A Control-Theory Standpoint for the Non-emptiness Problem of Automata

Sophie Pinchinat

OMEGA 2007

SIXTH FRAMEWORK PROGRAMME THE AUSTRALIAN NATIONAL UNIVERSITY ANU UNIVERSITE DE RENNES I

Control Problems

Partial Observation

Application to Non-deterministic Word Automata

Application to Alternating Word Automata

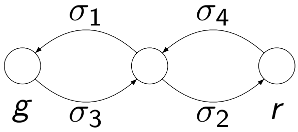
A word on Tree Automata

Possible Directions

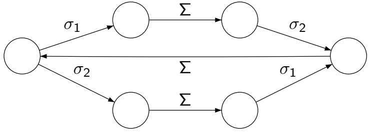
Control Problems

- $\mathcal{S}$ a Labeled Transition Systems
- The Control Objective is given, e.g. by a formula ϕ
- The problem to solve is:
  Find an LTS $\mathcal{C}$ such that the executons of $\mathcal{C} \times \mathcal{S}$ satisfy ϕ

Control Objective $= \forall(\overset{\infty}{\mathbf{F}}\, g \Rightarrow \overset{\infty}{\mathbf{F}}\, r)$

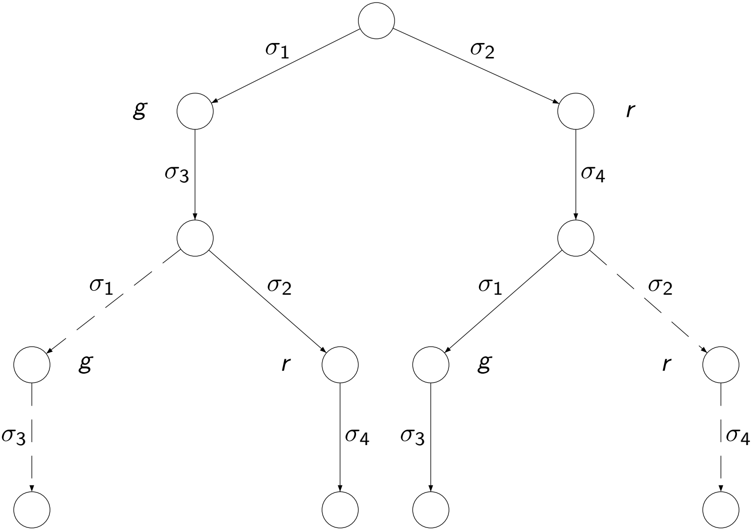


A possible controller $\mathcal{C}$



By the product with $\mathcal{C}$, the system behaviour is pruned.

The computation tree of $\mathcal{C} \times \mathcal{S}$ looks like (solid lines):



and indeed $\mathcal{C} \times \mathcal{S} \models \forall(\overset{\infty}{\mathbf{F}}\ g \Rightarrow \overset{\infty}{\mathbf{F}}\ r)$

- Additionally, we may require properties on $\mathcal{C}$:

Controllability: $\Sigma = \Sigma_c \uplus \Sigma_{uc}$

$$\mathbf{AG}\,(\bigwedge_{\sigma \in \Sigma_{uc}} \langle \sigma \rangle \text{true})$$

Non-blocking: $\mathbf{AG}\,(\bigvee_{\sigma \in \Sigma} \langle \sigma \rangle \text{true})$

Observability: $\Sigma_o \subseteq \Sigma$ (a projection)

Indistinguishability: $M : \Sigma \rightarrow \Delta \cup \{\tau\}$

Controllers as Strategies

- $\mathcal{G} = (V, E)$ an arena (turned-based two-player) $V = V_0 \uplus V_1$
- Alphabet of moves: $\Sigma_0 \uplus \Sigma_1$
- A winning condition (e.g. Buchi, a proposition *acc*)
- Find a controller $\mathcal{C}$ such that

$$\begin{cases} \mathcal{C} \times \mathcal{G} \models \forall \overset{\infty}{\mathbf{F}}\, acc \\ \mathcal{C} \models \mathbf{AG} \left(\bigwedge_{\sigma \in \Sigma_1} \langle \sigma \rangle \mathrm{true} \right) \end{cases}$$

Controller Synthesis for Deterministic Systems

[AVW03]

- A quotient construction

$$\mathcal{C} \times \mathcal{S} \models \Phi \text{ iff } \mathcal{C} \models \Phi/\mathcal{S}$$

  Based on automata: $\Phi \rightsquigarrow \mathcal{A}_\Phi$. Then $\mathcal{A}_\Phi/\mathcal{S}$ is a product between $\mathcal{A}_\Phi$ and $\mathcal{S}$ which updates the properties required in $\mathcal{A}_\Phi$ according to those of $\mathcal{S}$.

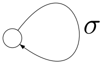
- Synthesis: Find a model of

$$\Phi/\mathcal{S} \wedge \Psi$$

Control Problems under Partial Observation

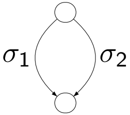
- [AVW03,Briand06] Impose the structure of the controller

(non-)Observability



if $\sigma \notin \Sigma_o$

Indistinguishability



if $M(\sigma_1) = M(\sigma_2)$

- Strict extension of tree automata,
  not bisimulation invariant, BUT still decidable!

Partial Observation Specifications

- Observability ($\Sigma_o \subseteq \Sigma$)
  - [AVW03]

$$\mathbf{AG}\,(\bigwedge_{\sigma \notin \Sigma_o} \circlearrowleft^{\sigma})$$

  - [PR05] use no loop but another product:

$$(\mathcal{C} \text{ is over } \Sigma_o \text{ and } \mathcal{C} \otimes \mathcal{S} \models \Phi) \text{ iff } \mathcal{C} \models \Phi /\!\!/ \mathcal{S}$$

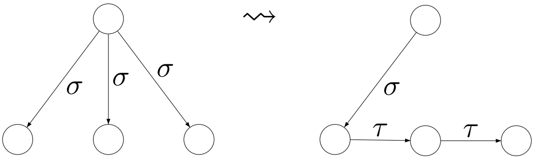
    Can express maximal permissiveness

- Indistinguishability ($M : \Sigma \rightarrow \Delta \cup \{\tau\}$)
  - [Briand06]

$$\mathbf{AG}\,(\bigwedge_{M(\sigma_1)=M(\sigma_2)} \sigma_1 \downdownarrows \sigma_2 \wedge \bigwedge_{M(\sigma)=\tau} \circlearrowleft^{\sigma})$$

Controller Synthesis for Non-deterministic Systems

$\mathcal{S}$ over $\Sigma \rightsquigarrow \quad \widetilde{\mathcal{S}}$ over $\Sigma \cup \{\tau\}$



Lemma[RP05] If $\mathcal{S}$ is finite, so is $\widetilde{\mathcal{S}}$ (polynomial).

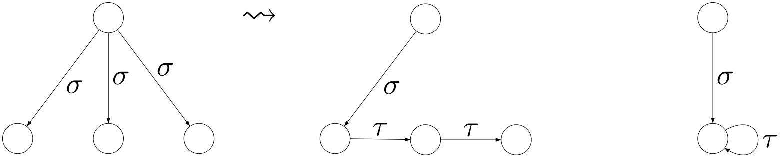
Transform Φ into $\widetilde{\Phi}$ according to $\widetilde{(\langle\sigma\rangle\Phi')} = \langle\sigma\tau^*\rangle\widetilde{(\Phi')}$

Theorem [RP05]

$$\mathcal{C} \times \mathcal{S} \models \Phi \quad \Leftrightarrow \quad \widetilde{\mathcal{C}} \times \widetilde{\mathcal{S}} \models \widetilde{\Phi}, \text{ and}$$

$$\widetilde{\mathcal{C}} \models \mathbf{AG}\,(\circlearrowleft^{\tau} \wedge \langle \tau \rangle \text{true})$$

($\mathcal{C}$ does not observe τ nor control it)



$\mathcal{S}$ $\qquad$ $\widetilde{\mathcal{S}}$ $\qquad$ $\widetilde{\mathcal{C}}$

$\mathcal{C}$ is obtained by removing the τ-loops in $\widetilde{\mathcal{C}}$

Non-deterministic Word Automata

- $\mathcal{B} = (A, Q, q_0, \delta, Acc)$ where $\delta \subseteq Q \times \Sigma \times Q$

$$\delta(q, a) = q_1 \vee \ldots \vee q_n$$

- w is accepted by $\mathcal{B}$ if there exists a run $q_0 q_1 q_2 \ldots \in Acc$ with $q_i \in \delta(q_{i-1}, w(i))$, $\forall i \leq 1$.
- Consider the following (decidable) problems:
  - Emptiness: "is there a word accepted by $\mathcal{B}$?"
  - Universality: "is there a word rejected by $\mathcal{B}$?"

Emptiness and Universality

- $\mathcal{B} = (A, Q, q_0, \delta, Acc)$ becomes a non-deterministic system
- $\mathcal{S}_\mathcal{B} = (Q, \delta, q_0, \rightarrow)$ over $\Sigma = A$, and define ϕ_{Acc}
- The controller chooses the letter(s) a, hence $\Sigma_c = A$
- The system chooses the run (a path)

The product $\mathcal{C} \times \mathcal{S}_\mathcal{B}$ is the tree of all runs on the word chosen by $\mathcal{C}$.

Emptiness and Universality

Theorem

(Emptiness) $L(\mathcal{B}) \neq \emptyset$ *iff there exists* $\widetilde{\mathcal{C}}$ *such that*

$$\begin{cases} \widetilde{\mathcal{C}} \times \widetilde{\mathcal{S}_{\mathcal{B}}} \models \exists \phi_{Acc} \\ \widetilde{\mathcal{C}} \models \mathbf{AG}\,(\circlearrowleft^{\tau} \wedge \langle \tau \rangle \mathit{true}) \\ (\widetilde{\mathcal{C}} \textit{ is non-blocking} \ldots) \end{cases}$$

Theorem

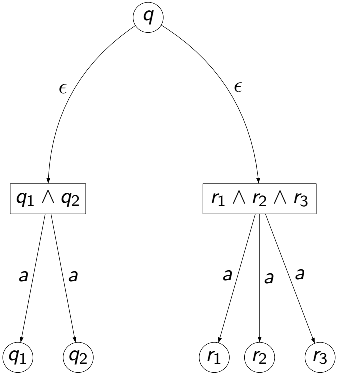
(Universality) $L(\mathcal{B}) \neq A^{\omega}$ *iff there exists* $\widetilde{\mathcal{C}}$ *such that*

$$\begin{cases} \widetilde{\mathcal{C}} \times \widetilde{\mathcal{S}_{\mathcal{B}}} \models \forall \neg \phi_{Acc} \\ \widetilde{\mathcal{C}} \models \mathbf{AG}\,(\circlearrowleft^{\tau} \wedge \langle \tau \rangle \mathit{true}) \\ (\widetilde{\mathcal{C}} \textit{ is non-blocking} \ldots) \end{cases}$$

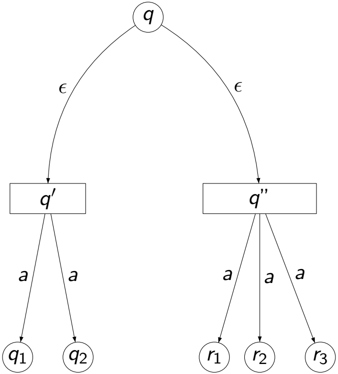
Alternating Word Automata

$$\mathcal{A} = (A, Q, q_0, \delta, Acc) \text{ where } \delta : Q \times \Sigma \rightarrow B^+(Q)$$

$$\delta(q, a) = (q_1 \wedge q_2) \vee (r_1 \wedge r_2 \wedge r_3)$$



Write T for the set of transitions of the automaton $(q, \epsilon, q'), (q', a, q_1), \ldots$



Alternation $\rightsquigarrow$ Partial Observation

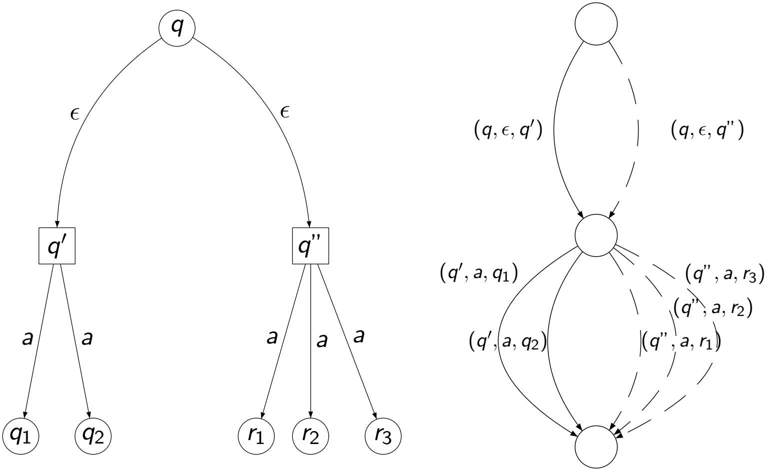
- $\mathcal{A}$ becomes a mere system $\mathcal{S}_{\mathcal{A}}$ over T (the transitions of $\mathcal{A}$)
- Alternation is captured by a game between the controller and the system; the controller only knows the word built so far.
  - The controller chooses the letter(s) a, and the disjunct in $\delta(q, a)$:

    $$\Sigma_c = \{(q, \epsilon, q') \mid (q, \epsilon, q') \in T\}$$

    and

    $M((q_1, \epsilon, q')) = M((q_2, \epsilon, q"))$, as no additional information is given
  - The system chooses the a-successor state:

    $$M((q', a, \ldots)) = M((q", a, \ldots))$$



The product $\mathcal{C} \times \mathcal{S}_{\mathcal{A}}$ is a run (a tree) of $\mathcal{A}$ on the word chosen by $\mathcal{C}$.

Emptiness and Universality

Let $\mathcal{A} = (A, Q, q_0, \delta, Acc)$, and define ϕ_{Acc}

Theorem

(Emptiness) *$L(\mathcal{A}) \neq \emptyset$ iff there exists $\mathcal{C}$ such that*

$$\begin{cases} \mathcal{C} \times \mathcal{S}_{\mathcal{A}} \models \forall \phi_{Acc} \\ \mathcal{C} \models \mathbf{AG}\,(\bigwedge_{a \in A} \langle (q, a, q') \rangle true \wedge \bigwedge (q_1, \epsilon, q') \downarrow\downarrow (q_2, \epsilon, q'') \wedge \bigwedge (q', a, q) \downarrow\downarrow (q'', a, r)) \\ \ldots \end{cases}$$

Theorem

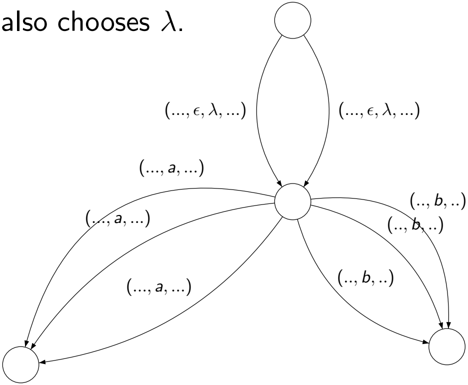
(Universality) *$L(\mathcal{A}) \neq A^\omega$ iff there exists $\mathcal{C}'$ such that*

$$\begin{cases} \mathcal{C}' \times \mathcal{S}_{\mathcal{A}} \models \forall \neg \phi_{Acc} \\ \mathcal{C}' \models \mathbf{AG}\,(\bigwedge \langle (q, \epsilon, q') \rangle true \wedge \bigwedge (q_1, \epsilon, q') \downarrow\downarrow (q_2, \epsilon, q'') \wedge \bigwedge (q', a, q) \downarrow\downarrow (q'', a, r)) \\ \mathcal{C}' \models \mathbf{AG}\,(\bigwedge_{q', a} \bigvee \langle (q', a, q) \rangle true) \ldots \end{cases}$$

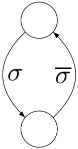
The product $\mathcal{C}' \times \mathcal{S}_{\mathcal{A}}$ is a tree which contains a branch of every possible run of $\mathcal{A}$ on the word chosen by $\mathcal{C}'$.

Tree Automata

- Have to take propositions into account:
  $\delta(q,\lambda) = [(a,q_1) \wedge (a,q_2) \wedge (b,q_3)] \vee [(a,q'_1) \wedge (b,q'_2) \wedge (b,q'_3)]$
- Alphabet $(q,\epsilon,\lambda,q'), (q',a,\lambda,q_1), \ldots$
- The controller also chooses λ.



Possible Directions



- Two-way Automata
  - [Briand06] with $\circlearrowleft^{\eta}$, where $\eta \in \Sigma^*$.
  - Undecidable for $|\eta| \geq 3$ (grids: $\circlearrowleft^{abc}, \circlearrowleft^{bac}, \circlearrowleft^{cd}$)
  - BUT open problem for $|\eta| \leq 2$.
- Automata with Constraints
  - Automata with Equality and Disequality Constraints [TATA book] Undecidable
  - Decidable subclasses, e.g. Automata With Constraints Between Brothers

Conclusion

- A novel standpoint to improve our insights
  In particular, regarding Alternation
- Extend this standpoint to other classes of automata.
- Exploit controllers properties in this setting (maximal permissiveness, fairness, etc.).