Formal Methods in Software Development
Resume of the 20/11/2019 lesson

Igor Melatti

1 Symbolic Model Checkers: NuSMV

- From a NuSMV model $\mathcal{M}$ (defined with the `ASSIGN` section) to the corresponding Kriepke structure $M = (S, S_0, R, L)$
  - $V = \langle v_1, \ldots, v_n \rangle$ is the set of variables defined inside the `main` module of $\mathcal{M}$, with domains $\langle D_1, \ldots, D_n \rangle$
    - note that each D_i may be the instantiation of other modules
    - in which case, again, all variables must be considered as *unfolded*
    - that is, if a variable v is the instantiation of a module with k variables, then v counts as k variables instead of one
    - if one of such k variables is another instantiation, this procedure must be recursively repeated
    - NuSMV calls this operation *hierarchy flattening*
    - essentially, it is the same as for records in Murphi
    - simple types are the recursion base step
  - $S = D_1 \times \ldots \times D_n$ (as in Murphi)
  - S_0 is defined by looking at `init` predicates
    - $s \in S_0$ iff, for all variables $v \in V$, $s(v) \in \texttt{init}(v)$
    - if $\texttt{init}(v)$ is not specified in $\mathcal{M}$, then any value for v is ok
    - formally, if $s \in S_0$, then also $s' \in S_0$ being $s'(v') = s(v') \forall v' \neq v$
  - R is defined by looking at `next` predicates
    - we assume all `next` predicates to be defined by the `case` construct (if not, simply assume it is the `case` construct with just one `TRUE` condition)
    - for each (flattened) variable v, we name $g_1(v), \ldots g_{k_v}(v)$ the conditions (guards) of the `case` for `next(`v`)`, and $a_1(v), \ldots a_{k_v}(v)$ the resulting values (actions) of the `case` for `next(`v`)`
    - note that, by NuSMV syntax, each $a_i(v)$ is actually a set (possibly a singleton)

    * $(s, s') \in R$ iff, for all variables $v \in V$, if $g_i(s(v)) \wedge \forall j < i \neg g_j(s(v))$ then $s'(v) \in a_i(v)$
    * that is, s may go in s' iff, for all variables v, if the values of v in s satisfy the guard g_i (and none of the preceding guards for the same variable), then the value of v in s' is one of the values specified by the `case` for guard g_i

- $AP = \{(v = d) \mid v = v_i \in V \wedge d \in D_i\}$
- $(v = d) \in L(s)$ iff variable v has value d in s
- If, instead, the NuSMV model $\mathcal{M}$ is defined with the `TRACS` section, then
  - $V = \langle v_1, \ldots, v_n \rangle$ is the set of variables as above and $S = D_1 \times \ldots \times D_n$
  - S_0 is defined by looking at `INIT` section
    * $s \in S_0$ iff, for all variables $v \in V$ and for all `INIT` sections I, $I(s(v))$ holds
  - R is defined by looking at `TRANS` section
    * $(s, s') \in R$ iff, for all variables $v \in V$ and `TRANS` sections T, $T(s(v), s'(v))$ holds