Verifikation nebenläufiger Programme
Wintersemester 2004/05

Ulrich Hannemann Jan Bredereke

8 The Proof Method of Owicki & Gries

8.1 Basic intuition of the method of Owicki & Gries

Since the generalization of Floyd's method generates a set of verification conditions whose size is *exponential* in the number of processes, as an alternative a more manageable proof method is adopted which is based on *local* inductive assertion networks which additionally satisfy the *interference freedom test* formulated by Susan Owicki and David Gries [OG76]. We try to improve the situation by deriving predicates associated with global locations from predicates attached to local locations. First these local predicates in P_i are proved to be *locally correct*, i.e., partially correct for the sequential execution of P_i when P_i is considered in isolation as a separate process. We investigate what must be added to these proofs in order to achieve partial correctness of $P_1 \parallel \ldots \parallel P_n$.

Let $P \equiv P_1 \parallel \ldots \parallel P_n$. Associate predicates to *local* locations of P instead of to its global locations: assume that for every *local* location l_i in P_i there exists a predicate $\mathcal{Q}_{l_i}$. In order to apply the inductive assertion method, Definition 3.2, associate with every global location $l = \langle l_1, \ldots, l_n \rangle$ of P (where l_i denotes a location of P_i) the predicate $\mathcal{Q}_l \equiv \mathcal{Q}_{l_1} \wedge \ldots \wedge \mathcal{Q}_{l_n}$; the resulting inductive assertion network is called $\mathcal{Q}_1 \times \ldots \times \mathcal{Q}_n$. Next this assertion network is shown to be inductive by proving the verification conditions for all steps. That is, for each transition $b \to f$ leading from $l = \langle l_1, \ldots, l_n \rangle$ to $l' = \langle l'_1, \ldots, l'_n \rangle$ we have to prove

$$\models \mathcal{Q}_l \wedge b \to \mathcal{Q}_{l'} \circ f,$$

i.e.,

$$\models (\mathcal{Q}_{l_1} \wedge \ldots \wedge \mathcal{Q}_{l_n} \wedge b) \to (\mathcal{Q}_{l'_1} \wedge \ldots \wedge \mathcal{Q}_{l'_n}) \circ f.$$

By the definition of a transition in a parallel composition, l differs from l' in at most only one local location. Suppose this step is a transition in P_i. Then $l_j \equiv l'_j$, for $i \neq j$, and hence $\mathcal{Q}_{l_j} = \mathcal{Q}_{l'_j}$. We shall demonstrate that it is sufficient to prove:

1. $\models \mathcal{Q}_{l_i} \wedge b \to \mathcal{Q}_{l'_i} \circ f$,
   i.e., the *local* verification condition in P_i, and

2. $\models \mathcal{Q}_{l_j} \wedge \mathcal{Q}_{l_i} \wedge b \to \mathcal{Q}_{l_j} \circ f$, for all $j \neq i$,
   that is, all predicates $\mathcal{Q}_{l_j}$ associated with other processes P_j, with $j \neq i$, are *invariant* under execution of this particular transition in P_i. In other words, *executing a transition in P_i does not interfere with the validity of the local assertions $\mathcal{Q}_{l_j}$ chosen in the other processes.*

This can be understood as follows:

$$
\begin{array}{ll}
\mathcal{Q}_l \wedge b = & \text{(by definition and propositional logic)} \\
(\bigwedge_{j \neq i} \mathcal{Q}_{l_j} \wedge \mathcal{Q}_{l_i} \wedge b) \ \wedge \ (\mathcal{Q}_{l_i} \wedge b) \rightarrow & \text{(by 1 and 2 above)} \\
(\bigwedge_{j \neq i} \mathcal{Q}_{l_j} \circ f) \wedge \mathcal{Q}_{l'_i} \circ f = & \text{(by definition and propositional logic)} \\
\mathcal{Q}_{l'} \circ f. &
\end{array}
$$

Consequently, the combination of conditions 1 and 2 above leads to a sound proof method.

Condition 1 implies that process P_i is partially correct w.r.t. $< \mathcal{Q}_{s_i}, \mathcal{Q}_{t_i} >$ *in isolation*. We say that P_i is *locally correct* w.r.t. $< \mathcal{Q}_{s_i}, \mathcal{Q}_{t_i} >$, for $i = 1, \ldots, n$. Condition 2 corresponds to the *interference freedom test* of Owicki & Gries [OG76].

This leads to a more efficient method for proving partial correctness of $P_1 \parallel \ldots \parallel P_n$: first prove partial correctness for every process P_i in isolation, and then check interference freedom. In order to compute the complexity of this new method, again suppose that P_i has r locations and s edges. Now we have to find $n \times r$ local assertions and then we must prove for every edge

- local correctness: 1 verification condition, and
- interference freedom: there are $(n-1) \times r$ assertions in the other processes, so $(n-1) \times r$ verification conditions.

Since there are $n \times s$ edges in $P_1 \| \ldots \| P_n$, we obtain $n \times s \times (1 + (n-1) \times r)$ verification conditions. Clearly this improves upon the global method, which required $n \times s \times r^{n-1}$ verification conditions, and reflects the so-called *state explosion* associated with parallel composition.

Example 8.1 Consider program $P \equiv P_1 \parallel P_2$ as in Figure 1.

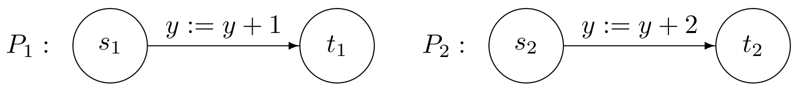

Figure 1: A very simple concurrent program.

We prove that P is partially correct w.r.t. specification $< y = 0, y = 3 >$, i.e.,
$\models \{y = 0\} \ P \ \{y = 3\}$. Take the assertion network $\mathcal{Q}$ defined in Figure 2.

1. It is easy to check that P_i is partially correct w.r.t. $< \mathcal{Q}_{s_i}, \mathcal{Q}_{t_i} >$, for $i \in \{1, 2\}$.
2. Verify interference freedom:
   - We show that $\mathcal{Q}_{s_1}$ and $\mathcal{Q}_{t_1}$ are invariant under $y := y + 2$, as follows.
     - Assume $\mathcal{Q}_{s_1} \wedge \mathcal{Q}_{s_2}$ holds. Then $y = 0$, and thus after executing $y := y + 2$ we have that $\mathcal{Q}_{s_1} \equiv y = 0 \vee y = 2$ holds.
     - Assume $\mathcal{Q}_{t_1} \wedge \mathcal{Q}_{s_2}$ holds. Then $y = 1$, and thus after executing $y := y + 2$ we have that $\mathcal{Q}_{t_1} \equiv y = 1 \vee y = 3$ holds.

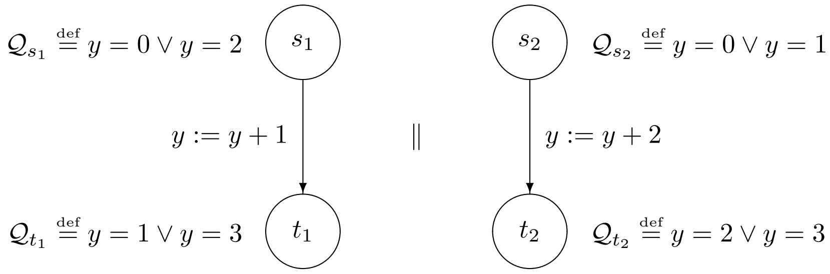

Figure 2: And its associated inductive assertion network.

- Similarly, $\mathcal{Q}_{s_2}$ and $\mathcal{Q}_{t_2}$ are invariant under $y := y + 1$.

3.
- $\models y = 0 \rightarrow \mathcal{Q}_s$, since $\mathcal{Q}_s \equiv \mathcal{Q}_{s_1} \wedge \mathcal{Q}_{s_2}$ and $\models \mathcal{Q}_{s_1} \wedge \mathcal{Q}_{s_2} \leftrightarrow y = 0$, and
- $\models \mathcal{Q}_t \rightarrow\ y = 3$, since $\mathcal{Q}_t \equiv \mathcal{Q}_{t_1} \wedge \mathcal{Q}_{t_2}$ and $\models \mathcal{Q}_{t_1} \wedge \mathcal{Q}_{t_2} \leftrightarrow y = 3$. $\square$

8.1.1 Incompleteness of the proposed method

Example 8.2 (Incompleteness of the proposed method) Consider $P \equiv P_1 \parallel P_2$ as in figure 3.

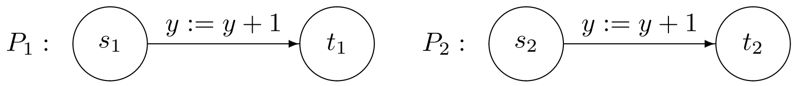

Figure 3: An even simpler concurrent program.

The aim is to prove that P is partially correct w.r.t. specification $< y = 0, y = 2 >$. Analogously to the previous example, we investigate whether the assertion network given in Figure 4 is interference free.

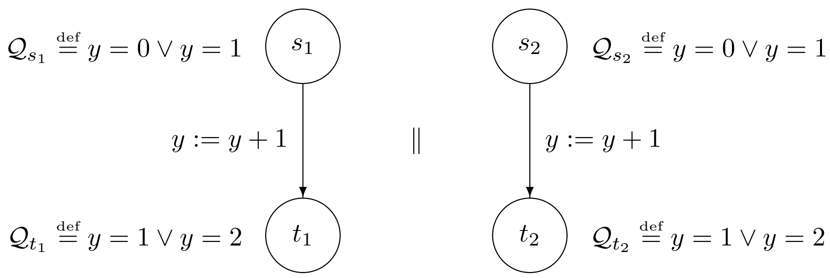

Figure 4: And a failed attempt at defining an interference free inductive assertion network for it.

Clearly P_i is partially correct w.r.t. $< \mathcal{Q}_{s_i}, \mathcal{Q}_{t_i} >$, for $i \in \{1, 2\}$. These predicates, however, are not interference free. For instance, assume that $\mathcal{Q}_{s_1} \wedge \mathcal{Q}_{s_2}$ holds. Then $y = 0 \vee y = 1$, and thus after executing $y := y + 1$ we have that $y = 1 \vee y = 2$ holds. Hence $\mathcal{Q}_{s_1} \equiv y = 0 \vee y = 1$ is *not* invariant under

execution of $y := y + 1$ in P_2.
A second problem is that $\mathcal{Q}_{t_1} \wedge \mathcal{Q}_{t_2}$ does not imply $y = 2$.

It is even impossible to find assertions that prove specification $< y = 0, y = 2 >$ for P using program variable y only! In order to show this, suppose we have $\mathcal{Q}_{s_i}$ and $\mathcal{Q}_{t_i}$ which are locally correct for P_i and, moreover, $\models y = 0 \rightarrow \mathcal{Q}_{s_1} \wedge \mathcal{Q}_{s_2}$, and $\models \mathcal{Q}_{t_1} \wedge \mathcal{Q}_{t_2} \rightarrow y = 2$. From the first implication, $\models y = 0 \rightarrow \mathcal{Q}_{s_1} \wedge \mathcal{Q}_{s_2}$, we obtain that $\mathcal{Q}_{s_1}$ and $\mathcal{Q}_{s_2}$ hold for a state which assigns the value 0 to y. Since we assumed local correctness, this implies that $\mathcal{Q}_{t_1}$ and $\mathcal{Q}_{t_2}$ hold for a state which assigns the value 1 to y, thus $\models y = 1 \rightarrow \mathcal{Q}_{t_1} \wedge \mathcal{Q}_{t_2}$. This, however, leads to a contradiction with the second implication, $\models \mathcal{Q}_{t_1} \wedge \mathcal{Q}_{t_2} \rightarrow y = 2$. $\square$

References

[OG76] S. Owicki and D. Gries. An axiomatic proof technique for parallel programs. *Acta Informatica*, 6:319–340, 1976.