Hyper-Minimization

Lossy compression of deterministic automata

Andreas Maletti

Institute for Natural Language Processing
University of Stuttgart

andreas.maletti@ims.uni-stuttgart.de

Debrecen — August 19, 2011

Collaborators

Reporting joint work with

- PAWEŁ GAWRYCHOWSKI — University of Wrocław
- MARKUS HOLZER — University of Giessen
- ARTUR JEŻ — University of Wrocław
- DANIEL QUERNHEIM — University of Stuttgart

Contents

Problem definition

Minimization

- given: DFA A
- return: minimal DFA B such that $L(B) = L(A)$

Hyper-minimization

- given: DFA A
- return: minimal DFA B such that $L(B)$ and $L(A)$ differ finitely

AFL 2008 (Balatonfüred, Hungary)

VILIAM GEFFERT spoke about:

- general problem and its structural characterization
- (inefficient) hyper-minimization

[BADR, GEFFERT, SHIPMAN: *Hyper-minimizing minimized deterministic finite state automata*. ITA 2009]

AFL 2008 (Balatonfüred, Hungary)

VILIAM GEFFERT spoke about:

- general problem and its structural characterization
- (inefficient) hyper-minimization

Unfortunately, I was not there

[BADR, GEFFERT, SHIPMAN: *Hyper-minimizing minimized deterministic finite state automata*. ITA 2009]

CIAA 2008 (San Francisco, CA, USA)

ANDREW BADR spoke about:

- faster (yet still inefficient) hyper-minimization
- combination with cover automata minimization

[BADR: *Hyper-minimization in* $O(n^2)$. CIAA 2008]

CIAA 2008 (San Francisco, CA, USA)

ANDREW BADR spoke about:

- faster (yet still inefficient) hyper-minimization
- combination with cover automata minimization

MARKUS HOLZER and I were there

[BADR: *Hyper-minimization in* $O(n^2)$. CIAA 2008]

© David Iliff

CIAA 2009 (Sydney, Australia)

MARKUS HOLZER spoke about:

- efficient hyper-minimization

[HOLZER, ∼: *An* $n \log n$ *algorithm for hyper-minimizing states in a (minimized) deterministic automaton*. CIAA 2009]

© David Iliff

CIAA 2009 (Sydney, Australia)

MARKUS HOLZER spoke about:

- efficient hyper-minimization

But at the same time ...

[HOLZER, ∼: *An* $n \log n$ *algorithm for hyper-minimizing states in a (minimized) deterministic automaton*. CIAA 2009]

MFCS 2009 (Novy Smokovec, Slovakia)

ARTUR JEŻ spoke about:

- efficient hyper-minimization (essentially the same algorithm)
- *k*-minimization

[GAWRYCHOWSKI, JEŻ: *Hyper-minimisation made efficient*. MFCS 2009]

— Best student paper

CIAA 2010 (Winnipeg, Canada)

I spoke about:

- error-optimal hyper-minimization

[∼: *Better hyper-minimization — not as fast, but fewer errors*. CIAA 2010]

FSTTCS 2010 (Chennai, India)

SVEN SCHEWE spoke about:

- hyper-minimization for DBA (deterministic Büchi automata)

[SCHEWE: *Beyond Hyper-Minimisation—Minimising DBAs and DPAs is NP-Complete*. FSTTCS 2010]

CIAA 2011 (Blois, France)

ARTUR JEŻ spoke about:

- efficient combination with cover automata minimization

[JEŻ, ∼: *Computing all l-cover automata fast*. CIAA 2011]

AFL 2011 (Debrecen, Hungary)

~~DANIEL QUERNHEIM~~ I spoke about:

- hyper-minimization for weighted DFA

[∼, QUERNHEIM: *Hyper-minimisation of deterministic weighted finite automata over semifields*. AFL 2011]

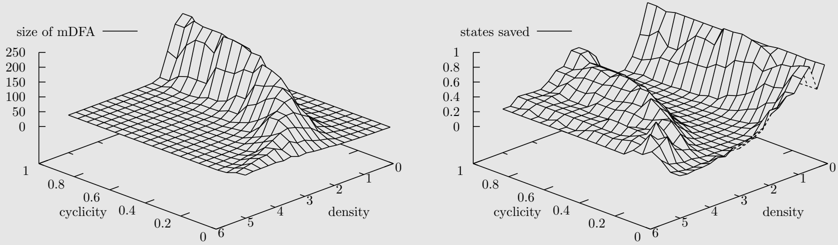
MFCS 2011 (Warsaw, Poland)

PAWEŁ GAWRYCHOWSKI will speak about:

- efficient hyper-minimization for partial DFA
- limits of hyper-minimization

[GAWRYCHOWSKI, JEŻ, ∼: *On minimising automata with errors*. MFCS 2011]

Experiments on random NFA

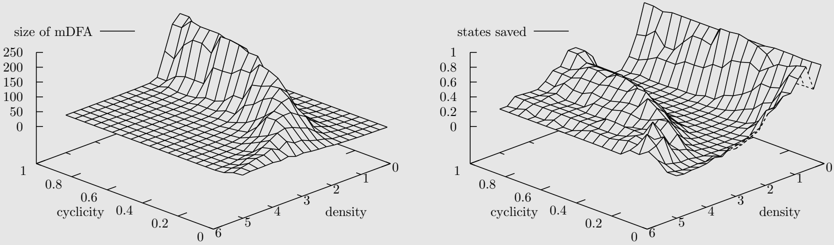


- Left: Size of minimal DFA
- Right: Ratio of saved states in hyper-minimization

Conclusion

- significant savings

Experiments on random NFA

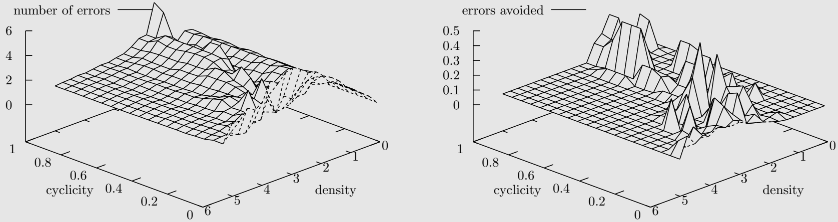


- Left: Size of minimal DFA
- Right: Ratio of saved states in hyper-minimization

Conclusion

- significant savings
- but only outside the difficult area for minimization

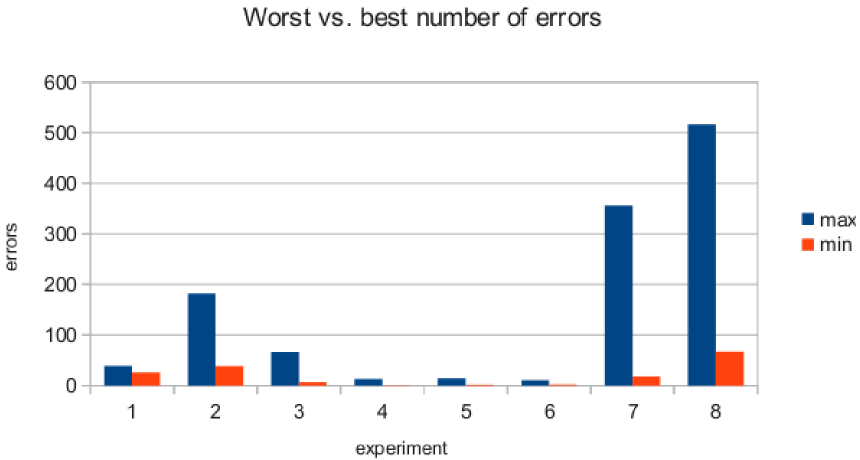
Error analysis



- Left: Average number of errors (100 NFA per data point)
- Right: Ratio of avoided errors in optimal hyper-minimization

[∼, QUERNHEIM: *Optimal hyper-minimization*. IJFCS 2011]

[TABAKOV, VARDI: *Experimental evaluation of classical automata constructions*. LPAR 2005]


Worst vs. best number of errors
600
500
400
300
200
100
0
errors
1
2
3
4
5
6
7
8
experiment
max
min

Contents

Basic definitions

Definition

- Two languages L_1, L_2 are **almost equal** if $L_1 \triangle L_2$ is finite

$$L_1 \triangle L_2 = (L_1 \setminus L_2) \cup (L_2 \setminus L_1)$$

- Two DFA A_1, A_2 are **almost equivalent** if $L(A_1)$ and $L(A_2)$ are almost equal

Example

- all finite languages are almost equal
- a^* and aaa^* are almost equal
- a^* and $(aa)^*$ are **not** almost equal

Overview

Common approach

1. Identify kernel states
2. Identify almost equivalent states
3. Merge states

Preamble and kernel states

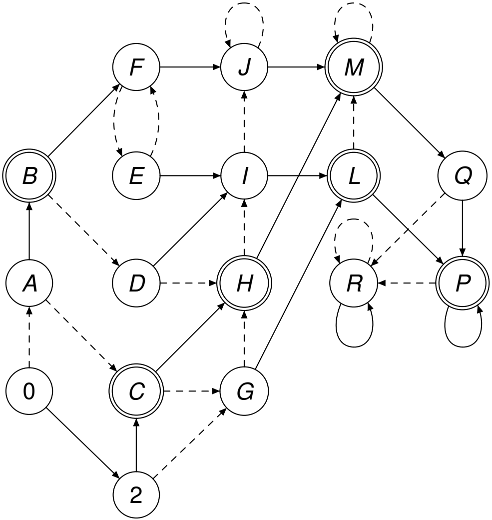
Definition

- preamble state: finitely many words lead to it
- kernel state: infinitely many words lead to it

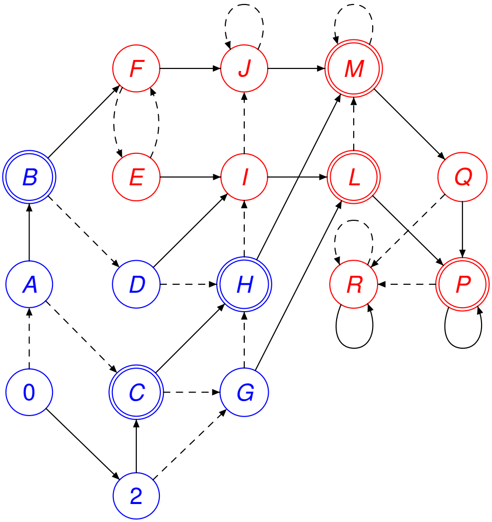
Words leading to state q:

$$\{w \in \Sigma^* \mid \delta(q_0, w) = q\}$$

Preamble and kernel states



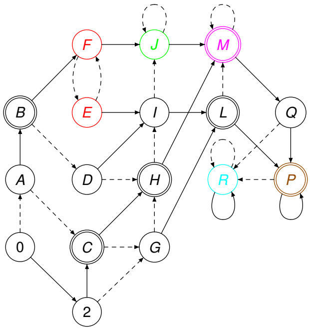
Preamble and kernel states



Computing kernel states

Step 1

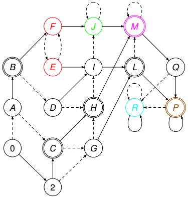
- Compute strongly connected components (using TARJAN'S algorithm)



Computing kernel states

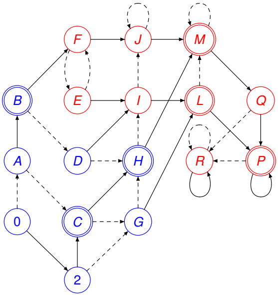
Step 1

- Compute strongly connected components (using TARJAN'S algorithm)



Step 2

- Mark all successors of nontrivial components



[TARJAN: *Depth-first search and linear graph algorithms*. SIAM J. Comput. 1972]

Computing kernel states

Theorem

We can compute the set of kernel states in linear time.

Almost equivalent states

Definition

Two states are **almost equivalent** if their right languages are almost equal

Right language of state q:

$$\{w \in \Sigma^* \mid \delta(q, w) \in F\}$$

Almost equivalent states

Definition

Two states are **almost equivalent** if their right languages are almost equal

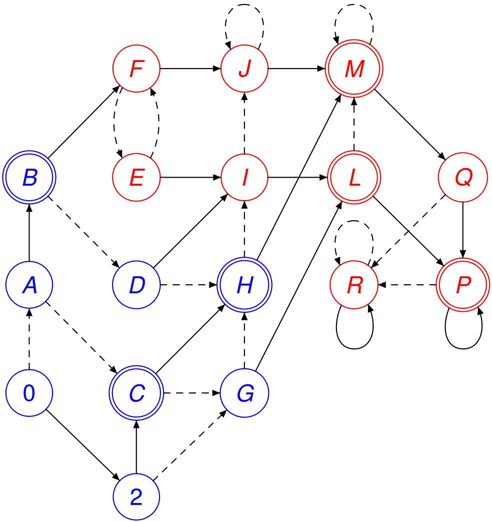
Right language of state q:

$$\{w \in \Sigma^* \mid \delta(q, w) \in F\}$$

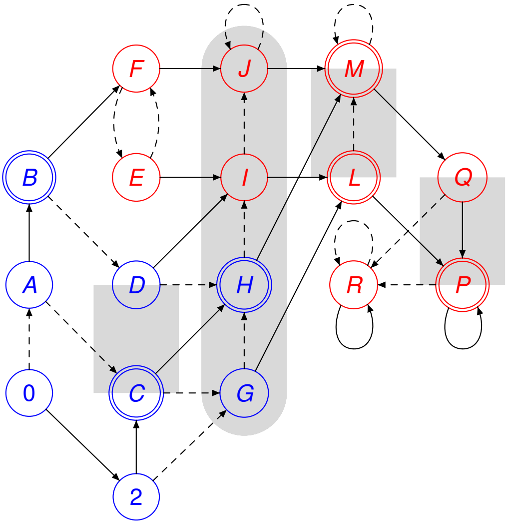
Consequence

For almost equivalent states p, q there is $k \in \mathbb{N}$ such that $\delta(p, w) = \delta(q, w)$ for all $|w| > k$

Almost equivalent states



Almost equivalent states



Computing almost equivalent states

Theorem

Almost equivalence is a congruence.

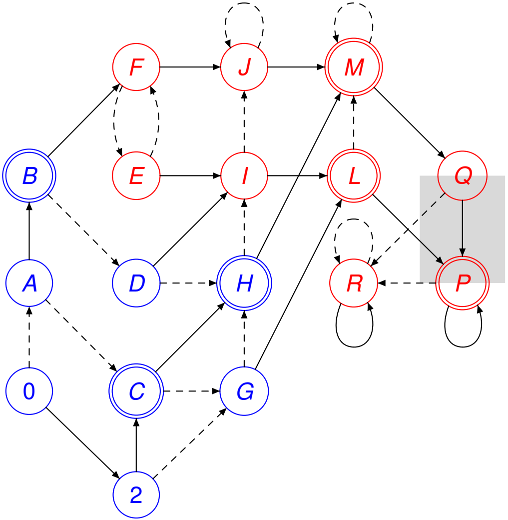
Theorem

If p, q are different, but almost equivalent, then there are p', q' such that

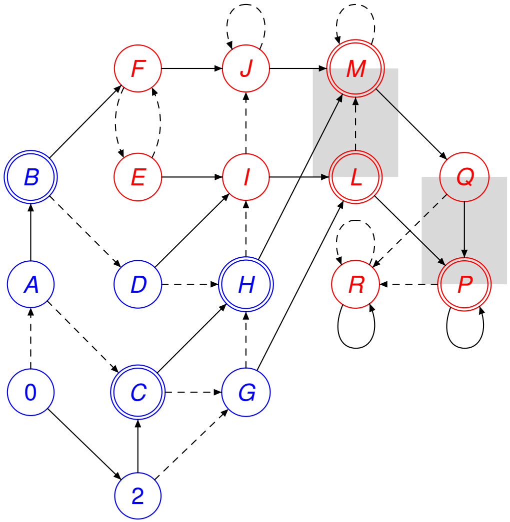
$$\delta(p', \sigma) = \delta(q', \sigma)$$

for all $\sigma \in \Sigma$.

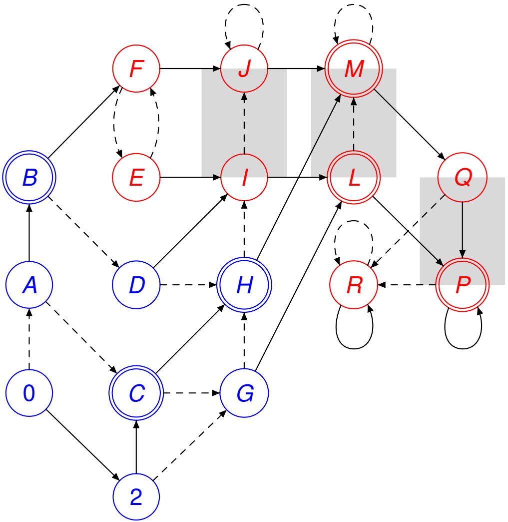
Computing almost equivalent states



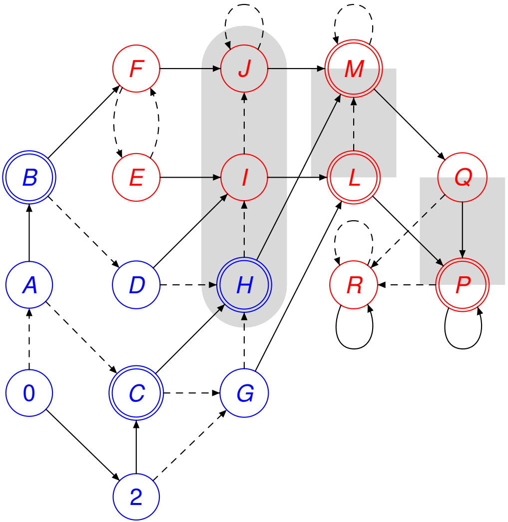
Computing almost equivalent states



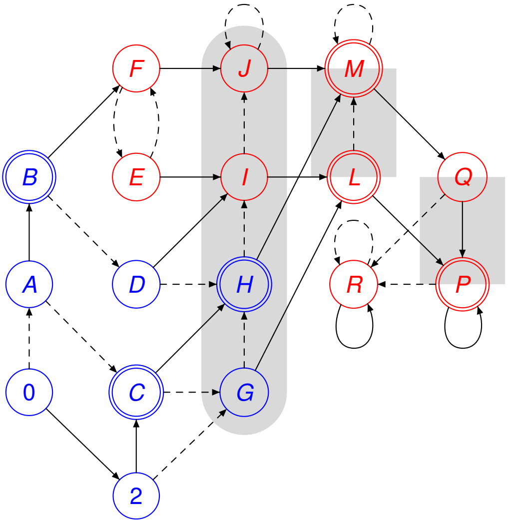
Computing almost equivalent states



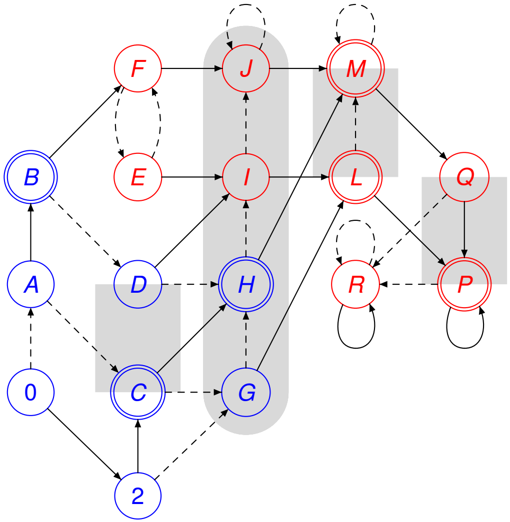
Computing almost equivalent states



Computing almost equivalent states



Computing almost equivalent states



Computing almost equivalent states

Theorem

The partition representing almost equivalent states can be computed in

- $O(n \log n)$ *using* $O(n^2)$ *space*
- $O(n \log^2 n)$ *using* $O(n)$ *space*

[HOLZER, $\sim$: *An* $n \log n$ *algorithm for hyper-minimizing states in a (minimized) deterministic automaton*. CIAA 2009]

[GAWRYCHOWSKI, JEŻ: *Hyper-minimisation made efficient*. MFCS 2009]

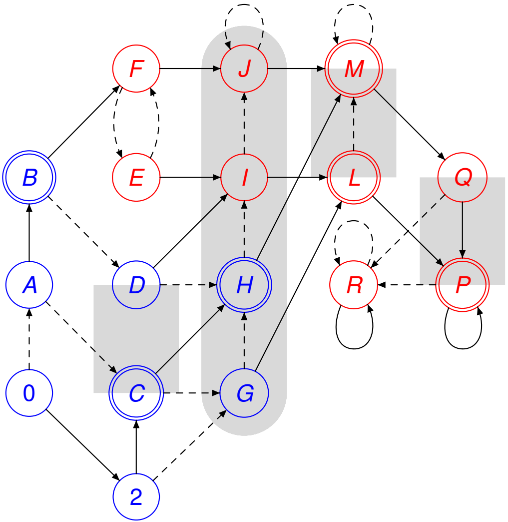
Merging states

Algorithm

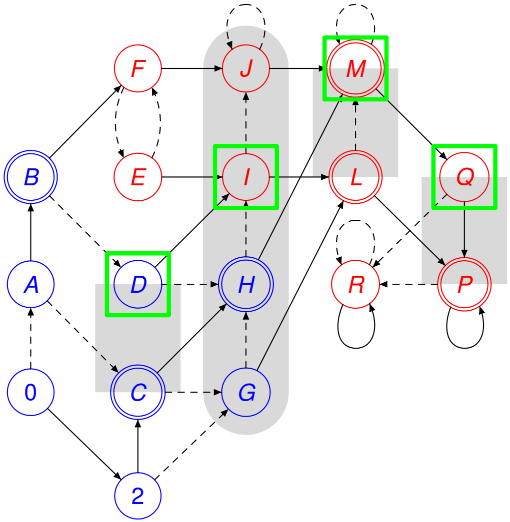
- don't-care nondeterministic
- select representative of each block; kernel state if possible
- merge all preamble states into their representative

[BADR, GEFFERT, SHIPMAN: *Hyper-minimizing minimized deterministic finite state automata*. ITA 2009]

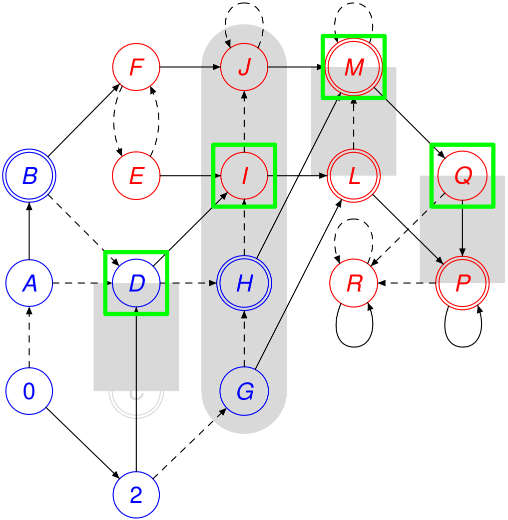
Merging states



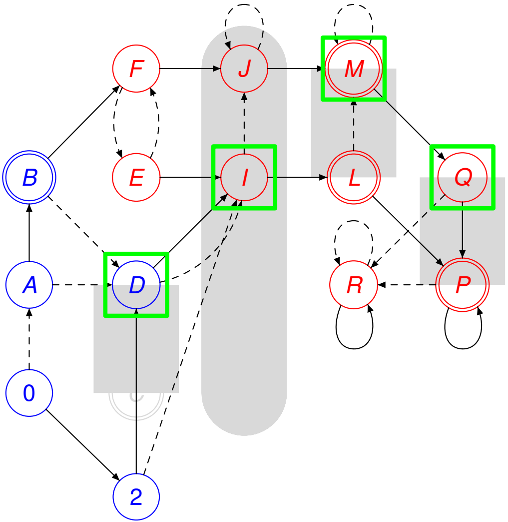
Merging states



Merging states



Merging states



Merging states

Definition

A DFA is **hyper-minimal** if all almost equivalent DFA are larger.

Merging states

Definition

A DFA is **hyper-minimal** if all almost equivalent DFA are larger.

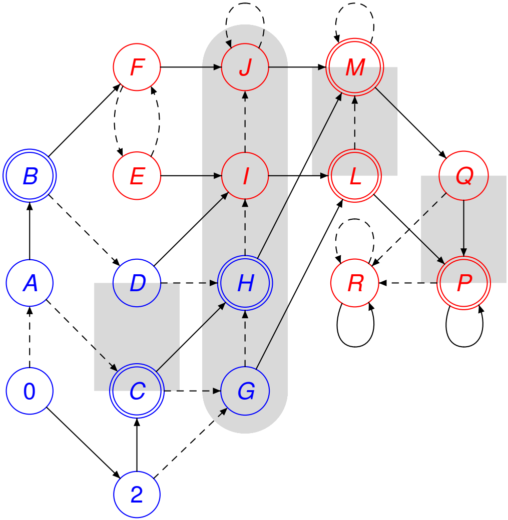
Theorem

A DFA is hyper-minimal if and only if

- *it is **minimal***
- *no preamble state is almost equivalent to another state*

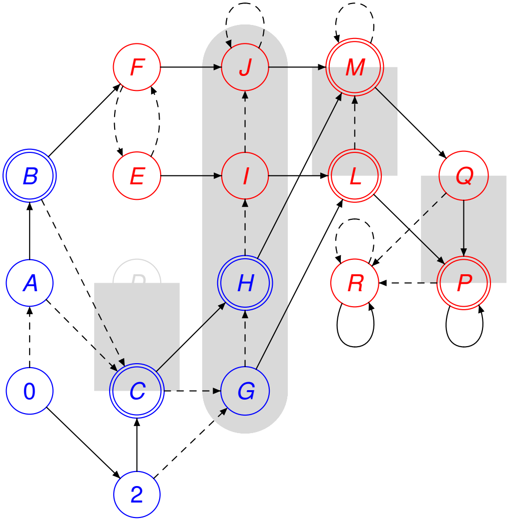
[BADR, GEFFERT, SHIPMAN: *Hyper-minimizing minimized deterministic finite state automata*. ITA 2009]

Merging states



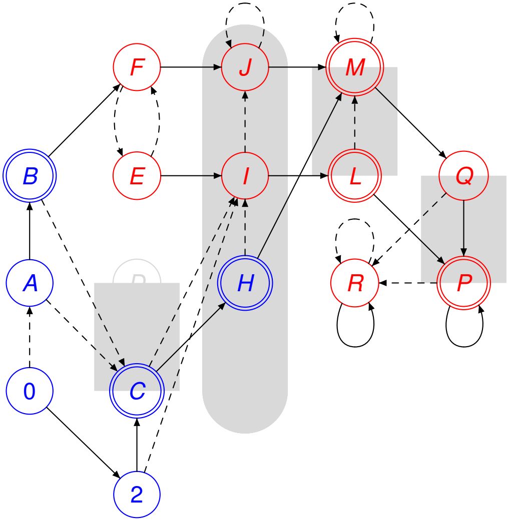
merges:
D into *C*

Merging states



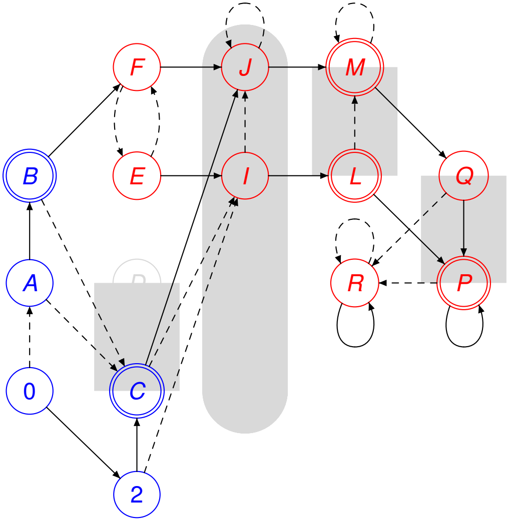
merges:
D into *C*
G into *I*

Merging states



merges:
D into *C*
G into *I*
H into *J*

Merging states



merges:

D into *C*

G into *I*

H into *J*

Merging states

Theorem

Merging can be done in linear time.

Merging states

Theorem

Merging can be done in linear time.

Theorem

Hyper-minimization can be achieved in time $O(n \log n)$.

Contents

Hyper-optimization

Goal

Obtain a DFA that

1. makes only finitely many mistakes
2. is as small as possible

Hyper-optimization

Goal

Obtain a DFA that

1. makes only finitely many mistakes
2. is as small as possible
3. **additionally makes minimal number of mistakes**

Question

- Can it be done in polynomial time?
  [BADR, GEFFERT, SHIPMAN 2009]
- Can it be done in $O(n \log n)$?

Hyper-optimization

Goal

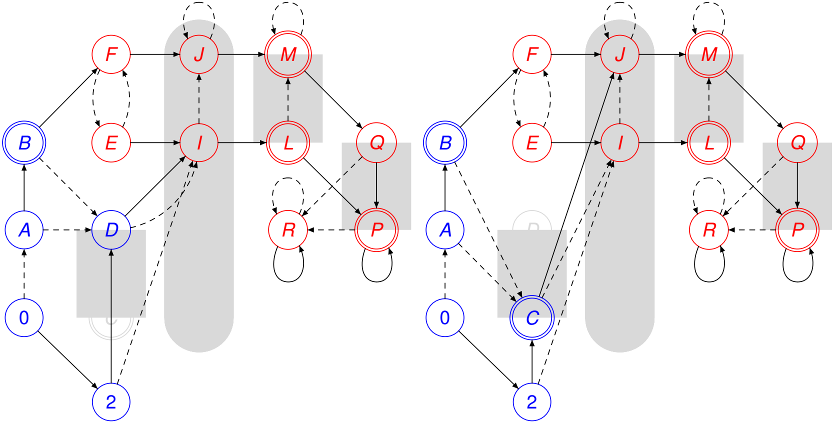
Obtain a DFA that

1. makes only finitely many mistakes
2. is as small as possible
3. **additionally makes minimal number of mistakes**

Question

- Can it be done in polynomial time? ✓
  [Badr, Geffert, Shipman 2009]
- Can it be done in $O(n \log n)$? ???

Comparison



Comparison

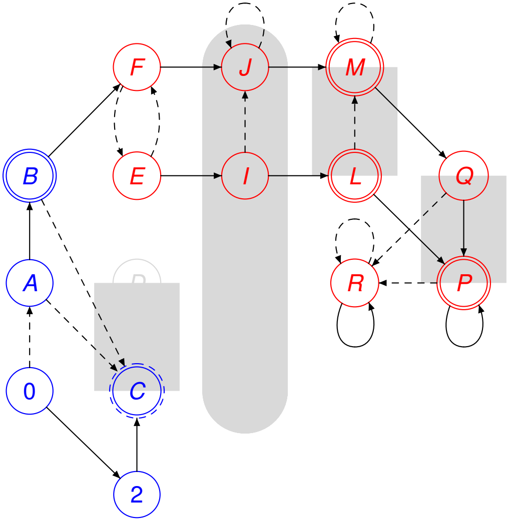
Theorem

Two almost equivalent, hyper-minimal DFA are isomorphic up to

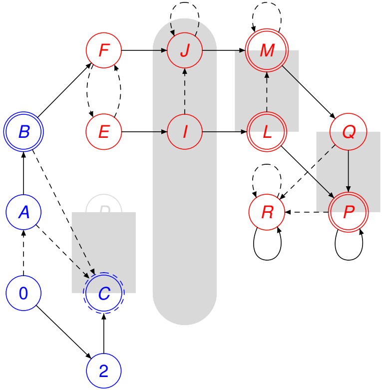
1. *finality of preamble states*
2. *transitions from preamble to kernel states*
3. *initial state*

[BADR, GEFFERT, SHIPMAN: *Hyper-minimizing minimized deterministic finite state automata*. ITA 2009]

Optimal merges



Finality of preamble states

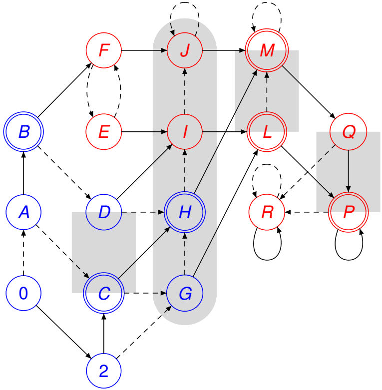


Question

Which words lead to C?

word w	$w \in L$
$\longrightarrow^2$	
$\dashrightarrow^2$	
$\dashrightarrow \longrightarrow \dashrightarrow$	

Finality of preamble states

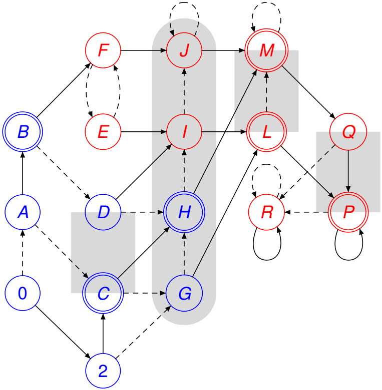


Question

Which words lead to *C*?

word w	$w \in L$
$\longrightarrow^2$	✓
$\dashrightarrow^2$	✓
$\dashrightarrow \longrightarrow \dashrightarrow$	✗

Finality of preamble states



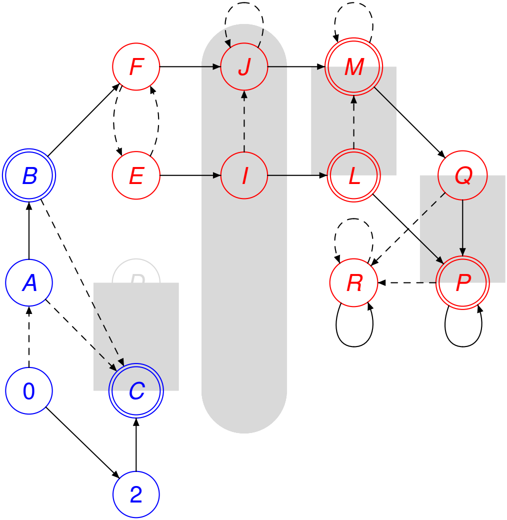
Question

Which words lead to C?

word w	$w \in L$
$\longrightarrow^2$	✓
$\dashrightarrow^2$	✓
$\dashrightarrow \longrightarrow \dashrightarrow$	✗

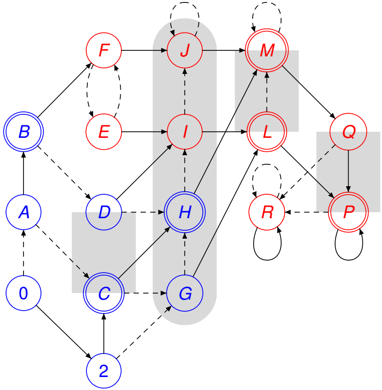
$\Rightarrow$ **make C final**

Optimal merges



Errors

Transitions from preamble to kernel states

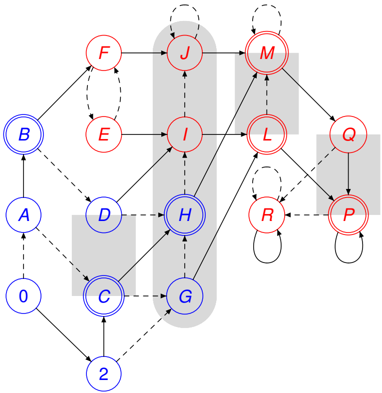


Question

On which words differ almost equivalent states?

states	words (number)
P–Q	ε (1)

Transitions from preamble to kernel states

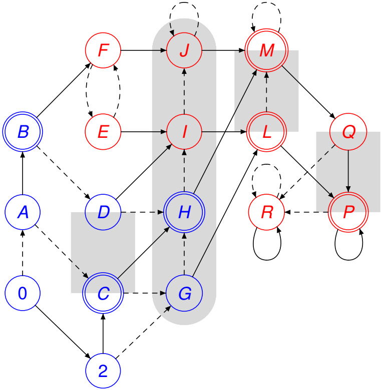


Question

On which words differ almost equivalent states?

states	words (number)
P–Q	ε (1)
L–M	$\longrightarrow$ (1)

Transitions from preamble to kernel states

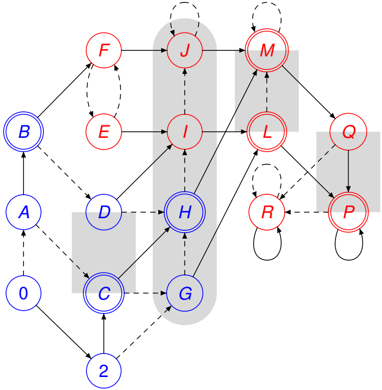


Question

On which words differ almost equivalent states?

states	words (number)
P–Q	ε (1)
L–M	$\longrightarrow$ (1)
I–J	$\longrightarrow^2$ (1)

Transitions from preamble to kernel states

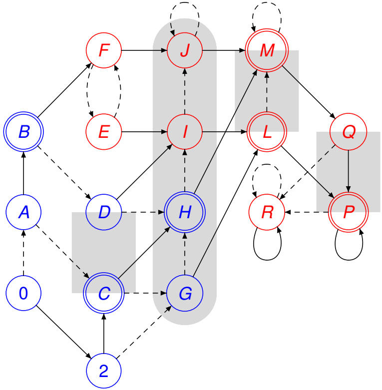


Question

On which words differ almost equivalent states?

states	words (number)
P–*Q*	ε (1)
L–*M*	$\longrightarrow$ (1)
I–*J*	$\longrightarrow^2$ (1)
H–*J*	ε, $\dashrightarrow$ $\longrightarrow^2$ (2)

Transitions from preamble to kernel states

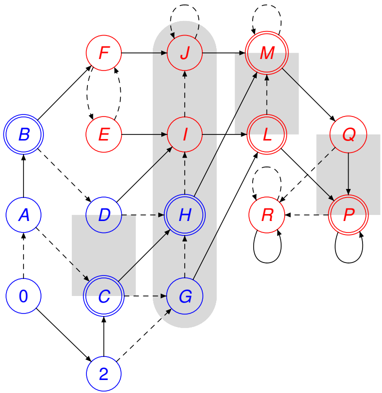


Question

On which words differ almost equivalent states?

states	words (number)
P–*Q*	ε (1)
L–*M*	$\longrightarrow$ (1)
I–*J*	$\longrightarrow^2$ (1)
H–*J*	ε, $\dashrightarrow\longrightarrow^2$ (2)
H–*I*	ε, $\dashrightarrow\longrightarrow^2$, $\longrightarrow^2$ (3)

Transitions from preamble to kernel states

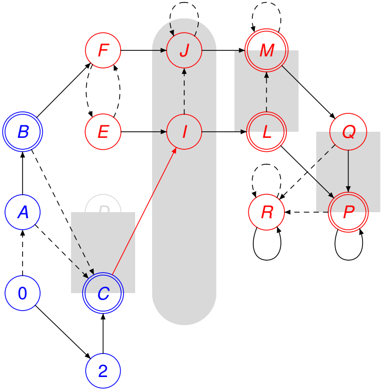


Question

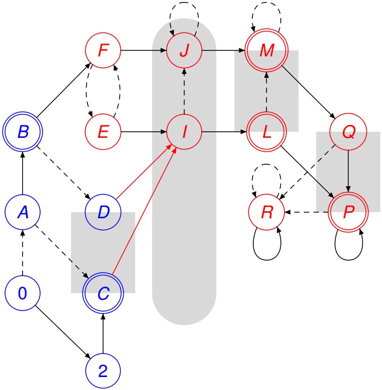
On which words differ almost equivalent states?

states	words (number)
P–Q	ε (1)
L–M	$\longrightarrow$ (1)
I–J	$\longrightarrow^2$ (1)
H–J	ε, $\dashrightarrow \longrightarrow^2$ (2)
H–I	ε, $\dashrightarrow \longrightarrow^2$, $\longrightarrow^2$ (3)
G–J	. . . (3)
G–I	. . . (2)
G–H	. . . (5)

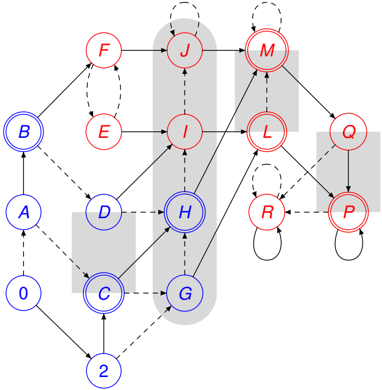
Transitions from preamble to kernel states



Transitions from preamble to kernel states



Transitions from preamble to kernel states



Errors

$u \longrightarrow w$

- u leads to C
- w error between H–I

$\longrightarrow^2 \longrightarrow$

$\longrightarrow^2 \longrightarrow \dashrightarrow \longrightarrow^2$

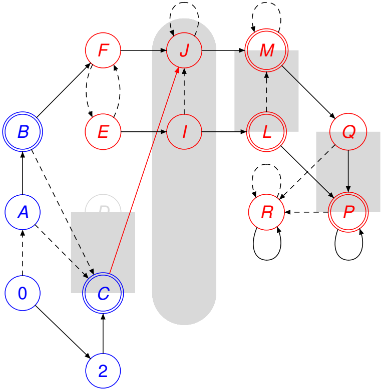
$\longrightarrow^2 \longrightarrow \longrightarrow^2$

$\dashrightarrow^2 \longrightarrow$

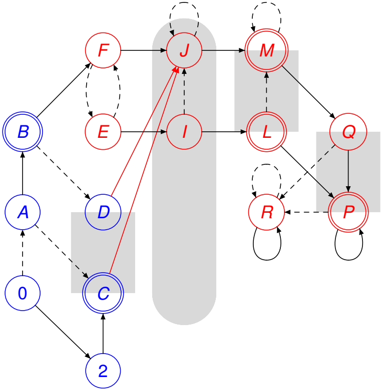
$\dashrightarrow^2 \longrightarrow \dashrightarrow \longrightarrow^2$

$\dashrightarrow^2 \longrightarrow \longrightarrow^2$ (6)

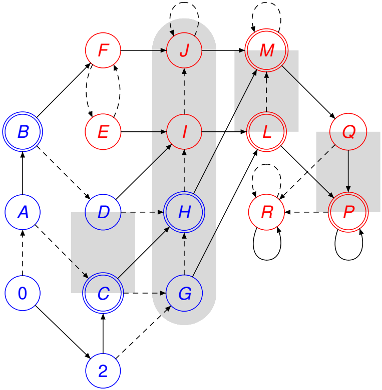
Transitions from preamble to kernel states



Transitions from preamble to kernel states



Transitions from preamble to kernel states



Errors

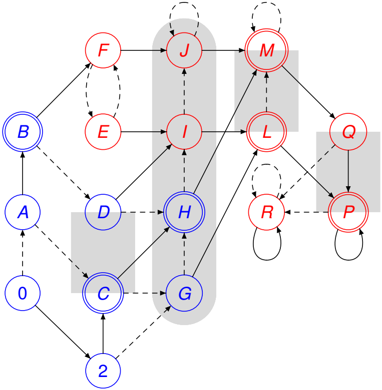
$u \longrightarrow w$

- u leads to C (2)
- $w \in H$–J (2)

or

- u leads to D (1)
- $w \in I$–J (1)

Transitions from preamble to kernel states



Errors

$u \longrightarrow w$

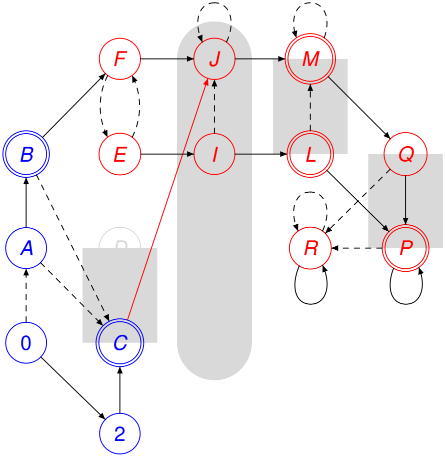
- u leads to C (2)
- $w \in H$–J (2)

or

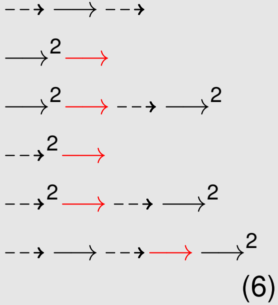
- u leads to D (1)
- $w \in I$–J (1)

$\Rightarrow$ **only**
$2 \cdot 2 + 1 \cdot 1 = 5$ **errors**

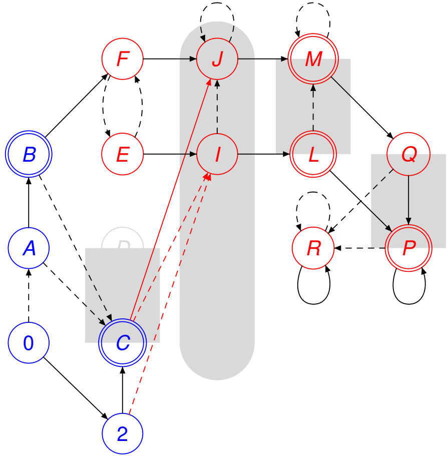
Optimal merges (cont'd)



Errors



Optimal merges (cont'd)



Errors

$\dashrightarrow \; \longrightarrow \; \dashrightarrow$

$\longrightarrow^2 \; \color{red}{\longrightarrow}$

$\longrightarrow^2 \; \color{red}{\longrightarrow} \; \dashrightarrow \; \longrightarrow^2$

$\dashrightarrow^2 \; \color{red}{\longrightarrow}$

$\dashrightarrow^2 \; \color{red}{\longrightarrow} \; \dashrightarrow \; \longrightarrow^2$

$\dashrightarrow \; \longrightarrow \; \dashrightarrow \; \color{red}{\longrightarrow} \; \longrightarrow^2$

. . . **(15)**

Result

Theorem

Hyper-optimization can be achieved in $O(n^2)$.

[$\sim$: *Better hyper-minimization — not as fast, but fewer errors*. CIAA 2010]

Result

Theorem

Hyper-optimization can be achieved in $O(n^2)$.

Theorem

We can compute the number of errors of a hyper-minimal DFA relative to any almost equivalent DFA in time $O(n^2)$.

[$\sim$: *Better hyper-minimization — not as fast, but fewer errors*. CIAA 2010]

Result

Theorem

Hyper-optimization can be achieved in $O(n^2)$.

Theorem

We can compute the number of errors of a hyper-minimal DFA relative to any almost equivalent DFA in time $O(n^2)$.

Open question

Can it also be done in $O(n \log n)$?

[$\sim$: *Better hyper-minimization — not as fast, but fewer errors*. CIAA 2010]

Contents

Canonical languages

Definition

Language L is **canonical** if recognized by a hyper-minimal DFA.

Open question

What are the closure properties of canonical languages?

[BADR, GEFFERT, SHIPMAN: *Hyper-minimizing minimized deterministic finite state automata*. ITA 2009]

Canonical languages

class	compl.	$\cup$	$\cap$	conc.	$*$	reversal
regular	✓	✓	✓	✓	✓	✓
canonical	✓	✗	✗	✗	?	?

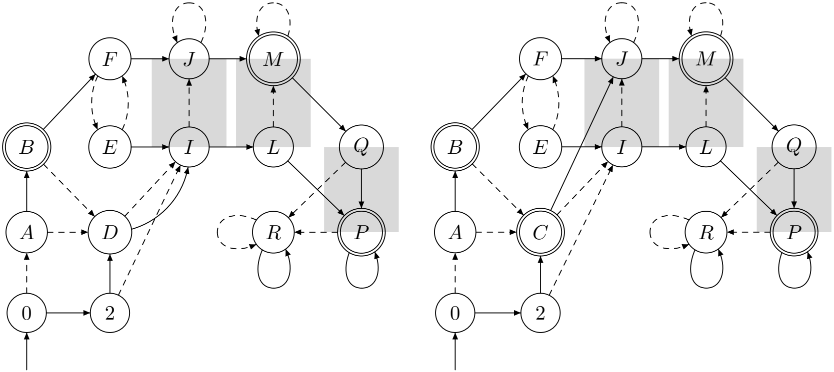
Table: Closure properties

[BADR, GEFFERT, SHIPMAN: *Hyper-minimizing minimized deterministic finite state automata*. ITA 2009]

[$\sim$: *Notes on hyper-minimization*. AFL 2011]

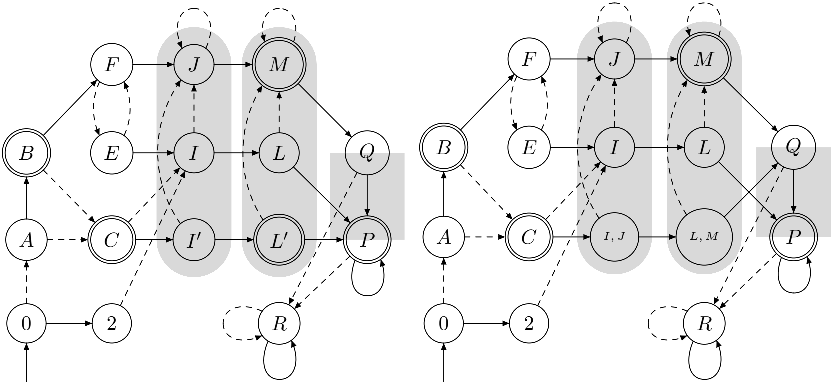
Canonical languages

Input hyper-minimal DFA



Canonical languages

Minimal DFA recognizing union and intersection



k-minimization

Definition

Two languages L_1, L_2 are k**-equivalent** if $L_1 \triangle L_2 \subseteq \Sigma^{<k}$.

Definition

The **gap** of two states p, q is

$$\text{gap}(p, q) = \sup\{|w| \mid \delta(p, w) \in F \text{ xor } \delta(q, w) \in F\}$$

with $\sup\ \emptyset = -\infty$

[GAWRYCHOWSKI, JEŻ: *Hyper-minimisation made efficient*. MFCS 2009]

k-minimization

Definition

Level of p is length of longest word leading to p

$$\text{level}(p) = \sup \{|w| \mid \delta(q_0, w) = p\}$$

k-minimization

Definition

Level of p is length of longest word leading to p

$$\text{level}(p) = \sup \{|w| \mid \delta(q_0, w) = p\}$$

Definition

Two states p, q are k-**similar** iff

$$\text{gap}(p, q) + \min(k, \text{level}(p), \text{level}(q)) < k$$

k-minimization

Note

k-similarity is not an equivalence!

[GAWRYCHOWSKI, JEŻ: *Hyper-minimisation made efficient*. MFCS 2009]
[GAWRYCHOWSKI, JEŻ, $\sim$: *On minimising automata with errors*. MFCS 2011]

k-minimization

Note

k-similarity is not an equivalence!

Theorem

k-minimization can be done in time $O(n \log n)$.

[GAWRYCHOWSKI, JEŻ: *Hyper-minimisation made efficient*. MFCS 2009]

[GAWRYCHOWSKI, JEŻ, $\sim$: *On minimising automata with errors*. MFCS 2011]

Optimize other criteria

So far

1. make it as small as possible
(allowing any finite number of errors)
2. minimize the number of errors

Optimize other criteria

So far

1. make it as small as possible
   (allowing any finite number of errors)
2. minimize the number of errors

More desirable

- optimize a ratio of saved states to committed errors

Error-bounded hyper-minimization

Problem

- given DFA A, integers m, s
- construct a DFA with
  - at most s states
  - making at most m errors

Note

trivial with only one restriction

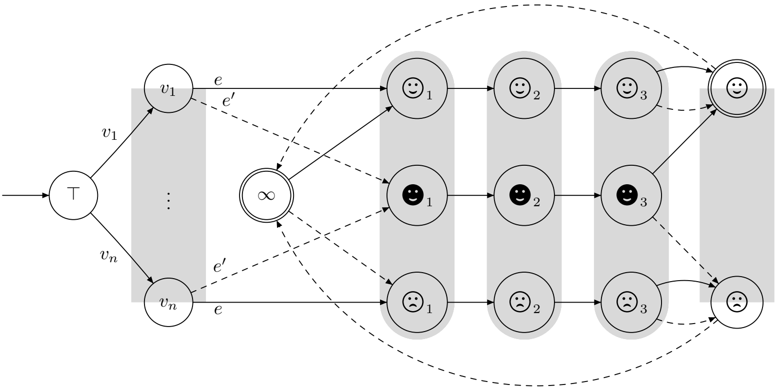
Bad news

Theorem

Error-bounded hyper-minimization is NP-complete.

[GAWRYCHOWSKI, JEŻ, ∼: *On minimising automata with errors*. MFCS 2011]

Bad news



- using reduction from 3-coloring problem
- redirection from middle to outside for 1 error
- redirection from outside to anywhere for 2 errors

Optimize other criteria (again)

So far for k-minimization

1. make it as small as possible
(allowing any number of errors of length at most k)

Optimize other criteria (again)

So far for k-minimization

1. make it as small as possible
(allowing any number of errors of length at most k)

More desirable

- minimize the number of committed errors

Optimal k-minimization

Problem

- given DFA A, integer k
- construct a DFA that
  - is k-minimal for A
  - commits the least number of errors for all such DFA

Note

Optimal hyper-minimization was possible in time $O(n^2)$.

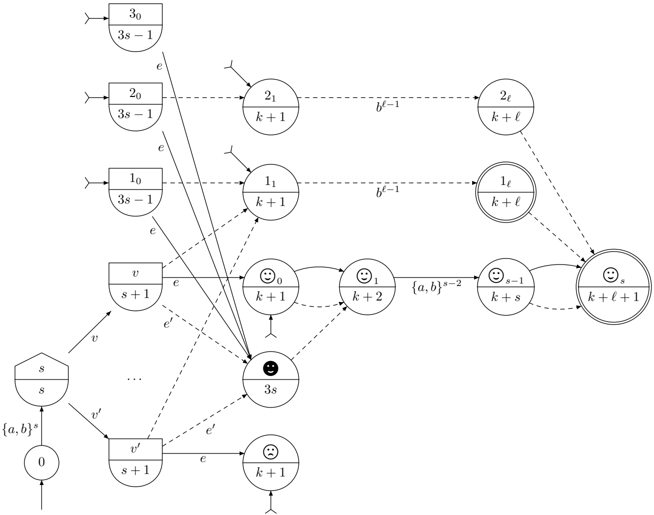
More bad news

Theorem

Optimal k-minimization is NP-complete.

[GAWRYCHOWSKI, JEŻ, $\sim$: *On minimising automata with errors*. MFCS 2011]

More bad news



Summary

Open questions [BADR, GEFFERT, SHIPMAN]

- ✓ Properties of canonical languages
- ✗ Asymptotic state complexity
- ✗ Hyper-minimization of NFA, AFA, 2FA, WDFA
- ✓ Efficient hyper-minimization algorithm
- ✓ Minimize the number of errors
- ✗ Minimize length of longest error
- ✓ k-minimization

[BADR, GEFFERT, SHIPMAN: *Hyper-minimizing minimized deterministic finite state automata*. ITA 2009]

That's all, folks!

Thank you for your attention!