Verifikation nebenläufiger Programme
Wintersemester 2004/05

Ulrich Hannemann   Jan Bredereke

# 14 Proof Methods for Partial Correctness of Synchronous Transition Diagrams

In order to prove partial correctness of synchronous diagram $P_1 \parallel \ldots \parallel P_n$ w.r.t. $< \varphi, \psi >$ find local predicates $\mathcal{Q}_l$ (associated with the local locations $l$ of $P_i$ and which do not involve any of the variables of the other components) and prove:

1. the local verification conditions of $P_i$ w.r.t. $\{\mathcal{Q}_l \mid l \textit{ is a location of } P_i\}$,
2. for every pair of syntactically-matching input-output transitions $l_i \overset{a_i}{\rightarrow} l'_i$ of $P_i$ and $l_j \overset{a_j}{\rightarrow} l'_j$ of $P_j$, with $a_i \equiv b; C!e \rightarrow f$ and $a_j \equiv b'; C?x \rightarrow g$:
$$\models \mathcal{Q}_{l_i} \wedge \mathcal{Q}_{l_j} \wedge b \wedge b' \rightarrow (\mathcal{Q}_{l'_i} \wedge \mathcal{Q}_{l'_j}) \circ h,$$
where $h \overset{\text{def}}{=} f \circ g \circ (x := e)$,
3. $\models \varphi \rightarrow \mathcal{Q}_{s_1} \wedge \ldots \wedge \mathcal{Q}_{s_n}$, with $s_i$ the initial location of $P_i$, and $\models \mathcal{Q}_{t_1} \wedge \ldots \wedge \mathcal{Q}_{t_n} \rightarrow \psi$, with $t_i$ the final location of $P_i$.

In the following example we show that there is a serious problem: the method sketched above is not complete.

**Example 14.1 (Incompleteness of the method above)** Consider again $P_1 \| P_2$ in Figure 1 from Session 13, where $l_0 \equiv s$, $l_t \equiv t$, $l'_0 \equiv s'$, and $l'_t \equiv t'$. With every local location we associate an assertion as shown in Figure 1.
Clearly $P_1 \parallel P_2$ is partially correct w.r.t. specification $< \mathit{true}, x = 2 >$. This cannot be proved, however, with the method described above. To show this, assume we have associated with all locations in the program predicates that satisfy the requirements of the method. Since predicates for $P_1$ may only refer to program variables of $P_1$, and there are no program variables in $P_1$, the predicates $\mathcal{Q}_{l_0}$, $\mathcal{Q}_{l_1}$, and $\mathcal{Q}_{l_t}$ do not contain free program variables and, as shown below, are identical to *true*. Similarly, $\mathcal{Q}_{l'_0}(x)$, $\mathcal{Q}_{l'_1}(x)$, and $\mathcal{Q}_{l'_t}(x)$ may only refer to program variable $x$.

From point (iii) of the method we obtain

$$\models \quad \mathit{true} \rightarrow \mathcal{Q}_{l_0} \wedge \mathcal{Q}_{l'_0}(x), \text{ and} \qquad (1)$$
$$\models \quad \mathcal{Q}_{l_t} \wedge \mathcal{Q}_{l'_t}(x) \rightarrow x = 2. \qquad (2)$$

According to point (ii), the cooperation test, the verification conditions for all syntactically-matching pairs should be valid. Thus one has validity of the following conditions:

$$\models \quad \mathcal{Q}_{l_0} \wedge \mathcal{Q}_{l'_0}(x) \rightarrow \mathcal{Q}_{l_1} \wedge \mathcal{Q}_{l'_1}(1), \qquad (3)$$
$$\models \quad \mathcal{Q}_{l_0} \wedge \mathcal{Q}_{l'_1}(x) \rightarrow \mathcal{Q}_{l_1} \wedge \mathcal{Q}_{l'_t}(1), \qquad (4)$$
$$\models \quad \mathcal{Q}_{l_1} \wedge \mathcal{Q}_{l'_0}(x) \rightarrow \mathcal{Q}_{l_t} \wedge \mathcal{Q}_{l'_1}(2), \text{ and} \qquad (5)$$
$$\models \quad \mathcal{Q}_{l_1} \wedge \mathcal{Q}_{l'_1}(x) \rightarrow \mathcal{Q}_{l_t} \wedge \mathcal{Q}_{l'_t}(2). \qquad (6)$$

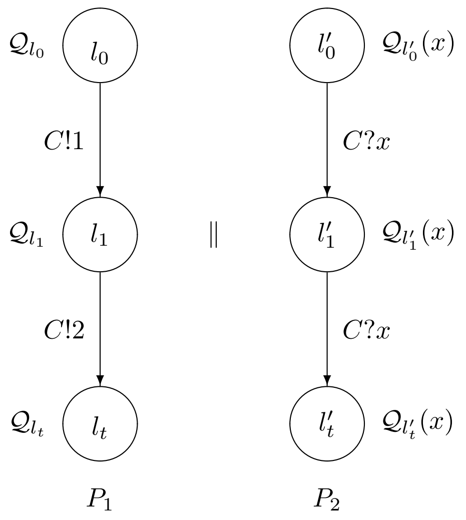


Figure 1: Adding assertions to program $P_1\|P_2$ from Figure 1 of Session 13.

From (1) we obtain $\models \mathcal{Q}_{l_0} \wedge \mathcal{Q}_{l'_0}(x)$, and by (3) this leads to $\models \mathcal{Q}_{l_1}$ and $\models \mathcal{Q}_{l'_1}(1)$. Then by (6) we obtain that $\models \mathcal{Q}_{l_t}$ holds. Since $\models \mathcal{Q}_{l_0}$ holds and $\models \mathcal{Q}_{l'_1}(1)$ is also *true*, (4) implies that $\models \mathcal{Q}_{l'_t}(1)$ holds. Hence, $\models \mathcal{Q}_{l_t} \wedge \mathcal{Q}_{l'_t}(1)$ is *true*, that is, if $x = 1$ then $\models \mathcal{Q}_{l_t} \wedge \mathcal{Q}_{l'_t}(x)$ holds. Thus $\models x = 1 \rightarrow \mathcal{Q}_{l_t} \wedge \mathcal{Q}_{l'_t}(x)$ holds. But then (2) would imply, by transitivity of implication, that $\models x = 1 \rightarrow x = 2$. This is a contradiction.

■

Two solutions are given for this problem:

1. Use *shared* auxiliary variables to relate local locations in different processes by expressing that certain combinations of locations will not occur during execution. (Observe that this is similar to the use of auxiliary variables in the method of Owicki & Gries.) In the case of Example 14.1 we can use a shared auxiliary variable $k$, *which counts the number of $C$-communications,* and the corresponding assertions, as illustrated in Figure 2, where $l_0 \equiv s$, $l_t \equiv t$, $l'_0 \equiv s'$, and $l'_t \equiv t'$.

   Then the semantically matching pairs cooperate:

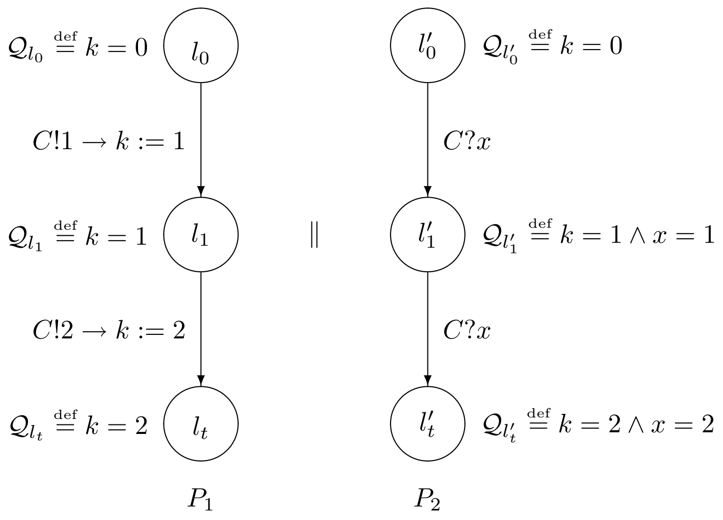
$$\models (\mathcal{Q}_{l_0} \wedge \mathcal{Q}_{l'_0})(x,k) \rightarrow (\mathcal{Q}_{l_1} \wedge \mathcal{Q}_{l'_1})(1,1) \quad \text{and}$$
$$\models (\mathcal{Q}_{l_1} \wedge \mathcal{Q}_{l'_1})(x,k) \rightarrow (\mathcal{Q}_{l_t} \wedge \mathcal{Q}_{l'_t})(2,2).$$

   Now the other, not semantically-matching, communication pairs pass the cooperation test because the conjunction of their preconditions evaluates to $false$:

$$\models (\mathcal{Q}_{l_0} \wedge \mathcal{Q}_{l'_1})(x,k) \rightarrow (\mathcal{Q}_{l_1} \wedge \mathcal{Q}_{l'_t})(1,2)$$

   since $\models k = 0 \wedge k = 1 \wedge x = 1 \rightarrow false$, and

$$\models (\mathcal{Q}_{l_1} \wedge \mathcal{Q}_{l'_0})(x,k) \rightarrow (\mathcal{Q}_{l_t} \wedge \mathcal{Q}_{l'_1})(2,1)$$



$\mathcal{Q}_{l_0} \stackrel{\text{def}}{=} k = 0$
$l_0$
$C!1 \rightarrow k := 1$
$\mathcal{Q}_{l_1} \stackrel{\text{def}}{=} k = 1$
$l_1$
$C!2 \rightarrow k := 2$
$\mathcal{Q}_{l_t} \stackrel{\text{def}}{=} k = 2$
$l_t$
$P_1$
$\|$
$l'_0$
$\mathcal{Q}_{l'_0} \stackrel{\text{def}}{=} k = 0$
$C?x$
$l'_1$
$\mathcal{Q}_{l'_1} \stackrel{\text{def}}{=} k = 1 \wedge x = 1$
$C?x$
$l'_t$
$\mathcal{Q}_{l'_t} \stackrel{\text{def}}{=} k = 2 \wedge x = 2$
$P_2$

Figure 2: Adding shared auxiliary variables plus corresponding assertions to program $P_1 \| P_2$.

since $\models k = 1 \wedge k = 0 \rightarrow false$.

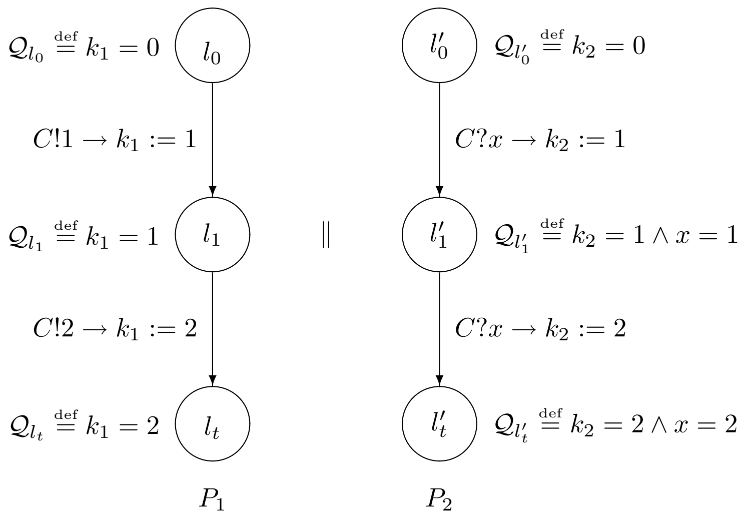
This approach leads to the method of Levin & Gries [LG81]. A systematic development of this method calls for defining a product of synchronous transition diagrams whose state spaces are no longer disjoint, since auxiliary variables now may be shared. Consequently, in addition to the cooperation test, this method again introduces the interference freedom test.

2. Use of the interference freedom test is avoided in the AFR-method [AFdR80] where only *local* auxiliary variables are used. In that method every process has its own set of local programming and local auxiliary variables, and assertions associated with a process should only refer to these local variables. Therefore, parallel composition remains formulated as in Definition 13.1.

In case of Example 14.1, the use of two auxiliary local counters $k_1$ and $k_2$ leads to program $P'_1 \| P'_2$ plus associated assertions as illustrated in Figure 3, where $l_0 \equiv s$, $l_t \equiv t$, $l'_0 \equiv s'$, and $l'_t \equiv t'$.

This, however, does not solve the problem that the conjunction of preconditions for a not semantically-matching pair, e.g., $\mathcal{Q}_{l_0} \wedge \mathcal{Q}_{l'_1} \equiv k_1 = 0 \wedge k_2 = 1 \wedge x = 1$, does not evaluate to $false$. Therefore, a *global invariant* $I$ (also called communication invariant) is introduced to relate the values of the local auxiliary variables in the various processes. *This invariant can be used to express which combinations of values for the local auxiliary variables occur during execution, and thus to indicate which combination of locations, and corresponding communications, occur during execution.*

In our example above we use the global invariant $I \stackrel{\text{def}}{=} k_1 = k_2$ to express that both processes have performed the same number of $C$-communications at any



$\mathcal{Q}_{l_0} \overset{\text{def}}{=} k_1 = 0$
$l_0$
$C!1 \rightarrow k_1 := 1$
$\mathcal{Q}_{l_1} \overset{\text{def}}{=} k_1 = 1$
$l_1$
$C!2 \rightarrow k_1 := 2$
$\mathcal{Q}_{l_t} \overset{\text{def}}{=} k_1 = 2$
$l_t$
$P_1$
$\|$
$l'_0$
$\mathcal{Q}_{l'_0} \overset{\text{def}}{=} k_2 = 0$
$C?x \rightarrow k_2 := 1$
$l'_1$
$\mathcal{Q}_{l'_1} \overset{\text{def}}{=} k_2 = 1 \wedge x = 1$
$C?x \rightarrow k_2 := 2$
$l'_t$
$\mathcal{Q}_{l'_t} \overset{\text{def}}{=} k_2 = 2 \wedge x = 2$
$P_2$

Figure 3: Adding local auxiliary variables plus corresponding assertions to $P_1 \| P_2$.

point during an execution. Of course we have to show that $I$ is indeed invariant and that it holds prior to executing $P'_1 \parallel P'_2$. Since the auxiliary variables are only added to communication steps, $I$ is trivially invariant under local steps. In order to prove that $I$ is invariant under all communication steps, one adds $I$ as follows to the cooperation test.

For every communication step leading from $l_1$ and $l'_1$ to $l_2$ and $l'_2$, prove:

$$\models (\mathcal{Q}_{l_1} \wedge \mathcal{Q}_{l'_1} \wedge I) \rightarrow (\mathcal{Q}_{l_2} \wedge \mathcal{Q}_{l'_2} \wedge I) \circ h,$$

where $h$ is the state transformation corresponding to the communication.

In the case of our example this leads to four verification conditions.

For the semantically-matching communication pairs, we have to check

$$\models (\mathcal{Q}_{l_0} \wedge \mathcal{Q}_{l'_0} \wedge I)(x, k_1, k_2) \rightarrow (\mathcal{Q}_{l_1} \wedge \mathcal{Q}_{l'_1} \wedge I)(1, 1, 1),$$

which holds because $\models (k_1 = 0 \wedge k_2 = 0 \wedge k_1 = k_2) \rightarrow (1 = 1 \wedge 1 = 1 \wedge 1 = 1)$ is *true*, and

$$\models (\mathcal{Q}_{l_1} \wedge \mathcal{Q}_{l'_1} \wedge I)(x, k_1, k_2) \rightarrow (\mathcal{Q}_{l_t} \wedge \mathcal{Q}_{l'_t} \wedge I)(2, 2, 2),$$

which holds because

$$\models (k_1 = 1 \wedge k_2 = 1 \wedge x = 1 \wedge k_1 = k_2) \rightarrow (2 = 2 \wedge 2 = 2 \wedge 2 = 2 \wedge 2 = 2)$$

is *true*.

The semantically-not-matching pairs pass the cooperation test as follows:

$$\models (\mathcal{Q}_{l_0} \wedge \mathcal{Q}_{l'_1} \wedge I) \rightarrow (\mathcal{Q}_{l_1} \wedge \mathcal{Q}_{l'_t} \wedge I)(1, 1, 2)$$

since $\models (k_1 = 0 \wedge k_2 = 1 \wedge x = 1 \wedge k_1 = k_2) \rightarrow \mathit{false}$, and

$$\models (\mathcal{Q}_{l_1} \wedge \mathcal{Q}_{l'_0} \wedge I) \rightarrow (\mathcal{Q}_{l_t} \wedge \mathcal{Q}_{l'_1} \wedge I)(2, 2, 1)$$

since $\models (k_1 = 1 \wedge k_2 = 0 \wedge k_1 = k_2) \rightarrow \mathit{false}$.

It remains to establish that $I$ holds initially. In the case of our example this is trivial: just choose $k_1 = 0 \wedge k_2 = 0$, then $\models k_1 = 0 \wedge k_2 = 0 \rightarrow \mathcal{Q}_{l_0} \wedge \mathcal{Q}_{l'_0} \wedge I$ holds. Since $\models \mathcal{Q}_{l_t} \wedge \mathcal{Q}_{l_t} \rightarrow x = 2$, we obtain that $P'_1 \parallel P'_2$ is partially correct w.r.t. $< k_1 = 0 \wedge k_2 = 0 \wedge k_1 = k_2, x = 2 >$.

Next we discuss how to derive from this result that $\{\mathit{true}\}\ P_1 \| P_2\ \{x = 2\}$ holds, using the initialisation and auxiliary variables rules from Session 9.

As in Example 8.12, we observe that the value of $x$ during any execution is not affected by the auxiliary variables $k_1$ and $k_2$, since these variables do not occur in the conditions and the assignments to $x$. Hence the postcondition $x = 2$ does not depend on the initial values of $k_1$ and $k_2$, and is established by the initialisation rule for any arbitrary initial value of $k_1$ and $k_2$, as long as $k_1 = k_2$ holds. This justifies choosing 0 as the value for $k_1$ and for $k_2$ in the precondition, and leads to: $P'_1 \parallel P'_2$ is partially correct w.r.t. $< \mathit{true}, x = 2 >$, since $\models \mathit{true} \leftrightarrow 0 = 0 \wedge 0 = 0 \wedge 0 = 0$, by soundness of the initialisation rule. Similarly to Lemma 10.2, we observe that for any execution of $P'_1 \parallel P'_2$ there exists a corresponding execution of $P_1 \parallel P_2$ with the same value for $x$ in the final state. Hence $\models \{\mathit{true}\} P_1 \parallel P_2 \{x = 2\}$ holds by soundness of the auxiliary variables rule.

This leads to the following formulation of the AFR-method.

**Definition 14.2 (Proof method of Apt, Francez & de Roever)** The proof method of Apt, Francez & de Roever (AFR-method) is formulated as follows.

Given synchronous transition diagram $P \equiv P_1 \parallel \ldots \parallel P_n$ with locations $L \equiv L_1 \times \ldots \times L_n$.

Prove as follows that $P$ is partially correct w.r.t. specification $< \varphi, \psi >$:

1. Augment $P_i$ by introducing auxiliary variables; every input-output transition $\alpha \rightarrow f$ is extended as follows: $\alpha \rightarrow f \circ g$, where $g$ is a state transformation such that its write variables are amongst the auxiliary variables $\bar{z}^i$, which should not occur in $P$, $\varphi$ and $\psi$; furthermore the auxiliary variables $\bar{z}^i$ associated with the components $P_i$ are required to be mutually disjoint for $i = 1, \ldots, n$. This leads to an augmented synchronous transition diagram $P' \equiv P'_1 \parallel \ldots \parallel P'_n$.

2. Associate a predicate $\mathcal{Q}_l$ with every location $l$ of $P'_i$, where $\mathcal{Q}_l$ does not involve any of the variables of $P'_j$, $j \neq i$.

3. Prove *local correctness* of every $P'_i$: for every internal transition $l \xrightarrow{a} l'$ of $P'_i$, assuming $a \equiv b \rightarrow f$, we have to prove

$$\models \mathcal{Q}_l \wedge b \rightarrow \mathcal{Q}_{l'} \circ f.$$

4. Choose a predicate $I(\bar{z})$, called *global invariant*, involving only the auxiliary variables $\bar{z} \equiv \bigcup_i \bar{z}^i$. Then prove *the cooperation test*, verify for every pair of transitions $l_1 \xrightarrow{a} l_2$ of $P'_i$ and $l'_1 \xrightarrow{a'} l'_2$ of $P'_j$, with $j \neq i$, $a \equiv b; C!e \rightarrow f$ and $a' \equiv b'; C?x \rightarrow g$, that

$$\models I \wedge \mathcal{Q}_{l_1} \wedge \mathcal{Q}_{l'_1} \wedge b \wedge b' \rightarrow (I \wedge \mathcal{Q}_{l_2} \wedge \mathcal{Q}_{l'_2}) \circ h,$$

where $h \overset{\text{def}}{=} f \circ g \circ (x := e)$.

5. Prove that

   - $\models \varphi \to (I \wedge \bigwedge_i \mathcal{Q}_{s_i}) \circ f$, for some state transformation $f$ whose write variables belong to the set of auxiliary variables $\bar{z}$ ($s_i$ denotes the initial location of $P_i'$), and
   - $\models I \wedge \bigwedge_i \mathcal{Q}_{t_i} \to \psi$, where $t_i$ denotes the final location of $P_i'$.

■

## References

[AFdR80] K.R. Apt, N. Francez, and W. P. de Roever. A proof system for communicating sequential processes. *ACM Transactions on Programming Languages and Systems*, 2:359–385, 1980.

[LG81] G.M. Levin and D. Gries. A proof technique for Communicating Sequential Processes. *Acta Informatica*, 15:281–302, 1981.