Appendix 3 to Lecture 20:
Two Symbolic Methods to prove Deadlock Freedom

J. Meseguer

Call an admissible rewrite theory $\mathcal{R} = (\Sigma, E \cup B, R)$ with constructor signature Ω *never terminating* iff it has no deadlocks, i.e., iff for each $t \in T_\Omega$ in $\vec{E}, B$-canonical form there always exists a $t' \in T_\Sigma$ such that $t \rightarrow_{R/B} t'$. Using this notion suggest two methods to symbolically prove that a topmost rewrite theory of the form $\mathcal{R} = (\Sigma, B, R)$ satisfies the deadlock freedom invariant from a symbolic initial state specified by a constructor pattern term $init \in T_\Omega(X)$:

Method 1: Check the Never Terminating Property Automatically

Obviously, if $\mathcal{R} = (\Sigma, B, R)$ is never terminating, it will automatically satisfy the deadlock freedom invariant from *any* symbolic initial state $init \in T_\Omega(X)$. If the rules $(l \rightarrow r) \in R$ are all *left-linear*, i.e., each variable x in l appears at a single position p in l, checking whether $\mathcal{R}$ is never terminating becomes decidable, since it reduces to checking that if $R = \{l_i \rightarrow r_i \mid 1 \leqslant i \leqslant k\}$, then the set $\{l_i \mid 1 \leqslant i \leqslant k\}$ is a *generator set* for the topmost sort *State* of $\mathcal{R}$. But this, as explained in pg. 12 of Lecture 15, can be automatically decided by the SCC tool by checking the sufficient completeness of the Maude functional module defining the *sort predicate* $State : State \rightarrow Bool$ associated to supposed generator set $\{l_i \mid 1 \leqslant i \leqslant k\}$.

Since $\mathcal{R}$ satisfying the deadlock free invariant from a symbolic initial state $init \in T_\Omega(X)$ is a *weaker property* than $\mathcal{R}$ being never terminating (since the invariant only involves states reachable from *init*), the above automatic SCC check may fail (so that $\mathcal{R}$ fails to be never terminating), whereas the deadlock free invariant may still hold for some symbolic initial state *init*. What can we do in this case? Several things. But, first of all, note that if the SCC test fails, then the SCC tool will give us a *ground term counterexample* of the form: $State(u)$, which exactly means that u is a *concrete deadlock state*. This opens up a second possibility for trying to *automatically check* that the deadlock freedom invariant fails for the symbolic initial state *init* as follows. Assuming that $\mathcal{R}$ satisfies the additional property that $\forall (l \rightarrow r) \in R,\ vars(l) = vars(r)$, its inverse theory $\mathcal{R}^{-1}$ (see Appendix 2 to Lecture 20) is *excutable by rewriting*, we will have proved that the deadlock freedom invariant fails for *init* if the Maude search command:

$$\texttt{search}\,[1]\ \ u\ \Rightarrow *\ \ init$$

in the system module `mod` $\mathcal{R}^{-1}$ `endm` finds a solution for this search query. But such a search command may not find a solution, even when $\mathcal{R}$ fails to be deadlock-free from *init*, since although u is a deadlock state, it may not be reachable from any of the ground states specified by *init*. Therefore, either failure to find a solution to the query in finite time, or infinite looping searching for such a solution do not allow us to settle whether $\mathcal{R}$ is actually deadlock-free from *init* or not: other methods are needed.

Method 2: Check the Deadlock Freedom Invariant by Narrowing Search

Assuming that the lefthand sides of rules in the topmost rewrite theory $\mathcal{R} = (\Sigma, B, R)$ are left-linear, and that all constructor symbols in such lefthand sides belong to a subsignature $\Sigma_0 \subseteq \Sigma$ of *absolutely free constructors*, i.e., constructors that do not obey any axioms in B, if we fail to prove $\mathcal{R}$ never terminating by the sufficient completeness check described above, we still have another alternative, namely, to specify the *complement* of the set of ground instances of the set of Ω_0-constructor patterns $\{l_i \mid 1 \leqslant i \leqslant k\}$ by another set of Ω_0-constructor patterns, say $\{u_j \mid 1 \leqslant j \leqslant n\}$. Then, $\mathcal{R}$ will be *deadlock free* from a symbolic initial state *init* iff none of the above patterns u_j can be reached from *init* by narrowing search using the `fvu-narrow` command. The terms $\{u_j \mid 1 \leqslant j \leqslant n\}$ can be chosen in two ways:

1. **The easy way**, by using the order-sorted pattern complement algoritm defined in [1].

2. **The hard way**, by actually *guessing* such patterns and then checking two properties automatically:

   (a) **Generation**: that the set $\{l_i \mid 1 \leqslant i \leqslant k\} \cup \{u_j \mid 1 \leqslant j \leqslant n\}$ is a *generator set* of sort *State* by the method already described above.

   (b) **Disjointness**: that for all i, j, $1 \leqslant i \leqslant k$ and $1 \leqslant j \leqslant n$, the equalities $l_i = u_j$ have no unifiers.

References

[1] J. Meseguer and S. Skeirik. Equational formulas and pattern operations in initial order-sorted algebras. *Formal Asp. Comput.*, 29(3):423–452, 2017.