

## Maximally Permissive Controllers in All Contexts

Stéphane Riedweg

LSV, ENS Cachan

61, avenue du Prsident Wilson  
F-94235 CACHAN Cedex - France  
riedweg@lsv.ens-cachan.fr

and

Sophie Pinchinat

IRISA-INRIA

Campus de Beaulieu  
F-35042, Rennes - France  
Sophie.Pinchinat@irisa.fr

## The Framework

- Discrete Events Systems

deterministic labeled transition systems with atomic propositions

- Centralized Supervision and Complete Observation

- Control Objectives

any **mu-calculus definable** property

- Optimal Control

**maximally permissive** controllers

We turn control problems into  $QL_\mu$  model-checking

## Processes for Systems

$$\mathcal{S} = \langle S, s^0, t, L \rangle \text{ on } \Gamma \subseteq AP$$

where

- $\Sigma = \{a, b, \dots\}$  events
- $AP = \{p, p', c, c', \dots\}$  (atomic) propositions
- $S, s^0 \in S$  set of states, initial state,
- $t : S \times \Sigma \rightarrow S$  transition (partial) function,
- $L : S \rightarrow 2^\Gamma$  labeling of states by propositions

## Centralized Supervision with Complete Observation

- $\Sigma = \Sigma_{uc} \uplus \Sigma_c$
- Controllers
- Admissible Controllers
- $\mathcal{C}$  is an admissible controller of  $\mathcal{S}$  for property  $P$  whenever  
 $\mathcal{C}$  is admissible and  $\mathcal{S} \times \mathcal{C}$  satisfies the property  $P$

## Synchronous Product

- $\mathcal{S}_1 = \langle S_1, s_1^0, t_1, L_1 \rangle$  on  $\Gamma_1$  and  $\mathcal{S}_2 = \langle S_2, s_2^0, t_2, L_2 \rangle$  on  $\Gamma_2$   
(with  $\Gamma_1 \cap \Gamma_2 = \emptyset$ )

$$\mathcal{S}_1 \times \mathcal{S}_2 = \langle S_1 \times S_2, (s_1^0, s_2^0), t, L \rangle$$

$t((s_1, s_2), a) = (s'_1, s'_2)$  whenever

$$\begin{cases} s'_1 = t_1(s_1, a), \text{ and} \\ s'_2 = t_2(s_2, a) \end{cases}$$

$$L(s_1, s_2) = L_1(s_1) \cup L_2(s_2)$$

## Centralized Supervision with Complete Observation

- $\Sigma = \Sigma_{uc} \uplus \Sigma_c$
- Controllers
- Admissible Controllers
- $\mathcal{C}$  is an admissible controller of  $\mathcal{S}$  for property  $P$  whenever  
 $\mathcal{C}$  is admissible and  $\mathcal{S} \times \mathcal{C}$  satisfies the property  $P$

## The Property $P$

- Mu-calculus  $L_\mu$   
[Kozen 1983], [Arnold & Niwinski 2001]
- Fix-point operators to build your own modalities  
 $L_\mu$  subsumes CTL, LTL, CTL\*, ...  
Safety, Liveness, Fairness, ...
- Equivalent to (parity) tree automata  
[Emerson & Jutla 1991]

## The Mu-Calculus $L_\mu$ and the Quantified Mu-Calculus $QL_\mu$

- Syntaxe of  $L_\mu$      $\top \mid p \mid \neg\beta \mid \beta \vee \beta' \mid \langle a \rangle\beta \mid X \mid \mu X.\beta(X)$

where  $p \in AP$ ,  $a \in \Sigma$ , conditions on  $\beta(X)$ , ...

- Syntaxe of  $QL_\mu$      $\exists c.\alpha \mid \neg\alpha \mid \alpha \vee \alpha' \mid \beta$

where  $c \in AP$  and  $\beta \in L_\mu$

- Semantics of  $QL_\mu$      $\mathcal{S}, s \models \beta$  for “state  $s$  of  $\mathcal{S}$  satisfies  $\beta$ ”

## The Semantics of $QL_\mu$

### ■ $L_\mu$ -formulas

$\mathcal{S}, s \models p$  iff  $s$  has label  $p$

$\mathcal{S}, s \models \langle a \rangle \beta$  iff there exists  $s' \in S$ ,  $\begin{cases} t(s, a) = s' \\ \mathcal{S}, s' \models \beta \end{cases}$

$\mathcal{S}, s \models \mu X. \beta(X)$  is technical ....

$[a]\alpha \stackrel{\text{def}}{=} \neg \langle a \rangle \neg \alpha$  means “if any  $a$ -successors then it satisfies  $\alpha$ ”

$\mu X. \langle a \rangle X \vee m$  means “some  $a^*$  trace reaches a marked state”

$\text{INV}(\alpha) \stackrel{\text{def}}{=} \neg \mu X. \neg ([ ] \neg X \wedge \beta)$  means “property  $\alpha$  is invariant”

### ■ Formulas $\exists c. \alpha$

we need a  $c$ -labeling process  $\mathcal{E} = \langle E, \varepsilon^0, t', L' \rangle \in \text{Lab}_c$

## $c$ -labeling Processes for the Semantics of $\exists c.\alpha$

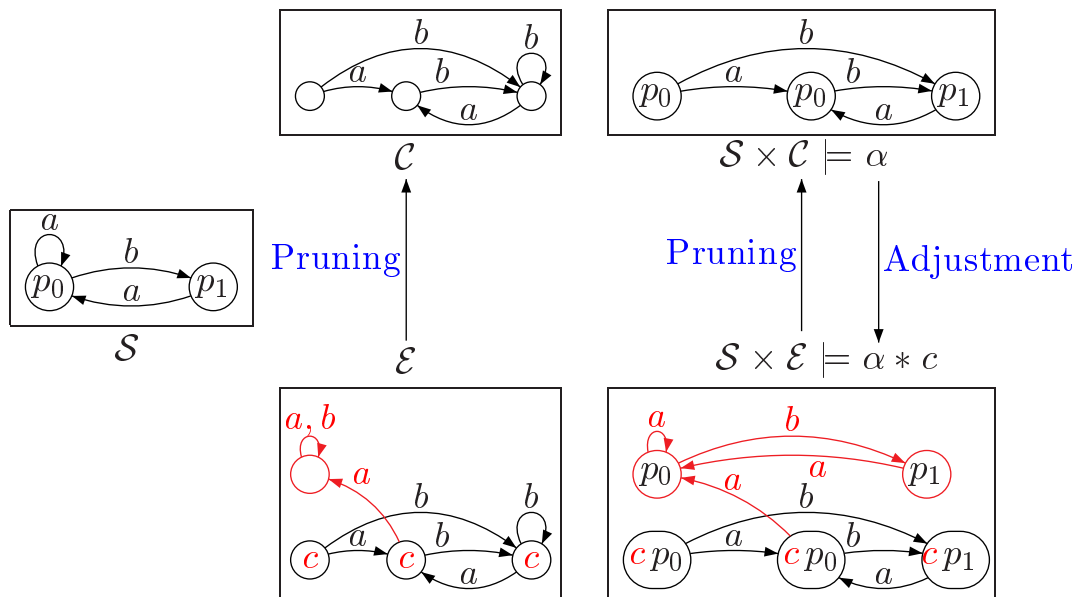
- A  $c$ -labeling process is a complete process  $\mathcal{E} = \langle E, \varepsilon^0, t', L' \rangle$  on  $\{c\}$ .  
 $Lab_c$  the set of  $c$ -labeling processes.

- Process  $\mathcal{S}$  on  $\Gamma$  and  $\mathcal{E} \in Lab_c$ ,  $\mathcal{S} \times \mathcal{E}$  is a  $c$ -labeling of  $\mathcal{S}$   
that is (an unfolding of)  $\mathcal{S}$  with the new label  $c$  put somehow.

- $\mathcal{S}, s \models \exists c.\alpha$  iff there exists  $\mathcal{E} = \langle E, \varepsilon^0, t', L' \rangle \in Lab_c$  s.t.

$$\mathcal{S} \times \mathcal{E}, (s, \varepsilon^0) \models \alpha$$

# Pruning and Adjustment



1

## The Proposition for Pruning and Adjustment

- $\mathcal{E}_{\rightarrow c}$  is the  $c$ -pruning of  $\mathcal{E}$   
just keep states of  $\mathcal{E}$  that remain “inside  $c$ ” and forget proposition  $c$
- $\alpha * c$  is the  $c$ -adjustment of  $\alpha$   
adjust the formula to talk only about states which remain inside  $c$   
 $(\langle a \rangle \dots) * c \stackrel{\text{def}}{=} \langle a \rangle (c \wedge \dots)$

$$\mathcal{S} \times \mathcal{E}_{\rightarrow c} \models \alpha \text{ iff } \mathcal{S} \times \mathcal{E} \models \alpha * c$$

## The Theorem for Basic Controllers

Write  $\text{Controller}(c) \in L_\mu$  for  $c \wedge \text{INV}(c \Rightarrow \bigwedge_{u \in \Sigma_{uc}} [u]c)$

For any  $\alpha \in QL_\mu$ ,

there exists an admissible controller  $\mathcal{C}$  of  $\mathcal{S}$  for  $\alpha$

if and only if

$$\mathcal{S} \models \exists c. \text{Controller}(c) \wedge \alpha * c$$

## Proof sketch for $\Leftarrow$ )

There exists an admissible controller  $\mathcal{C}$  of  $\mathcal{S}$  for  $\alpha$

$\Leftrightarrow$

$$\mathcal{S} \models \exists c. \text{Controller}(c) \wedge \alpha * c$$

1.  $\mathcal{S} \times \mathcal{E} \models \text{Controller}(c) \Leftrightarrow \mathcal{E}_{\rightarrow c}$  is admissible.
2.  $\mathcal{S} \times \mathcal{E} \models \alpha * c \Leftrightarrow \mathcal{S} \times \mathcal{E}_{\rightarrow c} \models \alpha$

## The Theorem for Maximally Permissive Controllers

For any  $\alpha \in QL_{\mu}$ ,

there exists a controller of  $\mathcal{S}$  for  $\alpha$  which is **maximally permissive**

iff

$$\mathcal{S} \models \exists c. \underbrace{\text{Controller}(c) \wedge \alpha * c}_{\text{Solution}(c, \alpha)} \wedge \forall c'. [c \sqsubseteq c' \Rightarrow \neg \text{Solution}(c', \alpha)]$$

## Proof sketch

- $\mathcal{C}$  is more permissive than  $\mathcal{C}'$  whenever  $\mathcal{S} \times \mathcal{C}' \preceq \mathcal{S} \times \mathcal{C}$   
( $\preceq$  means there exists a simulation)

- $c \sqsubseteq c'$  means “if a state is labeled by  $c$ , it is also labeled by  $c'$ ”

$$c \sqsubseteq c' \stackrel{\text{def}}{=} ([ ]\text{INV}(c')) * c \quad \text{and} \quad c \sqsubseteq c' \stackrel{\text{def}}{=} (c \sqsubseteq c') \wedge \neg(c' \sqsubseteq c)$$

- In fact,

$$\mathcal{S} \times \mathcal{E} \times \mathcal{E}' \models c \sqsubseteq c' \Leftrightarrow \mathcal{S} \times \mathcal{E}_{\rightarrow c} \preceq \mathcal{S} \times \mathcal{E}'_{\rightarrow c'}$$

## Conclusion

- Logical Characterization of Maximal Permissiveness  
for mu-calculus definable control objectives

- Decision and Synthesis

1. build a parity tree automaton  $\mathcal{A}$  for

$$\exists c. \text{Solution}(c, \alpha) \wedge \forall c'. [c \sqsubseteq c' \Rightarrow \neg \text{Solution}(c', \alpha)]$$

2. compute the parity game  $G(\mathcal{A}, \mathcal{S})$

3. find a winning strategy (if any)

4. derive  $\mathcal{E}$  from the strategy and compute  $\mathcal{C} = \mathcal{E}_{\rightarrow c}$

- Complexity  $O(|\mathcal{S}|^{2^{|\alpha|}} 2^{2^{|\alpha|}})$