

1 SOS rules for top level CIF models

To prove the main theorem we need to define the semantics of translatable CIF compositions. In CIF [1] top level declarations of models are not part of the core language (whose semantics is defined by means of SOS rules). These declarations are considered to be *modeling extensions*, and their semantics is given in terms of a syntactic transformation to the core language. However, for carrying our the proofs, it is more convenient to give SOS rules directly for the modeling extensions. Later on we give a lemma that shows that the rules presented here generates isomorphic transition relations.

We use notation $M_{p,u}$ to refer to the CIF composition:

```
[[ disc control  $xs$ 
; disc  $ys$ 
; clock  $ys$ 
; init  $u$ 
; urgent  $as$ 
::  $p$  ]]
```

For action transitions we define the following rule.

$$\frac{(p, \sigma) \xrightarrow{a,b,(xs \cup cs)} (p', \sigma'), \sigma \models u}{(M_{p,u}, \sigma) \xrightarrow{a,b,\emptyset} (M_{p',\text{true}}, \sigma')} \quad 1$$

For time transitions we define the following rule.

$$\frac{\begin{array}{l} (p, \sigma) \xrightarrow{\rho,A,\theta} (p', \sigma'), \\ \langle \forall s : s \in \text{dom}(\theta) : as \cap \theta(s) = \emptyset \rangle, \\ \langle \forall x : x \in xs \cup ys : (\rho_x, \rho_{\bar{x}}) \in G_{disc} \rangle, \\ \langle \forall c : c \in cs : (\rho_c, \rho_{\bar{c}}) \in G_{clock} \rangle, \\ \sigma \models u \end{array}}{(M_{p,u}, \sigma) \xrightarrow{\rho,A,\theta} (M_{p',\text{true}}, \sigma')} \quad 2$$

The rule for environment transitions is not relevant in the context of the present work, but it is needed to prove Lemma 2. Thus we present it next.

$$\frac{\begin{array}{l} (p, \sigma) \xrightarrow{A} (p', \sigma'), \\ \langle \forall x : x \in cs \cup xs : \sigma(x) = \sigma'(x) \rangle, \\ \sigma \models u \end{array}}{(M_{p,u}, \sigma) \xrightarrow{A} (M_{p',\text{true}}, \sigma')} \quad 3$$

2 Additional notations

Given a composition p , function application $actv(p)$ returns a sequence $xs++[l_u]$, where xs is the sequence of all the active locations of the automata contained in p , and l_u is the unique location of the receptive automaton.

In what follows, given a list xs we use also notation xs as a shorthand for $\text{ran}(xs)$ when no confusion arises.

Given a list of variables $xs = [x_0, \dots, x_{n-1}]$ and a list of expressions $es = [e_0, \dots, e_{n-1}]$, we use notation $xs = es$ to refer to the predicate

$$\langle \forall i : 0 \leq i < n : x_i = e_i \rangle$$

Similarly, $xs^+ = es$ denotes the predicate

$$\langle \forall i : 0 \leq i < n : x_i^+ = e_i \rangle$$

Making some abuse of notation, we use expression $xs = es$ to refer to the sequence of assignments

$$[x_0 = e_0, \dots, x_{n-1} = e_{n-1}]$$

and expression $xs = es^-$ to refer to the sequence of assignments

$$[x_0 = e_0^-, \dots, x_{n-1} = e_{n-1}^-]$$

In the present work it is clear when one or the other is intended.

Given a valuation σ , valuation σ^+ is defined as the set $\{(x^+, v) \mid (x, v) \in \sigma\}$.

3 Additional properties

In this section we present additional properties needed for proving the main theorem.

Lemma 1 (Translatable updates and assignments). *For any translatable update of the form $(zs, zs^+ = es)$, for any valuations σ, σ' , such that $\text{dom}(\sigma) = \text{dom}(\sigma')$ and $\langle \forall x : x \in \text{dom}(\sigma) \setminus zs : \sigma(x) = \sigma'(x) \rangle$, we have that*

$$\sigma'^+ \cup \sigma \models zs^+ = es \Leftrightarrow (zs = es^-)(\sigma) = \sigma'$$

The following lemma states that the SOS rules for the modeling extensions are consistent with the syntactic transformation function \mathcal{T}_s for modeling extensions.

Lemma 2. *Let \mathcal{T}_s be the function that translate modeling extensions to core CIF compositions, then for every modeling extension p , there is a transition*

$$(p, \sigma) \twoheadrightarrow (p', \sigma')$$

if and only if there is a transition

$$(\mathcal{T}_s(p), \sigma) \twoheadrightarrow (\mathcal{T}_s(p'), \sigma')$$

where \twoheadrightarrow is either an action, time, or environment transition.

In this work we use two functions, both named $actv()$, to extract the sequence of active locations of an CIF model, and from a UPPAAL network of automata. It is not hard to see that when the $actv()$ function is applied to a translated model p_0 , the result is the same as applying the function directly to p_0 . This is expressed in the following lemma.

Lemma 3. *For any translatable model p_0 , the following holds:*

$$actv(\mathcal{T}(p_0)) = actv(p_0).$$

4 The proof of the theorem

Let p_0 be a translatable composition. Then p_0 is of the form

```
[[ disc control  $xs$ 
; disc  $ys$ 
; clock  $ys$ 
; init  $u$ 
; urgent  $as$ 
::  $\alpha_{0_0} \parallel \dots \parallel \alpha_{0_{n-1}}$  ]]
```

We have to show $p_0 \Leftrightarrow \mathcal{T}(p_0)$. For this we need to find a bisimulation relation R such that:

$$(p_0, actv(\mathcal{T}(p_0)))$$

Or equivalently, by definition of function $actv()$ for CIF compositions, and by definition of \mathcal{T} :

$$(p_0, actv(p_0))$$

We define relation R as follows.

$$R \triangleq \{(p_0, actv(p_0))\} \cup \{(p, actv(p)) \mid (\exists \sigma, \sigma', p_0, \rho, A, \theta :: \llbracket p_0 \rrbracket \vdash (p_0, \sigma) \xrightarrow{\rho, A, \theta} (p, \sigma'))\} \cup \{(p, actv(p)) \mid (\exists \sigma, \sigma', p_0, a, b :: \llbracket p_0 \rrbracket \vdash (p_0, \sigma) \xrightarrow{a, b, \emptyset} (p, \sigma'))\}$$

Next we prove that R is indeed a bisimulation relation (as defined in our paper). We do so by checking that the for transfer conditions are satisfied.

Condition 1 Assume $(p, actv(p)) \in R$, and

$$\llbracket p_0 \rrbracket \vdash (p, \sigma) \xrightarrow{a, b, \emptyset} (p', \sigma') \tag{1}$$

Note that since p is in the transition system induced by p_0 , p is of the form:

```
[[ disc control  $xs$ 
; disc  $ys$ 
; clock  $ys$ 
; init  $u'$ 
; urgent  $as$ 
::  $\alpha_0 \parallel \dots \parallel \alpha_{n-1}$  ]]
```

where α_i and α_{i_0} may differ only in their initial locations, and $u \in \{u, \text{true}\}$.

Assume $\mathcal{T}(p_0) = \langle \bar{A}, [l_{0_0}, \dots, l_{0_{n-1}}], V, C, H, T_H, \mathcal{T}_i(u) \rangle$, where all the components are as in the definition of \mathcal{T} .

We have to do a case analysis depending on whether a is declared as urgent in p_0 , and depending on the value of b .

Case a is urgent and $\neg b$ Under this assumption, given that the model consists of a parallel composition of n automata (the model is translatable), we know that only one of these could have performed the action. Thus we get the following facts.

There is an index i , such that $0 \leq i < n$, α_i is of the shape

$$(L_i, \mapsto l_i, \text{inv}_i, \text{tcp}_i, E_i, \text{var}_{C_i}, \text{act}_{S_i}, \emptyset)$$

and the following holds.

$$\begin{aligned} p' &= p[\alpha'_i/\alpha_i] \\ \alpha'_i &= \alpha_i[\mapsto l'_i/\mapsto l_i] \\ (l_i, g, a, (zs, zs^+ = es), l'_i) &\in E_i \\ \sigma &\models g \\ \sigma &\models \text{inv}_i(l'_i) \\ \sigma'^+ \cup \sigma &\models zs^+ = es \\ \langle \forall x : x \in (cs \cup xs \cup \text{var}_{C_i}) \setminus zs : \sigma(x) = \sigma'(x) \rangle & \end{aligned} \quad (2)$$

All the remaining automata performed an environment transition. This is, for all j , $0 \leq j < n$ such that $j \neq i$, assuming α_j is of the shape

$$(L_j, \mapsto l_j, \text{inv}_j, \text{tcp}_j, E_j, \text{var}_{C_j}, \text{act}_{S_j}, \text{dtype}_j)$$

we have that

$$\begin{aligned} \sigma' &\models \text{inv}_i(l_j) \\ \langle \forall x : x \in \text{var}_{C_j} : \sigma(x) = \sigma'(x) \rangle & \end{aligned} \quad (3)$$

Using the above facts, we have that $\text{actv}(p') = \text{actv}(p)[l'_i/l_i]$.

According to the definition of the transformation function, we have an edge $(l_i, g, a!, zs = es^-, l'_i) \in E'_i$, where the UPPAAL automaton A_i is of the shape $(L_i, l_{i_0}, E'_i, \text{inv}_i, T_{L_i})$.

Since a is an urgent action, we know that there is an edge $(l_u, \text{true}, a?, [], l_u)$ in the receptive automaton.

Since in a translatable models all variables must be controlled, from (2) and (3) we know that¹:

$$\langle \forall x : x \in \text{dom}(\sigma) \setminus zs : \sigma(x) = \sigma'(x) \rangle \quad (4)$$

¹Remember that $\text{dom}(\sigma)$ contains exactly all variables declared in the model

Thus using Lemma 1 we get that:

$$(zs = es^-)(\sigma) = \sigma' \quad (5)$$

Since guard g is satisfied in σ , all the invariants of locations $\text{init}(p_0)$ are satisfied in σ' , there are no committed locations, and the assignment in the edge $(l_i, g, a!, zs = es^-, l'_i)$ transform σ into σ' , we have communication with the receptive automaton can take place, and thus we get:

$$\llbracket \mathcal{T}(p_0) \rrbracket \vdash (\text{actv}(p), \sigma) \xrightarrow{a} (\text{actv}(p'), \sigma')$$

Case a is urgent and b In this case we have that the synchronizing action a was executed synchronously by exactly two partners, by definition of translatable model. Moreover, the syntactic constraints guarantee that these two synchronizing actions correspond with send and receive actions in the UPPAAL model. The proof that there is a transition

$$\llbracket \mathcal{T}(p_0) \rrbracket \vdash (\text{actv}(p), \sigma) \xrightarrow{a} (\text{actv}(p'), \sigma')$$

goes very much along the lines of the previous case. Note that in this case, the requirement that the conjunction of the update predicates of the send and receive action is a translatable assignment plays an essential role.

Case a is not urgent The proof for this case is similar to the previous ones. For non-synchronizing actions we use UPPAAL rules for internal transitions instead of rules for channels.

Condition 2 Assume there is a transition

$$\llbracket \mathcal{T}(p_0) \rrbracket \vdash (\text{actv}(p), \sigma') \xrightarrow{a} (l', \sigma')$$

We have to do a case analysis depending on whether a is an (internal) action or a channel.

Internal action Then we have that there is an atomic UPPAAL automaton A_i , $A_i = (L_i, l_{i_0}, E'_i, \text{inv}'_i, T_{L_i})$, such that

$$\begin{aligned} (l_i, g, a, zs = es, l'_i) &\in E_i \\ \sigma &\models g \\ (zs = es)(\sigma) &= \sigma' \models \text{inv}'_i(l'_i) \\ \langle \forall j : 0 \leq j < n \wedge j \neq i : \sigma' \models \text{inv}_j(l_j) \rangle \\ l' &= \text{init}(p)[l'_i/l_i] \end{aligned} \quad (6)$$

Where we assume l_j to be the active location of automaton A_j , for all j , such that $0 \leq j < n$.

According to (6), we know that there must be an edge $(l_i, g, a, zs^+ = es', l'_i)$ in α_i , where $es'^- = es$.

By definition of UPPAAL assignment, we know that only the clocks and variables being assigned change, thus we can use Lemma 1 to infer:

$$\sigma'^+ \cup \sigma \models zs^+ = es' \quad (7)$$

Thus, automaton α_i can perform an action transition

$$(\alpha_i, \sigma) \xrightarrow{a, \text{false}, xs \cup cs} (\alpha'_i, \sigma')$$

where α'_i is the same automaton as α_i , except that the initial location is changed to l'_i .

Since, by definition of translatable model, all assigned variables in automaton α_i are not controlled in the other automata, and since, by (6), the invariants of all the active locations in p are satisfied, we have that all the other automata α_j , $j \neq i$ can perform environment transitions

$$(\alpha_j, \sigma) \xrightarrow{\text{act}_{S_j}} (\alpha_j, \sigma')$$

As a consequence, p can perform an action transition

$$(p, \sigma) \xrightarrow{a, \text{false}, \emptyset} (p', \sigma')$$

where $p' = p[\alpha'_i/\alpha_i]$, and thus $actv(p') = l'$, thus $(p', l') \in R$, which concludes the proof.

Communication If the receptive automaton is involved in a communication action, then one of the A_i 's performed a send action. The conditions for this send action are similar to those of (6), and thus, using a similar reasoning as in the previous case we can conclude that there is a corresponding action transition in p .

If the receptive automaton is not involved, then we have a communication action between two UPPAAL automata A_i and A_j , and the proof is similar to the case of synchronizing actions for the CIF automata.

Condition 3 Assume there is a transition

$$\llbracket p_0 \rrbracket \vdash (p, \sigma) \xrightarrow{\rho, A, \theta} (p', \sigma')$$

where $\text{dom}(\rho) = [0, t]$ for some $t \in \mathbb{T}$.

As stated before, note that since p is in the transition system induced by p_0 , it is of the form

```
[[ disc control xs
; disc ys
; clock ys
; init u
; urgent as
::  $\alpha_0$  || ... ||  $\alpha_{n-1}$  ]]
```

where α_i and α_{0_i} may differ only in their initial locations.

According to the SOS rules defined in Section 1, we have that

$$\begin{aligned}
(\alpha_0 \parallel \dots \parallel \alpha_{n-1}, \sigma) &\xrightarrow{\rho, A, \theta} (\alpha_0 \parallel \dots \parallel \alpha_{n-1}, \sigma') \\
&\langle \forall s : s \in [0, t] : as \cap \theta(s) = \emptyset \rangle \\
&\langle \forall x : x \in xs \cup ys : (\rho_x, \rho_x) \in G_{disc} \rangle \\
&\langle \forall c : c \in cs : (\rho_c, \rho_c) \in G_{clock} \rangle
\end{aligned} \tag{8}$$

Note that the time transition does not change the active locations of the automata involved in the parallel composition.

Consider state $(actv(p), \sigma)$. From the definition of function \mathcal{T} , we know that there are no automata in an committed location. Similarly, from (8), given the fact that a time transition is possible in the parallel composition of automata, we know that there cannot be any UPPAAL automaton in an urgent location, since that would imply that there is a corresponding CIF location in which the tcp predicate is false.

Next, we show that synchronization over urgent channels is not possible. If we assume in the UPPAAL automaton there are 2 edges $(l_i, g_i, a!, r_s, l'_i) \in E'_i$ and $(l_j, g_j, a!, r_s, l'_j) \in E'_j$ such that $\sigma \models g_i \wedge g_j$. But this implies that $a \in \theta(0)$ ², which contradicts the fact that a p did a time transition. Thus, synchronization between urgent channels cannot take place.

Furthermore, from (8) and the facts that the invariants are preserved by the transformation function, it is easy to see that the invariants of the active locations in $actv(p)$ are satisfied during the time delay.

Finally, we have to see that in the new valuation the clocks are incremented by t time units, and that the values of variables remain constant during the delay. This follows easily from Rule 3, and the definition of the dynamic types G_{disc} and G_{clock} . This is, for every clock c we have $\sigma'(c) = \sigma(c) + t$, and for every variable x we have $\sigma(x) = \sigma'(x)$.

Putting these facts together, we have that in the transition system induced by $\mathcal{T}(p_0)$ a time transition of t time units is possible, since all the requirements are satisfied. This is, we have a transition

$$\llbracket \mathcal{T}(p_0) \rrbracket \vdash (actv(p), \sigma) \xrightarrow{t} (actv(p), \sigma')$$

which concludes the proof of transfer condition 3.

Condition 4 Assume there is a transition

$$\llbracket \mathcal{T}(p_0) \rrbracket \vdash (actv(p), \sigma) \xrightarrow{t} (actv(p'), \sigma')$$

Considering the definition of function \mathcal{T} , this means that in the network of automata $\mathcal{T}(p_0)$ we have that:

1. No process is in a committed location.

²Note that we are also taking into account the receptive automaton case.

2. No process is in an urgent location.
3. No synchronization is possible over an urgent channel.
4. All invariants are satisfied during the time delay.
5. The new valuation advances the clocks t time units, and the values of variables are not changed.

From the above we get that, since no process is in a urgent location, the tcp predicate is true for all the corresponding locations in p . Similarly, we have that all the invariants are satisfied in the active locations of p during the delay.

Thus, since we assumed that the dynamic type mappings of the translatable automata are empty, every $\alpha_i \in p$ can do a time delay

$$(\alpha_i, \sigma) \xrightarrow{\rho, \text{act}_{S_i}, \theta} (\alpha_i, \sigma')$$

where

$$\begin{aligned} \text{dom}(\rho) &= [0, t] \\ \langle \forall s, s', x : s, s' \in [0, t] \wedge x \in V : \rho(s)(x) &= \rho(s')(x) \rangle \\ \langle \forall s, c : s \in [0, t] \wedge c \in C : \rho(s)(c) &= \rho(0)(c) + s \rangle \end{aligned} \quad (9)$$

and θ is constructed from ρ , using the SOS rules for atomic automata.

Since no urgent channel is enabled in valuation $\sigma = \rho(0)$, we have that

$$as \cap (\theta(0)) = \emptyset \quad (10)$$

Given the fact that the guard for urgent trajectories does not change over time (since it is not allowed to contain clocks), from (10) we conclude that

$$\langle \forall s : s \in [0, t] : as \cap (\theta(s)) = \emptyset \rangle$$

It is also clear from (9) that ρ satisfies the dynamic type constraints. Thus, from the above facts we can conclude that there is a time transition

$$\llbracket \mathcal{T}(p_0) \rrbracket \vdash (p, \sigma) \xrightarrow{\rho, A, \theta} (p, \sigma')$$

where $A = \text{act}_{S_0} \cup \dots \cup \text{act}_{S_{n-1}}$. And this concludes the proof.

References

- [1] J.C.M. Baeten, D.A. van Beek, D. Hendriks, A.T. Hofkamp, D.E. Ndales Agut, J.E. Rooda, and R.R.H. Schiffelers. Report describing the extended cif functionality. http://cms.multiform.bci.tu-dortmund.de/images/stories/multiform/deliverables/multiform_d112.pdf, 2010.