

# Specifying and Synthesizing Open Systems and their Controllers

Sophie Pinchinat  
MARIE CURIE FELLOW

Computer Sciences Laboratory - RSISE  
The Australian National University  
Canberra - Australia

IRISA - INRIA  
Campus de Beaulieu  
F-35042, Rennes - France

[Sophie.Pinchinat@{irisa.fr,rsise.anu.edu.au}](mailto:Sophie.Pinchinat@{irisa.fr,rsise.anu.edu.au})

# Plan

- The Framework
- Model-Checking Closed Systems
- Satisfiability
- Synthesis of Decisions for  $\varphi$  over  $\Lambda$
- Expressing Qualitative Properties Decisions
- Expressing Constraints on Decisions
- Perspectives
- Partial Observation

## The Framework

- A fixed set  $\Sigma = \{a, b, c, \dots\}$
- The full tree  $T$
- A labeling  $\lambda : \Sigma^* \rightarrow 2^\Lambda$
- A decision is a labeling  $p : \Sigma^* \rightarrow 2^p$

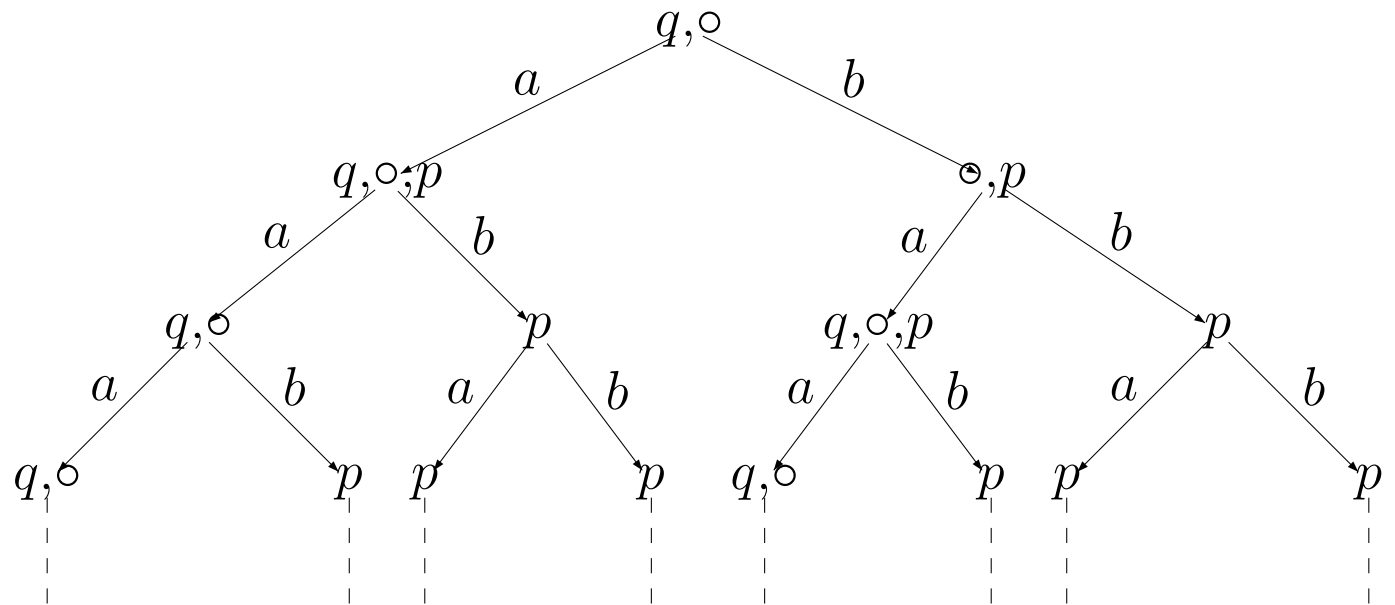
where  $\Lambda$  is a set of propositions

where  $p$  is a proposition



# Example

$$p : \Sigma^* \rightarrow 2^p$$





## The Logic

- Mu-calculus  $\varphi \rightsquigarrow \varphi \text{ IN } p$ , modalities refer to trajectories “inside  $p$ ”

$$\langle a \rangle \varphi \text{ IN } p = \langle a \rangle (p \wedge \varphi \text{ IN } p)$$

$$q \text{ IN } p = q$$

$$(\neg \varphi) \text{ IN } p = \neg(\varphi \text{ IN } p)$$

$$(\varphi \wedge \varphi') \text{ IN } p = \varphi \text{ IN } p \wedge \varphi' \text{ IN } p$$

$$(\mu Y. \varphi) \text{ IN } p = \mu Y. (\varphi \text{ IN } p)$$

$$Y \text{ IN } p = Y$$

- Theorem  $(T, \lambda) \text{ IN } p \models \varphi$  iff  $(T, \lambda) \models \varphi \text{ IN } p$

- Let  $\mathcal{S}$  be represented by  $(T, \lambda) \text{ IN } \circ$  with  $\Lambda \supseteq \{\circ\}$

$\mathcal{S} \models \varphi$  if and only if  $(T, \lambda) \text{ IN } \circ \models \varphi$  if and only if  $(T, \lambda) \models \varphi \text{ IN } \circ$

## Model-Checking Closed Systems

Given  $\lambda : \Sigma^* \rightarrow \Lambda$  and  $\varphi$  over  $\Lambda$

$$(T, \lambda) \models \varphi?$$

- 1) Provided  $(T, \lambda)$  finitely represented ( $\lambda$  is regular)
- 2) Consider the alternating (parity) tree automaton  $\mathcal{A}_\varphi$
- 3) Find a (memoryless) winning strategy in  $\mathcal{G}((T, \lambda), \mathcal{A}_\varphi)$

## Satisfiability of $\varphi$ over $\Lambda$

Does  $(T, \lambda) \models \varphi$  for some  $\lambda : \Sigma^* \rightarrow \Lambda$  ?

Find a memoryless w.s. in  $\mathcal{G}(\mathcal{A}_\varphi)$

$(T, \emptyset) \models \exists \Lambda. \varphi$  ?

- 1) Compute  $\mathcal{B}$  a ndta equivalent to  $\mathcal{A}_\varphi$  **Simulation Theorem**
- 2) Get  $\mathcal{B}'$  by projecting  $\mathcal{B}$  in order to forget  $\Lambda$
- 3) Find a memoryless w.s. in  $\mathcal{G}((T, \emptyset), \mathcal{B}')$  delivering the propositions in  $\Lambda$

**Complexity: EXPTIME in  $\varphi$  and PTIME in  $\lambda$**

## Synthesizing a Decision for $\varphi$ over $\Lambda$

Does  $(T, \lambda) \text{ IN } p \models \varphi$  for some decision  $p$  ?

Does  $(T, \lambda) \models \varphi \text{ IN } p$  for some decision  $p$  ?

$(T, \lambda) \models \exists p. \varphi \text{ IN } p$  ?

- 1) Compute  $\mathcal{B}'$  as  $\mathcal{A}_{\varphi \text{ IN } p} \downarrow p$       **Simulation Theorem**
- 2) Find a memoryless w.s. in  $\mathcal{G}((T, \lambda), \mathcal{B}')$  delivering the values of  $p$

**Complexity: EXPTIME in  $\varphi$  and PTIME in  $\lambda$**

## Expressing Qualitative Properties of the Decision $p$

$p \sqsubseteq p'$   $\left\{ \begin{array}{l} \text{means labeled by } p \text{ implies labeled by } p' \\ \text{is } L_\mu \text{ definable} \end{array} \right.$

Maximal permissiveness  $(T, \lambda) \models \exists p. \varphi \text{ IN } p \wedge \forall p'. [p \sqsubset p' \Rightarrow \neg \varphi \text{ IN } p']$

Minimal permissiveness

Uniqueness

## Expressing Constraints on the Decision

Taking into account e.g. inputs, environment's move, other players', decision of other agents... Given by some "states", some propositions or some events...

- Control of Closed systems  $\Sigma_u \subseteq \Sigma$  uncontrollable events

$$(T, \lambda) \models \exists p. AG(\bigwedge_{u \in \Sigma_u} \langle u \rangle p) \wedge \varphi \text{ IN } p$$

- Module Checking/MC Open Systems  $\{S, E\} \subseteq \Lambda$  exclusive

$$(T, \lambda) \models \forall e. AG(S \Rightarrow \bigwedge_{a \in \Sigma} [a]e) \wedge \varphi \text{ IN } e$$

- Control of Open Systems/Computing Strategies

$$(T, \lambda) \models \exists p. AG(E \Rightarrow \bigwedge_{a \in \Sigma} [a]p) \\ \wedge \forall e. AG(S \Rightarrow \bigwedge_{a \in \Sigma} [a]e) \wedge \varphi \text{ IN } (p \wedge e)$$

Complexity: 2EXPTIME in  $\varphi$  and PTIME in  $\lambda$

Maximal Permissiveness 3EXPTIME

## General Comments

- Model-Checking

$Q_1\Lambda_1.Q_2\Lambda_2.\dots.Q_n\Lambda_n.\varphi$  is  $|\lambda|^{(n-1)EXP(|\varphi|)} \times EXP(|\varphi|)$

- Satisfiability  $\begin{cases} \exists\Lambda_1.Q_2\Lambda_2.\dots.Q_n\Lambda_n.\varphi \text{ is } nEXP(|\varphi|) \\ \forall\Lambda_1.Q_2\Lambda_2.\dots.Q_n\Lambda_n.\varphi \text{ is } (n+1)EXP(|\varphi|) \end{cases}$

- Succinctness w.r.t. the mu-calculus

## Perspectives

- **Game Structures** with labelings. Take  $A = \{1, 2\} \subseteq \text{Agt} = \{1, 2, 3\}$   
 $\text{out}(F_A, q)$  is given by 3 propositions  $p_1$  (choice of gent 1),  $p_2$ , and  $p_3$   
 $\langle\langle A, a \rangle\rangle \bigcirc \varphi$  translates  $\exists p_1 \exists p_2 \forall p_3. (\langle a \rangle \varphi) \text{ IN } (p_1 \wedge p_2 \wedge p_3)$

**Complexity Issue (size a models)**

- **Components** Given classes  $\gamma_1, \gamma_2, \gamma_3$  for components

$$\exists p_1 \in \gamma_1. \forall p_2 \in \gamma_2. \forall p_3 \in \gamma_3. \varphi \text{ IN } (p_1 \wedge p_2 \wedge p_3)$$

- **Fusioning Decisions**

$$\exists p_1 \in \gamma_1. \exists p_2 \in \gamma_2. \forall p_3 \in \gamma_3. \varphi \text{ IN } \alpha(p_1, p_2, p_3).$$

e.g.  $\alpha(p_1, p_2, p_3) = p_1 \vee p_2 \vee p_3$

e.g.  $\alpha(p_1, p_2, p_3) = p_1 \wedge \langle a \rangle p_2 \wedge p_3$

**Expressiveness?**

## Partial Observation

- Given  $I \subseteq \Sigma$  of internal moves

Define  $\sim \subseteq \Sigma^*$  for the congruence generated by  $\epsilon \sim i$  for all  $i \in I$

- $(T, \lambda) \models \exists \sim p. \varphi$  means  $\left\{ \begin{array}{l} \text{there exists } p \text{ s.t. } (T, \lambda \cup p) \models \varphi, \text{ and} \\ \text{moreover } p(w) = p(w') \text{ whenever } w \sim w' \end{array} \right.$

$p$  is a uniform strategy

- The MC of  $Q_1^{\sim_1} p_1. Q_2^{\sim_2} p_2 \cdots Q_m^{\sim_m} p_m. \varphi$  is decidable whenever  $\sim_m \subseteq \cdots \subseteq \sim_2 \subseteq \sim_1$

[Pinchinat&Riedweg05]

- Corollary (with  $\sim_1 = \sim_2$ ) Maximal Permissiveness is decidable
- Corollary Open Systems under Partial Observation

## More Regarding Partial Observation

- Interplay between labelings over the Full Trees  $\Sigma' \cup \mathcal{I} \subseteq \Sigma$   
[Pinchinat&Riedweg05] Weak Synchronous Product  
[AVW03] [Briand06] Propositions  $\circlearrowleft^i$  and Loop Automata
- Observational equivalences between labelings ( $\tau$ -bisimulations)  
Non-deterministic Models with deterministic decisions [Pinchinat&Raclet05]  
Undistinguishable moves as in [Briand06]
- Games to decide Observability Properties
  - ★ Exhibit  $w \sim w'$  but  $w \in Acc$  and  $w' \notin Acc$   
( $w \sim_1 w_1$  and  $w \sim_2 w_2$  but  $w \in Acc$  and  $w_1, w_2 \notin Acc$ )
  - ★ Extend/Restrict a property to make it observable
  - ★ Extend the perception to observe certain properties