Verifikation nebenläufiger Programme
Wintersemester 2004/05

Ulrich Hannemann Jan Bredereke

5 Proving Convergence

A program is called convergent, if it does not have any infinite computations, that is, it either terminates or becomes deadlocked, but will never go on forever. Of course, in general we can only expect a program to converge when started from certain initial states. Let φ be a precondition. We define a program to be φ-convergent, if it does not admit infinite φ-computations. In this section, we look at methods for proving that a program is convergent.

The importance of this notion is that in order to prove a program to be totally correct, in the present framework one has to prove that it is both partially correct and convergent and has no failing computations.

5.1 Wellfounded Sets

The basic idea for proving convergence is to show that *there is a bound on the remaining computation steps from any state that the program reaches.* This notion of bound is formalised by the concept of *wellfounded* set in mathematics.

Let W be a set and $\prec$ a binary relation on $W \times W$. We say $\prec$ is an ordering if it is:

1. *Irreflexive,* i.e., $a \not\prec a$ for any $a \in W$,
2. *Asymmetric,* i.e., $a \prec b$ implies $b \not\prec a$ for any $a, b \in W$,
3. *Transitive,* i.e., $a \prec b$ and $b \prec c$ implies that $a \prec c$ for any $a, b, c \in W$.

Note that conditions 1 and 3 imply condition 2 (which in its turn implies condition 1). Such an ordering can be partial in the sense that there may exist pairs of unrelated elements in the set. The partially ordered set $(W, \prec)$ is called *wellfounded* if there exists NO infinitely descending sequence in $(W, \prec)$:

$$\ldots \prec w_2 \prec w_1 \prec w_0, \text{ with } w_i \in W.$$

One of the simplest wellfounded sets is the set of nonnegative integers $I\!N = \{0, 1, 2, \ldots\}$, with the ordering of the "less than" relation '$<$'. As we shall see, we (only) use this set in the completeness proof in this chapter. Hence, in principle, it suffices to use nonnegative integers to prove that the kind of program we study here is convergent. Nevertheless, forcing the use of nonnegative integers in practice often leads to complicated assertions and ranking functions, a notion introduced later. Therefore, other sorts of wellfounded sets are used, too. One such example is the product of two (simpler) wellfounded sets with the lexicographical order. More precisely, let $(W_1, \prec_1)$ and $(W_2, \prec_2)$ be two wellfounded sets, then the partially ordered set $(W, \prec)$ defined by

$$W \overset{\text{def}}{=} W_1 \times W_2 \text{ and}$$
$$(m_1, n_1) \prec (m_2, n_2) \overset{\text{def}}{=} (m_1 \prec_1 m_2) \vee ((m_1 = m_2) \wedge (n_1 \prec_2 n_2)),$$

is also wellfounded. This construction can be easily extended to the product of an arbitrary (finite) number of wellfounded sets. This associated partial order is called *lexicographical ordering.*

5.2 A Proof Method for Convergence

In this section we give a proof method for establishing convergence of a program. This method builds directly on the one for partial correctness developed in the previous sections.

Definition 5.1 (Floyd's wellfoundedness method) Given a transition diagram $P = (L, T, s, t)$, in order to verify that it is φ-convergent with respect to a precondition φ we use Floyd's *wellfoundedness* method for proving convergence (i.e., either termination or deadlock) of sequential programs as formulated below:

1. Find an assertion network $\mathcal{Q}$, show that it is inductive, and that $\models \varphi \rightarrow \mathcal{Q}_s$ holds.

2. Choose a wellfounded set $(W, \prec)$ and a network $\rho \overset{\text{def}}{=} \{\rho_l \mid l \in L\}$ of *partially* defined *ranking functions* $\rho_l : \Sigma \rightarrow W$ for every $l \in L$, and prove points 3 and 4 below:

3. $\mathcal{Q}_l$ implies that ρ_l *is defined*, i.e., for every σ,
$$\models \mathcal{Q}_l(\sigma) \text{ implies } \rho_l(\sigma) \ \in \ W.$$

4. Every transition $l \xrightarrow{a} l'$, with $a = c \rightarrow f$, *decreases* the ranking function, i.e.,
$$\models \mathcal{Q}_l \wedge c \rightarrow \rho_l \succ (\rho_{l'} \circ f).$$
Here $\rho_l \succ (\rho_{l'} \circ f)$ denotes the predicate defined by
$$(\rho_l \succ (\rho_{l'} \circ f))(\sigma) \overset{\text{def}}{=} tt \text{ iff } \rho_l(\sigma) \succ \rho_{l'}(f(\sigma)). \qquad \square$$

Example 5.2 As a first step, consider proving convergence of the following simple program in Figure 1 with respect to the trivial precondition *true.* To make it a little more interesting, we assume that x belongs to the set of real numbers, which is not wellfounded w.r.t. the "less than" relation '<'.

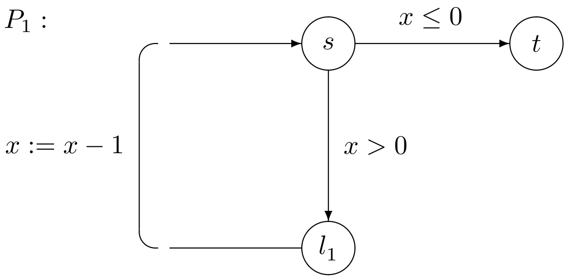

Figure 1: A terminating program over the real numbers.

The assertion network is chosen as:

$$\begin{aligned}\mathcal{Q}_s(x) &\stackrel{\text{def}}{=} \textit{true}\\ \mathcal{Q}_1(x) &\stackrel{\text{def}}{=} x > 0\\ \mathcal{Q}_t(x) &\stackrel{\text{def}}{=} \textit{true}.\end{aligned}$$

Next we consider the ranking function network. We have to map real numbers to a wellfounded set. Let $nni(x)$ be the function which returns the next nonnegative integer for a real number x (so for $-\sqrt{2}$ and π, it returns 0 and 4, respectively). The ranking function network defined over the lexicographical ordering $\mathbb{N} \times \mathbb{N}$ of nonnegative integers $\mathbb{N}$ is as follows:

$$\begin{aligned}\rho_s(x) &\stackrel{\text{def}}{=} (nni(x), 1)\\ \rho_1(x) &\stackrel{\text{def}}{=} (nni(x), 0)\\ \rho_t(x) &\stackrel{\text{def}}{=} (0, 0).\end{aligned}$$

Checking validity of the verification conditions is trivial; the ranking functions are always defined, and decrease along each transition, as shown below:

$\pi_1 : s \to l_1$, the first component of the ranking function does not increase, whereas the second decreases by one,
$\pi_2 : l_1 \to s$, the first component of the ranking function decreases at least by one when $x > 0$, and
$\pi_3 : s \to t$, the first component of the ranking function does not increase, whereas the second decreases by one. □

Example 5.3 Consider again the integer root-finding program from Examples 3.1 and 3.3. Now we want to show that it is convergent with respect to the precondition $\varphi \stackrel{\text{def}}{=} y_1 \geq 0$. We take the same assertion network as before, namely,

$$\begin{aligned}\mathcal{Q}_0(\bar{y}) &\stackrel{\text{def}}{=} y_1 \geq 0\\ \mathcal{Q}_1(\bar{y}) &\stackrel{\text{def}}{=} (y_2^2 \leq y_1) \wedge (y_3 = y_2^2) \wedge (y_4 = 2 * y_2 + 1)\\ \mathcal{Q}_2(\bar{y}) &\stackrel{\text{def}}{=} (y_2^2 \leq y_1) \wedge (y_3 = (y_2+1)^2) \wedge (y_4 = 2 * y_2 + 1)\\ \mathcal{Q}_3(\bar{y}) &\stackrel{\text{def}}{=} y_2^2 \leq y_1 < (y_2+1)^2.\end{aligned}$$

We already proved in Example 3.3 that it is inductive. The initial condition $\models \varphi \to \mathcal{Q}_0$ holds trivially. Next we choose as wellfounded set the lexicographically ordered product $\mathbb{N} \times \mathbb{N} \times \mathbb{N}$ over the set of nonnegative integers $\mathbb{N}$, and define the ranking functions as follows:

$$\begin{aligned}\rho_0(\bar{y}) &\stackrel{\text{def}}{=} (y_1 + 1, 0, 0)\\ \rho_1(\bar{y}) &\stackrel{\text{def}}{=} (y_1 - y_3, y_1 - y_2, 2)\\ \rho_2(\bar{y}) &\stackrel{\text{def}}{=} (max(y_1 - y_3, 0), y_1 - y_2, 1)\\ \rho_3(\bar{y}) &\stackrel{\text{def}}{=} (0, 0, 0).\end{aligned}$$

We now show that $\mathcal{Q}_i$ implies that ρ_i is defined. The only problematic cases are ρ_1 and ρ_2, because the expressions $y_1 - y_3$ and $y_1 - y_2$ may not be defined over the set of nonnegative integers. First, look at ρ_1. We immediately have $y_1 \geq y_2^2 = y_3$ from $\mathcal{Q}_1$. It is also obvious that $y_1 \geq y_2$ follows from $y_1 \geq y_2^2$. Secondly, consider ρ_2. From $\mathcal{Q}_2$, it follows that $y_1 \geq y_2$ holds for the same reason.

What remains to be proved is that the ranking functions decrease along every transition. This is proved as follows:

$\pi_1 : l_0 \to l_1$, the first component of the ranking function decreases at least by one,

$\pi_2 : l_1 \to l_2$, the first and second components of the ranking function do not increase, whereas the third decreases by one,

$\pi_3 : l_2 \to l_1$, the first component of the ranking function does not increase, whereas the second decreases by one,

$\pi_4 : l_2 \to l_3$, the first and second components of the ranking function do not increase, whereas the third decreases by one. □

5.3 Soundness and Semantic Completeness

Next we prove soundness and semantic completeness of the method.

Theorem 5.4 (Soundness)
Let $P = (L, T, s, t)$. If $\mathcal{Q}$ is an inductive assertion network for P, ρ is a ranking function network over the wellfounded set $(W, \prec)$ satisfying points 3 and 4 of Definition 5.1, and $\models \varphi \to \mathcal{Q}_s$, then P is φ-convergent.

Proof
Proving soundness is straightforward. Just consider an arbitrary φ-computation

$$\eta : \langle l_0; \sigma_0 \rangle \longrightarrow \langle l_1; \sigma_1 \rangle \longrightarrow \ldots,$$

with $l_0 = s$, then inductiveness of $\mathcal{Q}$ implies that $\mathcal{Q}$ is invariant, and hence $\models \varphi \to \mathcal{Q}_s$ implies, using conditions 3 and 4 of Definition 5.1, that $\rho_{l_0}(\sigma_0), \rho_{l_1}(\sigma_1), \ldots$ are all defined. Furthermore, the chain

$$\rho_{l_0}(\sigma_0) \succ \rho_{l_1}(\sigma_1) \succ \ldots$$

is decreasing. Due to the wellfoundedness of W, the above chain is finite, and, hence, η is also finite.

■

The completeness proof is more complicated. Define a tree to be of *finite degree* if each of its nodes has no more than a finite number of direct descendant nodes (children). First, we prove the following lemma due to König [Kön32].

Lemma 5.5 (König's lemma) An infinite tree of finite degree must have an infinite path.

Proof
Let n_0 be the root of an infinite tree T of finite degree. Let the descendants of n_0 be $n_1, \ldots, n_m$. Each node $n_i, i = 1, \ldots, m$ is the root of a subtree T_i. Since the number of nodes in the complete tree is infinite and there are finitely many T_i's, at least one of them must contain an infinite number of nodes. Let n_{i_1} be the root of a subtree T_{i_1} which is infinite. We now repeat the argument with respect to n_{i_1} and its immediate descendants $n'_1, \ldots, n'_t$. At least one of them

must be the root of an infinite tree. Let us denote it by n_{i_2}. Repeating the argument we trace in the tree T a path

$$n_0, n_{i_1}, n_{i_2}, \dots$$

of nodes each of which is the root of an infinite tree. The process will never terminate since we are continuously examining roots of infinite subtrees. Consequently the traced path is an infinite path in T.

■

An immediate corollary of König's lemma is that *a tree of finite degree which has no infinite paths must be finite.* In other words, in such a case there exists a constant such that all paths in the tree are not longer than that constant. We need this result to define the ranking functions.

Theorem 5.6 (Semantic completeness)
If P is φ-convergent, then there exist assertion and ranking-function networks satisfying the verification conditions for proving convergence.

Proof
Let $P = (L, T, s, t)$. We choose the same assertion network $\mathcal{Q}$ as in the semantic completeness proof of the partial correctness method, namely, the one consisting of semantic φ-minimal predicates. By Theorem 4.4, it is inductive, and $\models \varphi \rightarrow \mathcal{Q}_s$ holds.

As wellfounded set we choose $(I\!N, <)$, and define the ranking functions $\rho_l : \Sigma \rightarrow W$ as

$$\rho_l(\sigma) \overset{\text{def}}{=} \textit{the length of the longest computation path starting at } l \textit{ in state } \sigma.$$

The two extra conditions, namely, that the minimal predicate $\mathcal{Q}_l$ implies that the ranking function ρ_l is defined, and that the ranking functions decrease along each transition, remain to be shown.

Obviously, if there exists an infinite computation starting at l with state σ then $\rho_l(\sigma)$ is undefined. König's lemma ensures that in the other case, namely, when there is no infinite computation starting at l with σ, $\rho_l(\sigma)$ is indeed defined. To see this, *we construct the computation tree starting from l with σ*. The nodes of the tree are the configurations in the computation, and one node is an immediate descendant of another if and only if it is a configuration which is the result of one transition from the latter. To see why this construction leads to a tree, note that there is always at least one node, namely, the root $\langle l; \sigma \rangle$. The degree of this tree is finite, because one configuration can only lead to a finite number of direct descendant (follow-up) configurations in one step. (This follows from the finiteness of T.) By the assumption that there are no infinite computations from $\langle l;\ \sigma \rangle$, the tree also does not have infinite paths. Therefore, by König's lemma the tree is finite, and, consequently, $\rho_l(\sigma)$ is defined. To establish the first verification condition, we now only need to show that, when the minimal predicate $\mathcal{Q}_l$ holds in state σ, there are no infinite computations from $\langle l; \sigma \rangle$. By the definition of $\mathcal{Q}_l$, there exists a partial computation η starting from a state which satisfies φ and reaching l with σ. Clearly, the partial computation η can be continued by any computation η' starting from $\langle l; \sigma \rangle$. Since P is φ-convergent, this implies that any computation from $\langle l; \sigma \rangle$ is also finite.

Establishing the second condition now becomes straightforward, because it is easy to see that by the definition of ρ_l the value of the defined ranking function decreases by at least one along each transition.

We have shown semantic completeness of the method by using the same assertion network as before and constructing a particular ranking-function network, which satisfies the verification conditions 3 and 4 of Definition 5.1.

■

The kind of completeness shown here is *semantic* completeness, because we (i) define the ranking functions ρ_l mathematically, by giving an existence proof (and not by expressing them, e.g., using first-order predicate logic), and (ii) prove that the verification conditions for proving convergence are valid mathematically, i.e., we did not prove the implications within some formal system.

References

[Kön32] D. König. Theorie der endlichen und unendlichen Graphen. Technical report, Leipzig, 1932.