Abstraction of State Languages in Automata Algorithms

David Chocholatý

Supervisor: doc. Mgr. Lukáš Holík, Ph.D.

BRNO UNIVERSITY OF TECHNOLOGY
FACULTY OF INFORMATION TECHNOLOGY

June 13, 2022

Motivation

- Finite automata
- Useful operations with automata,
  but expensive
  - Union $\cup$
  - Intersection $\cap$
    - Emptiness test
  - Complement $\overline{M}$
  - $\ldots$
- Regular model-checking, string solving, Presburger arithmetic, WS1S, $\ldots$

Problem

- NFA M_1 (and $M_2, \dots$)
- Computation of the result of automata operations

$$M = f(M_1)$$
$$L(M) = f(L(M_1))$$

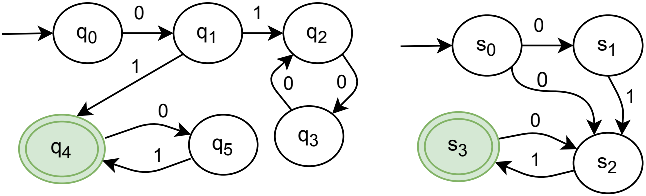
$$M = f(M_1, M_2, \dots)$$
$$L(M) = f(L(M_1), L(M_2), \dots)$$

- Large amount of states and transitions
- Nonterminating states

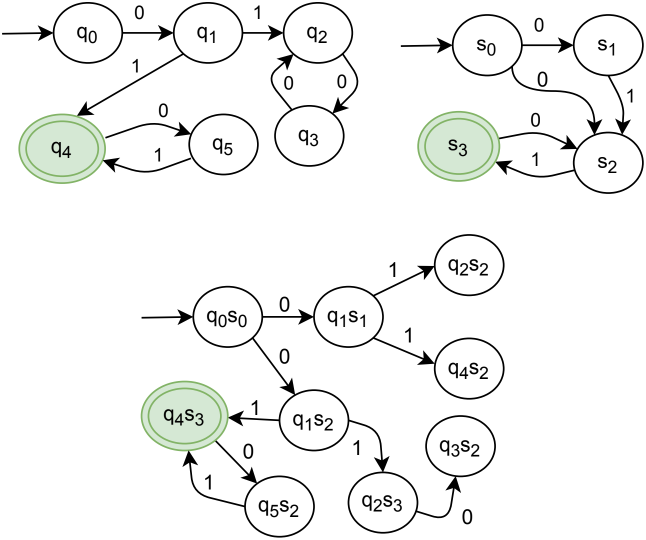
$\Longrightarrow$ Expensive generation of the result

Solution

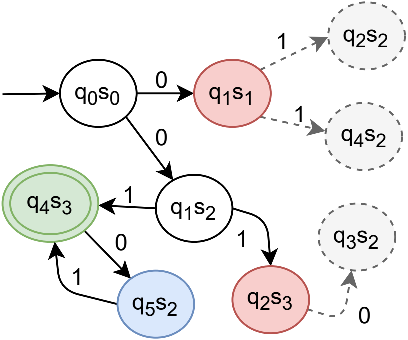
- Abstraction of state languages – different approaches
- Length abstraction
  - Lasso automata
  - Linear length formulae for potential product states
  - SMT solver – satisfiability test of linear length formulae
- Parikh's image – Parikh's theorem
  - Higher state pruning capabilities
  - Parikh's image semi-linear formulae for potential product states
  - SMT solver – satisfiability test of Parikh's image formulae
- Combined approaches
  - Minimization of necessary computations of Parikh's image formulae
  - Extensibility options for another abstractions
- Mintermization – combination with optimization methods


q0
0
q1
1
q2
1
0
0
q4
0
q5
1
q3
s0
0
s1
0
1
s3
0
s2
1

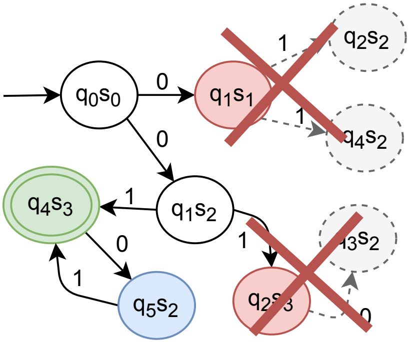
Product Generation

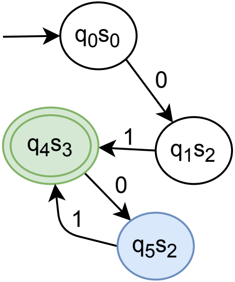


Optimized Product



Optimized Product



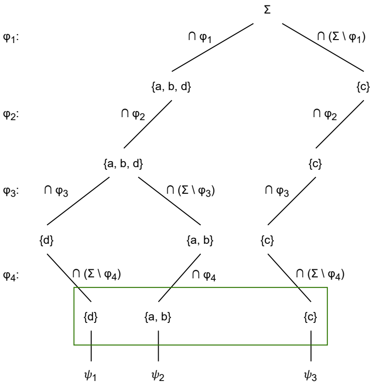

q_0s_0
0
q_1s_2
1
q_4s_3
0
1
q_5s_2

- Satisfiability of linear length and/or Parikh's image formulae
- Return values: satisfiable, unsatisfiable
- SMT solver limitation: computational time requirements, memory usage
  - Number of clauses to be solved and their complexity
  - Length abstraction formulae – small and cheap to compute
  - Parikh's image formulae – extensive formulae and expensive to compute

Common Constraints	Full Constraints	Product States	Time(s)
1782	2652	434	2092

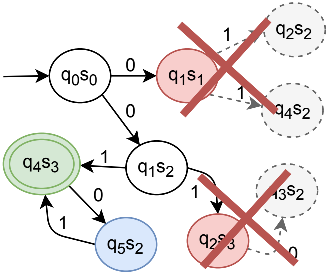
Future work and different approaches

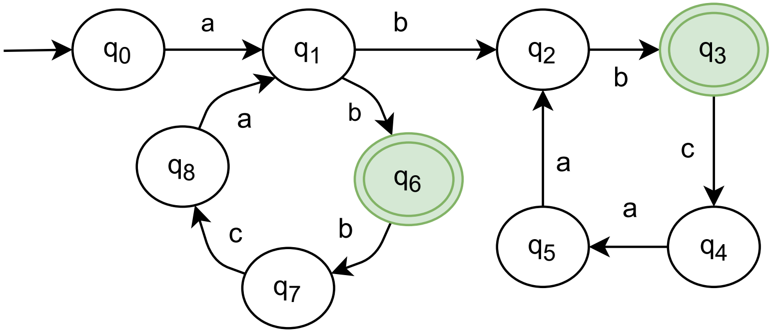
- Try different benchmarks
- Try different abstraction techniques
  - CEGAR, predicate abstraction, IMPACT, IC3/PDR
  - Attempts at applications on finite automata
    - IC3
    - Interpolar approach McMillan
- Parallelization possibilities
- States abstraction



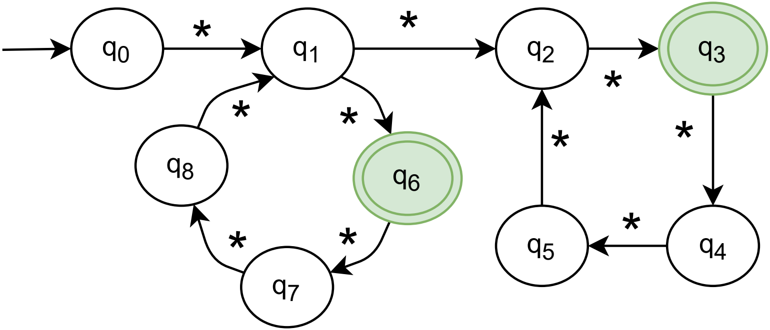
$$\varphi : \exists k(|w| = 2 \vee |w| = 4 + 2 \cdot k)$$
$$\psi : \exists l(|w| = 2 + 1 \cdot l)$$

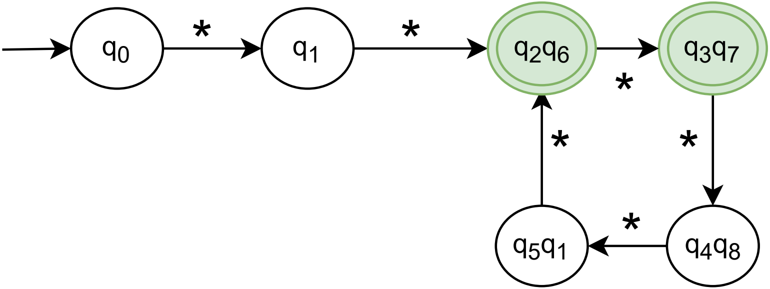
$$\exists k \exists l(2 \vee 4 + 2 \cdot k = 2 + 1 \cdot l)$$



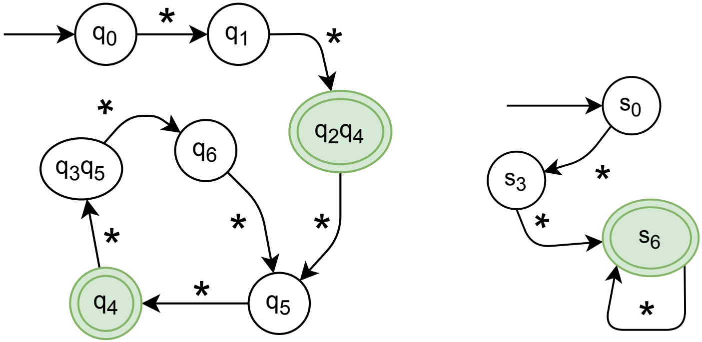

q0
q1
q2
q3
q4
q5
q6
q7
q8
a
b
b
c
a
a
b
b
c
a

NFA with a unary alphabet





$$\varphi : \exists k(|w| = 2 + 4 \cdot k \vee |w| = 3 + 4 \cdot k)$$



- SMT solver – satisfiability test of linear length abstraction formulae

$$\exists k \exists l (2 \vee 4 + 4 \cdot k = 2 + 1 \cdot l)$$

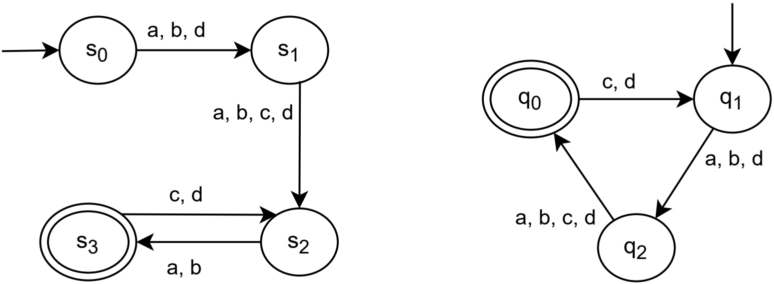
- Satisfiability of linear length formulae

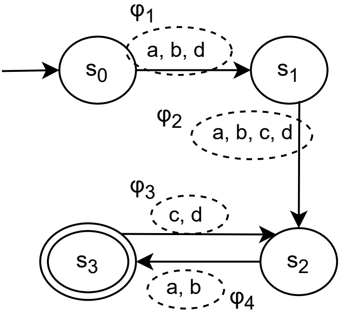
$$\phi : \exists k(|w| = 2 + 4 \cdot k \vee |w| = 3 + 4 \cdot k)$$
$$\psi : \exists l(|w| = 1 + 2 \cdot l \vee |w| = 5 + 5 \cdot l \vee |w| = 8)$$

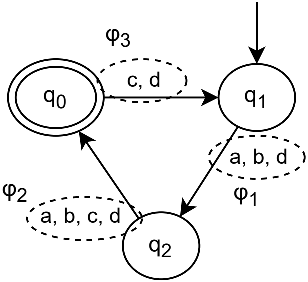
$$\exists k \exists l(2 + 4 \cdot k \vee 3 + 4 \cdot k = 1 + 2 \cdot l \vee 5 + 5 \cdot l \vee 8)$$

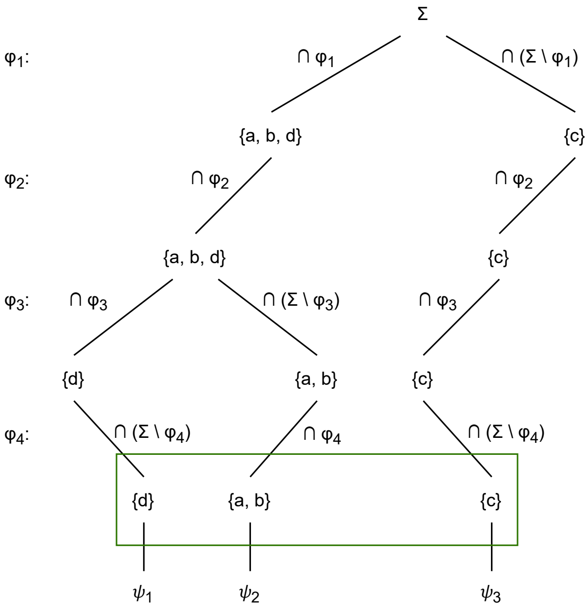
- Return values: satisfiable, unsatisfiable

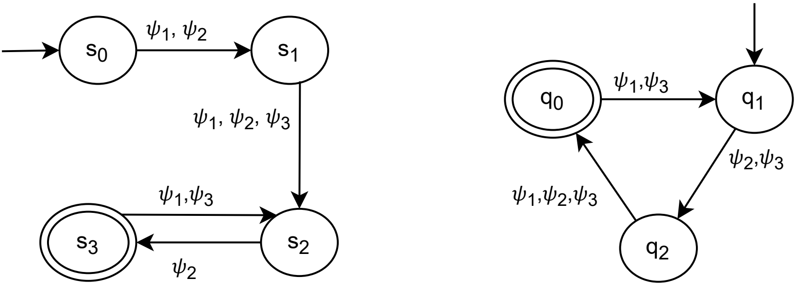
Mintermization

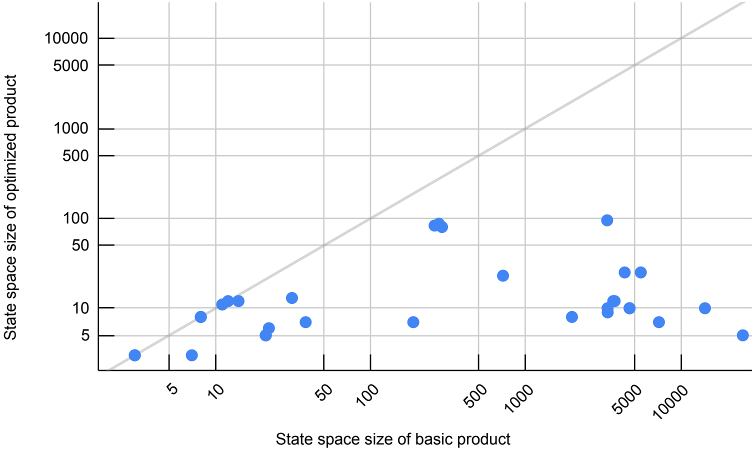



φ1
a, b, d
s0
s1
φ2
a, b, c, d
φ3
c, d
s3
s2
a, b
φ4

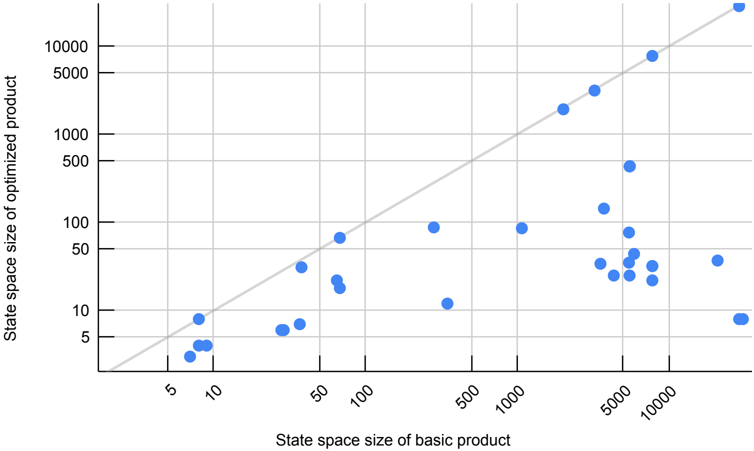

φ3
c, d
q0
q1
a, b, d
φ1
φ2
a, b, c, d
q2

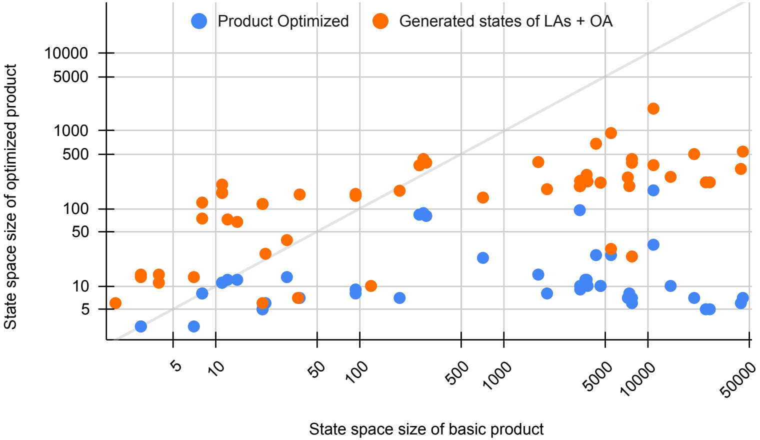

Σ
φ1:
∩ φ1
∩ (Σ \ φ1)
{a, b, d}
{c}
φ2:
∩ φ2
∩ φ2
{a, b, d}
{c}
φ3:
∩ φ3
∩ (Σ \ φ3)
∩ φ3
{d}
{a, b}
{c}
φ4:
∩ (Σ \ φ4)
∩ φ4
∩ (Σ \ φ4)
{d}
{a, b}
{c}
ψ1
ψ2
ψ3


s0
s1
s2
s3
ψ1, ψ2
ψ1, ψ2, ψ3
ψ1,ψ3
ψ2
q0
q1
q2
ψ1,ψ3
ψ2,ψ3
ψ1,ψ2,ψ3



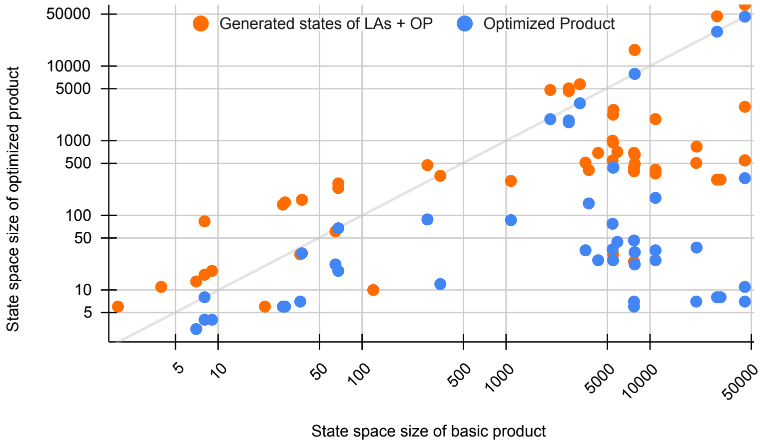
10000
5000
1000
500
100
50
10
5
State space size of optimized product
5
10
50
100
500
1000
5000
10000
State space size of basic product

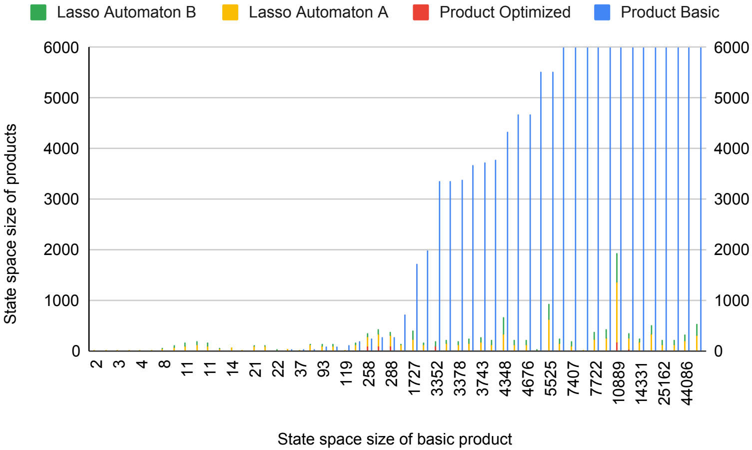

10000
5000
1000
500
100
50
10
5
State space size of optimized product
5
10
50
100
500
1000
5000
10000
State space size of basic product



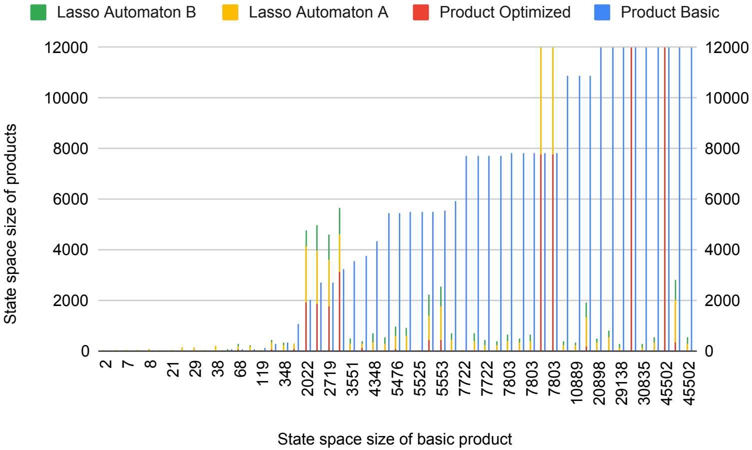
Product Optimized
Generated states of LAs + OA
State space size of optimized product
10000
5000
1000
500
100
50
10
5
5
10
50
100
500
1000
5000
10000
50000
State space size of basic product

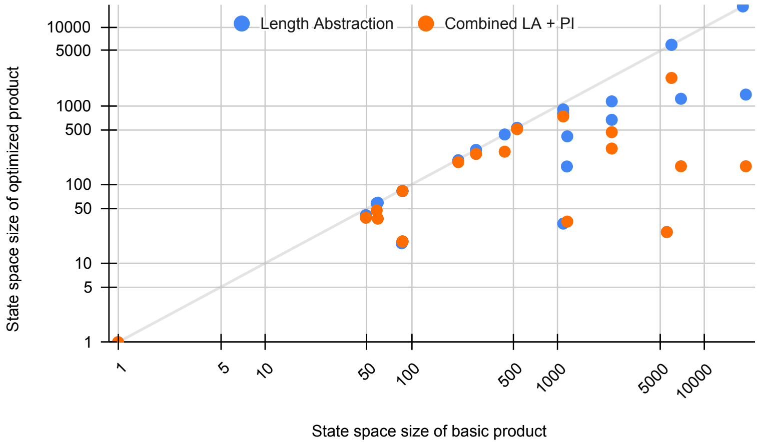
Optimized Product with Lasso Automata



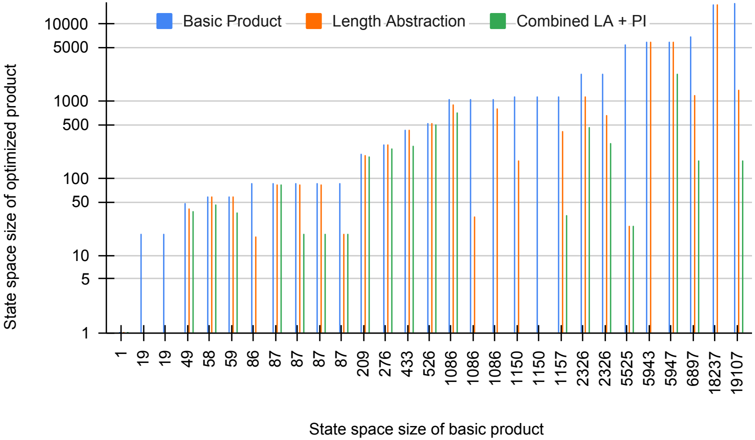


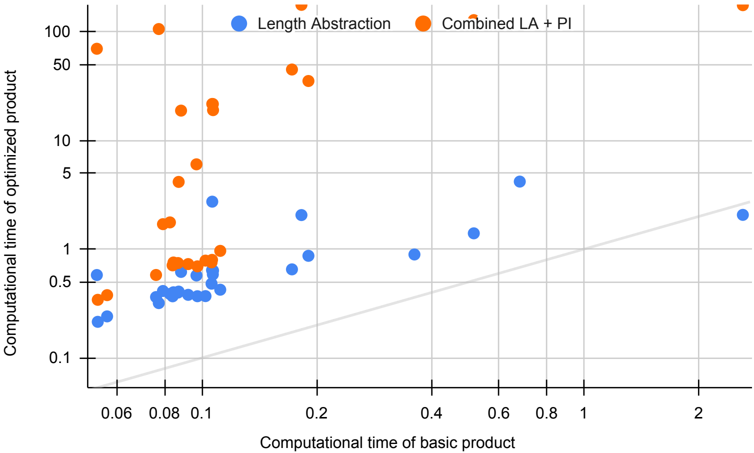
Lasso Automaton B
Lasso Automaton A
Product Optimized
Product Basic
State space size of products
6000
5000
4000
3000
2000
1000
0
2
3
4
8
11
11
14
21
22
37
93
119
258
288
1727
3352
3378
3743
4348
4676
5525
7407
7722
10889
14331
25162
44086
State space size of basic product


Lasso Automaton B
Lasso Automaton A
Product Optimized
Product Basic
State space size of products
12000
10000
8000
6000
4000
2000
0
2
7
8
21
29
38
68
119
348
2022
2719
3551
4348
5476
5525
5553
7722
7722
7803
7803
7803
10889
20898
29138
30835
45502
45502
State space size of basic product

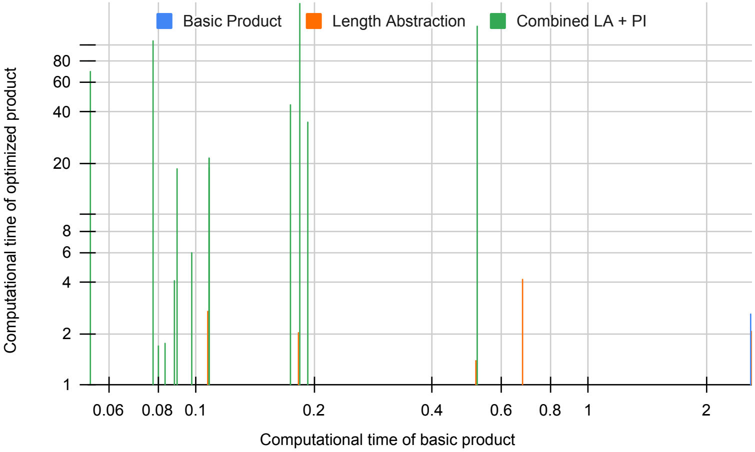

Length Abstraction
Combined LA + PI
10000
5000
1000
500
100
50
10
5
1
1
5
10
50
100
500
1000
5000
10000
State space size of optimized product
State space size of basic product

Abstractions Comparison




Length Abstraction
Combined LA + PI
Computational time of optimized product
Computational time of basic product
100
50
10
5
1
0.5
0.1
0.06
0.08
0.1
0.2
0.4
0.6
0.8
1
2

Abstractions Comparison



Classic Product Construction Algorithm

Input : NFA $A_1 = (Q_1, \Sigma, \delta_1, I_1, F_1)$,
NFA $A_2 = (Q_2, \Sigma, \delta_2, I_2, F_2)$

Output: NFA $(A_1 \cap A_2) = (Q, \Sigma, \delta, I, F)$
with $L(A_1 \cap A_2) = L(A_1) \cap L(A_2)$

$Q, \delta, F \leftarrow \emptyset$
$I \leftarrow I_1 \times I_2$
$W \leftarrow I$
while $W \neq \emptyset$ **do**
  pick $[q_1, q_2]$ **from** W
  add $[q_1, q_2]$ **to** Q
  if $q_1 \in F_1$ *and* $q_2 \in F_2$ **then**
    add $[q_1, q_2]$ **to** F
  forall $a \in \Sigma$ **do**
    forall $q_1' \in \delta_1(q_1, a), q_2' \in \delta_2(q_2, a)$ **do**
      if $[q_1', q_2'] \notin Q$ **then**
        add $[q_1', q_2']$ **to** W
      add $[q_1', q_2']$ **to** $\delta([q_1, q_2], a)$

Optimized Product Construction Algorithm

Input : NFA $A_1 = (Q_1, \Sigma, \delta_1, I_1, F_1)$,
NFA $A_2 = (Q_2, \Sigma, \delta_2, I_2, F_2)$

Output: NFA $(A_1 \cap A_2) = (Q, \Sigma, \delta, I, F)$
with $L(A_1 \cap A_2) = L(A_1) \cap L(A_2)$

$Q, \delta, F \leftarrow \emptyset$
$I \leftarrow I_1 \times I_2$
$W \leftarrow I$
$sat \leftarrow False$
$solved \leftarrow \emptyset$
while $W \neq \emptyset$ **do**
    picklast $[q_1, q_2]$ **from** W
    add $[q_1, q_2]$ **to** $solved$
    $sat \leftarrow$ **satisfiable**$([q_1, q_2])$
    if sat **then**
        add $[q_1, q_2]$ **to** Q
        if $q_1 \in F_1$ ***and*** $q_2 \in F_2$ **then**
            add $[q_1, q_2]$ **to** F
        forall $a \in \Sigma$ **do**
            forall $q_1' \in \delta_1(q_1, a), q_2' \in \delta_2(q_2, a)$ **do**
                if $[q_1', q_2'] \notin solved$ ***and*** $[q_1', q_2'] \notin W$ **then**
                    add $[q_1', q_2']$ **to** W
                add $[q_1', q_2']$ **to** $\delta([q_1, q_2], a)$

Input : NFA $A_1 = (Q_1, \Sigma, \delta_1, I_1, F_1)$,
NFA $A_2 = (Q_2, \Sigma, \delta_2, I_2, F_2)$
Output: NFA $(A_1 \cap A_2) = (Q, \Sigma, \delta, I, F)$
with $L(A_1 \cap A_2) = L(A_1) \cap L(A_2)$

$Q, \delta, F \leftarrow \emptyset$
$I \leftarrow I_1 \times I_2$
$W \leftarrow I$
$sat \leftarrow False$
$solved \leftarrow \emptyset$
while $W \neq \emptyset$ **do**
  picklast $[q_1, q_2]$ **from** W
  add $[q_1, q_2]$ **to** $solved$
  if ***not skippable***$([q_1, q_2])$ **then**
    $sat \leftarrow$ **satisfiable**$([q_1, q_2])$
  else
    $sat \leftarrow True$
  if sat **then**
    add $[q_1, q_2]$ **to** Q
    if $q_1 \in F_1$ ***and*** $q_2 \in F_2$ **then**
      add $[q_1, q_2]$ **to** F
    forall $a \in \Sigma$ **do**
      forall $q'_1 \in \delta_1(q_1, a), q'_2 \in \delta_2(q_2, a)$ **do**
        if $[q'_1, q'_2] \notin solved$ ***and*** $[q'_1, q'_2] \notin W$ **then**
          add $[q'_1, q'_2]$ **to** W
        add $[q'_1, q'_2]$ **to** $\delta([q_1, q_2], a)$