Formal Methods in Software Development
Resume of the 14/10/2020 lesson

Igor Melatti and Ivano Salvo

- Summing up, the relations between Kriepke structures and Murphi models are as follows. We are given a Murphi model, let's call it $\mathcal{M}$ and let's assume that:
  - $V = \langle v_1, \ldots, v_n \rangle$ is the set of global variables of $\mathcal{M}$, with domains $\langle D_1, \ldots, D_n \rangle$
    * note that each D_i may be a Cartesian product of other domains (if the corresponding type is an array or a record)
    * question: which is the difference, in terms of the definition of the domain, between an array and a record?
    * however, in the following we will take a different road: we will consider all variables *unfolded*
    * that is, if a variable is an array with q elements, then it is actually to be considered as q different variables
    * the same for records
    * for combination of arrays and records, all possible combinations must be considered
    * simple types are ok
    * as an example: `var a : array [1..n] of record begin b : 1..m; c: 1..k; endrecord`
    * then there will be $2n$ variables as follows: $a1b, \ldots, anb, a1c, \ldots, anc$
  - let $I = \{I_1, \ldots, I_k\}$ be the set of `startstate` sections in $\mathcal{M}$
    * startstates may be defined inside rulesets; here we suppose all rulesets are *unfolded*
    * thus, if a startstate I is inside m nested rulesets $\mathcal{R}_1, \ldots, \mathcal{R}_m$, and each ruleset $\mathcal{R}_i$ is defined on an index j_i spanning on a domain $\mathcal{D}_i$ (note that $\mathcal{D}_i$ must be a simple type), then there actually are $\prod_{l=1}^{m} |\mathcal{D}_l|$ startstates to be considered, instead of just one
    * of course, in each of these startstates definitions, the tuple $j_1, \ldots, j_m$ takes all possible values of $\mathcal{R}_1 \times \ldots \times \mathcal{R}_m$
  - let $T = \{T_1, \ldots, T_p\}$ be the set of `rule` sections

        * same as above: must be *unfolded* if in rulesets

- Then, the Kriepke structure $M = (S, S_0, R, L)$ described by $\mathcal{M}$ is such that:
  - $S = D_1 \times \ldots \times D_n$
  - $s \in S_0$ iff s may be obtained by applying the body of a startstate in I
  - $(s, t) \in R$ iff there is a rule $T_i \in T$ s.t. T_i guard is true in s and T_i body changes s to t
    * that is: in the body of T_i, variables starting values are those of s
    * note that there may be two or more rules defining the same transition from s to t; no problem with this
    * note that there is no assurance that R is total: Murphi can check this at run-time
      · "total" means that every state has at least a successor
      · if this is not true, i.e., if a state s does not have successors, Murphi calls s a *deadlock* state
      · note that there may exist deadlock states that are not reachable from the initial states: Murphi cannot find them
      · a state s is a deadlock state for two possible reasons:
        1. $(s, t) \notin R$ for all $t \in S$, i.e., the values for the variables in s do not satisfy any ruleset guard
        2. $(s, t) \in R \rightarrow t = s$, i.e., there is some ruleset guard which is satisfied by s, but its body do not change any of the global variables (e.g., the body is empty)
  - $AP = \{(v = d) \mid v = v_i \in V \wedge d \in D_i\}$
  - $(v = d) \in L(s)$ iff variable v has value d in s