Better Hyper-Minimization
Not as Fast, but Fewer Errors

Andreas Maletti

Universitat Rovira i Virgili
Tarragona, Spain

andreas.maletti@urv.cat

Winnipeg — August 13, 2010

Minimization

Observation

- minimal DFA too large to handle

Remedy

To make minimal DFA even smaller:

- sacrifice determinism
- sacrifice correctness

Hyper-minimization (Badr, Geffert, Shipman 2009)

Obtain a DFA that

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

Minimization

Observation

- minimal DFA too large to handle

Remedy

To make minimal DFA even smaller:

- ~~sacrifice determinism~~
- sacrifice correctness (allow a finite number of mistakes)

Hyper-minimization (Badr, Geffert, Shipman 2009)

Obtain a DFA that

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

Minimization

Observation

- minimal DFA too large to handle

Remedy

To make minimal DFA even smaller:

- ~~sacrifice determinism~~
- sacrifice correctness (allow a finite number of mistakes)

Hyper-minimization (Badr, Geffert, Shipman 2009)

Obtain a DFA that

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

Hyper-minimization

Algorithms

- $O(n^3)$ [BADR, GEFFERT, SHIPMAN 2009]
- $O(n^2)$ [BADR 2009]
- $O(n \log n)$ [GAWRYCHOWSKI, JEŻ 2009], [HOLZER, ∼ 2009]

Results (data by [QUERNHEIM 2010])

	errors					errors		
line	max	min	%		line	max	min	%
3	39.5	26.0	34.2		7	14.9	1.7	88.6
4	182.6	39.0	78.6		8	11.4	2.3	79.8
5	66.5	6.4	90.4		15	356.4	18.2	94.9
6	13.5	0.5	96.3		16	516.2	67.4	86.9

Hyper-minimization

Algorithms

- $O(n^3)$ [Badr, Geffert, Shipman 2009]
- $O(n^2)$ [Badr 2009]
- $O(n \log n)$ [Gawrychowski, Jeż 2009], [Holzer, ∼ 2009]

Results (data by [Quernheim 2010])

line	errors max	errors min	%
3	39.5	26.0	34.2
4	182.6	39.0	78.6
5	66.5	6.4	90.4
6	13.5	0.5	96.3

line	errors max	errors min	%
7	14.9	1.7	88.6
8	11.4	2.3	79.8
15	356.4	18.2	94.9
16	516.2	67.4	86.9

Hyper-optimization

Hyper-optimization

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

Hyper-optimization

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 $O(n \log n)$?

Contents

1. Motivation
2. Hyper-Minimization
3. Hyper-Optimization

Basic definitions

Definition (almost-equivalent)

- Two languages are almost-equivalent if their difference is finite.
- Two DFA are almost-equivalent if their languages are.

Example

- all finite languages are almost-equivalent
- $\{a^n \mid n \in \mathbb{N}\}$ and $\{aaa^n \mid n \in \mathbb{N}\}$ are almost-equivalent
- $\{a^n \mid n \in \mathbb{N}\}$ and $\{a^{2n} \mid n \in \mathbb{N}\}$ are not almost-equivalent

Definition (hyper-minimal)

A DFA is hyper-minimal if there is no smaller almost-equivalent DFA.

Basic definitions

Definition (almost-equivalent)

- Two languages are almost-equivalent if their difference is finite.
- Two DFA are almost-equivalent if their languages are.

Example

- all finite languages are almost-equivalent
- $\{a^n \mid n \in \mathbb{N}\}$ and $\{aaa^n \mid n \in \mathbb{N}\}$ are almost-equivalent
- $\{a^n \mid n \in \mathbb{N}\}$ and $\{a^{2n} \mid n \in \mathbb{N}\}$ are not almost-equivalent

Definition (hyper-minimal)

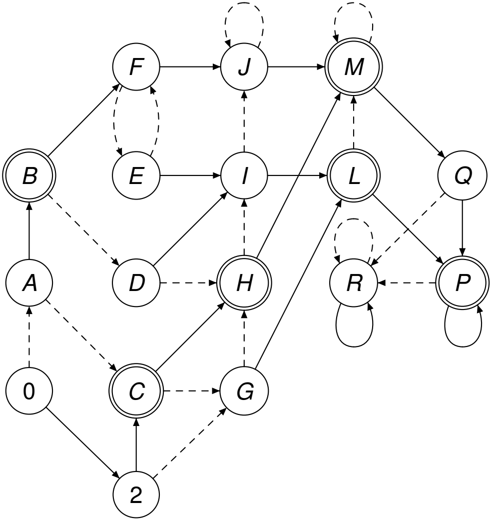
A DFA is hyper-minimal if there is no smaller almost-equivalent DFA.

Preamble and kernel states

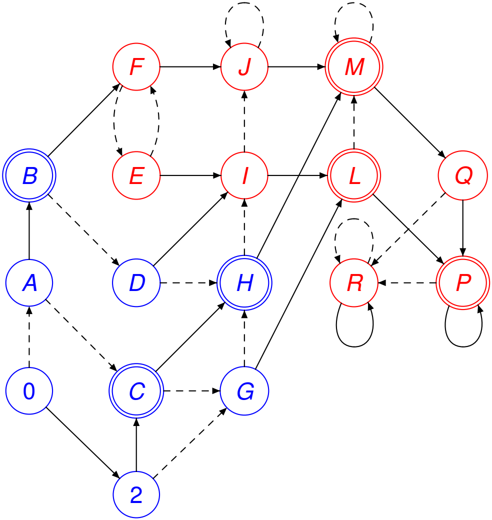
Definition

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

Preamble and kernel states (cont'd)



Preamble and kernel states (cont'd)



Almost-equivalent states

Definition

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

Consequence

Almost-equivalent states have almost-equivalent right-languages.

Almost-equivalent states

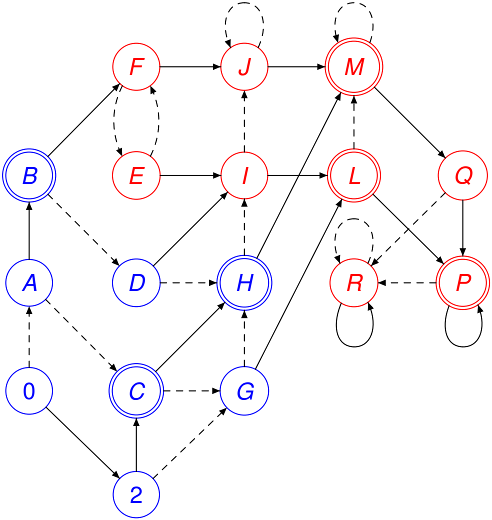
Definition

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

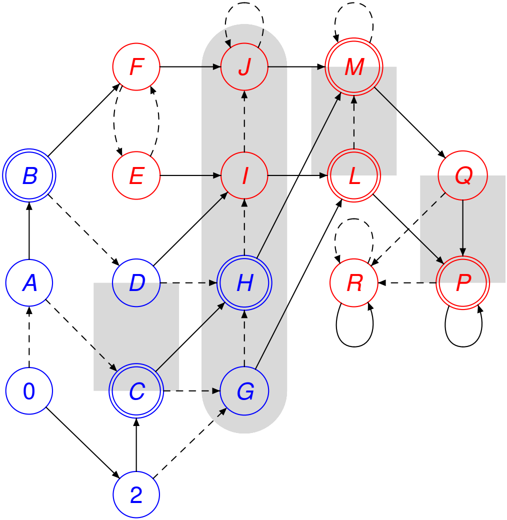
Consequence

Almost-equivalent states have almost-equivalent right-languages.

Almost-equivalent states (cont'd)



Almost-equivalent states (cont'd)

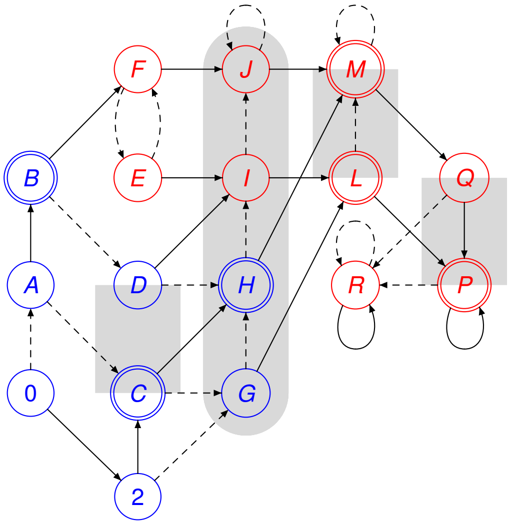


Merging states

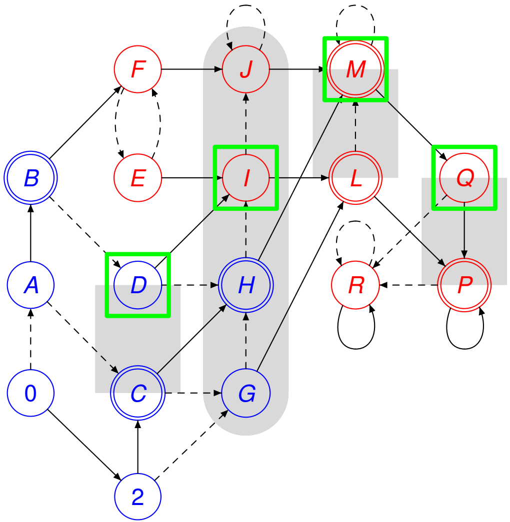
Algorithm (Badr, Geffert, Shipman 2009)

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

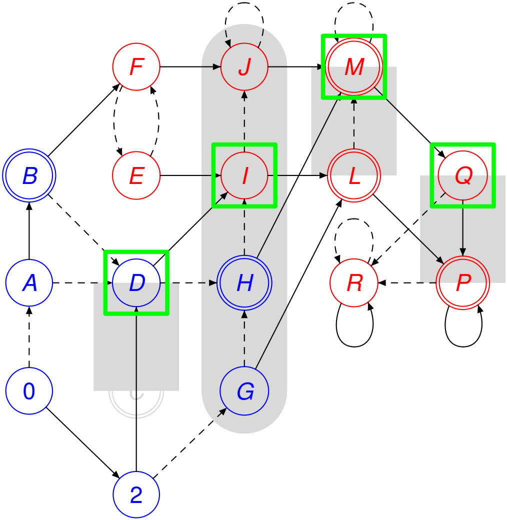
Merging states (cont'd)



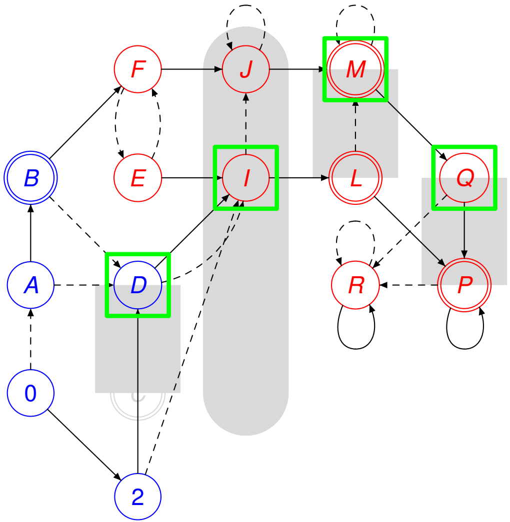
Merging states (cont'd)



Merging states (cont'd)



Merging states (cont'd)



Merging states (cont'd)

Theorem (BADR, GEFFERT, SHIPMAN 2009)

DFA is hyper-minimal if and only if

- *no unreachable states*
- *no equivalent states*
- *no preamble state is almost-equivalent to another state*

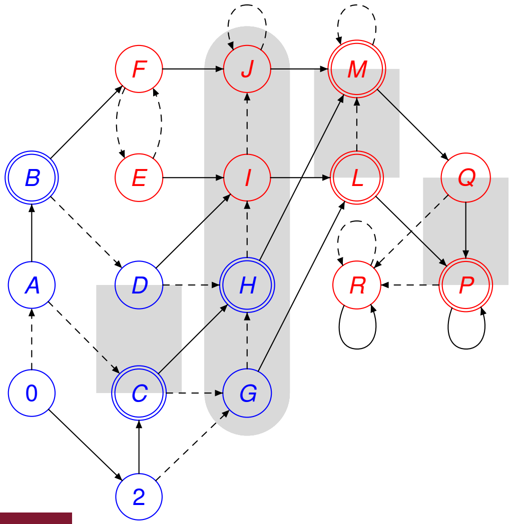
Merging states (cont'd)

Theorem (BADR, GEFFERT, SHIPMAN 2009)

DFA is hyper-minimal if and only if

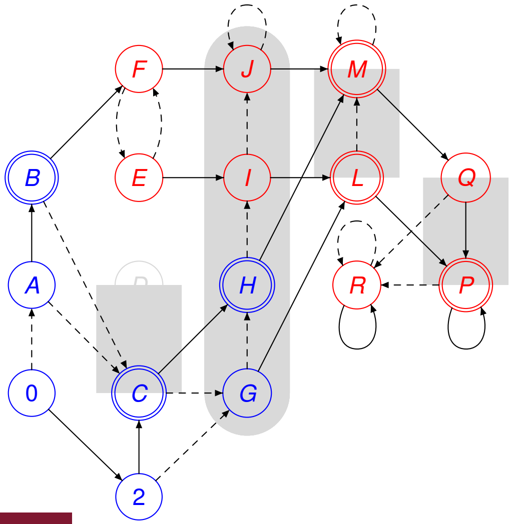
- *minimal*
- *no preamble state is almost-equivalent to another state*

Merging states (cont'd)



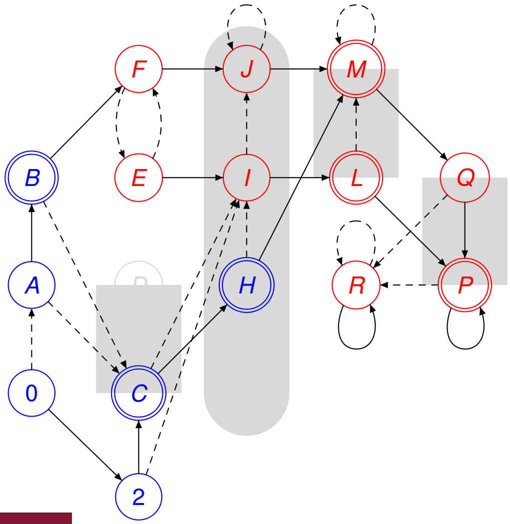
merges:
D into *C*

Merging states (cont'd)



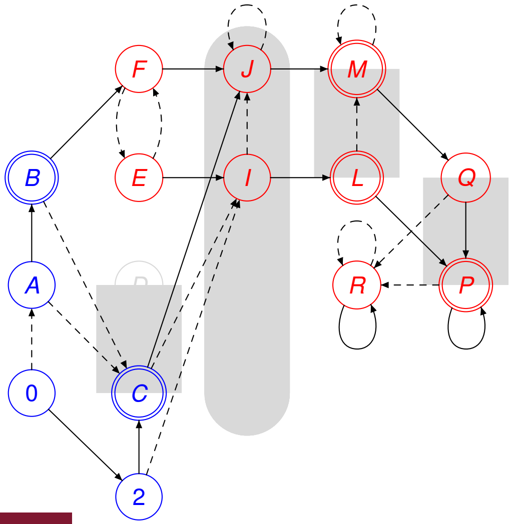
merges:
D into *C*
G into *I*

Merging states (cont'd)



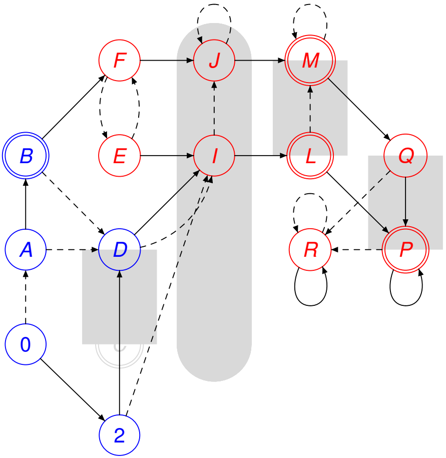
merges:
D into *C*
G into *I*
H into *J*

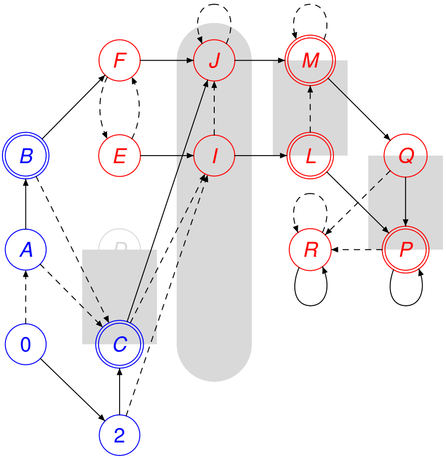
Merging states (cont'd)



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

Comparison





Comparison (cont'd)

Theorem (Badr, Geffert, Shipman 2009)

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*

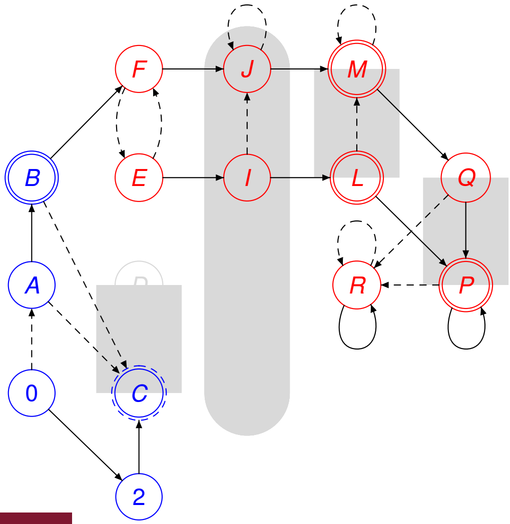
Comparison (cont'd)

Theorem (Badr, Geffert, Shipman 2009)

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

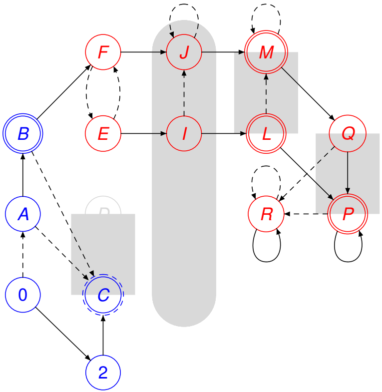
1. *finality of preamble states*
2. *transitions from preamble to kernel states*
3. *initial state*

Optimal merges



Errors

Finality of preamble states

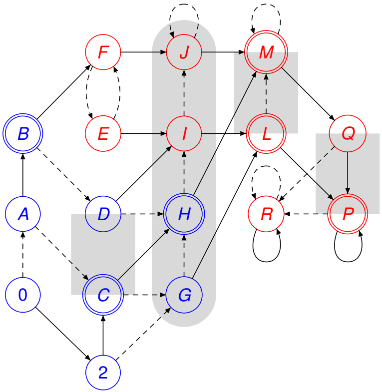


Question

Which words lead to C?

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

Finality of preamble states

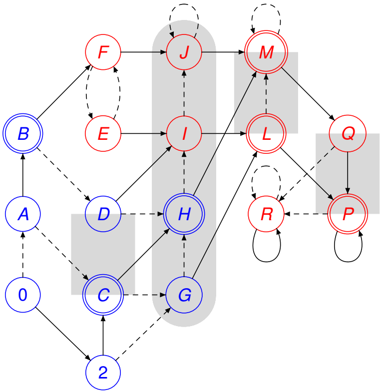


Question

Which words lead to C?

word w	$w \in L$
$\longrightarrow$ $\longrightarrow$	✓
⇢ ⇢	✓
⇢ $\longrightarrow$ ⇢	✗

Finality of preamble states



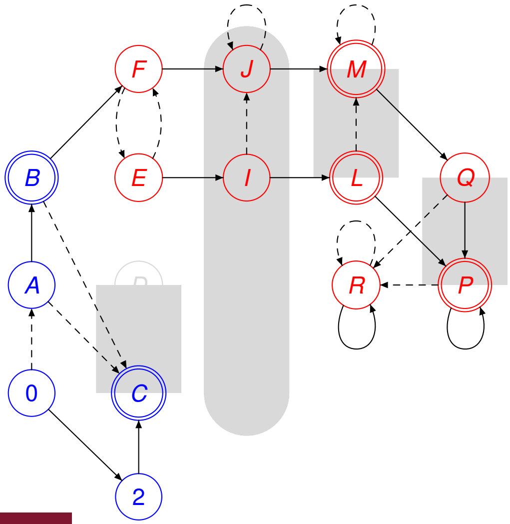
Question

Which words lead to C?

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

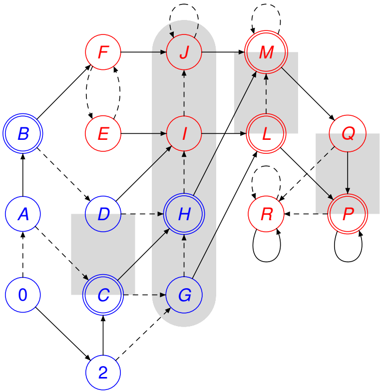
$\Rightarrow$ make C final

Optimal merges (cont'd)



Errors

Transitions from preamble to kernel states

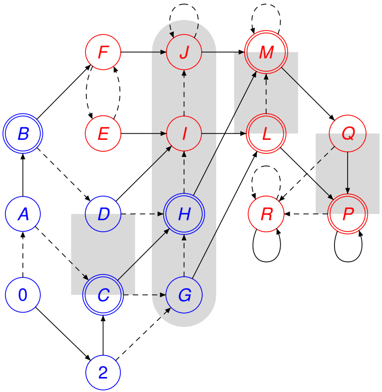


Question

On which words differ almost-equivalent states?

states	words (number)
P–Q	ε (1)

Transitions from preamble to kernel states

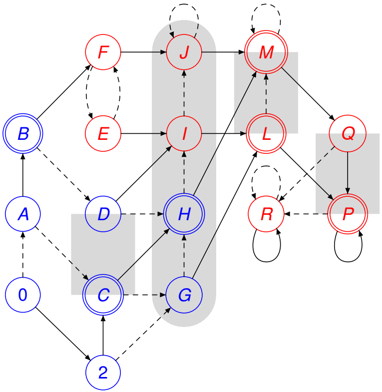


Question

On which words differ almost-equivalent states?

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

Transitions from preamble to kernel states

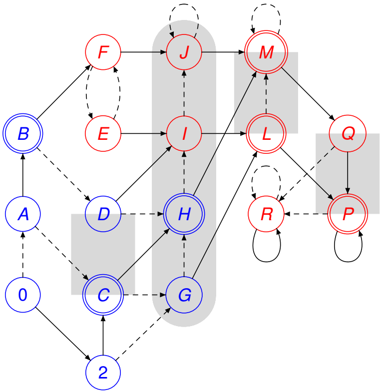


Question

On which words differ almost-equivalent states?

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

Transitions from preamble to kernel states

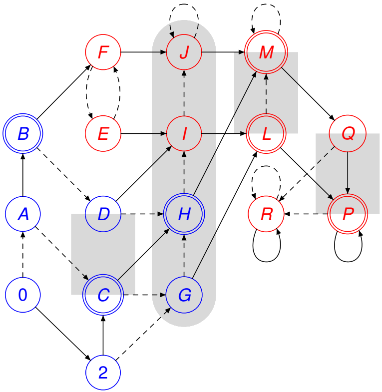


Question

On which words differ almost-equivalent states?

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

Transitions from preamble to kernel states

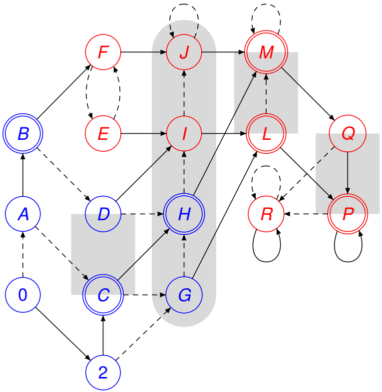


Question

On which words differ almost-equivalent states?

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

Transitions from preamble to kernel states

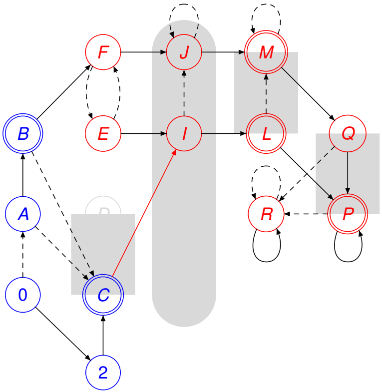


Question

On which words differ almost-equivalent states?

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

Transitions from preamble to kernel states (cont'd)



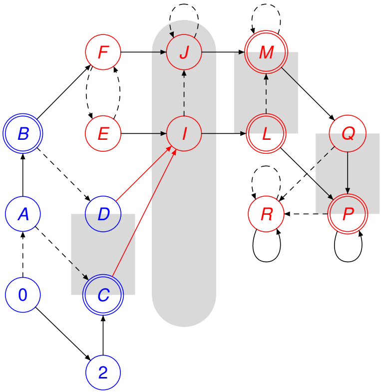
Errors

$$u \longrightarrow w$$

- u leads to C
- w error between H–I

(6)

Transitions from preamble to kernel states (cont'd)



Errors

$u \longrightarrow w$

- u leads to C
- w error between H–I

$\longrightarrow \longrightarrow \longrightarrow$

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

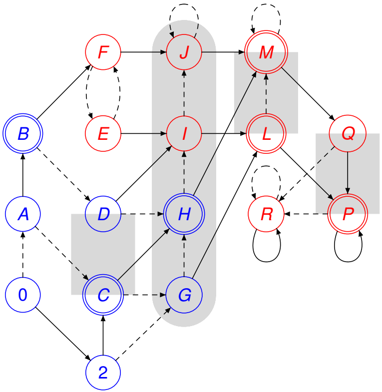
$\longrightarrow \longrightarrow \longrightarrow \longrightarrow \longrightarrow$

$\dashrightarrow \dashrightarrow \longrightarrow$

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

$\dashrightarrow \dashrightarrow \longrightarrow \longrightarrow \longrightarrow$ (6)

Transitions from preamble to kernel states (cont'd)



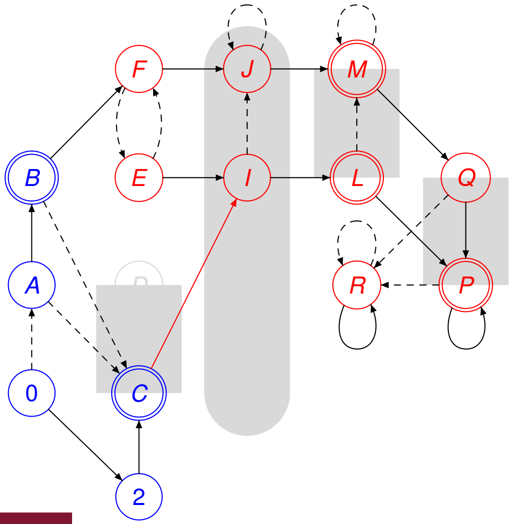
Errors

$$u \longrightarrow w$$

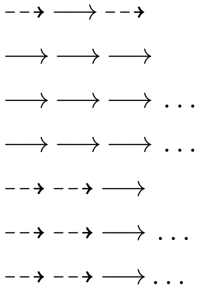
- u leads to C
- w error between H–I

$$\longrightarrow \longrightarrow \longrightarrow$$
$$\longrightarrow \longrightarrow \longrightarrow \dashrightarrow \longrightarrow \longrightarrow$$
$$\longrightarrow \longrightarrow \longrightarrow \longrightarrow \longrightarrow$$
$$\dashrightarrow \dashrightarrow \longrightarrow$$
$$\dashrightarrow \dashrightarrow \longrightarrow \dashrightarrow \longrightarrow \longrightarrow$$
$$\dashrightarrow \dashrightarrow \longrightarrow \longrightarrow \longrightarrow \qquad (6)$$

Optimal merges (cont'd)

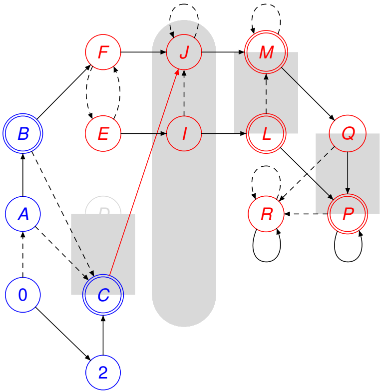


Errors



(7)

Transitions from preamble to kernel states (cont'd)



Errors

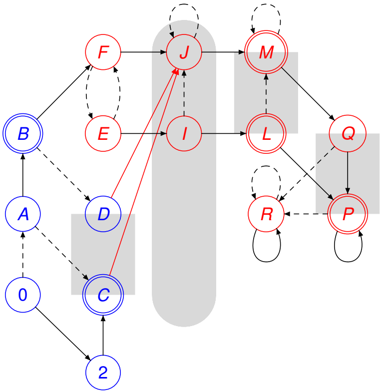
$$u \longrightarrow w$$

- 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 (cont’d)



Errors

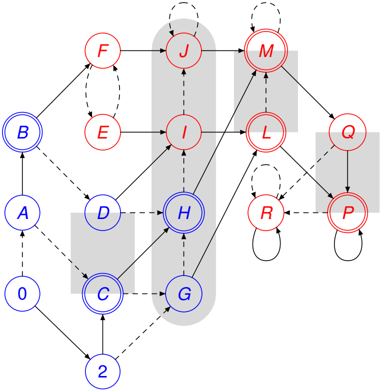
$$u \longrightarrow w$$

- u leads to C (2)
- $w \in H\text{–}J$ (2)

or

- u leads to D (1)
- $w \in I\text{–}J$ (1)

Transitions from preamble to kernel states (cont'd)



Errors

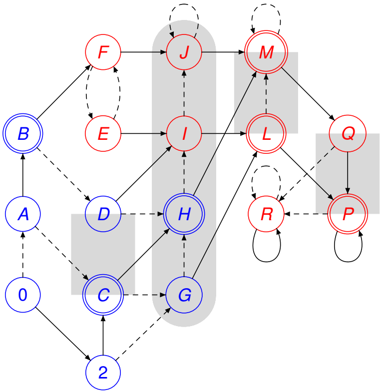
$$u \longrightarrow w$$

- u leads to C (2)
- $w \in H\text{–}J$ (2)

or

- u leads to D (1)
- $w \in I\text{–}J$ (1)

Transitions from preamble to kernel states (cont'd)



Errors

$$u \longrightarrow w$$

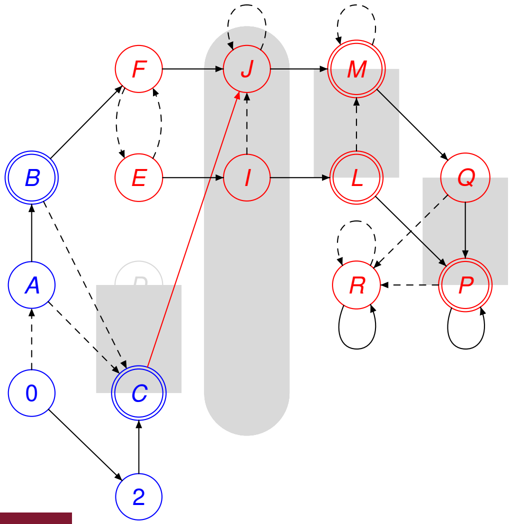
- u leads to C (2)
- $w \in H$–J (2)

or

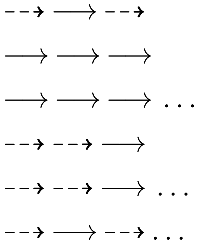
- u leads to D (1)
- $w \in I$–J (1)

$\Rightarrow$ only 5 errors

Optimal merges (cont'd)

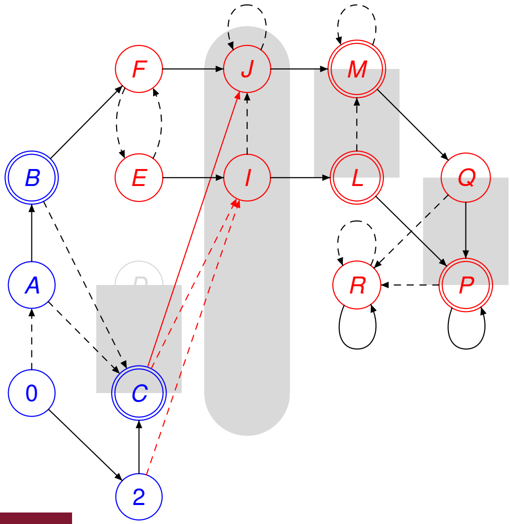


Errors

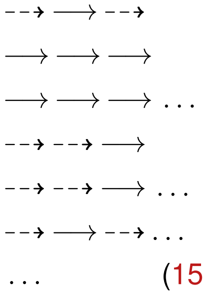


(6)

Optimal merges (cont'd)



Errors



Main result

Theorem

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

Open question

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

Main result

Theorem

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

Open question

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

References

- BADR: *Hyper-minimization in $O(n^2)$.* Int. J. Found. Comput. Sci. 20, 2009
- BADR, GEFFERT, SHIPMAN: *Hyper-minimizing minimized deterministic finite state automata*. ITA 43, 2009
- GAWRYCHOWSKI, JEŻ: *Hyper-minimisation made efficient*. MFCS 2009
- HOLZER, MALETTI: *An $n \log n$ algorithm for hyper-minimizing states in a (minimized) deterministic automaton*. CIAA 2009
- QUERNHEIM: *Hyper-minimisation of weighted finite automata*. Master thesis, 2010

Thank you for your attention!