About the Beamer class in presentation making

A short story

E. Cominato 137396[1]

[1]Dipartimento di Scienze Matematiche, Informatiche e Fisiche
Università degli studi di Udine

Very Large Conference, April 2013

Discrete case

Definition 1 (*Plant*)

A plant automaton is a tuple $\mathcal{P} = (Q, \Sigma_c, \delta, q)$ where Q is a finite set of states, Σ_c is a set of controller commands, $\delta : Q \times \Sigma_c \longmapsto 2^Q$ is the transition function and $q_o \in Q$ is an initial state.

Definition 2 (*Controllers*)

A controller (strategy) for a plant specified by $\mathcal{P} = (Q, \Sigma_c, \delta, q)$ is a function $C : Q^+ \longmapsto \Sigma_c$. A simple controller is a controller that can be written as a function $C : Q \longmapsto \Sigma_c$.

We are interested in the simpler cases of controllers that base their decisions on a finite memory.

Definition 3 (*Trajectories*)

Let $\mathcal{P}$ *be a plant and let* $C : Q^+ \longmapsto \Sigma_c$ *be a controller. An infinite sequence of states* $\alpha : q[0], q[1], \ldots$ *such that* $q[0] = q_0$ *is called a trajectory of* $\mathcal{P}$ *if*

$$q[i+1] \in \bigcup_{\sigma \in \Sigma_c} \delta(q[i], \sigma)$$

and a C-trajectory if $q[i+1] \in \delta(q[i], C[\alpha[0..i]])$ *for every* $i \geq 0$. *The corresponding sets of trajectories are denoted by* $L(\mathcal{P})$ *and* $L_C(\mathcal{P})$.

For every infinite trajectory $\alpha \in L(\mathcal{P})$:

- $\mathit{Vis}(\alpha)$ denote the set of all states appearing in α
- $\mathit{Inf}(\alpha)$ denote the set of all states appearing in α infinitely many times

Definition 4 (*Acceptance Condition*)

Let $\mathcal{P} = (Q, \Sigma_c, \delta, q)$ be a plant. An acceptance condition for $\mathcal{P}$ is

$$\Omega \in \{(F, \Box), (F, \Diamond), (F, \Diamond\Box), (F, \Box\Diamond), (\mathcal{F}, \mathcal{R}_n)\}$$

where $\mathcal{F} = \{(F_i, G_i)\}_{i=1}^{n}$ and F, F_i and G_i are certain subsets of Q referred as the good states. The set of sequences of $\mathcal{P}$ that are accepted accordig to Ω is defined as follows:

$L(\mathcal{P}, F, \Box)$	$\{\alpha \in L(\mathcal{P}) : Vis(\alpha) \subseteq F\}$	α always remains in F
$L(\mathcal{P}, F, \Diamond)$	$\{\alpha \in L(\mathcal{P}) : Vis(\alpha) \cap F \neq \emptyset\}$	α eventually visits F
$L(\mathcal{P}, F, \Diamond\Box)$	$\{\alpha \in L(\mathcal{P}) : Inf(\alpha) \subseteq F\}$	α eventually remains in F
$L(\mathcal{P}, F, \Box\Diamond)$	$\{\alpha \in L(\mathcal{P}) : Inf(\alpha) \cap F \neq \emptyset\}$	α visits F infinitely often
$L(\mathcal{P}, \mathcal{F}, \mathcal{R}_n)$	$\{\alpha \in L(\mathcal{P}) : \exists i \alpha \in L(\mathcal{P}, F_i, \Box\Diamond) \cap L(\mathcal{P}, G_i, \Diamond\Box)\}$	α visits F_i infinitely often and eventually stays in G_i

Definition 5 (*Controller Synthesis Problem*)

For a plant $\mathcal{P}$ and an acceptance condition Ω, the problem ***Synth***$(\mathcal{P}, \Omega)$ *is: Find a controller C such that $L_C(\mathcal{P}) \subseteq L(\mathcal{P}, \Omega)$ ot otherwise show that such a controller does not exists.*

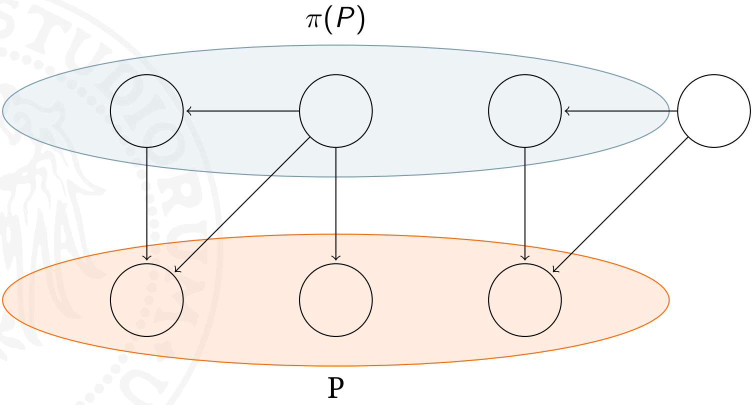
Definition 6 (*Controllable Predecessors*)

Let $\mathcal{P} = (Q, \Sigma_c, \delta, q)$ be a plant and a set of states $P \subseteq Q$. The controllable predecessors of P is the set of states from which the controller can "force" the plant into P in one step:

$$\{q : \exists \sigma \in \Sigma_c \,.\, \delta(q, \sigma) \subseteq P\}$$

We define a function $\pi : 2^Q \longmapsto 2^Q$, mapping a set of states $P \subseteq Q$ into the set of its Controllable predecessors:

$$\pi(P) = \{q : \exists \sigma \in \Sigma_c \,.\, \delta(q, \sigma) \subseteq P\}$$


$\pi(P)$
P

Discrete case

Theorem 1

For every $\Omega \in \{(F, \Box), (F, \Diamond), (F, \Diamond\Box), (F, \Box\Diamond), (\mathcal{F}, \mathcal{R}_n)\}$ *the problem* ***Synth***$(\mathcal{P}, \Omega)$ *is solvable. Moreover, if* $(\mathcal{P}, \Omega)$ *is controllable then it is controllable by a simple controller.*