Unique Fixpoint Induction for Message–Passing Process Calculi*

M. Hennessy & H. Lin†

where b is a boolean expression and

This means that our language for message-passing processes is extended in a functional manner to allow abstractions over data domains, and the application of these abstractions to data expressions. In this extended language Unique Fixpoint Induction is only applied to abstractions and we show that restricted in this manner it is sound. Although this

our approach is to work modulo these evaluations. We also assume, for example, that each expression e has associated with it a set of variables $fv(e)$ such that if ρ and ρ' agree on $fv(e)$ then $\rho(e) = \rho'(e)$. If an expression e has no variables, *i.e.* it is *clo ed*, then $\rho(e)$ is independent of ρ and we use $\llbracket e \rrbracket$ to denote its value. For expressions of type *bool* we use the suggestive notation $b \models b'$ to indicate that for every evaluation if $\rho(b)$ is true then so is $\rho(b')$. As a

$$\begin{array}{lll}
\tau.P \xrightarrow{\tau}_l P & & \\
c!e.P \xrightarrow{c!v}_l P & & \text{where } [\![e]\!] = v \\
c?.F \xrightarrow{c?}_l F & & \\
P \xrightarrow{a}_l P' & \text{implies} &
\end{array}$$

EQUIV $$\frac{}{\vdash_D \mathit{true} \rhd T = T} \qquad \frac{\vdash_D b \rhd T = U}{\vdash_D b \rhd U = T} \qquad \frac{\vdash_D b \rhd T = U \quad \vdash_D b \rhd U = V}{\vdash_D b \rhd T = V}$$

EQN $$\frac{}{\vdash_D T\theta = U\theta} \quad T = U \text{ is an axiom}$$

CONGR $$\frac{\vdash_D b \rhd T_i = U_i \quad i = 1, 2}{\vdash_D b \rhd T_1 + T_2 = U_1 + U_2}$$

α-CONV $$\frac{}{\vdash_D T = U} \quad T \equiv_\alpha U$$

INPUT $$\frac{\vdash_D b \rhd T = U}{\vdash_D b \rhd c?x.T = c?x.U} \quad x \notin fv(b)$$

dec-I $$\frac{\vdash_D b \rhd T = U}{\vdash_{D \cup E} b \rhd T = U}$$

dec-E $$\frac{\vdash_{D \cup E} b \rhd T = U}{\vdash_D b \rhd T = U} \qquad T,\ U \in \mathcal{T}_D$$

UNFOLD $$\frac{}{\vdash_D \ \rhd X = F} \qquad X \Leftarrow F \in D$$

UFI $$\frac{\vdash_D G_i = F_i[\overline{G}/\overline{X}],\ 1 \leq i \leq n}{\vdash_{D \cup E} G_1 = X_1} \qquad \begin{array}{l} E = \{\, X_i \Leftarrow F_i \mid 1 \leq i \leq n \,\} \\ \text{is a guarded declaration} \end{array}$$

λ-I $$\frac{\vdash_D b \rhd Tx = Ux}{\vdash_D b \rhd T = U} \qquad x \notin fv(b, T, U)$$

λ-E $$\frac{b \models e = e',\ \vdash_D b \rhd T = U}{\vdash_D Te = Ue'}$$

β $$\frac{}{\vdash_D \ \rhd (\lambda x T)e = T[e/x]}$$

Figure 4: The New Inference Rules

out, and their elimination, which is allowed provided the definitions being eliminated do not concern abstraction identifiers which occur in the conclusion. This is followed by the UNFOLD rule, also discussed in the Introduction, and a version of Unique Fixpoint Induction. Finally we have very standard rules for the introduction, application and elimination of λ-abstractions and β-reduction.

As with the rules for process manipulation in Figure 3 these rules form a basis for a proof system for manipulating abstractions and recursive definitions, and on top of which more interesting rules can be derived. Two such examples are:

- η $\quad \vdash_D \lambda x(Tx) = T$
- λ-cong $\quad \dfrac{T = U}{\lambda x T = \lambda x U}$

whose derivation we leave to the reader.

It is interesting to re-examine, in the light of these inference rules, the unsound reasoning in the Introduction which leads to the false conclusion

$$\vdash_D x \geq 0 \rhd P(x) = Q(x)$$

where P, Q are defined by

$$P\langle x\rangle \Longleftarrow c!x.c?y.P(y) \quad \text{and} \quad Q\langle x\rangle \Longleftarrow c!|x|c?y.Q(y)$$

where D' contains the definition of A. We can now apply λ-I to reduce this to

$$\vdash_{D'} (\lambda y A(y+1))w = (\lambda y c!(y+1).c?z.A(y+z+1))w,$$

which by β-reduction reduces to

$$\vdash_{D'} A(w+1) = c!(w+1).c?z.A(w+z+1),$$

which follows in a straightforward fashion by an instance of UNFOLD and β-reduction.

This is a somewhat laborious derivation of a relatively simple result but many of the proof steps are trivial applications of β-reduction and λ-introduction and elimination, which can be handled in a semi-automatic way in any implementation of the system. The proof is however complicated by the fact that UFI can only be applied to abstractions, in the sense that one of the terms in the conclusion must be an abstraction identifier. But this restriction can be relaxed a little by using the following derivable proof rule:

If $E = \{ X_i \Leftarrow \lambda \overline{x}_i(b_i \rightarrow T_i) \mid 1 \leq i \leq n \}$ is a guarded declaration then

$$\text{UFI-O} \qquad \frac{\vdash_D b_i \rhd U_i = T_i[\overline{F}/\overline{X}],\ 1 \leq i \leq n}{\vdash_{D \cup E} b_1 \rhd U_1 = X_1(\overline{x}_1)}$$

where $F_i \equiv \lambda \overline{x}_i(b_i \rightarrow U_i),\ 1 \leq i \leq n.$

Here the conclusion can involve an abstraction identifier, X_1 applied to a list of variables, $\overline{x}_1$, which makes the rule much easier to use. In particular when all b_i are *true* then the rule reduces to

$$\text{UFI-O-t} \qquad \frac{\vdash_D U_i = T_i[\overline{F}/\overline{X}],\ 1 \leq i \leq n}{\vdash_{D \cup E} U_1 = X_1(\overline{x}_1)}$$

where $F_i \equiv \lambda \overline{x}_i U_i,\ 1 \leq i \leq n.$

Now revisiting the proof above,

$$A(y+1) = B(y)$$

can be derived directly by one application of UFI-O-t from the judgement

$$\vdash_D A(y+1) = (c!(y+1).c?z.B(y+z))[\lambda y A(y+1)/B],$$

i.e.

$$\vdash_D A(y+1) = (c!(y+1).c?z.A(y+1+z),$$

which follows immediately from UNFOLD and β-reduction.

Pro osition 3.1 *The proof rule UFI-O i derivable.*

Proof: In this proof we assume some familiarity with the capabilities of the basic proof system, that based on the process manipulation rules. All of the properties we require are summarised in Proposition 3.2, stated below.

Suppose

$$\vdash_D b_i \rhd U_i = T_i[\overline{F}/\overline{X}],\ 1 \leq i \leq n.$$

$$\begin{array}{ll} \text{S1} & X + \mathbf{0} = X \\ \text{S2} & X + X = X \\ \text{S3} & X + Y = Y + X \\ \text{S4} & (X+Y)+Z = X+(Y+Z) \end{array}$$

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

Using elementary reasoning, as detailed in the just mentioned Proposition, this means we can infer

$$\vdash_D b_i \to U_i = b_i \to T_i[\overline{F}/\overline{X}],\ 1 \leq i \leq n.$$

By λ-cong we have

$$\vdash_D \lambda\overline{x}_i(b_i \to U_i) = \lambda\overline{x}_i(b_i \to T_i[\overline{F}/\overline{X}]),\ 1 \leq i \leq n.$$

Since $\vdash_D \lambda\overline{x}_i(b_i \to T_i[\overline{F}/\overline{X}]) = \lambda\overline{x}_i(b_i \to T_i)[\overline{F}/\overline{X}]$, applying UFI we obtain

$$\vdash_{D \cup E} \lambda\overline{x}_1(b_1 \to U_1) = X_1.$$

Using λ-E, Proposition 3.2 and λ-cong, we can derive $X_1 = \lambda\overline{x}_1(b_1 \to X_1(\overline{x}_1))$. Hence

$$\vdash_{D \cup E} \lambda\overline{x}_1(b_1 \to U_1) = \lambda\overline{x}_1(b_1 \to X_1(\overline{x}_1))$$

Applying λ-E we obtain

$$\vdash_{D \cup E} b_1 \to U_1 = b_1 \to X_1(\overline{x}_1),$$

which, again using Proposition 3.2, gives

$$\vdash_{D \cup E} b_1 \rhd U_1 = X_1(\overline{x}_1).$$

as required. □

This new form of UFI does make the system easier to use but there is still an apparent restriction to the application of UFI-O because conclusions must involve terms of the form $X(\overline{x})$. As an example of where this might cause problems consider the following definitions:

$$\begin{array}{lcl} A\langle x\rangle & \Leftarrow & c!(3x).A(x+2) \\ B\langle x\rangle & \Leftarrow & c!(2x).B(x+3). \end{array}$$

The two terms $A(2x)$ and $B(3x)$ are semantically equivalent but none of our versions of UFI can be used to directly conclude

$$\vdash A(2x) = B(3x).$$

However the way forward is to introduce a new definition

$$C\langle x\rangle \Leftarrow c!(6x).C(x+1)$$

and to use two applications of UFI-O-t to establish

$$\vdash A(2x) = C(x) \text{ and } \vdash B(3x) = C(x).$$

This is an instance of quite a general strategy which indicates the power of the derived rule UFI-O.

In general the usefulness of our proof system depends on the equations which we apply in the inference rule EQN. At the very least the equations in Figure 5 are necessary; these characterise strong bisimulation equivalence for *CCS*. Let $\vdash^L_D b \rhd T = U$ mean that $\vdash_D b \rhd T = U$ can be derived in the proof system using the equations $\mathcal{A}$ (the superscript L stands for "Late"). The following are some simple yet useful facts about $\vdash^L_D$ whose proofs can be found in [HL95a]:

Pro osition 3.2 *1.* $\vdash^L_D b \rightarrow b' \rightarrow T = b \wedge b' \rightarrow T$

2. $\vdash^L_D T = T + b' \rightarrow T$

3. $b \models b'$ *implie* $\vdash^L_D b \rhd T = b' \rightarrow T$

4. $\vdash^L_D b \wedge b' \rhd T = U$ *implie* $\vdash^L_D b \rhd b' \rightarrow T = b' \rightarrow U$

5. $\vdash^L_D b \rightarrow (T + U) = b \rightarrow T \; + \; b \rightarrow U$

6. $\vdash^L_D b \rightarrow U \; + \; b' \rightarrow U = b \vee b' \rightarrow U$

7. *If* $fv(b) \cap bv(\alpha) = \emptyset$ *then* $\vdash^L_D b \rightarrow \alpha.T = b \rightarrow \alpha.(b \rightarrow T)$

We end this section with another example where in addition the τ-laws of *CCS* and the Expansion Theorem, [Mil89], come into play. We take the version of the expansion theorem from [HL95a] and list it in Figure 6. We will only need the first τ-law in the example to follow:

$$\text{T1} \qquad \alpha.\tau.T = \alpha.T$$

We will also need two sound equations IF-PAR and IF-RES, to distribute the parallel and restriction operators over *if _ then _ el e*

This can be achieved by using IF-PAR and IF-RES, as well as Proposition 3.2, to push the parallel and restriction operators over *if _ then _ el e* in the left hand-side of the equation, followed by three applications of the expansion theorem and T1.

4 Soundness and Completeness

The soundness of the system is relatively straightforward. The only difficulty is the Unique Fixpoint Induction rule whose soundness depends on the following Proposition, a generalisation of Proposition 14, page 104 of [Mil89].

Pro osition 4.1 *Suppo e $\overline{H}$ i a equence of term of arbitrary type which only u e ab traction identifier from $\overline{X}$, and all occurrence of the e identifier are guarded. Let $\overline{F}$ and $\overline{G}$ be equence of data-clo ed term uch that $\overline{F} \sim_l \overline{H}[\overline{F}/\overline{X}]$ and $\overline{G} \sim_l \overline{H}[\overline{G}/\overline{X}]$. Then $\overline{F} \sim_l \overline{G}$.*

Proof: Let

$$R = \{\, (C[\overline{F}/\overline{X}]\rho,\ C[\overline{G}/\overline{X}]\rho) \mid \mathit{Id}(C) \subseteq \overline{X}, C[\overline{F}/\overline{X}], C[\overline{G}/\overline{X}] : \mathit{proce}\ \ \}.$$

First suppose R is a bisimulation. We show that it follows from this that $F_j \sim_l G_j$ for each j. Let the type of X_j be $\iota_1 \rightarrow \ldots \rightarrow \iota_k \rightarrow \mathit{proce}\ \ $. We need to demonstrate that for all $v_i \in \mathit{Val}_{\iota_i}$, $F_j v_1 \ldots v_k \sim_l G_j v_1 \ldots v_k$. Let $C[\ ]$ be the context $X_j z_1 \ldots z_k$, where z_i are fresh variables, and let ρ map z_i to v_i. Then $C[\overline{F}/\overline{X}]\rho$ is $F_j v_1 \ldots v_k$ and $C[\overline{G}/\overline{X}]\rho$ is $G_j v_1 \ldots v_k$.

So it remains to show that R is a (late) bisimulation. For this we shall show R is a *bi imulation up to $\sim_l$* ([Mil89]). By symmetry it is enough to prove

$$C[\overline{F}/\overline{X}]\rho \xrightarrow{a}_l U \text{ implies } C[\overline{G}/\overline{X}]\rho \xrightarrow{a}_l V \text{ with } U \sim_l R \sim_l V$$

For this we apply induction on why $C[\overline{F}/\overline{X}]\rho \xrightarrow{a}_l U$. Consider the possible cases for $C[]$.

- $C \equiv X_i(\overline{e})$. Then $C[\overline{F}/\overline{X}]\rho \equiv F_i\overline{e}\rho \xrightarrow{a}_l U$. Since $F_i \sim_l H_i[\overline{F}/\overline{X}]$, we have $H_i\overline{e}\rho[\overline{F}/\overline{X}] \xrightarrow{a}_l U' \sim_l U$. Since X_i is guarded in H_i, by Lemma 2.1 U' is of the form $C'[\overline{F}/\overline{X}]\rho$ and $H_i\overline{e}\rho[\overline{G}/\overline{X}] \xrightarrow{a}_l C'[\overline{G}/\overline{X}]\rho$. But $C[\overline{G}/\overline{X}]\rho \equiv G_i\overline{e}\rho \sim_l H_i\overline{e}\rho[\overline{G}/\overline{X}]$, so $C[\overline{G}/\overline{X}]\rho \xrightarrow{a}_l V \sim_l C'[\overline{G}/\overline{X}]\rho$. Hence $U \sim_l R \sim_l V$.

- $C_1 \mid C_2$. There are three cases.

  - $C[\overline{F}/\overline{X}]\rho \equiv C_1[\overline{F}/\overline{X}]\rho \mid C_2[\overline{F}/\overline{X}]\rho \xrightarrow{a}_l U$ is because $C_1[\overline{F}/\overline{X}]\rho \xrightarrow{a}_l U'$ with $U \equiv U' \mid C_2[\overline{F}/\overline{X}]\rho$. By induction $C_1[\overline{G}/\overline{X}]\rho \xrightarrow{a}_l V'$ with $U' \sim_l R \sim_l V'$. Hence $C[\overline{G}/\overline{X}]\rho \xrightarrow{a}_l V \equiv V' \mid C_2[\overline{G}/\overline{X}]\rho$, and $U \equiv U' \mid C_2[\overline{F}/\overline{X}]\rho \sim_l R \sim_l V' \mid C_2[\overline{G}/\overline{X}]\rho \equiv V$.

  - $C[\overline{F}/\overline{X}]\rho \xrightarrow{a}_l U$ is because $C_2[\overline{F}/\overline{X}]\rho \xrightarrow{a}_l U'$ with $U \equiv C_1[\overline{F}/\overline{X}]\rho \mid U'$. This case is symmetric to the first case.

  - $C[\overline{F}/\overline{X}]\rho \xrightarrow{\tau}_l U$ is because $C_1[\overline{F}/\overline{X}]\rho \xrightarrow{c?}_l F'$, $C_2[\overline{F}/\overline{X}]\rho \xrightarrow{c!v}_l U'$ and $U \equiv F'v \mid U'$. By induction $C_1[\overline{G}/\overline{X}]\rho \xrightarrow{c?}_l G'$, $C_2[\overline{G}/\overline{X}]\rho \xrightarrow{c!v}_l V'$ with $F'v \sim_l R \sim_l G'v$, $U' \sim_l R \sim_l V'$. Then $C[\overline{G}/\overline{X}]\rho \xrightarrow{\tau}_l G'v \mid V'$ and $F'v \mid U' \sim_l R \sim_l G'v \mid V'$.

The other cases are similar. □

Pro osition 4.2 *(Soundne of $\vdash^L_D$) $\vdash^L_D b \rhd T = U$ implie $T\rho \sim_e U\rho$ for any $\rho \models b$.*

Proof: It is sufficient to show that each axiom in $\mathcal{A}$ is sound and each of the proof rules preserves soundness. We concentrate on UFI.

Suppose $\overline{G} \sim_l \overline{F}[\overline{G}/\overline{X}]$. Directly from the operational semantics we can check that $\overline{X} \sim_l \overline{F}[\overline{X}/\overline{X}]$ and so, since the declaration is guarded, we can immediately apply the previous Proposition to conclude $\overline{G} \sim_l \overline{X}$. □

It is unrealistic to expect that the system is complete. Even pure *CCS*, or our language with a trivial one point message-domain, is Turing complete in the presence of the parallel and restriction operators. However in [Mil84, Mil89, BK88] complete proof systems are obtained for *regular* processes, where action prefixing and choice, $+$, are the only operators allowed in declarations. This leads to the following definition.

Definition 4.3 A declaration

$$D = \{\, X_i \Leftarrow F_i \mid 1 \le i \le n \,\}$$

is called *regular* if the only operators allowed in F_i are

- action prefixing, $c?x.\,\_$, $c!e.\,\_$ and $\tau.\,\_$;
- choice, $\_ + \_$;
- guards, $b \rightarrow \_$.

It is called *re tricted regular* if in addition every occurrence of an abstraction identifier X is of the form $X(\overline{v})$: *proce* , such that each v_i is either a variable or a data constant. A term is called *re tricted regular* if it can be used as part of a restricted regular definition. □

Theorem 4.4 *Let D be a re tricted regular declaration and T, U are re tricted regular term in $\mathcal{T}_D$. If $T\rho \sim_l U\rho$ for every ρ uch that $\rho \models b$, then $\vdash^L_D b \rhd T = U$.* □

This completeness theorem is not true in general for arbitrary unguarded regular declarations; a counter-example can be found in the conclusion. However the question is still open for guarded regular declarations.

The remainder of the section is devoted to proving this result which follows closely the corresponding result in [Mil84], but technically working at a *ymbolic* level.

The first step in the completeness proof is to outline a series of transformations on restricted regular processes which make them easier to handle. We may assume that definitions, and therefore associated terms, are formed by applying prefixing, choice or boolean guard to terms of the form $X(\overline{v})$ or $\mathbf{0}$. Moreover all use of constants can be eliminated by introducing appropriate new abstraction identifiers with fewer parameters. For example the declaration

$$A\langle x\rangle \quad \Leftarrow \quad c?y.x > y \rightarrow c!y.A(0) \; + y > x \rightarrow c!y.A(y)$$

can be replaced by

$$\begin{array}{rcl} A\langle x\rangle & \Leftarrow & c?y.x > y \rightarrow c!y.B \; + \; y > x \rightarrow c!y.A(y) \\ B & \Leftarrow & c?y.0 > y \rightarrow c!y.B \; + \; y > 0 \rightarrow c!y.A(y) \end{array}$$

without affecting the provability relation between the original terms; these are called *equivalent definition* , which is clarified below. The same technique may be used to eliminate multiple occurrences of the same data

$$
\begin{array}{llll}
\alpha.T \xrightarrow{true,\alpha} T & & & \alpha \in \{\, \tau, c!e \mid c \in \ Chan, e \in \ Exp \} \\
c?x.T \xrightarrow{true,c?y} T[y/x] & & & y \notin fv(c?x.T) \\
T \xrightarrow{b',\alpha} T' & \text{implies} & & b \rightarrow T \xrightarrow{b \wedge b',\alpha} T' \\
T \xrightarrow{b,\alpha} T' & \text{implies} & & T + U \xrightarrow{b,\alpha} T' \\
 & & & U + T \xrightarrow{b,\alpha} T' \\
U[\overline{x}/\overline{z}] \xrightarrow{b,\alpha} T' & \text{implies} & & X(\overline{x}) \xrightarrow{b,\alpha} T' \\
 & & & X \Leftarrow \lambda\overline{z}U \text{ is a definition}
\end{array}
$$

Figure 7: Symbolic Operational Semantics

Definition 4.5 A restricted regular declaration $\{\, X_i\langle\overline{x}_i\rangle \Leftarrow T_i \mid 1 \leq i \leq n \,\}$ is called *uniform* if all the $\overline{x}_i$ are of the same length and type. □

Proposition 4.6 *Every restricted regular declaration can be transformed into a uniform declaration.*

Proof: By systematic application of the above transformations. □

So for the rest of this section we can assume that we are working with respect to a uniform declaration, using a fixed sequence of variables $z_1, \ldots z_n$. For terms with respect to these kinds of definitions it is straightforward to develop a version of *symbolic semantics* as defined in [HL95a, HL95b], to which we refer the reader for details. The *symbolic operational semantics* is given in Figure 7 which uses *abstract actions* of the form $\{c?x, c!e, \tau\}$. Based on these relations we define *symbolic bisimulations*, which requires some auxiliary notation.

Proof: Following the lines in the proofs of **Theorem 4.5** in [HL95b] and **Theorem 3.6** in [HL95a]. □

It can be seen from the above theorem that the free data variables appearing in $T \sim_L^b U$ is interpreted *univer ally*. This fact is stated in the following proposition which can be easily proved using the theorem.

Pro osition 4.9 *$T \sim_L^b U$ implie $T\sigma \sim_L^{b\sigma} U\sigma$ for any data ub titution σ.*

If $T \sim_L^b U$ then the definition of symbolic bisimulation requires a boolean partition for each symbolic transition from T or U. As there are only finite many such transitions (modulo α-equivalence), it is possible to find a "uniform" partition which works for *all* symbolic transitions from T or U. Here we show a slightly weaker result: There exists a "uniform" partition from all transitions from T or U which have the same type, as this is sufficient for our purpose. Symbolic actions α, β are of the same type, if either $\alpha \equiv \beta \equiv \tau$, or $\alpha \equiv \beta \equiv c?x$ for some x, or α has the form $c!e$ and β has the form $c!e'$.

Lemma 4.10 *Suppo e $T \equiv \sum_{i \in I} \alpha_i.T_i$, $U \equiv \sum_{j \in J} \beta_j.U_j$, where all α_i and β_j are of the ame type and $bv(\alpha_i) \cap bv(\beta_j) \cap fv(b, T, U) = \emptyset$. Then $T \sim_L^b U$ iff there exi t a b-partition B with $fv(B) \subseteq fv(b, T, U)$ uch that for each $b' \in B$ the following hold*

- *For each $i \in I$ there i a $j \in J$ .t. $\alpha_i =^{b'} \beta_j$ and $T_i \sim_L^{b'} U_j$.*
- *For each $j \in J$ there i an $i \in I$ .t. $\alpha_i =^{b'} \beta_j$ and $T_i \sim_L^{b'} U_j$.*

Proof: Since $T \sim_L^b U$, for each $i \in I$ there exists a b-partition B_i with $fv(B_i) \subseteq fv(T, U)$ such that for each $b_i \in B_i$, $\exists j \; \alpha_i =^{b_i} \beta_j$ and $T_i \sim_L^{b_i} U_j$; for each $j \in J$ there exists a b-partition B'_j with $fv(B'_j) \subseteq fv(b, T, U)$ such that for each $b'_j \in B'_j$, $\exists i \; \alpha_i =^{b_i} \beta_j$ and $T_i \sim_L^{b_i} U_j$.

Let D_I denote the set of booleans $\{ \wedge_{i \in I} b_i \mid b_i \in B_i \}$, D_J the corresponding set $\{ \wedge_{j \in J} b'_j \mid b'_j \in B_j \}$ and let B be the set $\{ b_1 \wedge b_2 \mid b_1 \in D_I, \, b_2 \in D_J \}$. Then $\bigvee B = b$, $fv(B) \subseteq fv(b, T, U)$ and each $b' \in B$ has the form $(\wedge_i b_i) \wedge (\wedge_j b'_j)$ with $b_i \in B_i$, $b'_j \in B'_j$. For each $i \in I$ $b' \models b_i$ for some $b_i \in B_i$, so there is a $j \in B'_j$ s.t. $\alpha_i =^{b'} \beta_j$ and $T_i \sim_L^{b'} U_j$. For each $j \in J$ $b' \models b'_j$ for some $b'_j \in B'_j$, so there is an $i \in B_i$ s.t. $\alpha_i =^{b'} \beta_j$ and $T_i \sim_L^{b'} U_j$. □

Remark 4.11 *The boolean in partition B in the above lemma can be made di joint a follow : Suppo e $B = \{ b_i \mid 1 \leq i \leq n \}$. Set $B' = \{ b'_i \mid 1 \leq i \leq n \}$ with $b'_i = b_i \wedge \bigwedge_{1 \leq j < i} \neg b_j$. It i ea y to check that $\bigvee B' = \bigvee B$, $b'_i \wedge b'_j = $ fal e for any $i \neq j$, and B' enjoy the ame property of B mentioned in the lemma.*

From The6remsymTf12.0:24Tf0m.33(erty)-1Tf11.28010Td[(J)-15999.7(b)]TJ/R2660.24ekf(the)Tj190. *mentioned in 5* b

Definition 4.12 A uniform declaration $D = \{X_i\langle\overline{x}_i\rangle \Leftarrow T_i\}_{i\in I}$ is *tandard* if each T_i has the form

$$\sum_{k\in K_i} b_{ik} \rightarrow \sum_{p\in P_{ik}} \alpha_{ikp}.X$$

As an illustrative example let us apply this procedure to each of the definitions in D'' above. For convenience we ignore resulting definitions where the body is guarded by the boolean *false*. The first definition $Z_1\langle y, x\rangle$ remains essentially the same, giving rise to

$$X_1\langle y, x\rangle \Leftarrow \quad c?x.X_2(y, x) + c!y.X_1(y, x)$$

while the second, $Z_2\langle y, x\rangle$, gives rise to:

$$\begin{array}{rl} X_2\langle y, x\rangle \Leftarrow & x \geq y \wedge x \geq 0 \rightarrow (d!x.X_1(x, y) + d!(x-1).\mathbf{0}) + \\ & \neg x \geq 0 \rightarrow d!x.X_1(x, y) + \\ & \neg x \geq y \rightarrow \mathbf{0} \end{array}$$

For an arbitrary $T \in \mathcal{T}_D$ with $fv(T) \subseteq \overline{w}$, we can first eliminate any unguarded identifier occurrences in T by unfolding them, obtaining a term T' provably equal to T. We then add a definition $X_0 \Leftarrow \lambda \overline{w} T'$ to D, and finally transform it to standard form. □

We say two vectors of variables $\overline{x} = x_1 x_2 \ldots x_n$ and $\overline{x}' = x'_1 x'_2 \ldots x'_n$ *differ at most at* a (possibly empty) set of variables V, if $x_i = x'_i$ for any $1 \leq i \leq n$ such that $x_i \notin V$. A standard declaration $D = \{X_i\langle \overline{x}_i\rangle \Leftarrow T_i\}_{i \in I}$, where $T_i \equiv \sum_{k \in K_i} b_{ik} \rightarrow \sum_{p \in P_{ik}} \alpha_{ikp}.X_{f(i,k,p)}(\overline{x}_{ikp})$, is *parameter-saturated* if $\overline{x}_{ikp}$ and $\overline{x}_{f(i,k,p)}$ differ at most at $bv(\alpha_{ikp})$ for every i, k, p.

Proposition 4.14 *Every standard declaration $D = \{X_i\langle \overline{x}_i\rangle \Leftarrow T_i\}_{i \in I}$ can be transformed into an equivalent standard and parameter-saturated declaration.*

Proof: For each i and each permutation $\overline{x}'$ of $\overline{x}_i$ add a definition $X'\langle \overline{x}'\rangle \Leftarrow T_i[\overline{x}'/\overline{x}_i]$ into D, and replace each occurrence of $X_i(\overline{x}')$ by $X'(\overline{x}')$ in the enlarged declaration. Let the

Let $b_{ikjl} = c_{ik} \wedge d_{jl} \wedge b_{ij}$. Since $X_i(\bar{}$

where $\theta \equiv [\lambda\overline{z}(b_{ij} \rightarrow X_i(\overline{z}))/Z_{ij}|i,j]$. If this can be done then by UFI-O we obtain the required

$$\vdash^L_{D_1 \cup E} b_{11} \rhd X_1(\overline{z}) = Z_{11}(\overline{z}).$$

By Proposition 3.2, (1) is equivalent to

$$\vdash^L_{D_1} b_{ij} \rhd b_{ij} \rightarrow X_i(\overline{z}) = \sum_{k,l} c_{ik} \wedge d_{jl} \wedge b_{ij} \rightarrow (V^\tau + \sum_c V^{c!} + \sum_c V^{c?})\theta. \tag{2}$$

Since $\bigvee_{k,l}(c_{ik} \wedge d_{jl}) = (\bigvee_k c_{ik}) \wedge (\bigvee_l d_{jl}) = true$,

$$\vdash^L_{D_1} b_{ij} \rhd b_{ij} \rightarrow X_i(\overline{z}) = \sum_{k,l} c_{ik} \wedge d_{jl} \wedge b_{ij} \rightarrow T_{ik}.$$

So (2), hence (1), will hold if we can show

$$\vdash^L_{D_1} b_{ij} \rhd \sum_{k,l} b_{ikjl} \rightarrow T_{ik} = \sum_{k,l} b_{ikjl} \rightarrow (V^\tau + \sum_c V^{c!} + \sum_c V^{c?})\theta.$$

Since both $\{\, c_{ik} \mid k \,\}$ and $\{\, d_{jl} \mid l \,\}$ are sets of disjoint booleans, this reduces to: for each k, l

$$\vdash^L_{D_1} b_{ikjl} \rhd T_{ik} = (V^\tau + \sum_c V^{c!} + \sum_c V^{c?})\theta$$

which further reduces to

$$\vdash^L_{D_1} b_{ikjl} \rhd T^\tau = V^\tau\theta \tag{3}$$
$$\vdash^L_{D_1} b_{ikjl} \rhd T^{c!} = V^{c!}\theta \tag{4}$$
$$\vdash^L_{D_1} b_{ikjl} \rhd T^{c?} = V^{c?}\theta \tag{5}$$

for each $c \in Chan(T_{ik})$.

We first consider (5).

Now

$$V^{c?}\theta \equiv \sum_{b' \in B^{c?}} b' \rightarrow V^{c?}_{b'}\theta. \tag{6}$$

By the construction of $V^{c?}_{b'}$, for each $(p,q) \in I^{c?}_{b'}$ it holds that

$$X_{f(i,k,p)}(\overline{z}_{ikpjlq}) \sim^{b'}_L Y_{g(j,l,q)}(\overline{z}_{ikpjlq}).$$

By Proposition 4.9

$$X_{f(i,k,p)}(\overline{z}) \sim^{b'[\overline{z}/\overline{z}_{ikpjlq}]}_L Y_{g(j,l,q)}(\overline{z}).$$

By the definition of b_{ij}

$$b'[\overline{z}/\overline{z}_{ikpjlq}] \models b_{f(i,k,p)g(j,l,q)}.$$

Hence

$$b' \models b_{f(i,k,p)g(j,l,q)}[\overline{z}_{ikpjlq}/\overline{z}]. \tag{7}$$

Therefore

$$\begin{aligned}
\vdash^L_{D_1} b' \rhd\ & V^{c?}_{b'}\theta \\
= & \ (\sum_{(p,q)\in I^{c?}_{b'}} c?z.Z_{f(i,k,p)g(j,l,q)}(\overline{z}_{ikpjlq}))\theta \\
= & \sum_{(p,q)\in I^{c?}_{b'}} c?z.(b_{f(i,k,p)g(j,l,q)}[\overline{z}_{ikpjlq}/\overline{z}] \rightarrow X_{f(i,k,p)}(\overline{z}_{ikpjlq})) \\
= & \sum_{(p,q)\in I^{c?}_{b'}} b' \rightarrow c?z.(b_{f(i,k,p)g(j,l,q)}[\overline{z}_{ikpjlq}/\overline{z}] \rightarrow X_{f(i,k,p)}(\overline{z}_{ikpjlq})) \\
= & \sum_{(p,q)\in I^{c?}_{b'}} b' \rightarrow c?z.(b' \rightarrow b_{f(i,k,p)g(j,l,q)}[\overline{z}_{ikpjlq}/\overline{z}] \rightarrow X_{f(i,k,p)}(\overline{z}_{ikpjlq})) \\
\overset{(7)}{=} & \sum_{(p,q)\in I^{c?}_{b'}} b' \rightarrow c?z.(b' \rightarrow X_{f(i,k,p)}(\overline{z}_{ikpjlq})) \\
= & \sum_{(p,q)\in I^{c?}_{b'}} b' \rightarrow c?z.X_{f(i,k,p)}(\overline{z}_{ikpjlq}) \\
= & \sum_{(p,q)\in I^{c?}_{b'}} c?z.X_{f(i,k,p)}(\overline{z}_{ikpjlq}).
\end{aligned}$$

Since $I^{c?}_{b'}$ is total $T^{c?}$ can be obtained by duplicating and reordering the summands of the last line above, using $S2$, $S3$, $S4$. This means

$$\vdash^L_{D_1} b' \rhd V^{c?}_{b'}\theta = T^{c?}. \tag{8}$$

Therefore

$$\begin{aligned}
\vdash^L_{D_1} & \ V^{c?}\theta \\
\overset{(6)}{=} & \sum_{b'\in B^{c?}} b' \rightarrow V^{c?}_{b'}\theta \\
\overset{(8)}{=} & \sum_{b'\in B^{c?}} b' \rightarrow T^{c?} \\
\overset{3.2}{=} & \ b_{ikjl} \rightarrow T^{c?},
\end{aligned}$$

Using Proposition 3.2 again we then obtain

$$\vdash^L_{D_1} b_{ikjl} \rhd V^{c?}\theta = T^{c?},$$

which is the required (5) above. The proofs for (3) and (4) are similar.

This completes the proof of (1). In a symmetric way we can prove $\vdash^L_{D_2\cup E} b \rhd Y_1(\overline{z}) = Z_{11}(\overline{z})$. □

Theorem 4.16 *(Completene of* $\vdash^L$*) Let* T*,* $U \in \mathcal{T}_D$*,* D *a guarded declaration. Then* $T \sim^b_L U$ *e2339(Td(S)TjR132.24Tf7.91992Td(3)TjR1932.2)Tj8249.96.24T.24Tf22.8-1.688Td())Tj-171.3*

ThDTdTTjRTeTdDTjRTfTdoXTdETclarredTdTjGTfHTjdTjdTjRTfTdTdTTjRRTf

Hence $\vdash^L$

Theorem 5.3 *(Soundne and completene of* $\sim_E$*)* $T \sim_E^b U$ *iff* $T\rho$

We argue as follows: we know

$$T^{c?} \equiv \sum_{\{\, \alpha_{ikp} \equiv c?z | p \in P_{ik} \,\}} \alpha_{ikp}.X_{f(i,k,p)}(\overline{z}_{ikpjlq})$$

and we have (7) by the same argument as in the proof of Proposition 4.15. Hence

$$\begin{aligned}
\vdash^E_{D_1} & \; V_1^{c?}\theta \\
&= (\sum_{\alpha_{ikp} \equiv c?z} \alpha_{ikp}. \sum_{\substack{b' \in B^{c?} \\ (p,q) \in I^{c?}_{b'}}} b' \to Z_{f(i,k,p)g(j,l,q)}(\overline{z}_{ikpjlq}))\theta \\
&= \sum_{\alpha_{ikp} \equiv c?z} \alpha_{ikp}. \sum_{\substack{b' \in B^{c?} \\ (p,q) \in I^{c?}_{b'}}} b' \to (b_{f(i,k,p)g(j,l,q)}[\overline{z}_{ikpjlq}/\overline{z}_{f(i,k,p)g(j,l,q)}] \to X_{f(i,k,p)}(\overline{z}_{ikpjlq})) \\
&\overset{(7)}{=} \sum_{\alpha_{ikp} \equiv c?z} \alpha_{ikp}. \sum_{\substack{b' \in B^{c?} \\ (p,q) \in I^{c?}_{b'}}} b' \to X_{f(i,k,p)}(\overline{z}_{ikpjlq}) \\
&= \sum_{\alpha_{ikp} \equiv c?z} \alpha_{ikp}.(b_{ikjl} \to X_{f(i,k,p)}(\overline{z}_{ikpjlq}))
\end{aligned}$$

The last step of the above derivation uses the fact that $I^{c?}_{b'}$ is total.

Similarly we can derive

$$\vdash^E_{D_1} V_2^{c?}\theta = \sum_{\beta_{jlq} \equiv c?z} \beta_{jlq}. \sum_{\substack{b' \in B^{c?} \\ (p,q) \in I^{c?}_{b'}}} b' \to X_{f(i,k,p)}(\overline{z}_{ikpjlq})$$

Now for each q such that $\beta_{jlq} \equiv c?z$, we know that $\bigvee B^{c?} = b_{ikjl}$. Also by Remark 4.11 we may assume the booleans in $B^{c?}$ are mutual disjoint. And, finally, we know $I^{c?}_{b'}$ is surjective. So we can apply Proposition 5.4 to obtain

$$\begin{aligned}
\vdash^E_{D_1} b_{ikjl} \rhd & \sum_{\alpha_{ikp} \equiv c?z} \alpha_{ikp}.X_{f(i,k,p)}(\overline{z}_{ikpjlq}) = \\
& \sum_{\alpha_{ikp} \equiv c?z} \alpha_{ikp}.X_{f(i,k,p)}(\overline{z}_{ikpjlq}) + \beta_{jlq}. \sum_{\substack{b' \in B^{c?} \\ (p,q) \in I^{c?}_{b'}}} b' \to X_{f(i,k,p)}(\overline{z}_{ikpjlq})
\end{aligned}$$

Repeating this process for each q we obtain

$$\vdash^E_{D_1} b_{ikjl} \rhd V_1^{c?}\theta = V_1^{c?}\theta + V_2^{c?}\theta$$

Because $z \notin fv(b_{ikjl})$, from this (5) follows immediately:

$$\begin{aligned}
\vdash^E_{D_1} b_{ikjl} \rhd V^{c?}\theta &= \sum_{\alpha_{ikp} \equiv c?z} \alpha_{ikp}.(b_{ikjl} \to X_{f(i,k,p)}(\overline{z}_{ikpjlq})) \\
&= \sum_{\alpha_{ikp} \equiv c?z} \alpha_{ikp}.X_{f(i,k,p)}(\overline{z}_{ikpjlq}) \\
&= T^{c?}
\end{aligned}$$

This completes the proof for the early version of Proposition 4.15, thus giving the completeness result for the early case:

Theorem 5.5

$[\overline{x}'/$