Proof Systems for Message–Passing Process Algebras*

M. Hennessy, H. Lin
University of Sussex

Abstract

We give sound and complete proof systems for a variety of bisimulation based equivalences over a message-passing process algebra. The process algebra is a generalisation of *CCS* where the actions consist of receiving and sending messages or data on communication channels; the standard prefixing operator $a.p$ is replaced by the two operators $c?x.p$ and $c!e.p$ and in addition messages can be tested by a conditional construct. The various proof systems are parameterised on auxiliary proof systems for deciding on equalities or more general boolean identities over the expression language for data. The completeness of these proof systems are thus relative to the completeness of the auxiliary proof systems.

1 Introduction

In standard or *pure* process algebras processes are described in terms of their ability to perform atomic unanalysed actions. For example

$$P \Leftarrow a.P + b.c.P$$

describes a process which can continually either perform the action a or the sequence of actions b, c . By a *message–passing process algebra* we mean a process algebra in which these actions are given some structure; namely the reception or emission of data values on communication channels. Thus

$$Q \Leftarrow c?x.\ if\ x \geq 0\ then\ d!x.Q\ else\ c!(x+1).Q$$

describes a process which can cyclically input a value along the channel c and either output it along the channel d unchanged or output its successor along c, depending on whether or not the value concerned is greater than or equal to 0.

The standard approach to providing a semantic basis for these message–passing algebras, advocated for example in [Mil89], is to translate them into an underlying pure algebra. The central feature of this translation, mapping p to $[\![p]\!]$, is that the input expression $c?x.p$ is mapped into the term

$$\sum_{v \in Val} c?v.[\![p[v/x]]\!]$$

where Val is the domain of all data values. Thus the translation of the process Q above is

$$R \Longleftarrow \sum_{n \geq 0} c?n.d!n.R \; + \; \sum_{n<0} c?n.c!(n+1).R$$

This may be taken to be a description in a pure process algebra where we assume that for each channel name c and for every data-value v, in this case every integer, there are atomic actions $c?v$ and $c!v$.

There are two disadvantages in this approach. The first is that descriptions which are in some intuitive sense finite are translated into processes which are inherently infinite, at least if the domain of possible values, Val, is infinite; it is necessary to have in the underlying pure process algebra a summation operator Σ_I where I has the same cardinality as the value domain. Such process algebras are difficult to use. For example the standard algorithms and verification tools, see e.g. [CPS89], do not apply and equational reasoning is difficult since any proof system based on this approach is of necessity infinite. The second disadvantage is that with such translations *uniformities* which exist in the object description disappear in the translation. For example the subsequent behaviour of Q above after the reception of an input v is described functionally by the term $\lambda x.if\ x \geq 0 \rightarrow d!x.Q\ else\ c!(x+1).Q$. this uniform treatment of inputs is not apparent in the translation, R. Although this notion of *uniformity* is difficult to define precisely, it should play a central role in proving properties of message passing systems. The object of this paper is to develop a semantic theory of message–passing processes which takes advantage of this uniformity. In particular our semantic theory will apply directly to the syntax of message–passing processes and will not be mediated by a translation into an infinitary language. As a result the associated proof systems will be in some sense finitary.

Such theories already exist for value–passing processes. In [HIar] a fully-abstract denotational model is presented while in [Hen91] anotdisadv]

[Hen91] we design a proof system whose judgements are *guarded equations* of the form

$$b \rhd t = u$$

where b is a boolean expression and t, u are process terms that may contain free data variables. Semantically this should be read as "under any evaluation of free data variables that satisfies b t is semantically equivalent to u". The completeness of the proof system is thus relative to that for the data domain involved. Moreover rather than getting embroiled in the details of an actual proof system for data expressions we simply assume the existence of some all powerful mechanism for answering arbitrary questions about data. On the one hand this enables us to concentrate on the behaviour of processes and on the other it reflects what would be a reasonable implementation strategy for a proof system based on our results; the main proof system would be based on the proof rules whose applicability is determined by the structure of processes and this main system would periodically call auxiliary proof systems to establish facts about data expressions. A simple example of a proof rule from the main system is

$$\frac{b \rhd t_i = u_i \quad i = 1,2}{b \rhd t_1 + t_2 = u_1 + u_2}$$

while

$$\frac{b \models e = e', \quad b \rhd t = u}{b \rhd c!e.t = c!e'.u}$$

is a rule which depends on a call to

can not be matched by a corresponding $c?$ move from the second. Each of these generalisations of strong bisimulation equivalence has a corresponding "weak" version in which internal moves are abstracted. Thus in all we have four reasonable semantic equivalences and for each of these we present a corresponding proof system. In the strong cases the difference between early and late is manifested in the slightly different methods for inferring identities involving input prefixes; the early equivalence requires a stronger proof rule. On the other hand the weak version of both equivalences can be obtained by adding to the corresponding proof system the standard τ-laws from [Mil89].

The judgements of the proof systems involve *open* process terms, i.e. terms in which data variables need to be instantiated before any operational significance can be associated with them, but the observational equivalences are only defined on closed terms. Therefore in order to even express the soundness and completeness of the proof systems we need to generalise these equivalences to open terms. For each of these semantic equivalences, $\simeq$, we design a proof system with the property that

$$b \rhd t = u \text{ if and only if } t\rho \simeq u\rho \text{ for every evaluation } \rho \text{ satisfying } b.$$

As usual establishing soundness is straightforward but completeness requires some ingenuity. Here we use the approach of [HL92] and introduce symbolic versions of each of the semantic equivalences which are defined directly on open terms. These are expressed in terms of families of relations over open terms parameterised on boolean expressions and we show that, for each semantic equivalence $\simeq$ we consider,

$$t \simeq^b u \text{ if and only if } t\rho \simeq u\rho \text{ for every evaluation } \rho \text{ satisfying } b.$$

Thus soundness and completeness of the proof systems can be established relative to the semantic relations $\simeq^b$. Using this approach the completeness theorems in particular now become "symbolic versions" of the standard completeness theorems of [Mil89], although the details are somewhat more complicated.

We now give a brief outline of the content of the remainder of the paper. In the next section we define the simple language, give it a *concrete* operational semantics and define (early) strong bisimulation. This is followed by a discussion of the proof system for proving processes bisimilar. We state the soundness theorem for the system and indicate the difficulty in proving completeness. In the following section, Section , we define the symbolic semantics and the associated symbolic bisimulation equivalence and prove that it captures precisely the concrete bisimulation equivalence over processes. We then use this result to show the completeness of the proof system.

In Section 4 we repeat these results for *weak* bisimulation equivalence where again it is necessary to develop an appropriate definition of *weak* symbolic equivalence. The following section outlines corresponding results for a *late* operational semantics and considers both the strong and weak cases. We end by discussing briefly how to extend these results to other language constructs.

Related Work: We end this section with a brief discussion of related work. As stated previously the approach we have taken is based on that of [Hen91] where a sound and complete proof system for *testing equivalence* is developed. Here we tackle various bisimulation based equivalences and an essential ingredient of the completeness theorems is the notion of *symbolic bisimulation equivalence*. This has already been used in [HL92]

to develop an algorithm for checking whether two message-passing processes are equivalent and in [HL9] for developing a proof system to verify that such processes satisfy properties described by formulae from a first-order modal logic.

The more standard approach to message-passing processes is to translate them into "pure processes" as outlined at the beginning of this section [Mil89, HR88]. Indeed in [Bur91] a front-end for the Concurrency Workbench is described which translates message-passing processes from a language such as ours into "pure processes" which can be accepted by the Concurrency Workbench and various examples treated using this approach may be found in [Wal89]. However these approaches require the set of values to be finite and even using the boolean value space of two elements leads to an exponential blow-up in the size of descriptions. We hope that

where a ranges over $\{\tau,\ c?v,\ c!v\}$. We use $\sim_e$ to denote the maximal (early) strong bisimulation. This relation generalizes naturally to open terms by letting $t \sim_e u$ iff

EQUIV $\dfrac{}{true \rhd t = t} \quad \dfrac{b \rhd t = u}{b \rhd u = t} \quad \dfrac{b \rhd t = u, u = v}{b \rhd t = v}$

EQN $\dfrac{}{true \rhd t\sigma = u\sigma}$ $t = u$ is an axiom

CONGR $\dfrac{b \rhd t_i = u_i \quad i = 1, 2}{b \rhd t_1 = t_2 \Rightarrow u_1 = u_2}$

S1 $X + nil = X$
S2 $X + X = X$
S $X + Y = Y + X$
S4 $(X + Y) + Z = X + (Y + Z)$

Figure : The Axioms $\mathcal{A}_1$

Proposition 2.3 *Suppose* $x \notin fv(b, c_i, d_j),\ i \in I, j \in J$. *Then from*

$$b \rhd \Sigma_{i\in I} c_i \rightarrow \tau.t_i = \Sigma_{j\in J} d_j \rightarrow \tau.u_j$$

infer

$$b \rhd \Sigma_{i\in I} c_i \rightarrow c?x.t_i = \Sigma_{j\in J} d_j \rightarrow c?x.u_j$$

Proof: For each non-empty $K \subseteq I$ let c_K be the boolean expression $\wedge_{k\in K} c_k \ \wedge \wedge_{k'\in I-K} \neg c_{k'}$. Then $\bigvee c_K = true, c_K \wedge c_{K'} = false$ whenever $K \neq K'$. Using Proposition 2.2 we can show that

$$\vdash_1 \Sigma_{i\in I} c_i \rightarrow \tau.t_i = t^\tau$$

where t^τ denotes $\Sigma_K c_K \rightarrow t^\tau_K$ and t^τ_K denotes $\Sigma_{k\in K} c_k \rightarrow \tau.t_k$. Let $u^\tau = \Sigma_L d_L \rightarrow u^\tau_L$ be defined in a similar manner.

We know $b \rhd t^\tau = u^\tau$ and therefore for each K, L, $b \wedge c_K \wedge d_L \rhd t^\tau = u^\tau$. Again using Proposition 2.2 we can prove

$$b \wedge c_K \wedge d_L \rhd t^\tau = \Sigma_{k\in K} \tau.t_k$$

and

$$b \wedge c_K \wedge d_L \rhd u^\tau = \Sigma_{l\in L} \tau.u_l.$$

Therefore

$$b \wedge c_K \wedge d_L \rhd \Sigma_{k\in K} \tau.t_k = \Sigma_{l\in L} \tau.u_l.$$

Now we can apply E-INPUT to obtain

$$b \wedge c_K \wedge d_L \rhd \Sigma_{k\in K} c?x.t_k = \Sigma_{l\in L} c?x.u_l.$$

By reversing the above argument we have

$$b \wedge c_K \wedge d_L \rhd t^x = u^x$$

where t^x and u^x denote $\Sigma_K c_K \rightarrow \Sigma_{k\in K} c?x.t_k$ and $\Sigma_L d_L \rightarrow \Sigma_{l\in L} c?x.u_l$, respectively. Since $\bigvee_{K,L} c_K \wedge d_L = true$, we can apply CUT to obtain $b \rhd t^x = u^x$. Finally Proposition 2.2 can be used to transform t^x and u^x into the required form. □

The soundness of the $\vdash_1$ is given by the following proposition:

Proposition 2.4 *If* $\vdash_1 b \rhd t = u$ *and* $\rho \models b$ *then* $t\rho \sim_e u\rho$

$\alpha.t \xrightarrow{true,\alpha}_E t$ $\quad \alpha \in \{\, \tau, c!e \mid c \in Chan, e \in Exp \,\}$

$c?x.t \xrightarrow{true,c?y}_E t[y/x]$ $\quad y \notin fv(c?x.t)$

$t \xrightarrow{b',\alpha}_E t'$ implies $b \to t \xrightarrow{b \wedge b',\alpha}_E t'$

$t \xrightarrow{b,\alpha}_E t'$ implies $t + u \xrightarrow{b,\alpha}_E t'$

$(t,u) \in \mathcal{ESB}(\mathbf{S})^b$ if whenever $t \xrightarrow{b_1,\alpha}_E t'$ with $bv(\alpha) \cap fv(b,t,u) = \emptyset$, there is a $b \wedge b_1$-partition B with the property that for each $b' \in B$ there exists a $u \xrightarrow{b_2,\alpha'}_E u'$ such that $b' \models b_2$ and

1. if $\alpha = c!e$ then $\alpha' = c!e'$, $b' \models e = e'$ and $(t',u') \in S^{b'}$
2. otherwise $\alpha = \alpha'$ and $(t',u') \in S^{b'}$

Definition 3.1 (Symbolic Bisimulations)
S is an (early) strong symbolic bisimulation if $\mathbf{S} \subseteq \mathcal{ESB}(\mathbf{S})$, where $\subseteq$ is point-wise inclusion. □

Let $\sim_{\mathbf{E}} = \{\sim_E^b\}$ be the largest (early) strong symbolic bisimulation.

The interest in symbolic bisimulations lies in the fact they are defined with respect to the abstract operational semantics, which for finite terms can be represented as a finite transition graph; in contrast with the standard "concrete" bisimulations are defined over infinite transitions graphs, at least if the set of values is infinite. In [HL92] we give an algorithm for checking for this symbolic equivalence. Here we use it to show completeness of the proof systems. First we relate symbolic and concrete bisimulation equivalence.

Theorem 3.2 *1. (Soundness) $t \sim_E^b u$ implies $t\rho \sim_e u\rho$ for every evaluation ρ such that $\rho \models b$*

2. (Completeness) if $t\rho \sim_e u\rho$ for every evaluation ρ such that $\rho \models b$ then $t \sim_E^b u$

Proof: (Outline) The proof follows the corresponding result in [HL92], Theorem 6.5; it consists in establishing a relationship between symbolic bisimulations and concrete ones. If $\mathbf{S} = \{S^b\}$ is a strong symbolic bisimulation let

$$R_{\mathbf{S}} = \{\, (t\rho, u\rho) \mid \exists b,\ \rho(b) = \ true \ and\ (t,u) \in S^b \,\}$$

Soundness follows immediately if we can prove that $R_{\mathbf{S}}$ is a bisimulation. Conversely if R is a strong bisimulation let

$$S^b = \{\, (t,u) \mid \rho \models b \iff (t\rho, u\rho) \in R \,\}$$

for any boolean expression b. Completeness follows if we can show that $\mathbf{S} = \{S^b\}$ is a symbolic bisimulation.

The proof of these two subsidiary results depends on relating the abstract actions to the concrete actions. We simply state the required relationships and leave the proofs to the reader.

1. $t\rho \xrightarrow{\tau}_e q$ if and only if there exist $b,\ t'$ s.t. $\rho \models b,\ q \equiv_\alpha t'\rho$ and $t \xrightarrow{b,\tau}_E t'$.
2. $t\rho \xrightarrow{c!v}_e q$ if and only if there exist $b,\ e,\ t'$ and ρ s.t. $\rho \models b,\ \rho(e) = v,\ q \equiv_\alpha t'\rho$ and $t \xrightarrow{b,c!e}_E t'$.
3. . $t\rho \xrightarrow{c?v}_e q$ if and only if there exist $b,\ x,\ t'$ and ρ s.t. $x \notin fv(t),\ \rho \models b,\ r \equiv_\alpha t'\rho\{v/x\}$ and t $\xrightarrow{b,c?x}_E t'$.

□

With this theorem we can now return to the proof system and show its completeness by proving

$$t \sim_E^b u \text{ implies } \vdash_1 b \rhd t = u \qquad (*)$$

This provides the converse to Proposition 2.4. The proof of $(*)$ follows the standard proof of the corresponding "concrete" result, as given in [Mil89], except that now we work at the symbolic level.

The proof is by induction on the size of terms which is defined as follows:

1. $\mid nil \mid = 0$
2. $\mid t + u \mid = max\{\mid t \mid, \mid u \mid\}$
3. $\mid b \rightarrow t \mid = \mid t \mid$
4. $\mid \alpha.t \mid = 1 + \mid t \mid$

We also need the notion of normal form:

Definition 3.3 t is a *normal form*, or *in normal form*, if it has the form $\Sigma_i b_i \rightarrow \alpha_i.t_i$ and each t_i is in normal form. □

Lemma 3.4 *For every term t there exists a normal form t' such that $fv(t) = fv(t'), \mid t \mid = \mid t' \mid$ and $\vdash_1 t = t'$.*

Proof: By structural induction on terms using the elementary facts about the proof system given in Proposition 2.2 □

Theorem 3.5 *(Completeness) $t \sim_E^b u$ implies $\vdash_1 b \rhd t = u$*

Proof: By induction on the joint size of t and u. We may assume that both are normal forms, $t \equiv \Sigma_{i \in I} c_i \rightarrow \alpha_i.t_i$ and $u \equiv \Sigma_{j \in J} d_j \rightarrow \beta_j.u_j$. Call a prefix of type $\gamma \in \{\, \tau, c!, c? \mid c \in Chan \,\}$ if it has the form $\tau, c!e, c?x$, respectively. Let $I_\gamma = \{\, i \in I \mid \alpha_i \text{ has type } \gamma \,\}$, $J_\gamma = \{\, j \in J \mid \beta_j \text{ has type } \gamma \,\}$. Let also $t_\gamma = \Sigma_{i \in I_\gamma} c_i \rightarrow \alpha_i.t_i$, $u_\gamma = \Sigma_{j \in J_\gamma} d_j \rightarrow \beta_j.u_j$. We show $b \rhd t_\gamma = u_\gamma$ for each type γ. Clearly $t_\gamma \sim_E^b u_\gamma$. We examine the cases $\gamma = \tau$ and $\gamma = c?$ here and leave the case $\gamma = c!$ to the reader.
(Case $\gamma = \tau$). By symmetry we need only to show

$$b \rhd u_\tau + c_i \rightarrow \tau.t_i = u_\tau.$$

for each $i \in I_\tau$. Note that $b \wedge \neg c_i \rhd c_i \rightarrow \tau.t_i = nil$ so by CUT it is sufficient to show

$$b \wedge c_i \rhd u_\tau + c_i \rightarrow \tau.t_i = u_\tau.$$

Now $t_\tau \xrightarrow{c_i,\tau}_E t_i$. So there exists a $b \wedge c_i$-partition B such that for each $b' \in B$ there is $u_\tau \xrightarrow{d_j,\tau}_E u_j$ such that b

Now let $\mathbf{S} = \{ S^b \mid b \in Exp \}$ be a family of relations over terms, indexed by boolean expressions b. Then $\mathcal{EWB}(\mathbf{S})$ is the family of symmetric relations defined by:

$(t,u) \in \mathcal{EWB}(\mathbf{S})^b$ if whenever $t \xrightarrow{b_1,\alpha}_E t'$ with $bv(\alpha) \cap fv(b,t,u) = \emptyset$, then there is a $b \wedge b_1$-partition B such that for each $b' \in B$ there exists a $u \stackrel{b_2,\hat{\alpha'}}{\Longrightarrow}_E u'$ such that $b' \models b_2$ and

1. if $\alpha \equiv c!e$ then $\alpha' \equiv c!e'$, $b' \models e = e'$ and $(t',u') \in S^{b'}$
2. otherwise $\alpha \equiv \alpha'$ and $(t',u') \in S^{b'}$

Definition 4.2 (Weak Symbolic Bisimulations)
$\mathbf{S}$ is a weak symbolic bisimulation if $\mathbf{S} \subseteq \mathcal{EWB}(\mathbf{S})$ □

Let $\approx_{\mathbf{E}} = \{\approx_E^b\}$ be the largest (early) weak symbolic bisimulation.

The two versions of weak bisimulation can be related as in the case of strong bisimulation.

Theorem 4.3 *$t \approx_E^b u$ if and only if $t\rho \approx_e u\rho$ for every ρ which satisfies b.*

Proof: Similar to that of Theorem .2. □

The aim of this section is to extend the proof system of Section 2 to weak bisimulation equivalence. However it is well-known that $\approx_e$ is not preserved by $+$ and so we have to work with the modified relation:

Definition 4.4 Two closed terms t, u are *early observation equivalent*, written $t \simeq_e u$, if for all $a \in \{\tau, c?v, c!v\}$

- Whenever $t \xrightarrow{a}_e t'$ then u

T1	$\alpha.\tau.X = \alpha.X$
T2	$X + \tau.X = \tau.X$
T	$c!e.(X + \tau.Y) + c!e.Y = c!e.(X + \tau.Y)$

Figure 5: The Axioms $\mathcal{A}_2$

Lemma 4.6 *If* $fv(b) \cap bv(\alpha) = \emptyset$ *then* $\vdash_2 \alpha.(X + b \rightarrow \tau.Y) = \alpha.(X + b \rightarrow \tau.Y) \; + \; b \rightarrow \alpha.Y$

Proof: Since $X = X + b \rightarrow X$ we need only to show $b \rightarrow \alpha.(X + b \rightarrow \tau.Y) = b \rightarrow \alpha.(X + b \rightarrow \tau.Y) \; + \; b \rightarrow \alpha.Y$ which can be derived as follows (using Proposition 2.2):

$$
\begin{array}{lll}
 & b \rightarrow \alpha.(X + b \rightarrow \tau.Y) & () \\
= & b \rightarrow \alpha.(b \rightarrow (X + b \rightarrow \tau.Y)) & (2.2.7) \\
= & b \rightarrow \alpha.(b \rightarrow (X + \tau.Y)) & (2.2.5,1) \\
= & b \rightarrow \alpha.(X + \tau.Y) & (2.2.7) \\
= & b \rightarrow (\alpha.(X + \tau.Y) + \alpha.Y) & (\mathrm{T}\ ) \\
= & b \rightarrow \alpha.(X + \tau.Y) + b \rightarrow \alpha.Y & (2.2.5) \\
= & b \rightarrow \alpha.(X + b \rightarrow \tau.Y) + b \rightarrow \alpha.Y & (\text{previous steps reversed})
\end{array}
$$

□

The main result of this section is

Theorem 4.7 $\vdash_2 b \rhd t = u$ *if and only if* $t\rho \simeq_e u\rho$ *for every* ρ *such that* $\rho \models b$.

The soundness is straightforward, by induction on the length of the proof of $b \rhd t = u$ and we prove the completeness by relying on the symbolic version of weak bisimulation. Again we have to modify it so that it is preserved by $+$:

Definition 4.8 Two terms t, u are *symbolic observation equivalent* with respect to b, written $t \simeq^b_E u$, if whenever $t \xrightarrow{b_1,\alpha}_E t'$ with $bv(\alpha) \cap fv(b,t,u) = \emptyset$, then there is a $b \wedge b_1$-partition B with the following property: for each $b' \in B$ there exists a $u \stackrel{b_2,\alpha'}{\Longrightarrow}_E u'$ such that $b' \models b_2$ and

1. if $\alpha \equiv c!e$ then $\alpha' \equiv c!e'$, $b' \models e = e'$ and $t' \approx^{b'}_E u'$
2. otherwise $\alpha \equiv \alpha'$ and $t' \approx^{b'}_E u'$

and symmetrically for u. □

Proposition 4.9 *The relation* $\simeq^{true}_E$ *is preserved by all the operators in the language.*

Theorem 4.10 $t \simeq^b_E u$ *if and only if* $t\rho$ { \≡→{¬

1. $\|nil\| = 0$
2. $\|\tau.t\| = \|t\|$
3. $\|\alpha.t\| = 1 + \|t\|$ if $\alpha \not\equiv \tau$
4. $\|t + u\| = max\{\|t\|, \|u\|\}$
5. $\|b \rightarrow t\| = \|t\|$

The crucial lemma is the following:

Lemma 4.11 *(Absorption) If* $t \stackrel{b,\alpha}{\Longrightarrow}_E t'$ *with* $fv(b) \cap bv(\alpha) = \emptyset$, *then* $\vdash_2 t = t + b \rightarrow \alpha.t'$.

Proof: By induction on why $t \stackrel{b,\alpha}{\Longrightarrow}_E t'$.

1. $t \stackrel{b,\alpha}{\longrightarrow}_E t'$.
   - $\alpha'.t_1 \stackrel{true,\alpha}{\longrightarrow}_E t'$ with $\alpha'.t_1 \equiv_\alpha \alpha.t'$. Use S .
   - $b' \rightarrow t_1 \stackrel{b' \wedge b'',\alpha}{\longrightarrow}_E t'$ because $t_1 \stackrel{b'',\alpha}{\longrightarrow}_E t'$. By induction $t_1 = t_1 + b'' \rightarrow \alpha.t'$. So

$$\begin{aligned} b' \rightarrow t_1 &= b' \rightarrow (b'' \rightarrow \alpha.t') \\ &= b' \rightarrow t_1 + b' \wedge b'' \rightarrow \alpha.t' \end{aligned}$$

   - The other cases are similar.
2. $t \stackrel{b',\alpha}{\longrightarrow}_E t_1 \stackrel{b'',\tau}{\Longrightarrow}_E t'$ with $b \equiv b' \wedge b''$. By induction $t_1 = t_1 + b'' \rightarrow \tau.t'$ and $t = t + b' \rightarrow \alpha.t_1$. So, since $bv(\alpha$ bv!ff

Lemma 4.13 *For any normal form t there is a full normal form t' such that $fv(t) = fv(t'), \|t\| = \|t'\|$ and $\vdash_2 t = t'$.*

Proof: By structural induction on t using the absorption lemma. □

By this lemma and Lemma .4, every term can be transformed into a full normal form of equal weak size.

The following proposition relates symbolic observation equivalence to weak bisimulation. It will be used in the proof of the completeness theorem.

Proposition 4.14 *$t \approx_E^b u$ if and only if there is a b-partition B such that for all $b' \in B$, $t \simeq_E^{b'} u$ or $t \simeq_E^{b'} \tau.u$ or $\tau.t \simeq_E^{b'} u$*

Proof: The "if" part is trivial. For the "only if" part, because of Lemma .4 we can

To prove $\vdash_2 b \rhd t_{c?} = u_{c?}$ it is sufficient to establish

$$\vdash_2 b \wedge c_i \rhd u_{c?} + c_i \rightarrow c?x_i.t_i = u_{c?}$$

for each $i \in I_{c?}$.

Now $t_{c?} \xrightarrow{c_i, c?z}_E t_i[z/x_i]$, so there is a $b \wedge c_i$-partition B with the property that for each $b' \in B$ there is $u_{c?} \xrightarrow{d_j, c?z}_E u_j[z/y_j] \stackrel{d, \hat{\tau}}{\Longrightarrow}_E u'$ s.t. $b' \models d_j \wedge d'$ and $t_i[z/x_i] \approx^{b'}_E u'$. By Proposition 4.14 there exists a b'-partition B', for each $b'' \in B'$ $t_i[z/x_i] \simeq^{b''}_E u'$ or $t_i[z/x_i] \simeq^{b''}_E \tau.u'$ or $\tau.t_i[z/x_i] \simeq^{b''}_E u'$. By induction, together with TAU and T1, in each case we can derive

$$b'' \rhd \tau.u' = \tau.t_i[z/x_i]$$

Applying CUT on B' we get $b' \rhd \tau.u' = \tau.t_i[z/x_i]$.

If $u_j[z/y_j] \equiv u'$, then $b' \rhd \tau.u_j[z/y_j] = \tau.t_i[z/x_i]$ and hence $b' \rhd \tau.u_j[z/y_j] = \tau.u_j[z/y_j]$7Tj/R1980.t

[

$$
\begin{array}{lll}
a.t \xrightarrow{a}_l t & & a \in \{\tau\} \cup \{\, c!v \mid c \in Chan \,\} \\
c?x.t \xrightarrow{c?x}_l \lambda x.t & & \\
t \xrightarrow{\alpha}_l t' & \text{implies} & t + u \xrightarrow{\alpha}_l t' \\
t \xrightarrow{\alpha}_l t', b = true & \text{implies} & b \to t \xrightarrow{\alpha}_l t'
\end{array}
$$

Figure 6: Late Operational Semantics - closed terms

Definition 5.1 A symmetric relation R

It is important to note that we now require $fv(B) \subseteq fv(b)$; hence when $\alpha \equiv c?x$ it is guaranteed $x \notin fv(B)$. So we can not partition over the value space for an input variable. This makes all the differences between early and late bisimulations!

Late weak symbolic observation equivalence is defined in terms of weak symbolic bisimulation:

Definition 5.4 Two terms t, u are *late weak symbolic observation equivalence* over b, written $t \simeq_L^b u$, if whenever $t \xrightarrow{b_1,\alpha}_L t'$ with $bv(\alpha) \cap fv(b,t,u) = \emptyset$, then there is a $b \wedge b_1$-partition B such that $fv(B) \subseteq fv(b)$ and for each $b' \in B$ there exists a $u \stackrel{b_2,\alpha'}{\Longrightarrow}_L u'$ such that $b' \models b_2$ and

1. if $\alpha \equiv c!e$ then $\alpha' \equiv c!e'$, $b' \models e = e'$ and $t' \approx_L^{b'} u'$
2. if $\alpha \equiv \tau$ then $\alpha' \equiv \tau$ and $t' \approx_L^{b'} u'$
3. if $\alpha \equiv c?x$ then $\alpha' \equiv c?x$ and there is a b'-partition B' s.t for each $b'' \in B'$ there is $u' \stackrel{b'_2}{\Longrightarrow}_L u''$ s.t $b'' \models b'_2$ and $t' \approx_L^{b''} u''$.

and symmetrically for u. □

The late versions of Theorems 4. and 4.10 can be proved similarly as their early counterparts:

Theorem 5.5 *$t \simeq_L^b u$ if and only if $t\rho \simeq_l u\rho$ for every $\rho \models b$.*

The inference system for late symbolic observation equivalence can be obtained by replacing E-INPUT in Figure 2 with the following simpler rule

$$\text{L-INPUT} \qquad \frac{b \rhd t = u}{b \rhd c?x.t = c?x.u} \quad x \notin fv(b)$$

As the inference system is weakened, the τ-law T can no longer be generalised to the case of input prefix. So we have to replace it with

$$\text{T L} \quad \alpha.(X + \tau.Y) + \alpha.Y = \alpha.(X + \tau.Y)$$

Let $\mathcal{A}_{2L}$ be the set of axioms consisting of T1, T2 and T L. We write $\vdash_{2L} b \rhd t = u$ to denote $b \rhd t = u$ can be derived from the new inference system using axioms in $\mathcal{A}_1$ and $\mathcal{A}_{2L}$.

We have the soundness theorem:

Theorem 5.6 *(Soundness) $\vdash_{2L} b \rhd t = u$ implies $t\rho \simeq_l u\rho$ for every ρ such that $\rho \models b$.*

For the completeness theorem, we use essentially the same form of full normal form as in the early case (keep in mind that now double input arrows only absorb those τ moves before it):

Definition 5.7 A normal form $t \equiv \Sigma_i b_i \rightarrow \alpha_i.t_i$ is a late full normal form if

1. $t \stackrel{b,\alpha}{\Longrightarrow}_L t'$ implies $t \xrightarrow{b,\alpha}_L t'$.

2. Each t_i is in late full normal form.

□

The absorption lemma still holds (note that now α can not be an input action in the second case in the proof of the lemma). Every term can be transformed to late normal form and the appropriate version of Proposition 4.14 holds.

Theorem 5.8 *(Completeness)* $t \simeq_L^b u$ *implies* $\vdash_{2L} b \rhd t = u$.

Proof: Assume t, u are in late full normal form and apply induction on the joint weak size of t and u. For the non-trivial case when the size is not 0 let $t \equiv \Sigma_{i\in I} c_i \rightarrow \alpha_i.t_i$, $u \equiv \Sigma_{j\in J} d_j \rightarrow \beta_j.u_j$. We need to show

$$b \wedge c_i \rhd u + c_i \rightarrow \alpha_i.t_i = u$$

for each $i \in I$. We only consider the case when $\alpha_i \equiv c?x$ here (the other two cases are the same as in the early case). Let z be a fresh variable, i.e. $z \notin fv(b, t, u)$.

Now $t \xrightarrow{c_i, c?z}_L t_i'[z/x]$. So there exists a $b \wedge c_i$-partition B with $fv(B) \subseteq fv(b \wedge c_i)$ s.t for all $b' \in B$, $b' \models c_i$ and there is $u \xrightarrow{d_j, c?z}_L u_j[z/y]$ s.t. $b' \models d_j$ and there exists a b'-partition B' s.t for all $b'' \in B'$ there is $u_j[z/y] \stackrel{d', \hat{\tau}}{\Longrightarrow}_L u'$ s.t. $b'' \models d'$ and $t_i[z/x] \approx_L u'$.

By Proposition 4.14 and induction, together with TAU and T1, we can derive

$$b'' \rhd \tau.u' = \tau.t_i[z/x]$$

By an argument similar to that used in Theorem 4.15, using CUT on B', we obtain

$$b' \rhd \tau.u_j[z/y] = \tau.u_j[z/y] + \tau.t_i[z/x].$$

Now, since $z \notin fv(b')$, we can apply L-INPUT to get

$$\begin{aligned} b' \rhd c?z.\tau.u_j[z/y] &= c?z.(\tau.u_j[z/y] + \tau.t_i[z/x]) \\ &= c?z.(\tau.u_j[z/y] + \tau.t_i[z/x]) + c?x.t_i[z/x] \\ &= c?z.\tau.u_j[z/y] + c?z.t_i[z/x] \end{aligned}$$

By T1 and α-CONV, $b' \rhd c?y.u_j = c?y.u_j + c?x.t_i$. Since $b' \models c_i \wedge d_j$, we can derive $b' \rhd d_j \rightarrow c?y.u_j = d_j \rightarrow c?y.u_j + c_i \rightarrow c?x.t_i$. Hence $b' \rhd u = u + c_i \rightarrow c?x.t_i$. Finally, an application of CUT on B gives the required $b \wedge c_i \rhd u = u + c_i \rightarrow c?x.t_i$. □

6 Extensions

So far we have concentrated on the core language of Section 2. As said in the Introduction it can be easily extended by adding the | (parallel) and \ (restriction) operators. The concrete operational semantics for these operators are standard and we only give their symbolic operational semantics (Figure 7, where symmetric rules have been omitted),

$$
\begin{array}{lll}
t \xrightarrow{b,\alpha} t' & \text{implies} & t \mid u \xrightarrow{b,\alpha} t' \mid u \\
 & & \alpha \in \{\, \tau, c!e \mid c \in Chan, e \in Exp \,\} \\
t \xrightarrow{b,c?x} t' & \text{implies} & t \mid u \xrightarrow{b,c?x} t' \mid u \\
 & & x \notin fv(u) \\
t \xrightarrow{b,c?x} t',\ u \xrightarrow{b',c!e} u' & \text{implies} & t \mid u \xrightarrow{b \wedge b',\tau} t'[e/x] \mid u' \\
t \xrightarrow{b,\alpha} t' & \text{implies} & t \backslash c \xrightarrow{b,\alpha} t' \backslash c \\
 & & \text{if } chan(\alpha) \neq c
\end{array}
$$

Figure 7: Symbolic Operational Semantics – continued

followed by the equational laws reducing them to the core language. As the symbolic operational semantics is the same for both early and late cases, the “E1ob6999.7(an(;)-8000L (“E1ob6(subs

References

[Bur91] G. Burns. A language for value-passing CCS.