On Properties and State Complexity of Deterministic State-Partition Automata

Galina Jirásková [1] Tomáš Masopust [2]

[1] Mathematical Institute, Slovak Academy of Sciences

[2] Institute of Mathematics, Academy of Sciences of the Czech Republic

TCS 2012

Motivation – hierarchical supervisory control

- Supervisory control
  – given a plant G and a specification K
  – find supervisor S such that $L(S/G) = K$
- ...with partial observation (modeled by projections)
- ...computed on abstractions (modeled by projections)

Motivation – hierarchical supervisory control

- Supervisory control
  – given a plant G and a specification K
  – find supervisor S such that $L(S/G) = K$
- ... with partial observation (modeled by projections)
- ... computed on abstractions (modeled by projections)

Motivation – hierarchical supervisory control

- Supervisory control
  – given a plant G and a specification K
  – find supervisor S such that $L(S/G) = K$
- ...with partial observation (modeled by projections)
- ...computed on abstractions (modeled by projections)

Motivation – hierarchical supervisory control

Definition (Observation consistency)

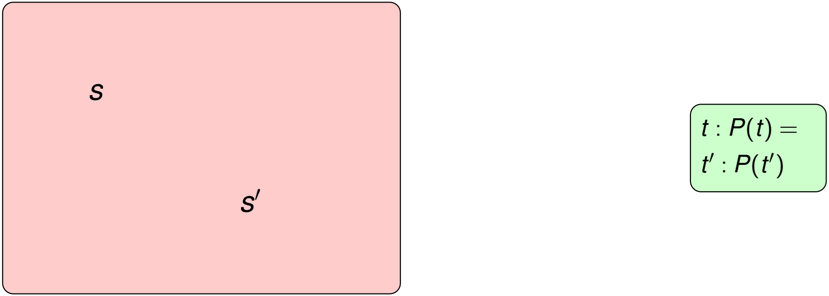
A language L is observation consistent wrt projections Q and P if
$\forall t, t' \in Q(L)$ such that $P(t) = P(t')$,
$\exists s, s' \in L$ such that $Q(s) = t$, $Q(s') = t'$, and $P(s) = P(s')$.

$t : P(t) =$
$t' : P(t')$

Motivation – hierarchical supervisory control

Definition (Observation consistency)

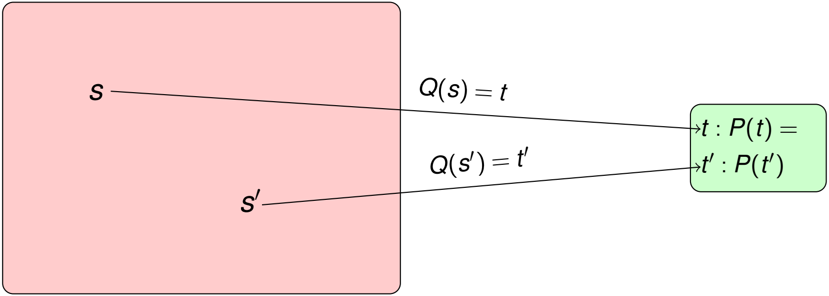
A language L is observation consistent wrt projections Q and P if
$\forall t, t' \in Q(L)$ such that $P(t) = P(t')$,
$\exists s, s' \in L$ such that $Q(s) = t$, $Q(s') = t'$, and $P(s) = P(s')$.



Motivation – hierarchical supervisory control

Definition (Observation consistency)

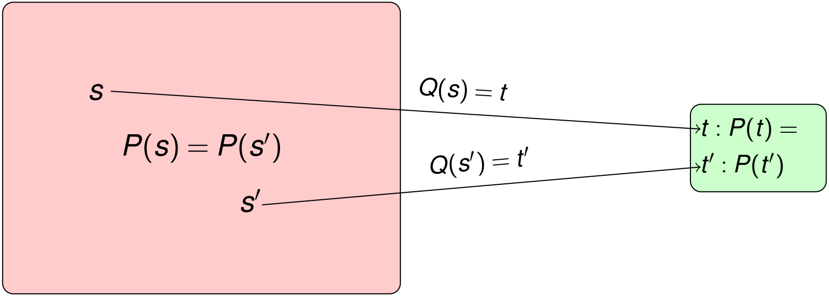
A language L is observation consistent wrt projections Q and P if
$\forall t, t' \in Q(L)$ such that $P(t) = P(t')$,
$\exists s, s' \in L$ such that $Q(s) = t$, $Q(s') = t'$, and $P(s) = P(s')$.



Motivation – hierarchical supervisory control

Definition (Observation consistency)

A language L is observation consistent wrt projections Q and P if
$\forall t, t' \in Q(L)$ such that $P(t) = P(t')$,
$\exists s, s' \in L$ such that $Q(s) = t$, $Q(s') = t'$, and $P(s) = P(s')$.



Projection

Definition (Projection)

A homomorphism

$$P : \Sigma^* \to \Sigma_o^*$$

defined by

$$P(a) = \begin{cases} a & \text{if } a \in \Sigma_o \\ \varepsilon & \text{if } a \in \Sigma \setminus \Sigma_o \end{cases}$$

Subset Automaton

Definition (Subset automaton)

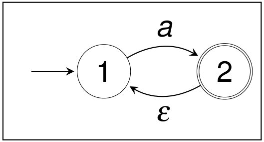
For an NFA $N = (Q, \Sigma, \delta, S, F)$, the DFA

$$\det(N) = (2^Q, \Sigma, \delta_d, s_d, F_d)$$

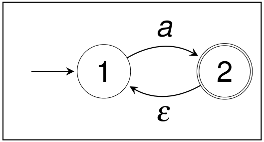
constructed by *"subset construction"* is the subset automaton corresponding to N.

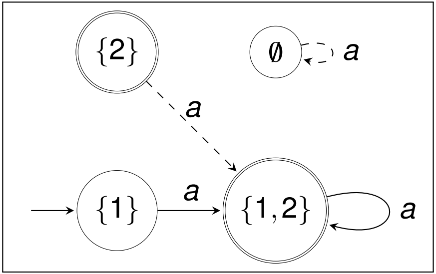
The state set of the subset automaton is the set of all subsets of Q, even though some of them may be unreachable.

Subset automaton – example



Subset automaton – example



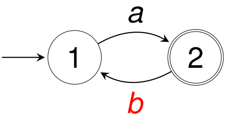


Projected automaton

Definition (Projected automaton)

Let $G = (Q, \Sigma, \delta, s, F)$ be a DFA.

1. Construct an NFA N_G by replacing all transitions labeled by $\Sigma \setminus \Sigma_o$ with ε-transitions
2. Construct $\det(N_G)$
3. $P(G)$ is the reachable part of $\det(N_G)$

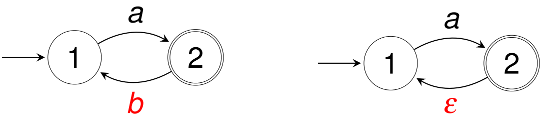


Projected automaton

Definition (Projected automaton)

Let $G = (Q, \Sigma, \delta, s, F)$ be a DFA.

1. Construct an NFA N_G by replacing all transitions labeled by $\Sigma \setminus \Sigma_o$ with ε-transitions
2. Construct $\det(N_G)$
3. $P(G)$ is the reachable part of $\det(N_G)$

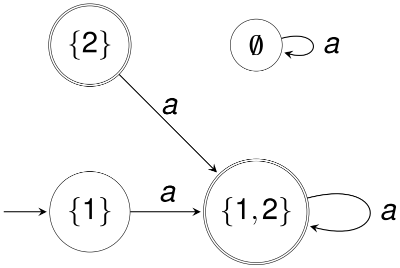


Projected automaton

Definition (Projected automaton)

Let $G = (Q, \Sigma, \delta, s, F)$ be a DFA.

1. Construct an NFA N_G by replacing all transitions labeled by $\Sigma \setminus \Sigma_o$ with ε-transitions
2. Construct $\det(N_G)$
3. $P(G)$ is the reachable part of $\det(N_G)$

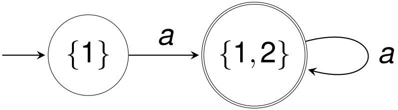


Projected automaton

Definition (Projected automaton)

Let $G = (Q, \Sigma, \delta, s, F)$ be a DFA.

1. Construct an NFA N_G by replacing all transitions labeled by $\Sigma \setminus \Sigma_o$ with ε-transitions
2. Construct $\det(N_G)$
3. $P(G)$ is the reachable part of $\det(N_G)$



State-partition automaton

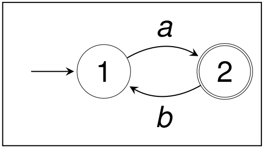
Definition (SPA)

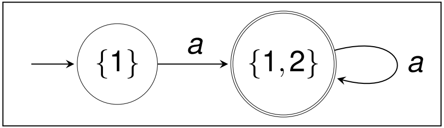
DFA G is a state-partition automaton with respect to projection P if the states of $P(G)$ are pairwise disjoint:

$$X \cap Y \neq \emptyset \implies X = Y$$

If G reachable, the state set of $P(G)$ defines a partition of its state set.

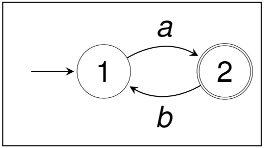
SPA – example; $P(b) = \varepsilon$





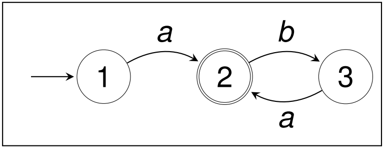
$$\{1\} \cap \{1,2\} \neq \emptyset$$

SPA – example II; $P(b) = \varepsilon$



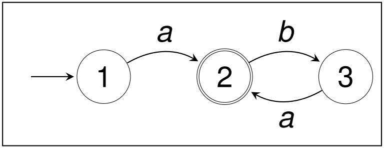
$$\{1\} \cap \{2,3\} = \emptyset$$

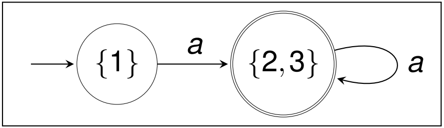
SPA – example II; $P(b) = \varepsilon$



$$\{1\} \cap \{2,3\} = \emptyset$$

SPA – example II; $P(b) = \varepsilon$





$$\{1\} \cap \{2,3\} = \emptyset$$

Why SPAs?

1. When abstraction modeled by projection. . .
2. Simplification of constructions. . .
3. . . . and proofs. . .
4. Not yet investigated in detail. . .
5. . . .

Why SPAs?

1. When abstraction modeled by projection. . .
2. Simplification of constructions. . .
3. . . . and proofs. . .
4. Not yet investigated in detail. . .
5. . . .

Why SPAs?

1. When abstraction modeled by projection…
2. Simplification of constructions…
3. …and proofs…
4. Not yet investigated in detail…
5. …

Why SPAs?

1. When abstraction modeled by projection. . .
2. Simplification of constructions. . .
3. . . . and proofs. . .
4. Not yet investigated in detail. . .
5. . . .

Why SPAs?

1. When abstraction modeled by projection...
2. Simplification of constructions...
3. ...and proofs...
4. Not yet investigated in detail...
5. ...

Simple observation

...the projected automaton is not bigger

Lemma

If G is an SPA with respect to P, then

$$\mathrm{sc}(P(G)) \leq \mathrm{sc}(G)\,.$$

$\mathrm{sc}(G) =$ number of states of G.

Checking the SPA property is easy. . .

Theorem

There is a linear algorithm deciding whether G is an SPA with respect to P.

Proof idea.

1. If G is an SPA, then each state of G belongs to exactly one state of $P(G)$
2. Construct $P(G)$ and take care of the set representation of states of $P(G)$

□

Checking the SPA property is easy. . .

Theorem

There is a linear algorithm deciding whether G is an SPA with respect to P.

Proof idea.

1. If G is an SPA, then each state of G belongs to exactly one state of $P(G)$
2. Construct $P(G)$ and take care of the set representation of states of $P(G)$

□

Checking the SPA property is easy. . .

Theorem

There is a linear algorithm deciding whether G is an SPA with respect to P.

Proof idea.

1. If G is an SPA, then each state of G belongs to exactly one state of $P(G)$
2. Construct $P(G)$ and take care of the set representation of states of $P(G)$

□

SPA for languages

...exists for all regular languages and projections

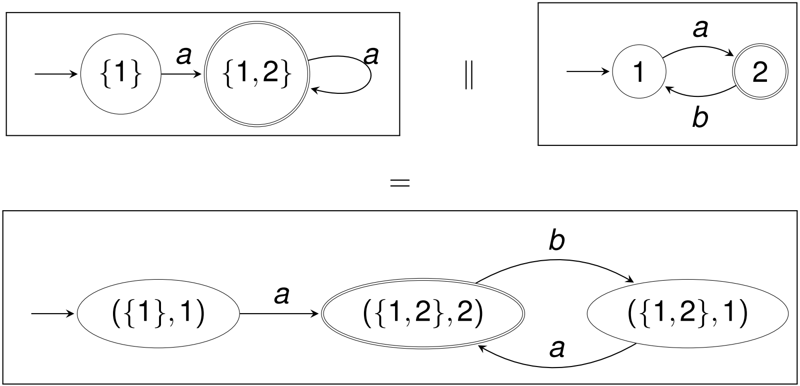
Theorem (Cho & Marcus 1989)

Let G be a DFA. The automaton

$$P(G) \parallel G$$

is an SPA with respect to projection P that accepts $L(G)$.

$P(G) \| G$ – example



Minimal-state SPA

Theorem

Let G be a minimal DFA. The DFA

$$P(G) \parallel G$$

is the unique (up to isomorphism) minimal SPA with respect to projection P that accepts $L(G)$.

No SPA for two projections

Lemma (Komenda, M., Schmidt, van Schuppen – personal com.)

There exist a regular language L and projections P and P' such that no DFA accepting L is an SPA with respect to both projections.

Proof.

Let P and P' be projections from $\{a, b\}^*$ to $\{a\}^*$ and $\{b\}^*$, respectively. Consider the language

$$L = (ab)^* .$$

□

Closure properties

Every regular language has an SPA representation

Problem

Given an SPA, is the DFA resulting from the standard construction for a regular operation an SPA with respect to the same projection?

Not closed

Theorem

SPAs are not closed under the operations of

- *complement,*
- *intersection,*
- *union,*
- *concatenation,*
- *star,*
- *reversal,*
- *cyclic shift,*
- *left quotient.*

Complement:

Proof.

To get a DFA for complement of a DFA,

1. add the dead state, if necessary,
2. interchange final and non-final states.

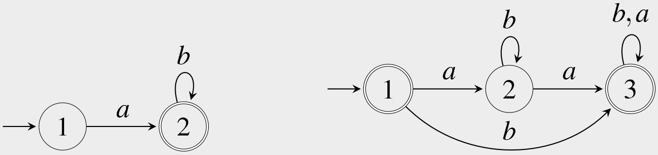


Figure: SPA G and DFA G^c for complement of $L(G)$; $P : \{a, b\}^* \to \{a\}^*$.

$\{1, 3\}$ reached by ε and $\{2, 3\}$ reached by a. □

not closed because. . .

The structure of the automaton is changed

Are closed. . .

Theorem

SPAs are closed under the operations of

- *right quotient and*
- *complement of complete state-partition automata.*

. . . because the structure of the automaton is NOT changed

State-partition complexity

Definition (State-partition complexity)

State-partition complexity of a language L with respect to a projection P,

$$\text{denoted by } \mathrm{sc}(L)\,,$$

is the smallest number of states in any SPA accepting L.

State-partition complexity of language L is the number of states of DFA

$$P(G) \parallel G$$

where G is the minimal incomplete DFA accepting language L.

Tight bounds

Theorem

Let L be a regular language accepted by the minimal incomplete DFA G with n states. Then

$$\mathrm{sc}(L) \leq 3n \cdot 2^{n-3}\,.$$

Tight bounds

Theorem

Let L be a regular language accepted by the minimal incomplete DFA G with n states. Then

$$\mathrm{sc}(L) \leq 3n \cdot 2^{n-3}.$$

Proof.

(X, q) a state of $P(G) \parallel G$ iff $q \in X$ and X is a state of $P(G)$. Then the sum of cardinalities of states of $P(G)$ gives the upper bound

$$\mathrm{sc}(L) \leq \sum_{X \in Q'} |X| = \sum_{i=0}^{n} \binom{n}{i} i - \sum_{i=0}^{n-2} \binom{n-2}{i} (i+1) = 3n \cdot 2^{n-3}$$

because $|Q'| \leq 3 \cdot 2^{n-2} - 1$ if the projection is not identity. □

Tight bounds

Theorem

Let L be a regular language accepted by the minimal incomplete DFA G with n states. Then

$$\mathrm{sc}(L) \leq 3n \cdot 2^{n-3}.$$

Proof.

(X, q) a state of $P(G) \parallel G$ iff $q \in X$ and X is a state of $P(G)$. Then the sum of cardinalities of states of $P(G)$ gives the upper bound

$$\mathrm{sc}(L) \leq \sum_{X \in Q'} |X| = \sum_{i=0}^{n} \binom{n}{i} i - \sum_{i=0}^{n-2} \binom{n-2}{i} (i+1) = 3n \cdot 2^{n-3}$$

because $|Q'| \leq 3 \cdot 2^{n-2} - 1$ if the projection is not identity. □

Tight bounds

Theorem

Let L be a regular language accepted by the minimal incomplete DFA G with n states. Then

$$\mathrm{sc}(L) \leq 3n \cdot 2^{n-3}.$$

Proof.

(X,q) a state of $P(G) \parallel G$ iff $q \in X$ and X is a state of $P(G)$. Then the sum of cardinalities of states of $P(G)$ gives the upper bound

$$\mathrm{sc}(L) \leq \sum_{X \in Q'} |X| = \sum_{i=0}^{n} \binom{n}{i} i - \sum_{i=0}^{n-2} \binom{n-2}{i} (i+1) = 3n \cdot 2^{n-3}$$

because $|Q'| \leq 3 \cdot 2^{n-2} - 1$ if the projection is not identity. □

Witness language

Theorem

For every $n \geq 3$, there is a regular language L accepted by the minimal incomplete DFA G with n states such that $\mathrm{sc}(L) = 3n \cdot 2^{n-3}$.

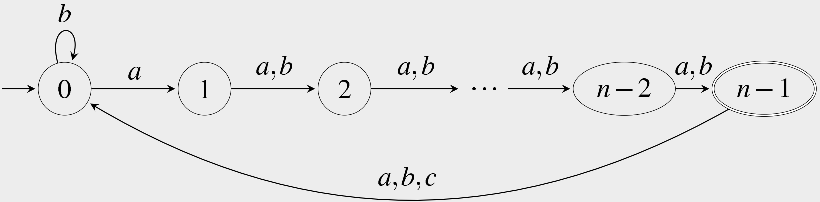
This bound cannot be reached using a smaller alphabet or a projection to a singleton.

Witness language

Theorem

For every $n \geq 3$, there is a regular language L accepted by the minimal incomplete DFA G with n states such that $\mathrm{sc}(L) = 3n \cdot 2^{n-3}$.

Proof.



Minimal DFA meeting the bound $3n \cdot 2^{n-3}$; $P : \{a,b,c\}^* \to \{a,b\}^*$. □

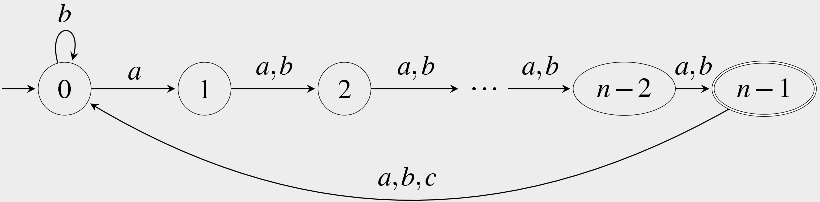
This bound cannot be reached using a smaller alphabet or a projection to a singleton.

Witness language

Theorem

For every $n \geq 3$, there is a regular language L accepted by the minimal incomplete DFA G with n states such that $\mathrm{sc}(L) = 3n \cdot 2^{n-3}$.

Proof.



Minimal DFA meeting the bound $3n \cdot 2^{n-3}$; $P : \{a,b,c\}^* \to \{a,b\}^*$. □

This bound cannot be reached using a smaller alphabet or a projection to a singleton.

Future work

- State-partition complexity of a language and its complement differs by one in case of complete DFAs, and
- by $3n$ if the DFAs are incomplete.

- Defining nondeterministic state-partition automata and investigating their properties may be of interest.

Future work

- State-partition complexity of a language and its complement differs by one in case of complete DFAs, and
- by $3n$ if the DFAs are incomplete.
- Defining nondeterministic state-partition automata and investigating their properties may be of interest.

Future work

- State-partition complexity of a language and its complement differs by one in case of complete DFAs, and
- by $3n$ if the DFAs are incomplete.

- Defining nondeterministic state-partition automata and investigating their properties may be of interest.

Thank You