On the synthesis of provably correct discrete controllers

Jüri Vain

Dept. of Computer Science/Institute of Cybernetics

Tallinn University of Technology

Controller synthesis problem (I)

- Given:
  - a dynamical system *P* (plant) with all its possible behaviors
  - a subset of plant's behaviors, defined as good (acceptable)
- Find:
  - a controller *C* interacting with *P* by observing the state of *P* and by issuing control actions that influence the behavior of *P* restricting it to be subset of good behaviors

Controller synthesis problem (II)

- CSP formulations differ in the kind:
  - How dynamics is considered
  - How acceptability criteria are specified
- Two extreme examples:
  - Reactive program synthesis
  - Classical control theory

CSP as reactive program synthesis problem

- Models base on discrete TS-s (automata):
  - Plant represents reactions to environment and control actions.
  - The program has control over some of the transitions (non-determinism).
- Control problem:

  find at each (plant's) state one among possible transitions s.t. exclude 'bad' behaviors.

CSP in classical control theory

- Models base on differential equations
  - The plant is a *continuous* dynamical system.
  - Plant's inputs express the non-determinism of environment (disturbances) and the effects of controller actions.
- Control problem:

  define a *feed-back law,* which *continuously* determines inputs to *P* s.t. specification is met.

Current approach to CSP

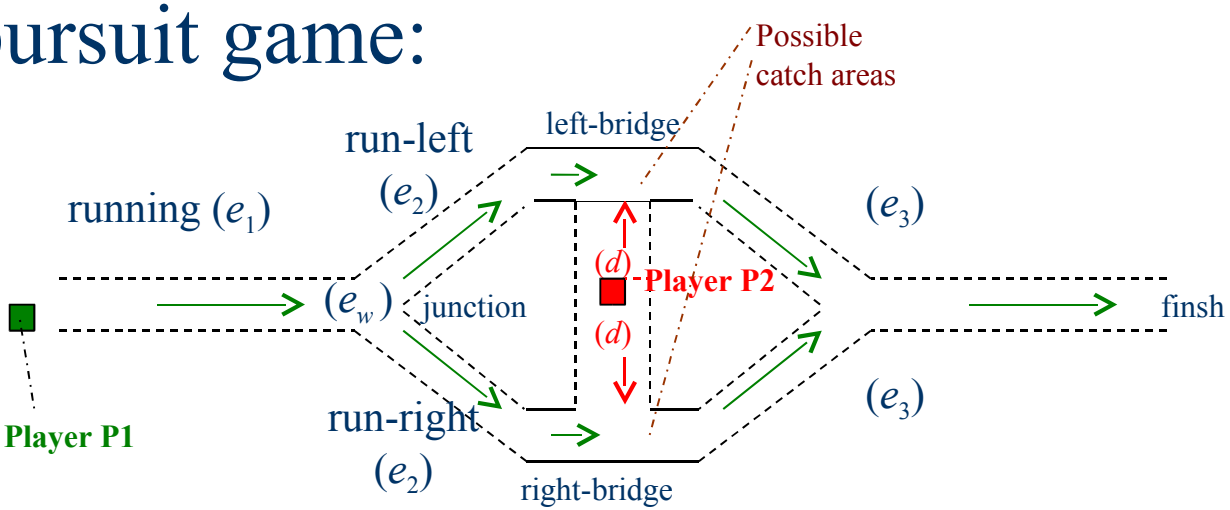
- Given:
  - Plant model (*timed automaton* T_P):
    - Discrete state transitions
    - Continuous passage of time
  - Correctness criteria φ stated in TL
- Find:

  the controller automaton T_C s.t.

$$T_P \,||\, T_C \,|\!=\, \varphi$$

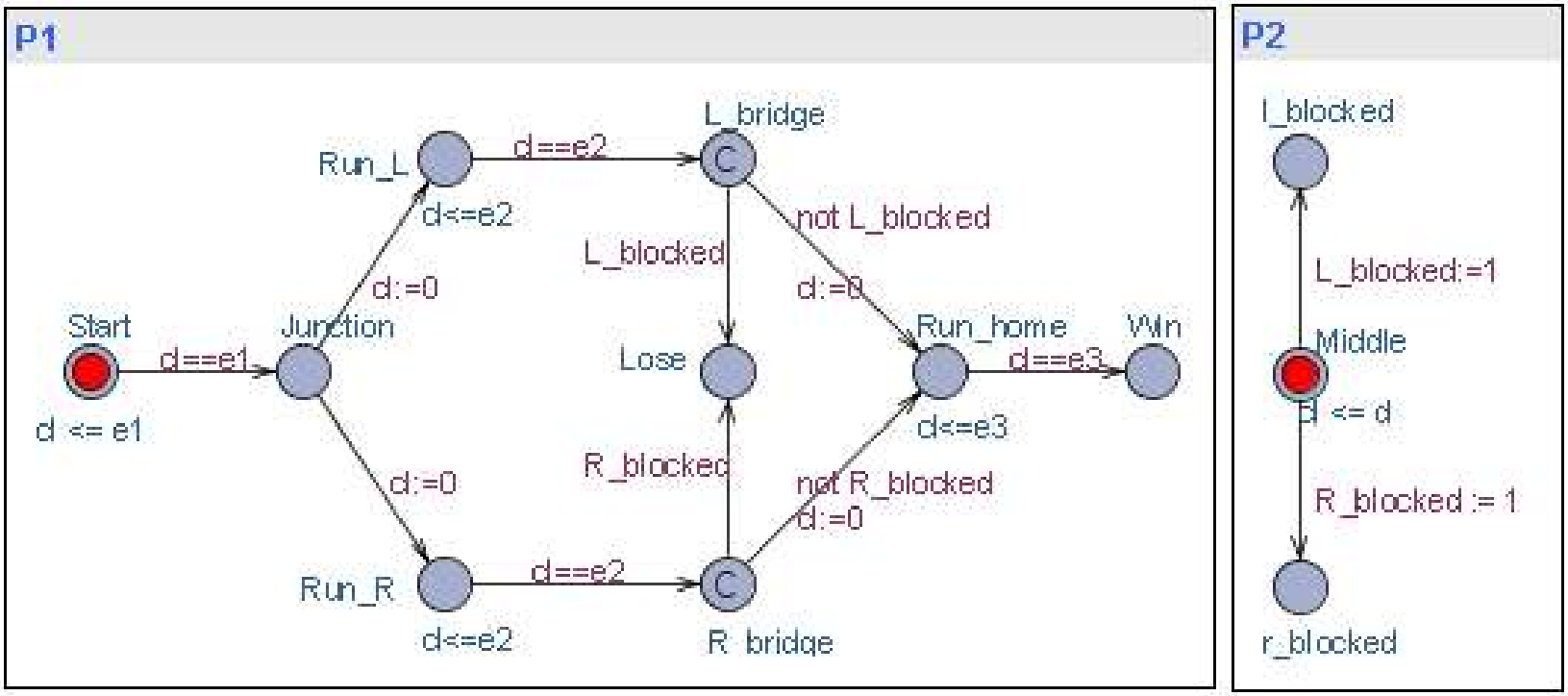
An example: RT game

" A pursuit game:



- Player P1 wins if finish is reached with $< c$ *sec*
- Winning strategy exists *iff* $max(d, e_1) + e_2 + e_3 < c$
- Strategy for P1: stay in junction until $max(d, e_1)$.

The game as two interacting TA



The discrete case

- Plant (automaton): $\mathsf{P} = (Q, \Sigma_c, \delta, q_0)$
  - Q – finite set of states
  - Σ_c – set of controller commands
  - δ – transition relation : $Q \times \Sigma_c \rightarrow 2^Q$
  - q_0 – initial state
- Controller automaton C for plant P implements a function C: $Q \rightarrow \Sigma_c$
- Memoryless controllers observe only current state of P, i.e.,

$$\forall q \in Q, w, w' \in Q^*, C(w\,q) = C(w'\,q)$$

Trajectories

- $L(\mathsf{P})$ - set of all (infinite) trajectories
- $L_c(\mathsf{P})$ – set of controlled trajectories

$$L_c(\mathsf{P}) \subseteq L(\mathsf{P})$$

How to define good trajectories?

Let for each $\alpha \in L(\mathsf{P})$:

$Vis(\alpha)$ – all states appearing in α.

$Inf(\alpha)$ – states appearing infinitely often in α.

Acceptance condition for P

” $\Omega \in \{(F,\Diamond), (F, \ ), (F,\Diamond\ ), (F, \ \Diamond)\}$,

where $F \subseteq Q$ (‘good’ state)

$L(\mathsf{P}, F, \ ) = \{\alpha \in L(\mathsf{P}): Vis(\alpha) \subseteq F\}$

$L(\mathsf{P}, F, \Diamond) = \{\alpha \in L(\mathsf{P}): Vis(\alpha) \cap F \neq \varnothing\}$

$L(\mathsf{P}, F, \Diamond\ ) = \{\alpha \in L(\mathsf{P}): Inf(\alpha) \subseteq F\}$

$L(\mathsf{P}, F, \ \Diamond) = \{\alpha \in L(\mathsf{P}): Inf(\alpha) \cap F \neq \varnothing\}$

CSP

” *Problem* **Synth**(P, Ω):

Find a controller C *s.t.* $L_c(\mathsf{P}) \subseteq L(\mathsf{P}, \Omega)$,

otherwise show that such C does not exist.

Theorem (Maler, Pnueli, Sifakis):

For every Ω the problem **Synth**(P, Ω) is decidable.

If (P, Ω) is controllable then it is controllable by a simple (memoryless) controller.

Sketch of proof (I)

- *Def. Controllable predecessors* of a state P is a set of states from which the controller can force the plant into P in one step:

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

- *Def. Winning states* W – states from which a controller C can enforce good behaviors (according to Ω).

Sketch of proof (II)

- Set W can be characterized by *fp* expressions:
  - $: \nu\, W(F \cap \pi(W))$ (1) ν - greatest *fp*
  - $\Diamond: \mu\, W(F \cup \pi(W))$ (2) μ - least *fp*
  - $\Diamond : \mu W \nu H(\pi(H) \cap (F \cup \pi(W)))$ (3)
  - $\Diamond: \nu W \mu H(\pi(H) \cup (F \cap \pi(W)))$ (4)

Sketch of proof (III)

- *For a given plant* P *and* π *it is straightforward to calculate W using* (1) - (4).
- Procedurally:

◊: $W_0 := \varnothing$

for $i = 0, 1, \ldots,$ **repeat**

$W_{i+1} := F \cup \pi(W_i)$

until $W_{i+1} = W_i$

: $W_0 := Q$

for $i = 0, 1, \ldots,$ **repeat**

$W_{i+1} := F \cap \pi(W_i)$

until $W_{i+1} = W_i$

Sketch of proof (IV)

- *The sequences of W_i are monotone over a finite domain*
- $\Rightarrow$ *convergence is guaranteed.*
- *Define the controller at q as $C(q) = \sigma$ if* $\exists \sigma \in \Sigma_c$ *s.t.* $\delta(q, \sigma) \subseteq W_i$
- The plant is controllable iff $q_0 \in W$.
- *When the process terminates the controller is synthesized for all winning states.*

Timed case (I)

- Timed automaton:

$$\mathcal{T} = (Q, X, \Sigma, I, G, R, q_0)$$

Q – *set of locations*

$X = (\boldsymbol{R}^{+d})$ – *clock domain* — *d - number of clocks*

$\Sigma = \Sigma_c \cup \{e\}$ — *e – environment action*

$I: Q \mapsto H_k$ — H_k – subregions of X

R - clock resets

$R \subseteq Q \times \Sigma \times G \times 2^C \times Q,$ — *where* C *– set of clocks*

Timed case (II)

- Timed trajectory
  - *Configuration*: $(q, \boldsymbol{x}) \in Q \times X$
  - *Transition* - pair of configurations $((q,\boldsymbol{x}),(q', \boldsymbol{x}'))$ s.t. either
    - *t-trasition*: $q = q'$ *and* $\exists t \in T.\ \boldsymbol{x}' = \boldsymbol{x} + \mathbf{1}t,\ \boldsymbol{x} \in I_q$ or
    - σ-*transition*: $\exists r \in R.\ \boldsymbol{x} \in g$ *and* $\boldsymbol{x}' = \boldsymbol{x}|_{xr=0}$
  - *Trajectory – sequence of configurations* $\langle (q_i, \boldsymbol{x}_i), i \geq 0 \rangle$ *s.t. for every* i $((q_i, \boldsymbol{x}_i), (q_{i+1}, \boldsymbol{x}_{i+1}))$ is a transition.

Timed case (III)

- *Simple timed controller*:

  $C: Q \times X \mapsto \Sigma_c^{\perp}$ $\qquad\qquad$ $\Sigma_c^{\perp} = \Sigma_c \cup \{\perp\}$

  $\forall \sigma \in \Sigma_c^{\perp}: C^{-1}(\sigma)$ is a polyhedral set

- Controlled trajectory: given a simple controller *C,* a pair $((q, \boldsymbol{x}),(q', \boldsymbol{x}'))$ is a *C-transition* if it is either
  - *e-transition* *or*
  - σ *-transition s.t.* $C(q, \boldsymbol{x}) = \sigma \in \Sigma_c$ *or*
  - *t-transition for some* $t \in T$ *s.t.* $\forall t' \in [0,t)$ $C(q, \boldsymbol{x} + \mathbf{1}t') = \perp$
- *C-trajectory* consists of *C*- transitions.

RT-CSP

" Given TA $\mathcal{T}$ and an acceptance condition Ω,

RT-Synth$(\mathcal{T}, \Omega)$**:**

find a controller C s.t. $L_C(\mathcal{T}) \subseteq L(\mathcal{T}, \Omega)$.

Def. (Extended transition relation):

$\forall t,\sigma \in T, \Sigma_c^{\perp}$. $\delta((q, \boldsymbol{x}),(t, \sigma)) = \{(q', \boldsymbol{x}')$ s.t. $(q', \boldsymbol{x}')$ is a (t, σ)-*successor* **or** (t', e)- *successor* of $(q, \boldsymbol{x})$ for some $t' \in [0, t]\}$.

As for discrete case, define π that indicates the configurations from which the controller can force the automaton into a given set of configurations.

Def. (*Controllable predecessor* π):

$$\forall K \subseteq Q \times X: \quad \pi(K) = \{(q, \boldsymbol{x}): \exists t, \sigma \in T, \Sigma_c^{\perp}.\ \delta((q, \boldsymbol{x}),(t, \sigma)) \subseteq K\}$$

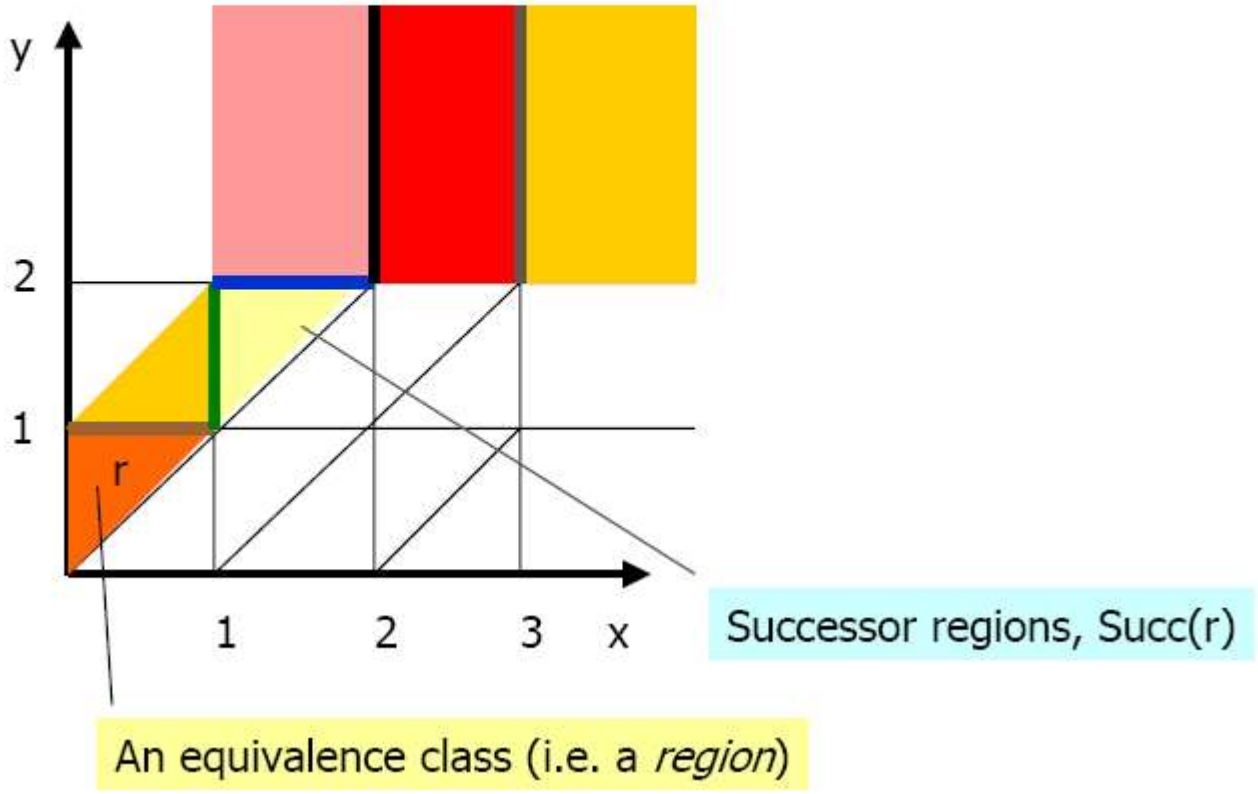
How to compute?

- Any set of configurations K can be expressed by a set tuple $K = \langle P_0 \times \ldots \times P_m \rangle$, *where* $P_0, \ldots, P_m \subseteq X$ are *polyhedra.*
- We have to show that π always maps a polyhedral set tuple to another polyhedral set tuple.
- Intuitive idea:

  Any predecessor can be efficiently constructed using linear clock constraints.

Thus the set of polyhedral regions $2^Q \times H$ is closed under π.

Successor Operation (wrt delay)



Decidability of RT-CSP

Theorem:

Given a TA $\mathcal{T}$ and an acceptance condition $\Omega \in \{(F,\Diamond), (F, \ ), (F,\Diamond\ ), (F, \ \Diamond)\}$,the problem ***RT-Synth***($\mathcal{T}$, Ω) is decidable.

Scetch of proof:

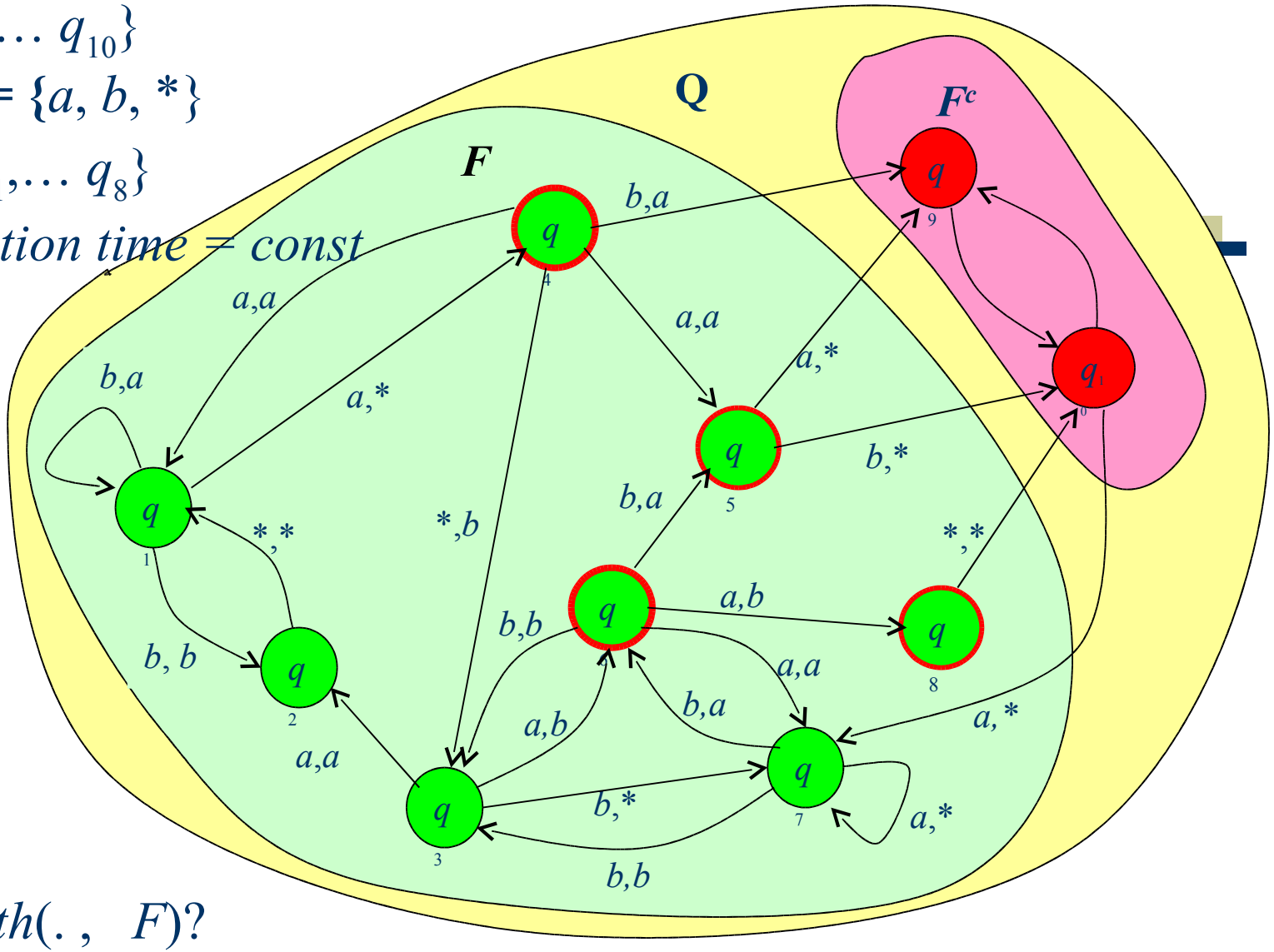
- Any of iterative processes for *fp equations* (1)-(4) starts with an element of $2^Q \times H$, *e.g.,* starts with $W_0 = Q \times F$
- Any iteration applies Boolean operations and π, i.e., every W_i is also an element of $2^Q \times H$ – finite set of linear constrs.
- By monotonicity, a fixed-point is eventually reached.

$\mathbf{Q} = \{q_1, \ldots q_{10}\}$

$\sum_c = D = \{a, b, *\}$

$F = \{q_1, \ldots q_8\}$

t_c – *reaction time = const*



Solve

RT-Synth(. , *F*)?