_1, \ldots, x_{i-1}\}$.* [1]

[1] In this paper, the term *environment* is used with two different meaning: in an informal context, it refers to a possibly hostile process with which principals exchange messages, while in a formal context it refers to a list $x_1 : T_1, \ldots, x_n : T_n$ with the properties specified above.

We note that $K(T_1) \subseteq \cdots \subseteq K(T_n)$. In figure 2, we define a basic system to rewrite symbolic configurations. As usual, we assume that all bound variables are distinct and different from the free variables. The rules symmetric to (m^s_{3-6}) are omitted. Moreover, the substitution $[t/x](P,T,E)$ is defined as $([t/x]P,[t/x]T,[t/x]E)$, and $[t/x]E$ is just an abbreviation defined in figure 2 in the special cases we need.

In defining the symbolic reduction, our strategy is to maintain the constraints $x_1 : T_1, \ldots, x_n : T_n$ in the form required to apply lemma 3.4. We note in particular, that in the rules (m^s_{5-6}) for equalities we do not follow the usual unification procedure. For instance, an equation $[x_i = \langle t_1, t_2\rangle]$ is not directly eliminated by a substitution $[\langle t_1, t_2\rangle / x_i]$ as t_i may contain variables of 'higher rank', *e.g.*, it may contain a variable x_j whose constraint T_j depends on x_i. Instead, we perform the substitution $[\langle x', x''\rangle / x_i]$ and solve the equations $[x' = t_1]$ and $[x'' = t_2]$ where x' and x'' are fresh variables with the same rank as x_i. We will present in section 4 a precise definition of rank and a proof that this non-standard unification procedure does terminate.

For the sake of simplicity, we consider first processes P which satisfy the following conditions:

(1) All terms occurring in P belong to $\mathcal{M}_V$.

(2) P does not contain the operator ν for name creation (thus in a configuration we omit the counter n).

(3) P does not contain the operator $|$ of parallel composition.

(4) All conditionals occurring in P are of the form $[t = t']P, 0$ that we abbreviate as $[t = t']P$.

Note that these conditions are preserved by reduction. We will see in section 5 that by simple extensions of the basic rewrite system the restrictions above can be lifted.

We will prove in section 4 that a symbolic configuration $(P,T,\emptyset)$ reduces to error iff the configuration (P,T) does. Moreover, we will show that all symbolic reductions are terminating and finitely branching. This entails a simple decision procedure for the reachability problem: explore all symbolic reductions and check whether they lead to an erroneous symbolic configuration. We consider two simple examples of application of this procedure.

Example 3.6 (1) *Consider the process*

$$P_1 \equiv ?x_1.!E(x_1, C_1).?x_2.\mathsf{case}\ E(x_3, C_1) = x_2\ \mathsf{in}\ \mathsf{case}\ E(x_4, C_0) = x_3\ \mathsf{in}\ err$$

where initially $T_1 = \{C_0\}$. *We show* $(P_1, T_1, \emptyset) \downarrow_* err$. *We have:*

$$\begin{array}{l}
(P_1, T_1, \emptyset) \ \rightarrow\ (!E(x_1, C_1) \ldots, T_1, x_1 : T_1)\ \textit{by}\ (?^s) \\
\rightarrow\ (?x_2. \ldots, T_2, x_1 : T_1)\ \textit{where}\ T_2 = T_1 \cup \{E(x_1, C_1)\},\ \textit{by}\ (!^s) \\
\rightarrow\ (\mathsf{case}\ E(x_3, C_1) = x_2\ \mathsf{in}\ \ldots, T_2, E_2)\ \textit{where}\ E_2 = x_1 : T_1, x_2 : T_2,\ \textit{by}\ (?^s) \\
\rightarrow\ [E(x_1, C_1)/x_2]([x_1/x_3]\mathsf{case}\ E(x_4, C_0) = x_3\ \mathsf{in}\ err, T_2, E_2),\ \textit{by}\ (\mathsf{c}^s_5) \\
\equiv\ (\mathsf{case}\ E(x_4, C_0) = x_1\ \mathsf{in}\ err, T_2, x_1 : T_1),\ \textit{by substitution} \\
\rightarrow\ (err, T_2, x_4 : T_1),\ \textit{by}\ (\mathsf{c}^s_4)\ .
\end{array}$$

Following the substitutions backwards, we can express the set of successful 'attacks' of the environment as $x_1 = E(x_4, C_0), x_2 = E(E(x_4, C_0), C_1)$ *where* $x_4 \in S(\{C_0\})$.

$(?^s)$ $(?x.P, T, E) \rightarrow (P, T, E, x : T)$
$(!^s)$ $(!t.P, T, E) \rightarrow (P, T \cup \{t\}, E)$
(l^s) $(\mathsf{let}\ x = t\ \mathsf{in}\ P, T, E) \rightarrow ([t/x]P, T, E)$

(c_1^s) $(\mathsf{case}\ \langle x', x''\rangle = \langle t_1, t_2\rangle\ \mathsf{in}\ P, T, E) \rightarrow ([t_1/x', t_2/x'']P, T, E)$
(c_2^s) $(\mathsf{case}\ \langle x', x''\rangle = x_i\ \mathsf{in}\ P, T, E) \rightarrow [\langle x', x''\rangle/x_i](P, T, E)$

(c_3^s) $(\mathsf{case}\ E(x, C) = E(t, C)\ \mathsf{in}\ P, T, E) \rightarrow ([t/x]P, T, E)$
(c_4^s) $(\mathsf{case}\ E(x, C) = x_i\ \mathsf{in}\ P, T, E) \rightarrow [E(x, C)/x_i](P, T, E)$ if $C \in K(T_i)$
(c_5^s) $(\mathsf{case}\ E(x, C) = x_i\ \mathsf{in}\ P, T, E) \rightarrow [E(t, C)/x_i]([t/x]P, T, E)$ if $E(t, C) \in I_{K(T_i)}(T_i)$

(m_1^s) $([f(t_1, \ldots, t_n) = f(s_1, \ldots, s_n)]P, T, E) \rightarrow ([t_1 = s_1] \ldots [t_n = s_n]P, T, E)$ f constructor
(m_2^s) $([x_i = x_i]P, T, E) \rightarrow (P, T, E)$
(m_3^s) $([x_i = x_j]P, T, E) \rightarrow [x_i/x_j](P, T, E)$ if $i < j$
(m_4^s) $([x_i = C]P, T, E) \rightarrow [C/x_i](P, T, E)$ if $C \in K(T_i)$
(m_5^s) $([x_i = \langle t_1, t_2\rangle]P, T, E) \rightarrow (\mathsf{case}\ \langle x', x''\rangle = x_i\ \mathsf{in}\ [x' = t_1][x'' = t_2]P, T, E)$ $x_i \notin Var(t_i)$
(m_6^s) $([x_i = E(t, C)]P, T, E) \rightarrow (\mathsf{case}\ E(x, C) = x_i\ \mathsf{in}\ [x = t]P, T, E)$ $x_i \notin Var(t)$

$$
\begin{aligned}
[\langle x', x''\rangle/x_i]E \equiv\ & x_1 : T_1, \ldots, x_{i-1} : T_{i-1}, x' : T_i, x'' : T_i, \\
& x_{i+1} : [\langle x', x''\rangle/x_i]T_{i+1}, \ldots, x_n : [\langle x', x''\rangle/x_i]T_n \\
\\
[E(x, C)/x_i]E \equiv\ & x_1 : T_1, \ldots, x_{i-1} : T_{i-1}, x : T_i, \\
& x_{i+1} : [E(x, C)/x_i]T_{i+1}, \ldots, x_n : [E(x, C)/x_i]T_n \\
\\
[E(t, C)/x_i]E \equiv\ & x_1 : T_1, \ldots, x_{i-1} : T_{i-1}, \\
& x_{i+1} : [E(t, C)/x_i]T_{i+1}, \ldots, x_n : [E(t, C)/x_i]T_n \\
\\
[x_i/x_j]E \equiv\ & x_1 : T_1, \ldots, x_{j-1} : T_{j-1}, \\
& x_{j+1} : [x_i/x_j]T_{j+1}, \ldots, x_n : [x_i/x_j]T_n \\
\\
[C/x_i]E \equiv\ & x_1 : T_1, \ldots, x_{i-1} : T_{i-1} \\
& x_{i+1} : [C/x_i]T_{i+1}, \ldots, x_n : [C/x_i]T_n\ .
\end{aligned}
$$

Figure 2: Basic symbolic reduction

(2) *Consider the process*

$$P_1 \equiv ?x_1.\mathsf{case}\ E(x_2,C_2)=x_1\ \mathsf{in}\ !x_2.?x_3.\mathsf{case}\ \langle x_4,x_5\rangle = x_3\ \mathsf{in}\ [x_4=C_0][x_5=C_1]err$$

where initially $T_1=\{E(C_0,C_2),E(C_1,C_2)\}$. *We show that* $(P_1,T_1,\emptyset)$ *does not reach error. We have:*

$$(P_1,T_1,\emptyset)\ \rightarrow\ (\mathsf{case}\ E(x_2,C_2)=x_1\ \mathsf{in}\ \ldots,T_1,x_1:T_1),\ \textit{by}\ (?^s)\ .$$

We then apply (c_5^s). *Since* $\sharp I_\emptyset(T_1)=2$ *we have to consider two cases:* $x_2=C_0$ *or* $x_2=C_1$. *We develop only the first one, the second being quite similar.*

$\rightarrow\ (!C_0.?x_3\ldots,T_1,\emptyset),\ \textit{by}\ (\mathsf{c}_5^s), x_2=C_0$
$\rightarrow\ (?x_3\ldots,T_2,\emptyset),\ \textit{where}\ T_2=T_1\cup\{C_0\},\ \textit{by}\ (!^s)$
$\rightarrow\ (\mathsf{case}\ \langle x_4,x_5\rangle=x_3\ \mathsf{in}\ \ldots,T_2,x_3:T_2),\ \textit{by}\ (?^s)$
$\rightarrow\ ([x_4=C_0][x_5=C_1]err,T_2,x_4:T_2,x_5:T_2),\ \textit{by}\ (\mathsf{c}_2^s)$
$\rightarrow\ ([x_5=C_1]err,T_2,x_5:T_2),\ \textit{by}\ (\mathsf{m}_4^s),\ \textit{and we are stuck as}\ C_1\notin K(T_2).$

4 Analysis

In this section, we prove the termination of the symbolic reduction relation which is not obvious due to the rules (m_{5-6}^s) that handle the conditional and we state and prove the soundness and completeness of the symbolic reduction system.

We divide the reduction rules into two sets: rules $(!^s),(?^s),(\mathsf{I}^s),(\mathsf{c}_{1-5}^s)$ which reduce the size of processes and rules (m_{1-7}^s) which deal with equations and may increase the size of processes. We note that rules (m_{5-6}^s) are followed by one of the rules $(\mathsf{c}_{2,4,5}^s)$ yielding configurations of the form $([s'=t']P',T',E')$.

We first consider the termination problem for certain 'simplified' symbolic configurations, which we take as pairs (Eq,E) where Eq is a possibly empty sequence of equations $[s=t]$, and E is an environment (cf. definition 3.5) containing all variables occurring in Eq.

The reduction rules on these pairs are given in figure 3 where the substitution of a term in an environment is defined as in figure 2. Given a pair (Eq,E), we associate a rank $rk(x)$ to every variable x occurring in an environment E' such that $(Eq,E)\stackrel{*}{\rightarrow}(Eq',E')$ as follows: if the variable is in E then its rank is its position in E, otherwise the variable inherits the rank of the variable it replaces. We note that the maximal rank of a variable occurring in a pair (Eq',E') reachable from (Eq,E) is bound by the size of the list E.

Assume every variable is assigned a rank ranging between 1 and n. Then we define the rank of a term t as 0 if the term is closed and i if i is the maximal rank of a variable occurring in t. We define the complexity of an equation $[s=t]$ with respect to the maximal rank n as

$$\mu([s=t])=(r_n,\ldots,r_1,max(|s|,|t|)) \tag{1}$$

where r_i is the number of occurrences of variables of rank i in $[s=t]$, and $|s|$ is the number of symbols in s. We define a well-founded partial ordering on pairs by setting

$$([s=t]Eq,E)\succ([s'=t']Eq',E')\ \ \text{iff}\ \ \mu([s=t])>\mu([s'=t']) \tag{2}$$

$(\mathtt{e}_1)$ $([x_i = x_i]Eq, E) \rightarrow (Eq, E)$

$(\mathtt{e}_2)$ $([x_i = x_j]Eq, E) \rightarrow [x_i/x_j](Eq, E)$ if $i < j$

$(\mathtt{e}_3)$ $([x_i = C]Eq, E) \rightarrow [C/x_i](Eq, E)$ if $C \in K(T_i)$

$(\mathtt{e}_4)$ $([x_i = \langle t_1, t_2\rangle]Eq, E) \rightarrow ([x' = t_1][x'' = t_2]([\langle x', x''\rangle/x_i]Eq), [\langle x', x''\rangle/x_i]E)$
x', x'' fresh, $x_i \notin Var(\langle t_1, t_2\rangle)$,
(here $rk(x') = rk(x'') = rk(x_i)$)

$(\mathtt{e}_5)$ $([x_i = E(t, C)]Eq, E) \rightarrow ([x = t]([E(x, C)/x_i]Eq, [E(x, C)/x_i]E)$
x fresh, $x_i \notin Var(t), C \in K(T_i)$
(here $rk(x) = rk(x_i)$)

$(\mathtt{e}_6)$ $([x_i = E(t, C)]Eq, E) \rightarrow ([t' = t][E(t', C)/x_i]Eq, [E(t', C)/x_i]E)$
x fresh, $x_i \notin Var(t), E(t', C) \in I_{K(T_i)}(T_i)$
(here $rk(x) = rk(x_i)$)

$(\mathtt{e}_7)$ $([f(t_1, \ldots, t_n) = f(s_1, \ldots, s_n)]Eq, E) \rightarrow ([t_1 = s_1] \ldots [t_n = s_n]Eq, E)$

Figure 3: Simplification rules for equalities

where $>$ denotes the lexicographic ordering. Remark that this ordering is not a lexicographic extension of the ordering on equations to sequences of equations (which is not well-founded), for instance pairs with an empty sequence of equations are incomparable with any other pair.

The domain of a substitution σ is the set $Dom(\sigma) = \{x \mid \sigma x \not\equiv x\}$. We say that a substitution σ is *decreasing* if $\forall x \in Dom(\sigma)\ rk(x) > rk(\sigma x)$. For instance, the identity substitution is decreasing since its domain is empty. We note the following properties of decreasing substitutions.

Lemma 4.1 (1) *The composition of decreasing substitutions is a decreasing substitution.*
(2) *If σ is decreasing then either $\sigma[t = t'] \equiv [t = t']$ or $\mu([t = t']) > \mu(\sigma[t = t'])$.*
(3) *The composition $\sigma \circ [t/x]$ of a decreasing substitution σ and of the substitution $[t/x]$ is decreasing if for all $y \in Var(t)$, $rk(y) \not> rk(x)$ and $y \in Dom(\sigma)$.*

The termination of rules $(\mathtt{e}_{1-7})$ in figure 3 relies on the following technical lemma which is proven in appendix A proceeding by induction on the order $\succ$.

Lemma 4.2 *Let $p \equiv ([s = t]Eq, E)$ be a pair which is reduced by rules $(\mathtt{e}_{1-7})$. Then two cases can arise: either $p \xrightarrow{*} ([s' = t']Eq', E')$ and no rule applies, or we can reduce p to a configuration $(Eq', E') \equiv \sigma(Eq, E)$, where σ satisfies:*
(i) σ is decreasing,
(ii) if $s \equiv x$ and $t \notin V$ then $x \in Dom(\sigma)$.

By iterating lemma 4.2, we can conclude that equations can be eliminated.

Proposition 4.3 *The simplification of pairs (Eq, E) always terminates.*

Moreover, lemma 4.2 entails that every reduction sequence starting from a symbolic configuration $([s = t]P, T, E)$ either terminates or reduces to an instance of (P, T, E). Since all the other rules reduce the size of the process P, an inductive argument on the size of P proves termination for basic symbolic reduction.

Theorem 4.4 *Basic symbolic reduction always terminates.*

Proof. Let $|P|$ be the number of instructions in the process P, *i.e.*, $|err| = 1$, $|\mathsf{let}\ x = t\ \mathsf{in}\ P| = 1 + |P|$, etc. All rules $(!^s), (?^s), (\mathsf{l}^s), (\mathsf{c}^s_{1-5})$ decrease $|P|$. By lemma 4.2, we know that every reduction starting from $([s = t]P, T, E)$ either stops or computes a configuration $\sigma(P, T, E)$. By definition, $|[s = t]P| > |\sigma P|$, which shows that every reduction using rules $(!^s), (?^s), (\mathsf{l}^s), (\mathsf{c}^s_{1-5}), (\mathsf{m}^s_{1-7})$ eventually terminates. $\diamond$

Next we examine the soundness and completeness of the symbolic reduction system.

Definition 4.5 *Let $k \equiv (P, T, E)$ be a symbolic configuration and σ be a ground substitution.*

(1) *We write $k \downarrow err$ if $P \equiv err$ and $k \downarrow_* err$ if $k \xrightarrow{*} k'$ and $k' \downarrow err$.*

(2) *We write $\sigma \models E$ iff σ is compatible with the sequence $T_1 \subseteq \cdots \subseteq T_n$ in E.*

Theorem 4.6 *$(P, T, E) \downarrow_* err$ iff $\exists\, \sigma \models E\ \ \sigma(P, T) \downarrow_* err$.*

Proof hint. For both implications we proceed by induction on the length of the reduction to an erroneous (symbolic) configuration. To show *completeness* (implication ($\Leftarrow$)), lemma 3.4 is instrumental. For instance, suppose $\exists\, \sigma \models E\ \ \sigma(\mathsf{case}\ E(x, C) = x_i\ \mathsf{in}\ P, T) \downarrow_* err$. By definition of compatibility, $\sigma(x_i) \in S(A(\sigma T_i))$ and for the reduction to error to be possible, it must be that $\sigma(x_i) = E(t, C)$ for some t. By lemma 3.4(1), we know that $\sigma T_i \in \mathcal{T}_{K(T_i)}$. We consider two cases.

(1) If $C \in K(T_i)$ then $t \in S(A(\sigma T_i))$ and rule (c^s_4) applies.

(2) If $C \notin K(T_i)$ then, by lemma 3.4(1), $C \notin K(\sigma T_i)$ and therefore $E(t, C) \in I_{K(T_i)}(\sigma T_i)$. By lemma 3.4(2), $\exists\, E(t', C) \in I_{K(T_i)}(T_i)\ \ t = \sigma t'$ and then rule (c^s_5) applies.

Full details of the proof are given in appendix B. Note that by taking (P, T) closed and $E = \emptyset$ we obtain that $(P, T) \downarrow_* err$ iff $(P, T, \emptyset) \downarrow_* err$. $\diamond$

5 Extensions of basic symbolic reduction

In this section, we show how to lift the four restrictions imposed in section 3 thus defining a symbolic reduction for the full model introduced in section 2. Let us first consider the four restrictions separately and present our basic ideas.

Variables in key position Suppose we have to deal with a symbolic configuration of the shape (let $x = E(t, x_i)$ in P, T, E). We note that for any substitution σ such that $\sigma \models E$, σ(let $x = E(t, x_i)$ in P, T) reduces iff $\sigma(E(t, x_i)) \in \mathcal{M}$.

In general, $\sigma t \in \mathcal{M}$ iff in every subterm $E(t', t'')$ of t, t'' is either a name or a variable, say x. In the second case, we say that x is a variable in 'key position' and $\sigma(x)$ ranges over names.

By the constraints imposed by the environment E, if $\sigma(x_i) \in \mathcal{N}$ then $x_i \in K(T_i)$ which is a finite set. Thus there are only a finite number of assignments of the variables in key positions that are compatible with E.

Formally, let $T_{\mathcal{M}} = \{t \in T_\Sigma(V) \mid \exists \sigma \ \sigma t \in \mathcal{M}\}$. We note that a term t is in $T_{\mathcal{M}}$ iff in all subterms $E(t, t')$, t' is either a name or a variable. If $t \in T_{\mathcal{M}}$ then let $Var_{key}(t)$ be the set of variables x which occur in t in 'key position', *i.e.*, such that $E(t', x)$ is a subterm of t.

Definition 5.1 *We write $\sigma \downarrow (t, E)$ if (i) $\sigma = id$ and $t \in \mathcal{M}_V$ or (ii) $t \in T_{\mathcal{M}}$, $Var_{key}(t) = \{x_{i_1}, \ldots, x_{i_m}\}$, $m \geq 1$, and $\sigma(x_l) \in K(T_l)$, if $x_l \in Var_{key}(t)$ and $\sigma(x_l) = x_l$, otherwise.*

With this notation, we rewrite the rules for let, output, case, and conditional by combining reduction and instantiation of variables in key position. For instance, the rule (l) becomes:

$$(\text{let } x = t \text{ in } P, T, E) \ \rightarrow \ \sigma([t/x]P, T, E) \ \text{ if } \sigma \downarrow (t, E) \ .$$

We should pause to note that this is a brute force method. In an implementation, it appears that a more efficient approach is to restrict the range of a variable in key position, say x_i, to the corresponding finite set of names $K(T_i)$ (provided it is not empty).

Restriction As in the reduction of (standard) configurations (figure 1), it is enough to introduce a counter. Then a symbolic configuration is a quadruple (P, n, T, E) where n is a natural number such that $cnst(P) \cup cnst(T) \subseteq \mathcal{N}(n)$. We then add the following rule for restriction.

$$(\nu x \, P, n, T, E) \ \rightarrow \ ([C_{n+1}/x]P, n+1, T, E) \ .$$

Parallel composition We assume that parallel composition is associative and commutative and that $P \equiv P \mid 0$. Then, without loss of generality, we always reduce the leftmost thread of a process and we assume that there is at least another thread running in parallel. For instance, the rule for output is rewritten as follows.

$$(!t.P \mid P', T, E) \ \rightarrow \ (P \mid P', T \cup \{t\}, E)$$

We note that the confluence of non-input reductions (proposition 2.5) does not apply – litterally– to symbolic reductions. For instance, consider the symbolic configuration

$$(\text{let } x = E(t, x_1) \text{ in } 0 \mid [x_1 = \langle C_0, C_0 \rangle] err, \{C_0\}, x_1 : \{C_0\}) \ .$$

The output and the conditional do not commute as to fire the (l) rule we have to instantiate the variable x_1 with a name thus precluding the reachability of error in the other parallel

$$(\mathrm{i}_1)\quad \{t \neq t\} \cup I \;\rightarrow\; \bot$$

$$(\mathrm{i}_2)\quad \{f(\vec{s}) \neq g(\vec{t})\} \cup I \;\rightarrow\; I$$

$$(\mathrm{i}_3)\quad \{f(\vec{s}) \neq f(\vec{t})\} \cup I \;\rightarrow\; \{s_i \neq t_i\} \cup I \quad 1 \leq i \leq arity(f)$$

$$(\mathrm{i}_4)\quad \{x \neq t\} \cup I \;\rightarrow\; I \quad \text{if } x \in Var(t) \text{ and } t \not\equiv x$$

Figure 4: Reduction for inequalities

component. We expect that useful confluence results can be obtained when the threads running in parallel do not share variables.

Conditional We enrich a symbolic configuration with another component I which is a finite set of inequalities $s \neq t$ or a special symbol $\bot$ (which is never satisfied). We write $\sigma \models I$ iff $I \neq \bot$ and σ satisfies all inequalities in I. We present in figure 4 a standard set of simplification rules for inequalities.

We note that each rule decreases the number of symbols. Then, we can state the following proposition.

Proposition 5.2 *Rewriting by the rules for inequalities* (i_{1-4}) *in figure 4 always terminates.*

We can now introduce our most general notion of symbolic reduction.

Definition 5.3 *We say that a set of inequalities is* simplified *if it is not equal to* $\bot$ *and the reduction rules* (i_{1-4}) *do not apply.*

Definition 5.4 *A symbolic configuration is a quintuple* (P, n, T, E, I) *where:*

(1) *P is a process (as specified above) and T is a non-empty finite set of terms in $\mathcal{M}_V$ such that for some set of variables $\{x_1, \ldots, x_n\}$, $FV(P) \cup Var(T) \subseteq \{x_1, \ldots, x_n\}$.*

(2) *E is an environment, $x_1 : T_1, \ldots, x_n : T_n$ as specified in definition 3.5.*

(3) *n is a natural number such that if $C_m \in cnst(P) \cup cnst(T) \cup cnst(I)$ then $m \leq n$.*

(4) *I is a finite set of inequalities or $\bot$.*

We present in figures 5, 6, and 7 the full system for symbolic reduction where we denote with u either a variable or a constant and with P a process which admits as subprocesses two special forms of the conditional, namely $[t = t']P$ and $[t \neq t']P$, provided $t, t' \in \mathcal{M}_V$.

Definition 5.5 *Let $k \equiv (P, n, T, E, I)$ be a symbolic configuration. We write $k \downarrow err$ if $P \equiv err \mid P'$ and I is simplified. As usual, we write $k \downarrow_* err$ if $k \xrightarrow{*} k'$ and $k' \downarrow err$.*

This definition of symbolic reachability of error is justified by the observation that if I is simplified then we can always find a substitution σ such that $\sigma \models E$ and $\sigma \models I$ (henceforth abbreviated as $\sigma \models E, I$). To see this, let us define the height $h(t)$ of a term t as $h(C) = h(x) = 1$ and $h(f(t_1, \ldots, t_n)) = 1 + max\{h(t_i) \mid i = 1, \ldots, n\}$. We note that if σ is a substitution then

$$h(\sigma t) \leq max(h(t), max\{h(t) - 1 + h(\sigma x) \mid x \in Var(t)\} \ .$$

Moreover, given a set $T \subset \mathcal{M}$, $T \neq \emptyset$ we can always find an integer h_0 such that $\forall\, h > h_0 \ \ \exists\, t \in S(A(T)) \ \ h(t) = h$. Take $h_0 = min\{h(t) - 1 \mid t \in T\}$ and $t_0 \in T$ such that $h(t_0) = h_0 + 1$. Then, using pairing over t_0, we can build terms of arbitrary height $h > h_0$.

Proposition 5.6 *Let I be a simplified set of inequalities and h_0 be an integer such that $Var(I) = \{x_1, \ldots, x_n\}$ and x_i ranges over a set R_i containing terms of height h for all $h > h_0$. Then there is a substitution σ such that $\sigma \models I$ and $\sigma x_i \in R_i$.*

Proof. If $I = \emptyset$ then we are done. So suppose I is not empty and simplified. Then it can be written as $\bigwedge_{i=1,\ldots,n} \bigwedge_{j=1,\ldots,m_i} x_i \neq t_{i,j}$ where $x_i \notin Var(t_{i,j})$. Let $H_m = max\{h(t_{i,j}) \mid 1 \leq i \leq n, 1 \leq j \leq m_i\}$ and let σ be any substitution such that $h(\sigma x_i) = (i+1)H$ with $H = max(H_m, h_0)$ and $\sigma x_i \in R_i$. There is at least one such substitution by hypothesis. We claim that $\sigma \models I$. We distinguish two cases:

$x_i \neq t_{i,j}, \ \ Var(t_{i,j}) \subseteq \{x_1, \ldots, x_{i-1}\}$. Then:

$$h(\sigma x_i) = (i+1)H > max_{j=1,\ldots,i-1}(H_m, H_m - 1 + (j+1)H) \geq h(\sigma t_{i,j}) \ .$$

$x_i \neq t_{i,j}, \ \exists\, j > i \ \ x_j \in Var(t_{i,j})$. Then:

$$h(\sigma t_{i,j}) \geq h(\sigma x_j) = (j+1)H > (i+1)H = h(\sigma x_i) \ . \diamond$$

Corollary 5.7 *If $(P, n, T, E, I) \downarrow err$ then $\exists\, \sigma \ \ \sigma \models E, I$.*

Next we comment the reduction rules. The first group of rules presented in figure 5 deals with all operators but conditional and applies the ideas sketched above to handle variables in key position, restriction, and parallel composition. The rules to handle the conditional are presented in figures 6 and 7. In the first rule (m_1^s), we instantiate the terms in the conditional $([t = t']P_1, P_2)$ and split it in two parts $\sigma([t = t']P_1)$ and $\sigma([t \neq t']P_1)$. Then, we develop the rules that handle the simpler conditionals $[t = t']P$ and $[t \neq t']P$ where $t, t' \in \mathcal{M}$. Equalities are reduced by the following rules of figure 6 (which are similar to the corresponding rules of figure 2) where the rules symmetric to (m_{4-7}^s) are omitted. By rule (i^s) in figure 7, inequalities $[t \neq t']P$ are shifted to I which is then simplified by rule (i_{1-4}^s).

The reduction process may diverge when we interleave the rules for equalities in two parallel threads with shared variables. For instance, consider the reduction

$$\begin{array}{l} ([x = \langle y, z\rangle]0 \mid [y = \langle x, w\rangle]0, \ldots) \\ \rightarrow ([x' = y][x'' = z]0 \mid [y = \langle\langle x', x''\rangle, w\rangle]0, \ldots, \ldots) \\ \rightarrow ([x' = \langle y', y''\rangle][x'' = z]0 \mid [y' = \langle x', x''\rangle][y'' = w]0, \ldots) \end{array}$$

$(?^s)$ $(?x.P \mid P', n, T, E, I) \rightarrow (P \mid P', n, T, E, x : T, I)$

$(!^s)$ $(!t.P \mid P', n, T, E, I) \ \rightarrow \ \sigma(P \mid P', n, T \cup \{t\}, E, I)$ if $\sigma \downarrow (t, E)$

(let^s) $(\mathsf{let}\ x = t\ \mathsf{in}\ P \mid P', n, T, E, I) \ \rightarrow \ \sigma([t/x]P \mid P', n, T, E, I)$ if $\sigma \downarrow (t, E)$

(ν^s) $(\nu x\, P \mid P', n, T, E, I) \ \rightarrow \ ([C_{n+1}/x]P \mid P', n+1, T, E, I)$

(c_1^s) $(\mathsf{case}\ \langle x', x''\rangle = \langle t_1, t_2\rangle\ \mathsf{in}\ P \mid P', n, T, E, I) \ \rightarrow \ \sigma([t_1/x', t_2/x'']P \mid P', n, T, E, I)$
if $\sigma \downarrow (\langle t_1, t_2\rangle, E)$

(c_2^s) $(\mathsf{case}\ \langle x', x''\rangle = x_i\ \mathsf{in}\ P \mid P', n, T, E, I) \ \rightarrow \ [\langle x', x''\rangle/x_i](P \mid P', n, T, E, I)$

(c_3^s) $(\mathsf{case}\ E(x, u) = E(t, u')\ \mathsf{in}\ P \mid P', n, T, E, I) \ \rightarrow \ \sigma([t/x]P \mid P', n, T, E, I)$
if $\sigma \downarrow (E(E(t, u'), u), E), \sigma(u) = \sigma(u')$

(c_4^s) $(\mathsf{case}\ E(x, C) = x_i\ \mathsf{in}\ P \mid P', n, T, E, I) \ \rightarrow \ [E(x, C)/x_i](P \mid P', n, T, E, I)$
if $C \in K(T_i)$

(c_5^s) $(\mathsf{case}\ E(x, x_j) = x_i\ \mathsf{in}\ P \mid P', n, T, E, I) \ \rightarrow \ [E(x, C)/x_i, C/x_j](P \mid P', n, T, E, I)$
if $i \neq j,\ C \in K(T_{min(i,j)})$

(c_6^s) $(\mathsf{case}\ E(x, C) = x_i\ \mathsf{in}\ P \mid P', n, T, E, I) \ \rightarrow [E(t, C)/x_i]([t/x]P \mid P', n, T, E, I)$
if $E(t, C) \in I_{K(T_i)}(T_i)$

(c_7^s) $(\mathsf{case}\ E(x, x_j) = x_i\ \mathsf{in}\ P \mid P', n, T, E, I) \ \rightarrow [E(t, C)/x_i, C/x_j]([t/x]P \mid P', n, T, E, I)$
if $E(t, C) \in I_{K(T_i)}(T_i), i < j, C \in K(T_j)$

Figure 5: Symbolic reduction (without conditional)

$$(\mathsf{m}_1^s) \quad ([t = t']P_1, P_2 \mid P', n, T, E, I) \rightarrow \begin{cases} \sigma([t = t']P_1 \mid P', n, T, E, I) \\ \sigma([t \neq t']P_2 \mid P', n, T, E, I) \end{cases} \quad \sigma \downarrow \langle t, t' \rangle$$

$$(\mathsf{m}_2^s) \quad ([f(\vec{t}) = f(\vec{s})]P \mid P', n, T, E, I) \rightarrow ([t \stackrel{\rightarrow}{=} s]P \mid P', n, T, E, I) \ f \text{ constr.}$$

$$(\mathsf{m}_3^s) \quad ([x_i = x_i]P \mid P', n, T, E, I) \rightarrow (P \mid P', n, T, E, I)$$

$$(\mathsf{m}_4^s) \quad ([x_i = x_j]P \mid P', n, T, E, I) \rightarrow [x_i/x_j](P \mid P', n, T, E, I) \text{ if } i < j$$

$$(\mathsf{m}_5^s) \quad ([x_i = C]P \mid P', n, T, E, I) \rightarrow [C/x_i](P \mid P', n, T, E, I) \text{ if } C \in K(T_i)$$

$$(\mathsf{m}_6^s) \quad ([x_i = \langle t_1, t_2 \rangle]P \mid P', n, T, E, I) \rightarrow (\mathsf{case}\ \langle x', x'' \rangle = x_i\ \mathsf{in}\ [x' = t_1][x'' = t_2]P \mid P', n, T, E, I) \text{ if } x_i \notin Var(t_i)$$

$$(\mathsf{m}_7^s) \quad ([x_i = E(t, C)]P \mid P', n, T, E, I) \rightarrow (\mathsf{case}\ E(x, C) = x_i\ \mathsf{in}\ [x = t]P \mid P', n, T, E, I) \text{ if } x_i \notin Var(t)$$

Figure 6: Symbolic reduction: rules for equalities

$$(\mathsf{i}^s) \quad ([s \neq t]P \mid P', n, T, E, I) \rightarrow (P \mid P', n, T, E, I \cup \{s \neq t\})$$

$$(\mathsf{i}_{1-4}^s) \quad (P \mid P', n, T, E, I) \rightarrow (P \mid P', n, T, E, I') \quad \text{if } I \rightarrow I' \text{ by } (\mathsf{i}_{1-4})$$

Figure 7: Symbolic reduction: rules for inequalities

After two reduction steps and modulo renaming, we are back to a configuration of the shape $([x = \langle y, z\rangle]P \mid [y = \langle x, w\rangle]Q, \ldots)$

To avoid this pathological loop, we enforce the following reduction strategy on the rules (m_{2-7}): if one of these rule is applied in a thread $[s = t]Q$ of the configuration $([s = t]Q \mid \ldots, n, T, E, I)$, then we keep applying these rules till we get stuck or we obtain a configuration $\sigma(Q \mid \ldots, n, T', E', I')$.

We call this strategy *eager reduction of equations*. Lemma 4.2 ensures that the reduction of a component $([s = t]Q \mid \ldots, n, T, E, I)$ always terminates, and yields a configuration $\sigma(Q \mid \ldots, n, T', E', I')$.

Theorem 5.8 (1) *Symbolic reduction with the eager reduction of equations always terminates.*

(2) $(P, n, T, E, I) \downarrow_* err$ *iff* $\exists\, \sigma\ \ (\sigma \models E, I$ *and* $\sigma(P, n, T) \downarrow_* err)$.

Proof. (1) We take as the size of a configuration the pair composed of the size of the process (as defined in the proof of theorem 4.4) and the size of the set of inequalities (defined as the number of symbols in it, cf. proposition 5.2). We observe that all rules decrease the size of a configuration but the rules for reduction of equations. The eager reduction strategy ensures that rules for equations eventually lead to a smaller configuration.

(2) The proof technique applied for basic symbolic reduction (theorem 4.6 and appendix B) can be easily adapted.

($\Rightarrow$) We proceed by induction on the length of the reduction to error. We consider two typical cases.

- Instantiation of variables in key position: case (I^s). Suppose

$$(\mathsf{let}\ x = t\ \mathsf{in}\ P \mid P', n, T, E, I) \to \sigma'([t/x]P \mid P', n, T, E, I) \downarrow_* err$$

where $\sigma' \downarrow (t, T)$. By inductive hypothesis,

$$\exists\, \sigma'' \models E, I\ \ (\sigma''(\sigma'([t/x]P \mid P', n, T)) \downarrow_* err)$$

Then we conclude taking $\sigma = \sigma'' \circ \sigma'$.

- Rules for inequalities. Suppose

$$([s \neq t]P, n, T, E, I) \to (P, n, T, E, I \cup \{s \neq t\}) \downarrow_* err$$

by (i^s). By inductive hypothesis,

$$\exists\, \sigma \models E, I \cup \{s \neq t\}\ \ \sigma(P, n, T) \downarrow_* err$$

and we conclude observing $\sigma([s \neq t]P, n, T) \to \sigma(P, n, T)$. On the other hand, if we apply rules (i^s_{1-4}), it is enough to observe that if $I \to I'$ and $\sigma \models I'$ then $\sigma \models I$.

($\Leftarrow$) Suppose $\sigma \models E, I$ and $\sigma(P, n, T) \downarrow_* err$. We proceed by induction on the length of the reduction to error. We consider three typical cases.

- If the reduction has length 0 then $\sigma(P, n, T) \equiv \sigma(err \mid P', n, T)$ for some P'. We observe that if $\sigma \models I$ and $I \rightarrow I'$ then $\sigma \models I'$. By proposition 5.2, we eventually reach an I' which is simplified. By iterated application of rule (i^s_{1-4}), we conclude that $(err \mid P', n, T, E, I) \downarrow_* err$.
- The case for the let. Assume

$$\sigma(\mathsf{let}\ x = t\ \mathsf{in}\ P \mid P', n, T) \rightarrow \sigma([t/x]P \mid P', n, T) \downarrow_* err\ .$$

Then $\sigma t \in \mathcal{M}$. Define $U = Var_{key}(t)$ and $U' = Var(E) \backslash U$ and denote with $\sigma_{|U}, \sigma_{|U'}$ the restriction of σ to U, U', respectively. From $\sigma \models E$, we derive that $\sigma_{|U} \downarrow (t, E)$ and $\sigma = \sigma_{|U'} \circ \sigma_{|U}$. Then

$$(\mathsf{let}\ x = t\ \mathsf{in}\ P \mid P', n, T, E, I) \rightarrow \sigma_{|U}([t/x]P \mid P', n, T, E, I)$$

and since $\sigma_{|U'} \models \sigma_{|U}(E, I)$ the inductive hypothesis applies.

- The case for inequalities. Assume

$$\sigma([t = s]P_1, P_2 \mid P', n, T) \rightarrow \sigma(P_2 \mid P', n, T) \downarrow_* err\ .$$

Then $\sigma t \neq \sigma s$ and $\sigma s, \sigma t \in \mathcal{M}$. If we define $U = Var_{key}(\langle t, s \rangle)$ and $U' = Var(E) \backslash U$ we obtain as above that $\sigma_{|U} \downarrow (\langle t, s \rangle, E)$ and $\sigma = \sigma_{|U'} \circ \sigma_{|U}$. At the symbolic level, we can perform the following reductions:

$$\begin{array}{ll} ([t = s]P_1, P_2 \mid P', n, T, E, I) & \rightarrow \sigma_{|U}([t \neq s]P_2 \mid P', n, T, E, I) \\ & \rightarrow \sigma_{|U}(P_2 \mid P', n, T, E, I \cup \{t \neq s\}) \end{array}$$

We can then apply the inductive hypothesis, observing that $\sigma_{|U'} \models \sigma_{|U}(I, E)$ and $\sigma_{|U'}\sigma_{|U}(P_2 \mid P', n, T) \downarrow_* err$. $\diamond$

6 Conclusion

We have proposed a simple decision procedure for the reachability problem which is based on a reduction system acting on symbolic configurations. While being conceptually simple, we can hardly claim that our procedure is efficient.

Consider the symbolic configuration $(P, T, \emptyset)$ where $T = \{E(C_0, C), E(C_1, C)\}$ and

$$\begin{array}{rl} P \equiv ?x_1 \ldots ?x_n. & \mathsf{case}\ E(y_1, C) = x_1\ \mathsf{in} \cdots \\ & \mathsf{case}\ E(y_n, C) = x_n\ \mathsf{in}\ P' \end{array}$$

Assuming $P' \equiv 0$, the symbolic reduction of this configuration contains a complete binary tree of depth n which has to be explored completely to conclude that an erroneous configuration cannot be reached.

Actually, there is a good reason for this complexity, the reachability problem is NP-hard even if we restrict our attention to very simple programs (this seems to be a folk theorem). For instance, let us see how to reduce satisfaction of a boolean formula to the reachability problem. Let ϕ be a boolean formula depending on the variables $\{y_1, \ldots, y_n\}$. Suppose we code the boolean value 0 with C_0 and the boolean value 1 with C_1. We can rely on the program P above to select an arbitrary assignment ρ of boolean values to $y_1, \ldots, y_n$. We are then left with the problem of writing a process P' depending on $\{y_1, \ldots, y_n\}$ such that ϕ evaluates to true in ρ iff $\rho P' \downarrow_* err$.

We can do this in two ways. If ϕ is generated by the grammar $\phi ::= 0 \parallel 1 \parallel \mathsf{if}\ y_i\ \mathsf{then}\ \phi\ \mathsf{else}\ \phi$ then we translate 0 to 0, 1 to err, and $\mathsf{if}\ y_i\ \mathsf{then}\ \phi'_1\ \mathsf{else}\ \phi'_2$ to $[y_i = C_1]P'_1, P'_2$, where inductively, ϕ'_i translates to P'_i. On the other hand, if $\phi = \psi_1 + \ldots + \psi_m$ is in disjunctive normal form, then we build $P' = (Q_1 \mid \cdots \mid Q_m)$ where if, for instance, $\psi_i = (y_1 \overline{y_2} y_3)$, we set $Q_i = [y_1 = C_1][y_2 = C_0][y_3 = C_1]err$.

In collaboration with V. Vanackere, we are currently implementing our decision procedure to verify whether this complexity actually arises in practice when analysing protocols such as those described in [CJ97]. It seems also interesting to reconsider the automata-theoretic techniques advocated by [Mon99] in our framework (cf. proposition 2.2) and see whether they lead to a better performance either in the worst case or in practice. Finally, we plan to investigate whether our techniques entail decision procedures for suitable notions of bisimulation (cf. [AG97, BDNP99]).

References

[AG97] M. Abadi and A. Gordon. A calculus for cryptographic protocols: the spi calculus. In *Proc. ACM Computer and Comm. Security*, 1997.

[AP99] R. Amadio and S. Prasad. The game of the name in cryptographic tables. In *Proc. ASIAN99, Springer Lect. Notes in Comp. Sci. 1742*, pages 15–26, 1999.

[BDNP99] M. Boreale, R. De Nicola, and R. Pugliese. Proof techniques for cryptographic processes. In *Proc. IEEE Logic in Comp. Sci.*, 1999.

[Bol96] D. Bolignano. Formal verification of cryptographic protocols. In *Proc. ACM Conference on Computer Communication and Security*, 1996.

[Bor00] M. Boreale. Symbolic analysis of cryptographic protocols in the spi-calculus. Personal communication, 2000.

[CDG+] H. Comon, M. Dauchet, R. Gilleron, F. Jacquemard, D. Lugiez, S. Tison, and M. Tommasi. *Tree Automata Techniques and Applications.* Draft available at *http://www.grappa.univ-lille3.fr/tata.*

[CJ97] J. Clark and J. Jacob. A survey of authentication protocol literature: Version 1.0. Technical report, 1997. Available at http://www.cs.york.ac.uk/ ∼jac/papers/ drareview.ps.gz.

[CJM98] E. Clarke, S. Jha, and W. Marrero. Using state space exploration and a natural deduction style message derivation engine to verify security protocols. In *Proc. IFIP Working Conference on Programming Concepts and Methods (PROCOMET)*, 1998.

[DY83] D. Dolev and A. Yao. On the security of public key protocols. *IEEE Trans. on Information Theory*, 29(2):198–208, 1983.

[Hui99] A. Huima. Efficient infinite-state analysis of security protocols. In *Proc. Formal methods and security protocols, FLOC Workshop, Trento*, 1999.

[Low96] G. Lowe. Breaking and fixing the Needham-Schroeder public-key protocol using FDR. In *Proc. TACAS, Springer Lect. Notes in Comp. Sci. 1996*, 1996.

[Mea94] C. Meadows. A model of computation for the nrl protocol analyzer. In *Proc. IEEE Computer Security Foundations Workshop*, 1994.

[MMS97] J. Mitchell, M. Mitchell, and U. Stern. Automated analysis of cryptographic protocols using murϕ. In *Proc. IEEE Symposium on Security and Privacy*, 1997.

[Mon99] D. Monniaux. Abstracting cryptographic protocols with tree automata. In *Proc. Static Analysis Symposium, Springer Lect. Notes in Comp. Sci.*, 1999.

[Pau97] L. Paulson. Proving properties of security protocols by induction. In *Proc. IEEE Computer Security Foundations Workshop*, 1997.

[Pau99] L. Paulson. Proving security protocols correct. In *Proc. IEEE Logic in Comp. Sci.*, 1999.

[Sch96] S. Schneider. Security properties and CSP. In *Proc. IEEE Symp. Security and Privacy*, 1996.

[Wei99] C. Weidenbach. Towards an automatic analysis of security protocols in first-order logic. In *Proc. CADE 99*. Springer Lect. Notes in Comp. Sci. (LNAI) 1632, 1999.

A Proof of lemma 4.2

PROOF. Let n be the size of E. All the variables we consider will have a rank bound by n. We proceed by induction on the order $\succ$ defined with respect to the maximal rank n.

(e_{1-3}) $([x_i = t]Eq, E) \stackrel{e_{1-3}}{\rightarrow} \sigma(Eq, E)$ where σ is a decreasing substitution by construction. Moreover if $t \notin Var$ then $x_i \in Dom(\sigma)$.

(e_4) $([x_i = \langle t_1, t_2\rangle]Eq, E) \stackrel{\mathsf{e}_4}{\rightarrow} ([x' = t_1][x'' = t_2][\langle x', x''\rangle/x_i]Eq, [\langle x', x''\rangle/x_i]E)$.

By definition, we have (a) $\mu([x_i = \langle t_1, t_2\rangle]) > \mu([x' = t_1])$.
(b) $\mu([x_i = \langle t_1, t_2\rangle]) > \mu([x'' = t_2])$.

By (a), the induction hypothesis applies. Then either the reduction terminates or there exists a substitution σ_1 satisfying: (i) σ_1 is decreasing, (ii) $x' \in Dom(\sigma_1)$. By induction hypothesis, we have the following reduction:

$$([x_i = \langle t_1, t_2\rangle]Eq, E) \stackrel{*}{\rightarrow} ([\sigma_1 x'' = \sigma_1 t_2]\sigma_1[\langle x', x''\rangle/x_i]Eq, \sigma_1[\langle x', x''\rangle/x_i]E) \ .$$

Since σ_1 is decreasing, by lemma 4.1(2) either $\mu([x'' = t_2]) > \mu([\sigma_1 x'' = \sigma_1 t_2])$ or $\sigma_1 x'' = x''$ and $\sigma_1 t_2 = t_2$. By (b), the induction hypothesis applies. Then either the reduction terminates or there exists some substitution σ_2 satisfying (i) and (ii). We distinguish two cases depending on whether $\sigma_1 x'' = x''$ or not.

- $\sigma_1 x'' = x''$. The induction hypothesis yields: σ_2 is decreasing and $x'' \in Dom(\sigma_2)$.

  (i) Let $\sigma = \sigma_2\sigma_1[\langle x', x''\rangle/x_i]$. if $x' \in Dom(\sigma_1), x'' \in Dom(\sigma_2)$, $rk(x') = rk(x'') = rk(x_i)$, and σ_i decreasing for $i = 1, 2$, then $\sigma_2\sigma_1 x' \not\equiv x'$ and $\sigma_2\sigma_1 x'' \not\equiv x''$. Then $x' \in Dom(\sigma_2\sigma_1), x'' \in Dom(\sigma_2\sigma_1)$ and, by lemma 4.1(3) σ is decreasing.

  (ii) By definition, $x_i \in Dom(\sigma)$.

- $\sigma_1 x'' \not\equiv x''$. Let $\sigma = \sigma_2\sigma_1[\langle x', x''\rangle/x_i]$.

  (i) We have $x', x'' \in Dom(\sigma_1)$, and σ_1 decreasing implies that $rk(x') > rk(\sigma_1 x')$ and $rk(x'') > rk(\sigma_1 x'')$. Therefore $rk(x') > rk(\sigma_2\sigma_1 x')$ and $rk(x'') > rk(\sigma_2\sigma_1 x'')$ which proves that $x' \not\equiv \sigma_2\sigma_1 x'$ and $x'' \not\equiv \sigma_2\sigma_1 x''$. Then $x', x'' \in Dom(\sigma_2\sigma_1)$ and lemma 4.1(3) proves that σ is decreasing.

  (ii) By definition $x_i \in Dom(\sigma)$.

In each case, we get a substitution σ satisfying the requirements of the lemma.

(e_5) $([x_i = E(t, C)]Eq, E) \stackrel{\mathsf{e}_5}{\rightarrow}$ $([x = t][E(x, C)/x_i]Eq, [E(x, C)/x_i]E)$, with x fresh, $rk(x) = rk(x_i)$.

By definition, $\mu([x_i = E(t,C)]) > \mu([x = t])$ since $|E(t,C)| > |t|$. The induction hypothesis applies: either the reduction terminates or there exists a substitution σ_1 satisfying (i) and (ii). Let $\sigma = \sigma_1 \circ [E(x,C)/x_i]$.

(i) We observe that $x \in Dom(\sigma_1)$, σ_1 is decreasing, and $rk(x) = rk(x_i)$. Then σ is decreasing by lemma 4.1(3).

(ii) By definition, $x_i \in Dom(\sigma)$.

Therefore $([x_i = E(t,C)]Eq, E) \xrightarrow{*} \sigma(Eq, E)$ and σ satisfies the requirements of the lemma.

(e_6) $([x_i = E(t,C)]Eq, E) \xrightarrow{\mathsf{e}_6} ([t' = t] \quad [E(t',C)/x_i](Eq, E)$ with $rk(x_i) > rk(t')$.

By definition $\mu([x_i = E(t,C)]) > \mu([t' = t])$ since $rk(x_i) > rk(t')$. The induction hypothesis applies and either the reduction terminates or there is some σ_1 satisfying (i), (ii).

(i) Let $\sigma = \sigma_1 \circ [E(t',C)/x_i]$. By lemma 4.1(1), σ is decreasing. (ii) We observe that $x_i \in Dom(\sigma)$.

Therefore $([x_i = E(t,C)]Eq, E) \xrightarrow{*} \sigma(Eq, E)$ and σ satisfies the requirement of the lemma.

(e_7) $([f(\vec{s}) = f(\vec{t})]Eq, E) \xrightarrow{\mathsf{e}_7} ([s_1 = t_1] \ldots [s_m = t_m]Eq, E)$.

By definition, we have $\mu([s = t]) > \mu([s_i = t_i])$ for $i = 1, \ldots, m$. Then either the reduction terminates or there exists a substitution σ_1 satisfying (i)(ii) such that $([s_1 = t_1] \ldots [s_m = t_m]Eq, E) \rightarrow \sigma_1([s_2 = t_2] \ldots [s_m = t_m]Eq, E)$.

Since σ_1 is decreasing $\mu([s = t]) > \mu(\sigma_1[s_i = t_i])$ for $i = 2, \ldots, m$. Therefore there is some decreasing substitution σ_2 such that $\sigma_1([s_2 = t_2] \ldots [s_m = t_m]Eq, E) \xrightarrow{*} \sigma_2\sigma_1([s_3 = t_3] \ldots [s_m = t_m]Eq, E)$. Iterating the process, we obtain a substitution $\sigma = \sigma_n \circ \cdots \circ \sigma_1$ and lemma 4.1(1) proves that σ is decreasing. $\diamond$

B Proof of theorem 4.6

($\Rightarrow$) $(P,T,E) \xrightarrow{*} (err, T', E')$ for some T', E'. We proceed by induction on the length of the reduction to show that $\exists \sigma \models E \ (\sigma(P,T) \xrightarrow{*} (err, T''))$.

($P \equiv err$) Then $\exists \sigma \models E \ (\sigma(err,T) \equiv (err, \sigma T) \downarrow_* err)$.

($P \equiv !t.P$) Then $(!t.P, T, E) \rightarrow (P, T \cup \{t\}, E)$ by $(!^s)$. By inductive hypothesis, $\exists \sigma \models E \ (\sigma(P, T \cup \{t\}) \downarrow_* err)$ and we note that $\sigma(!t.P, T) \rightarrow \sigma(P, T \cup \{t\})$ since $\sigma t \in \mathcal{M}$.

($P \equiv ?x.P$) Then $(?x.P, T, E) \rightarrow (P, T, E, x : T)$ by $(?^s)$. By inductive hypothesis, $\exists \sigma \models E, x : T \ (\sigma(P,T) \downarrow_* err)$. Let $\sigma' = \sigma[x/x]$. Then $\sigma' \models E$, $\sigma(x) \in S(A(\sigma' T))$, and

$$\sigma'(?x.P, T) \rightarrow ([\sigma(x)/x]\sigma' P, \sigma' T) \equiv \sigma(P,T) \downarrow_* err \ .$$

$(P \equiv \mathsf{let}\ x = t\ \mathsf{in}\ P)$ Then $(\mathsf{let}\ x = t\ \mathsf{in}\ P, T, E) \to ([t/x]P, T, E)$, by (l^s). By inductive hypothesis, $\exists\, \sigma \models E\ \ (\sigma([t/x]P, T) \downarrow_* err)$. We observe that

$$\begin{array}{l}\sigma(\mathsf{let}\ x = t\ \mathsf{in}\ P, T) \equiv (\mathsf{let}\ x = \sigma t\ \mathsf{in}\ \sigma P, \sigma T)\\ \to ([\sigma t/x](\sigma P), \sigma T) \equiv \sigma([t/x]P, T) \downarrow_* err\ .\end{array}$$

$(P \equiv \mathsf{case}\ \langle x', x''\rangle = \langle t_1, t_2\rangle\ \mathsf{in}\ P)$ Then

$$(\mathsf{case}\ \langle x', x''\rangle = \langle t_1, t_2\rangle\ \mathsf{in}\ P, T, E) \to ([t_1/x', t_2/x'']P, T, E)$$

by (c_1^s). By inductive hypothesis:

$$\exists\, \sigma \models E\ \ (\sigma([t_1/x', t_2/x'']P, T) \downarrow_* err)\ .$$

We observe that

$$\begin{array}{l}\sigma(\mathsf{case}\ \langle x', x''\rangle = \langle t_1, t_2\rangle\ \mathsf{in}\ P, T) \equiv (\mathsf{case}\ \langle x', x''\rangle = \langle \sigma t_1, \sigma t_2\rangle\ \mathsf{in}\ \sigma P, \sigma T)\\ \to ([\sigma t_1/x', \sigma t_2/x'']\sigma P, \sigma T) \equiv \sigma([t_1/x', t_2/x'']P, T) \downarrow_* err\ .\end{array}$$

$(P \equiv \mathsf{case}\ \langle x', x''\rangle = x_i\ \mathsf{in}\ P)$ Then

$$(\mathsf{case}\ \langle x', x''\rangle = x_i\ \mathsf{in}\ P, T, E) \to [\langle x', x''\rangle/x_i](P, T, E),$$

by (c_2^s). By inductive hypothesis:

$$\exists\, \sigma \models [\langle x', x''\rangle/x_i]E\ \ (\sigma[\langle x', x''\rangle/x_i](P, T) \downarrow_* err)\ .$$

We define $\sigma' = \sigma[\langle\sigma(x'), \sigma(x'')\rangle/x_i, x'/x', x''/x'']$ and observe that $\sigma' \models [\langle x', x''\rangle/x_i]E$. To see this, we consider three cases:

$(j < i)$ $\sigma'(x_j) = \sigma(x_j) \in S(A(\sigma T_j)) = S(A(\sigma' T_j))$.

$(j = i)$ $\sigma'(x_i) = \langle\sigma(x'), \sigma(x'')\rangle \in S(A(\sigma' T_i)) = S(A(\sigma T_i))$. As $\sigma(x'), \sigma(x'') \in S(A(\sigma T_i))$.

$(j > i)$ $\sigma'(x_j) = \sigma(x_j) \in S(A(\sigma[\langle x', x''\rangle/x_i]T_j)) = S(A(\sigma' T_j))$.

We then observe that:

$$\begin{array}{l}\sigma'(\mathsf{case}\ \langle x', x''\rangle = x_i\ \mathsf{in}\ P, T) \equiv (\mathsf{case}\ \langle x', x''\rangle = \langle \sigma x', \sigma x''\rangle\ \mathsf{in}\ \sigma' P, \sigma' T)\\ \to ([\sigma x'/x', \sigma x''/x''](\sigma' P), \sigma' T) \equiv \sigma[\langle x', x''\rangle/x_i](P, T) \downarrow_* err\ .\end{array}$$

$(P \equiv \mathsf{case}\ E(x, C) = E(t, C)\ \mathsf{in}\ P)$ Then

$$(\mathsf{case}\ E(x, C) = E(t, C)\ \mathsf{in}\ P, T, E) \to ([t/x]P, T, E)$$

by (c_3^s). By inductive hypothesis, $\exists\, \sigma \models E\ \ (\sigma([t/x]P, T) \downarrow_* err)$. We observe that

$$\begin{array}{l}\sigma(\mathsf{case}\ E(x, C) = E(t, C)\ \mathsf{in}\ P, T) \equiv (\mathsf{case}\ E(x, C) = E(\sigma t, C)\ \mathsf{in}\ \sigma P, \sigma T)\\ \to ([\sigma t/x]\sigma P, \sigma T) \equiv \sigma([t/x]P, T) \downarrow_* err\ .\end{array}$$

$(P \equiv \mathsf{case}\ E(x,C) = x_i\ \mathsf{in}\ P,\ C \in K(T_i))$ Then

$$(\mathsf{case}\ E(x,C) = x_i\ \mathsf{in}\ P, E, T) \to [E(x,C)/x_i](P,E,T)$$

by (c_4^s). By inductive hypothesis:

$$\exists\, \sigma \models [E(x,C)/x_i]E\ \ \sigma[E(x,C)/x_i](P,T) \downarrow_* err\ .$$

We define $\sigma' = \sigma[E(\sigma(x),C)/x_i, x/x]$ and remark that $\sigma' \models E$. To show this, we consider three cases:

$(j < i)$ $\sigma'(x_j) = \sigma(x_j) \in S(A(\sigma T_j)) = S(A(\sigma' T_j))$.

$(j = i)$ $\sigma'(x_i) = E(\sigma(x),C) \in S(A(\sigma T_i)) = S(A(\sigma' T_i))$, as $\sigma(x) \in S(A(\sigma T_i))$ and $C \in K(T_i)$.

$(j > i)$ $\sigma'(x_j) = \sigma(x_j) \in S(A(\sigma[E(x,C)/x_i]T_j)) = S(A(\sigma' T_j))$.

Then we observe that:

$$\begin{array}{l}\sigma'(\mathsf{case}\ E(x,C) = x_i\ \mathsf{in}\ P,T) \equiv (\mathsf{case}\ E(x,C) = E(\sigma(x),C)\ \mathsf{in}\ \sigma' P, \sigma' T)\\ \to ([\sigma(x)/x]\sigma' P, \sigma' T) \equiv \sigma[E(x,C)/x_i](P,T) \downarrow_* err\ .\end{array}$$

$(P \equiv \mathsf{case}\ E(x,C) = x_i\ \mathsf{in}\ P,\ E(t,C) \in I_{K(T_i)}(T_i))$ Then $\mathsf{case}\ E(x,C) = x_i\ \mathsf{in}\ P \to [E(t,C)/x_i]([t/x]P, T, E)$, by (c_5^s). By inductive hypothesis

$$\exists\, \sigma \models [E(t,C)/x_i]E\ \ (\sigma[E(t,C)/x_i]([t/x]P,T) \downarrow_* err)\ .$$

We define $\sigma' = \sigma[E(\sigma t, C)/x_i]$ and note that $\sigma' \models [E(t,C)/x_i]E$. To see this, we consider three cases:

$(j < i)$ $\sigma'(x_j) = \sigma(x_j) \in S(A(\sigma T_j)) = S(A(\sigma' T_j))$.

$(j = i)$ $\sigma'(x_i) = E(\sigma t, C) \in S(A(\sigma' T_i))$ as $E(t,C) \in I_{K(T_i)}(T_i)$.

$(j > i)$ $\sigma'(x_j) = \sigma(x_j) \in S(A(\sigma[E(t,C)/x_i]T_j)) = S(A(\sigma' T_j))$.

We observe that

$$\begin{array}{l}\sigma'(\mathsf{case}\ E(x,C) = x_i\ \mathsf{in}\ P,T) \equiv (\mathsf{case}\ E(x,C) = E(\sigma t, C)\ \mathsf{in}\ \sigma' P, \sigma' T)\\ \to ([\sigma t/x]\sigma' P, \sigma' T) \equiv \sigma[E(t,C)/x_i]([t/x]P,T) \downarrow_* err\ .\end{array}$$

$(P \equiv [f(t_1,\ldots,t_n) = f(s_1,\ldots,s_n)]P)$ Then

$$([f(t_1,\ldots,t_n) = f(s_1,\ldots,s_n)]P, T, E) \to ([t_1 = s_1]\ldots[t_n = s_n]P, T, E)$$

by (m_1^s). By inductive hypothesis, $\exists\, \sigma \models E\ \ \sigma([t_1 = s_1]\ldots[t_n = s_n]P,T) \downarrow_* err$. This implies that $\sigma t_i = \sigma s_i$ for $i = 1,\ldots,n$ and $\sigma([t_1 = s_1]\ldots[t_n = s_n]P,T) \xrightarrow{*} \sigma(P,T) \downarrow_* err$. We observe that

$$\sigma([f(t_1,\ldots,t_n) = f(s_1,\ldots,s_n)]P,T) \to \sigma(P,T) \downarrow_* err\ .$$

$(P \equiv [x_i = x_i]P)$ Then $([x_i = x_i]P, T, E) \to (P, T, E)$ by (m_2^s) and by inductive hypothesis, $\exists\, \sigma \models E\ \ (\sigma(P,T) \downarrow_* err)$. We observe that

$$\sigma([x_i = x_i]P, T) \equiv ([\sigma(x_i) = \sigma(x_i)]\sigma P, \sigma T) \to \sigma(P,T) \downarrow_* err\ .$$

$(P \equiv [x_i = x_j]P,\ i < j)$ Then $([x_i = x_j]P, T, E) \to [x_i/x_j](P, T, E)$, by (m_3^s). By inductive hypothesis, $\exists\, \sigma \models [x_i/x_j]E\ \ (\sigma[x_i/x_j](P,T) \downarrow_* err)$. We define $\sigma' = \sigma[\sigma(x_i)/x_j]$ and note that $\sigma' \models E$. This follows, by the usual case analysis where we note in particular that $\sigma'(x_j) = \sigma(x_i) \in S(A(\sigma T_i)) \subseteq S(A(\sigma T_j))$. We observe that

$$\sigma'([x_i = x_j]P, T) \equiv ([\sigma x_i = \sigma x_i]\sigma' P, \sigma' T) \to \sigma'(P,T) \equiv \sigma[x_i/x_j](P,T) \downarrow_* err\ .$$

$(P \equiv [x_i = C]P,\ C \in K(T_i))$ Then $([x_i = C]P, T, E) \to [C/x_i](P, T, E)$ by (m_4^s). By inductive hypothesis, $\exists\, \sigma \models E\ \ (\sigma[C/x_i](P,T) \downarrow_* err)$. We define $\sigma' = \sigma[C/x_i]$ and observe that $\sigma' \models E$ by the usual case analysis, where we note in particular that $\sigma'(x_i) = C \in K(T_i) \subseteq S(A(\sigma' T_i))$. We observe that

$$\sigma'([x_i = C]P, T) \equiv ([C = C]\sigma' P, \sigma' T) \to \sigma'(P,T) \equiv \sigma[C/x_i](P,T) \downarrow_* err\ .$$

$(P \equiv [x_i = \langle t_1, t_2\rangle]P \quad x_i \notin Var(t_i))$ Then

$$([x_i = \langle t_1, t_2\rangle]P, T, E) \to (\mathsf{case}\ \langle x', x''\rangle = x_i\ \mathsf{in}\ [x' = t_1][x'' = t_2]P, T, E)$$

by (m_5^s). By inductive hypothesis

$$\exists\, \sigma \models E\ \ \sigma(\mathsf{case}\ \langle x', x''\rangle = x_i\ \mathsf{in}\ [x' = t_1][x'' = t_2]P, T) \downarrow_* err\ .$$

This implies that $\sigma(x_i) = \langle \sigma t_1, \sigma t_2\rangle$ and $\sigma(P,T) \downarrow_* err$. We observe that

$$\sigma([x_i = \langle t_1, t_2\rangle]P, T) \equiv ([\sigma x_i = \sigma x_i]\sigma P, \sigma T) \to \sigma(P,T) \downarrow_* err\ .$$

$(P \equiv [x_i = E(t,C)]P,\ x_i \notin Var(t))$ Then $([x_i = E(t,C)]P, T, E) \to (\mathsf{case}\ E(x,C) = x_i\ \mathsf{in}\ [x = t]P, T, E)$, by (m_6^s). By inductive hypothesis, $\exists\, \sigma \models E\ \ (\sigma(\mathsf{case}\ E(x,C) = x_i\ \mathsf{in}\ [x = t]P, T) \downarrow_* err)$. This implies that $\sigma(x_i) = E(\sigma t, C)$ and $\sigma(P,T) \downarrow_* err$. We observe that

$$\sigma([x_i = E(t,C)]P, T) \equiv ([\sigma x_i = \sigma x_i]P, \sigma T) \to \sigma(P,T) \downarrow_* err\ .$$

$(\Leftarrow)$ We now turn to the proof of completeness. Suppose $\exists\, \sigma \models E\ \ \sigma(P,T) \downarrow_* err$. We proceed by induction on the length of a reduction to an erroneous state and analysis of the form of σP to show that $(P, T, E) \downarrow_* err$.

$(\sigma P \equiv err)$ Then $P \equiv err$ and by definition $(err, T, E) \downarrow_* err$.

$(\sigma P \equiv 0)$ This case cannot arise.

$(\sigma P \equiv\, !\sigma t.\sigma P)$ Then $\sigma(!t.P, T) \to \sigma(P, T \cup \{t\})$, by $(!)$. By inductive hypothesis, $(P, T \cup \{t\}, E) \downarrow_* err$ and we observe that $(!t.P, T, E) \to (P, T \cup \{t\}, E)$ by $(!^s)$.

($\sigma P \equiv ?x.\sigma P$) Then $\sigma(?x.P,T) \rightarrow ([t/x]\sigma P, \sigma T) \equiv \sigma[t/x](P,T) \downarrow_* err$ by (?) where $t \in S(A(\sigma T))$. We define $\sigma' = \sigma[t/x]$ and observe that $\sigma' \models E, x : T$ and $(?x.P,T,E) \rightarrow (P,T,E,x : T)$ by $(?^s)$. Since $\sigma'(P,T) \equiv \sigma[t/x](P,T) \downarrow_* err$ by inductive hypothesis $(P,T,E,x:T) \downarrow_* err$.

($\sigma P \equiv \mathsf{let}\ x = \sigma t\ \mathsf{in}\ \sigma P$) Then $\sigma(\mathsf{let}\ x = t\ \mathsf{in}\ P,T) \rightarrow ([\sigma t/x]\sigma P, \sigma T)$ by (l). We observe that $(\mathsf{let}\ x = t\ \mathsf{in}\ P,T,E) \rightarrow ([t/x]P,T,E)$, $\sigma[t/x]P \equiv [\sigma t/x]\sigma P$, and that by inductive hypothesis $([t/x]P,T,E) \downarrow_* err$.

($\sigma P \equiv \mathsf{case}\ \langle x', x''\rangle = \sigma t\ \mathsf{in}\ \sigma P$) Then it must be that $\sigma t = \langle t'_1, t'_2\rangle$ and

$$(\mathsf{case}\ \langle x', x''\rangle = \sigma t\ \mathsf{in}\ \sigma P, \sigma T) \rightarrow ([t'_1/x', t'_2/x'']\sigma P, \sigma T) \downarrow_* err\ .$$

Only two cases can arise: (i) $t \equiv \langle t_1, t_2\rangle$ and $\sigma t_i \equiv t'_i$ for $i = 1,2$, (ii) $t \equiv x_i$ and $\sigma(x_i) = \langle t'_1, t'_2\rangle \in S(A(\sigma T_i))$.

(i) We apply rule (c^s_1) to derive

$$(\mathsf{case}\ \langle x', x''\rangle = \langle t_1, t_2\rangle\ \mathsf{in}\ P,T,E) \rightarrow ([t_1/x', t_2/x'']P,T,E)\ .$$

We observe that $\sigma([t_1/x', t_2/x'']P,T) \equiv ([t'_1/x', t'_2/x'']\sigma P, \sigma T) \downarrow_* err$ and by inductive hypothesis we conclude that $([t_1/x', t_2/x'']P,T,E) \downarrow_* err$.

(ii) We apply rule (c^s_2) to derive

$$(\mathsf{case}\ \langle x', x''\rangle = x_i\ \mathsf{in}\ P,T,E) \rightarrow [\langle x', x''\rangle/x_i](P,T,E)\ .$$

We define $\sigma' = \sigma[t'_1/x', t'_2/x'', x_i/x_i]$ and claim that $\sigma' \models [\langle x', x''\rangle/x_i]E$. We observe that $\sigma'[\langle x', x''\rangle/x_i](P,T) \equiv ([t'_1/x', t'_2/x'']\sigma P, \sigma T) \downarrow_* err$, therefore the inductive hypothesis applies and we conclude that $[\langle x', x''\rangle/x_i](P,T,E) \downarrow_* err$.

($\sigma P \equiv \mathsf{case}\ E(x,C) = \sigma t\ \mathsf{in}\ \sigma P$) Then it must be that $\sigma t = E(t',C)$ and $(\mathsf{case}\ E(x,C) = E(t',C)\ \mathsf{in}\ \sigma P, \sigma T) \rightarrow ([t'/x]\sigma P, \sigma T) \downarrow_* err$. Only three cases can arise: (i) $t \equiv E(t'',C)$ and $\sigma t'' \equiv t'$, (ii) $t \equiv x_i$, $C \in K(T_i)$, $\sigma(x_i) = E(t',C)$, $t' \in S(A(\sigma T_i))$, (iii) $t \equiv x_i$, $C \notin K(T_i)$, and $\sigma(x_i) = E(t',C)$.

(i) We apply rule (c^s_3) to derive

$$(\mathsf{case}\ E(x,C) = E(t'',C)\ \mathsf{in}\ P,T,E) \rightarrow ([t''/x]P,T,E)\ .$$

We observe that $\sigma([t''/x]P,T) \equiv ([t'/x]\sigma P, \sigma T) \downarrow_* err$ and by inductive hypothesis we conclude that $([t''/x]P,T,E) \downarrow_*$.

(ii) We apply rule (c^s_4) to derive

$$(\mathsf{case}\ E(x,C) = x_i\ \mathsf{in}\ P,T,E) \rightarrow [E(x,C)/x_i](P,T,E)\ .$$

We define $\sigma' = \sigma[t'/x, x_i/x_i]$ and claim that $\sigma' \models [E(x,C)/x_i]E$. We observe that

$$\sigma'[E(x,C)/x_i](P,T) \equiv ([t'/x]\sigma P, \sigma T) \downarrow_* err\ .$$

$$(([x_i = x_i]P, T, E), \rho, \sigma) \rightarrow \quad ((P, T, E), \rho, \sigma)$$

$$(([x_i = x_j]P, T, E), \rho, \sigma) \rightarrow \quad (\rho'(P, T, E), \rho' \circ \rho, \sigma[x_j/x_j]) \quad \rho' = [x_i/x_j] \quad (i < j)$$

$$(([x_i = C]P, T, E), \rho, \sigma) \rightarrow \quad (\rho'(P, T, E), \rho' \circ \rho, \sigma[x_i/x_i]) \quad \rho' = [C/x_i], \ C \in K(T_i)$$

$$(([f(\vec{t}) = f(\vec{s})]P, T, E), \rho, \sigma) \rightarrow \quad (([t \stackrel{\rightarrow}{=} s]P, T, E), \rho, \sigma)$$

$$\begin{array}{c}(([x_i = \langle t_1, t_2\rangle]P, T, E), \rho, \sigma) \rightarrow \quad (([x_i' = t_1][x_i'' = t_2]\rho' P, \rho' T, \rho' E), \rho' \circ \rho, \sigma') \\ \rho' = [\langle x_i', x_i''\rangle/x_i], \sigma(x_i) = \langle t_1', t_2'\rangle, \sigma' = \sigma[t_1'/x_i', t_2'/x_i'', x_i/x_i]\end{array}$$

$$\begin{array}{c}(([x_i = E(t, C)]P, T, E), \rho, \sigma) \rightarrow \quad (([x_i' = t]\rho' P, \rho' T, \rho' E), \rho' \circ \rho, \sigma[t'/x_i']) \\ C \in K(T_i), \ \ \rho' = [E(x_i', C)/x_i], \sigma(x_i) = E(t', C)\end{array}$$

$$\begin{array}{c}(([x_i = E(t, C)]P, T, E), \rho, \sigma) \rightarrow \quad (([t' = t]\rho' P, \rho' T, \rho' E), \rho' \circ \rho, \sigma[x_i/x_i]) \\ E(t', C) \in I_{K(T_i)}(T_i), \ \ \rho' = [E(t', C)/x_i]\end{array}$$

Figure 8: Rewriting equations

Therefore the inductive hypothesis applies and we conclude that $[E(x, C)/x_i](P, T, E) \downarrow_* err$.

(iii) By lemma 3.4(1), $C \notin K(\sigma T_i)$ and therefore $E(t', C) \in I_{K(T_i)}(\sigma T_i)$. By lemma 3.4(2), $\exists\, E(t'', C) \in I_{K(T_i)}(T_i) \quad (t' = \sigma t'')$. Then by rule (c_5^s) $(\mathsf{case}\ E(x, C) = x_i\ \mathsf{in}\ P, T, E) \rightarrow [E(t'', C)/x_i]([t''/x]P, T, E)$. We define $\sigma' = \sigma[x_i/x_i]$ and claim that $\sigma' \models [E(t'', C)/x_i]E$. We observe that

$$\sigma'[E(t'', C)/x_i]([t''/x]P, T) \equiv ([t'/x]\sigma P, \sigma T) \downarrow_* err$$

therefore the inductive hypothesis applies and we conclude that

$$[E(t'', C)/x_i]([t''/x]P, T, E) \downarrow_* err\ .$$

$(\sigma P \equiv [\sigma t' = \sigma t'']P)$ Then $\sigma([t' = t']P, T) \rightarrow \sigma(P, T) \downarrow_* err$ and t' and t'' must be unifiable. We note that, up to symmetries, these are exactly the cases considered in the rules (m_{1-6}^s).

We introduce in figure 8 yet another variant of the rules for equalities (cf. figures 2 and 3) where we also keep track of a pair of substitutions ρ and σ. We prove by case analysis the following invariant of the reduction system: given a triple $(([s = t][s \stackrel{\rightarrow}{=} t]P, T, E), \rho, \sigma)$ such that (i) $\sigma \models E$, (ii) $\sigma \models [s = t][s \stackrel{\rightarrow}{=} t]$, and (iii) $\sigma \circ \rho = \sigma_0$ we have that

$$(([s = t][s \stackrel{\rightarrow}{=} t]P, T, E), \rho, \sigma) \rightarrow (([s' \stackrel{\rightarrow}{=} t']\rho'[s \stackrel{\rightarrow}{=} t]P, \rho' T, \rho' E), \rho' \circ \rho, \sigma')$$

where (i) $\sigma' \models \rho' E$, (ii) $\sigma' \models [s' \stackrel{\rightarrow}{=} t']\rho'[s \stackrel{\rightarrow}{=} t]$, and (iii) $\sigma' \circ \rho' \circ \rho = \sigma_0$.

We note that the reductions of the first component of the triple correspond to basic symbolic reductions (figure 2) where, as in the termination proof, we combine (m_5^s) with (c_2^s), (m_6^s) with (c_4^s), and (m_6^s) with (c_5^s).

Now consider the triple $(([s = t]P, T, E), id, \sigma)$ where σ plays the role of σ_0. By iterated application of the invariant above we have:

$$(([s = t]P, T, E), id, \sigma) \rightarrow \cdots \rightarrow (([s \stackrel{\rightarrow}{=} t]\rho' P, \rho' T, \rho' E), \rho', \sigma')$$

where (i) $\sigma' \models \rho' E$, (ii) $\sigma' \models [s \stackrel{\rightarrow}{=} t]$, and (iii) $\sigma' \circ \rho' = \sigma$. By the termination lemma 4.2, we eventually reduce to a triple $((\rho'' P, \rho'' T, \rho'' E), \rho'', \sigma'')$ where (i) $\sigma'' \models \rho'' E$, (ii) $\sigma'' \circ \rho'' = \sigma$.

Since $\sigma''(\rho'' P, \rho'' T) \equiv \sigma(P, T) \downarrow_* err$, we can apply the inductive hypothesis and conclude that $\rho''(P, T, E) \downarrow_* err$. $\diamond$

Unité de recherche INRIA Sophia Antipolis
2004, route des Lucioles - B.P. 93 - 06902 Sophia Antipolis Cedex (France)

Unité de recherche INRIA Lorraine : Technopôle de Nancy-Brabois - Campus scientifique
615, rue du Jardin Botanique - B.P. 101 - 54602 Villers lès Nancy Cedex (France)
Unité de recherche INRIA Rennes : IRISA, Campus universitaire de Beaulieu - 35042 Rennes Cedex (France)
Unité de recherche INRIA Rhône-Alpes : 655, avenue de l'Europe - 38330 Montbonnot St Martin (France)
Unité de recherche INRIA Rocquencourt : Domaine de Voluceau - Rocquencourt - B.P. 105 - 78153 Le Chesnay Cedex (France)

Éditeur
INRIA - Domaine de Voluceau - Rocquencourt, B.P. 105 - 78153 Le Chesnay Cedex (France)
http://www.inria.fr
ISSN 0249-6399