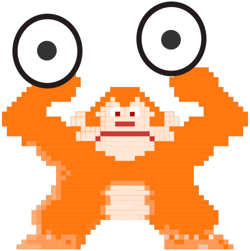
Kong: a Tool to Squash Concurrent Places *(and more...)*

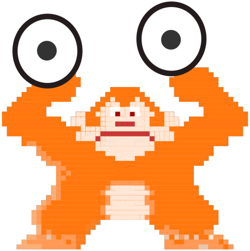
Nicolas Amat, **Louis Chauvet**

LAAS-CNRS

Petri Nets, June 22 2022



- A tool for **reachability problems** using **polyhedral reductions**



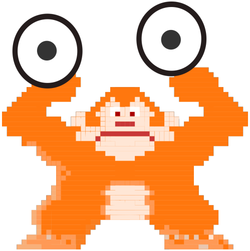
- A tool for **reachability problems** using **polyhedral reductions**

  Concurrent places problem: enumerate all pairs of places that can be marked together in some reachable marking

  Marking reachability: is a given marking reachable?

What is Kong?

Introduction



- A tool for **reachability problems** using **polyhedral reductions**

  Concurrent places problem: enumerate all pairs of places that can be marked together in some reachable marking

  Marking reachability: is a given marking reachable?

- **Freely available** under the GPLv3 license
  `github.com/nicolasAmat/Kong`

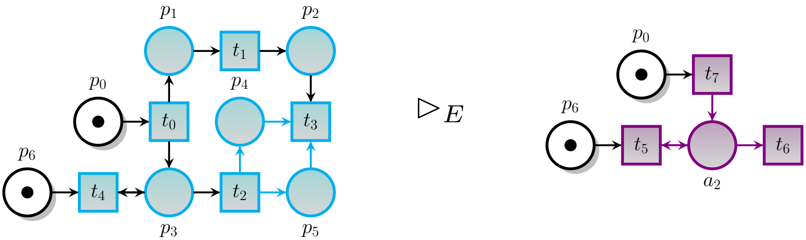
Outline

$$\underbrace{(N_1, m_1)}_{\text{initial net}} \quad \underbrace{\triangleright_E}_{\text{linear system}} \quad \underbrace{(N_2, m_2)}_{\text{reduced net}}$$

Correspondence between the set of reachable markings "modulo" the linear equations E

$$\underbrace{(N_1, m_1)}_{\text{initial net}} \quad \underbrace{\rhd_E}_{\text{linear system}} \quad \underbrace{(N_2, m_2)}_{\text{reduced net}}$$

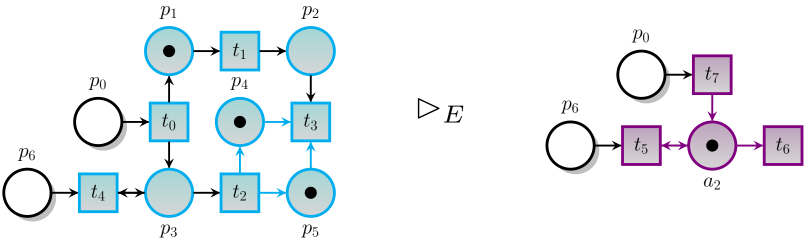
Correspondence between the set of reachable markings "modulo" the linear equations E



$$E = (p_5 = p_4) \wedge (a_1 = p_2 + p_1) \wedge (a_2 = p_4 + p_3) \wedge (a_1 = a_2)$$

Theorem (Reachability preservation)

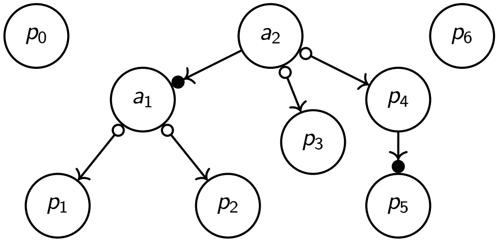
Assume m'_1, m'_2, E is satisfiable then m'_2 is reachable in (N_2, m_2) if and only if m'_1 is reachable in (N_1, m_1).



$$E = (p_5 = p_4) \wedge (a_1 = p_2 + p_1) \wedge (a_2 = p_4 + p_3) \wedge (a_1 = a_2)$$

A **Token Flow Graph** is a DAG that captures the specific structure of reduction equations

$$E = (p_5 = p_4) \wedge (a_1 = p_2 + p_1) \wedge (a_2 = p_4 + p_3) \wedge (a_1 = a_2)$$



Outline

- Basically a front-end to `accelerate` the computation of reachability problems
- Divided into three **subcommands**: `reach`, `conc` and `dead`
- **Input formats**: `.pnml`, `.net` and `.nupn`

```
$> ./kong.py reach model.pnml -m marking
```

Textual description of the marking: "`p1 p4 p5`"
(assume non-specified places do not contain tokens)

reach subcommand – Basic usage

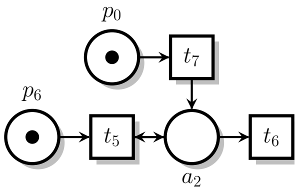
```
$> ./kong.py reach model.pnml -m marking

REACHABLE
```

Textual description of the marking: "`p1 p4 p5`"
(assume non-specified places do not contain tokens)

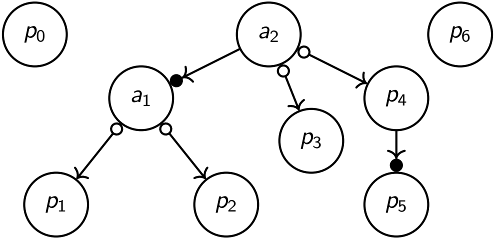
--show-equations

```
# System of equations
# R |- p5 = p4
# A |- a1 = p2 + p1
# A |- a2 = p4 + p3
# R |- a1 = a2
```

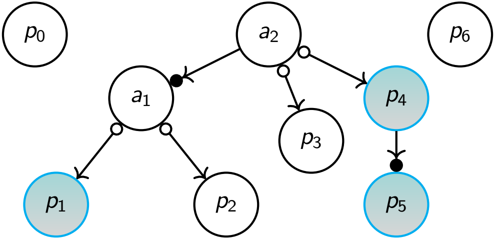
--save-reduced-net

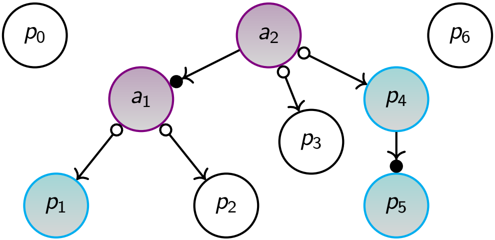


`--draw-graph`

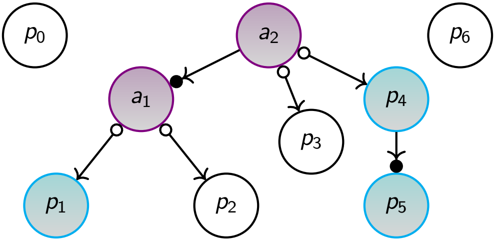


$$p_0 = 0 \wedge p_1 = 1 \wedge p_2 = 0 \wedge p_3 = 0 \wedge p_4 = 1 \wedge p_5 = 1 \wedge p_6 = 0$$





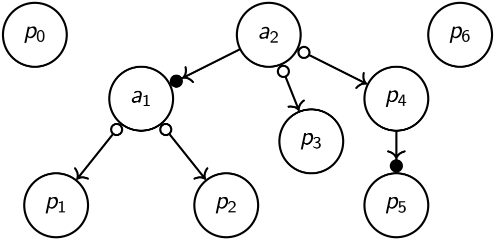
p_0
a_2
p_6
a_1
p_4
p_3
p_1
p_2
p_5



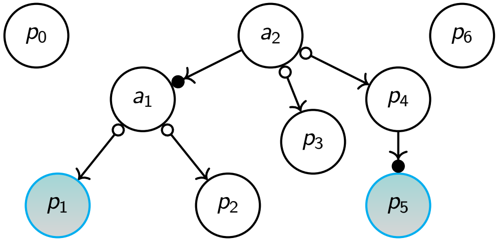
`--projected-marking`:

$$p_0 = 0 \wedge a_2 = 1 \wedge p_6 = 0$$

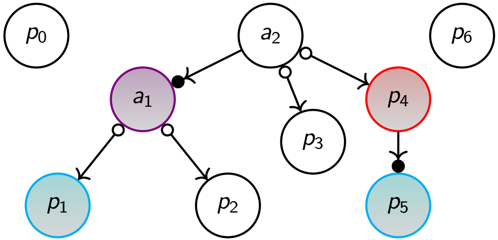
$p_0 = 0 \wedge p_1 = 1 \wedge p_2 = 0 \wedge p_3 = 0 \wedge p_4 = 0 \wedge p_5 = 1 \wedge p_6 = 0$



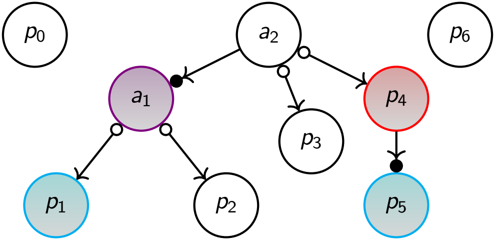
$$p_0 = 0 \land p_1 = 1 \land p_2 = 0 \land p_3 = 0 \land p_4 = 0 \land p_5 = 1 \land p_6 = 0$$



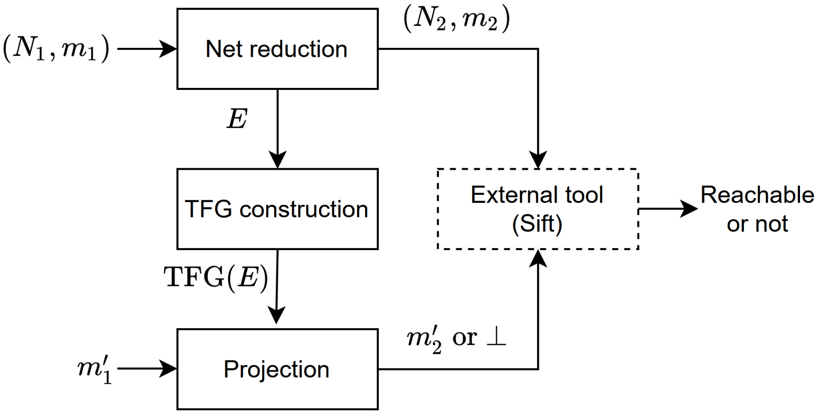
$$p_0 = 0 \land p_1 = 1 \land p_2 = 0 \land p_3 = 0 \land p_4 = 0 \land p_5 = 1 \land p_6 = 0$$



$$p_0 = 0 \land p_1 = 1 \land p_2 = 0 \land p_3 = 0 \land p_4 = 0 \land p_5 = 1 \land p_6 = 0$$

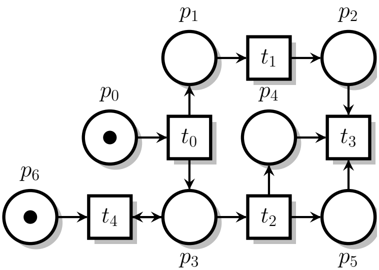


No projection! And so, the marking is trivially unreachable.


(N_1, m_1)
Net reduction
(N_2, m_2)
E
TFG construction
External tool
(Sift)
Reachable
or not
$\mathrm{TFG}(E)$
m'_2 or $\bot$
m'_1
Projection

```
$> ./kong.py conc model.pnml --place-names
```

model.pnml

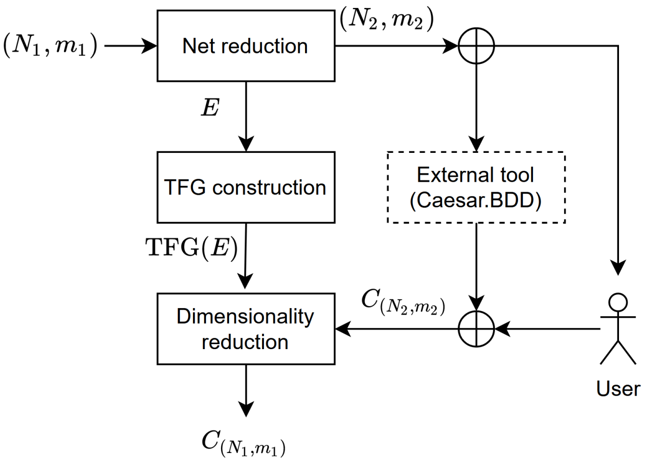


Output

```
p0  1
p1  01
p2  001
p3  0111
p4  01101
p5  011011
p6  1(7)
```

`--show-reduced-matrix`

```
# Reduced concurrency matrix
# a2  1
# p0  01
```



(N_1, m_1)
Net reduction
(N_2, m_2)
E
TFG construction
External tool
(Caesar.BDD)
$\mathrm{TFG}(E)$
$C_{(N_2,m_2)}$
Dimensionality
reduction
User
$C_{(N_1,m_1)}$

Outline

Models from the Model Checking Contest (MCC)

Concurrent places

- 424 instances with reduction opportunities (out of 562 safe)

Marking reachability

- Selected of 426 instances (out of 1 411)
- Generated 5 reachable markings as queries using a "random walk"

Models from the Model Checking Contest (MCC)

Concurrent places

- 424 instances with reduction opportunities (out of 562 safe)

Marking reachability

- Selected of 426 instances (out of 1 411)
- Generated 5 reachable markings as queries using a "random walk"

We compare: CAESAR.BDD and SIFT alone, on the initial net, and KONG + REDUCE + CAESAR.BDD or SIFT

Models from the Model Checking Contest (MCC)

Concurrent places

- 424 instances with reduction opportunities (out of 562 safe)

Marking reachability

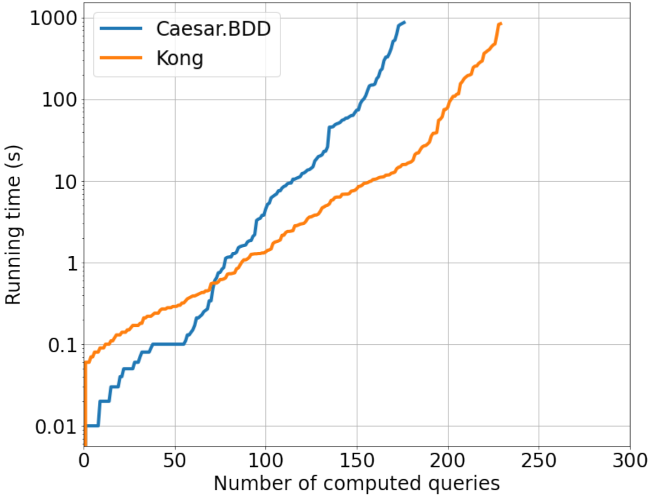
- Selected of 426 instances (out of 1 411)
- Generated 5 reachable markings as queries using a "random walk"

We compare: CAESAR.BDD and SIFT alone, on the initial net, and KONG + REDUCE + CAESAR.BDD or SIFT

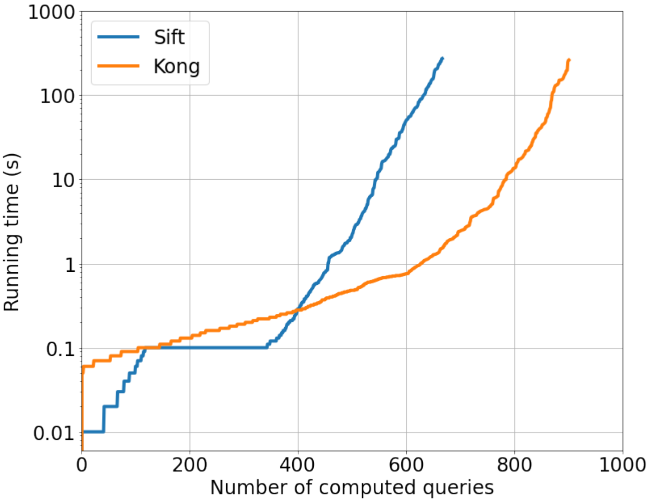
All benchmark scripts are available online!

Concurrent places

Performance



Minimal timeout to compute a given number of concurrency matrices



Minimal timeout to compute a given number of queries

Outline

Reduce & Shrink – Our reduction tools

Reduction tools

Reduce

https://projects.laas.fr/tina

- Available in the Tina Toolbox
  Since version 3.7 (January 20, 2022)
- Used in Tina and SMPT in the MCC

Reduce & Shrink – Our reduction tools

Reduction tools

Reduce

`https://projects.laas.fr/tina`

- Available in the Tina Toolbox
  Since version 3.7 (January 20, 2022)
- Used in Tina and SMPT in the MCC

Shrink

`https://github.com/Fomys/pnets`

- Freely available under MIT license
- Based on the pnets library

Outline

Perspectives

- Generalized Mutual Exclusion Constraints

  $\sum_{p\in P} w_p.m(p) \leqslant k$, with $w_1,\ldots,w_n, k$ constants in $\mathbb{Z}$

Perspectives

- Generalized Mutual Exclusion Constraints

  $\sum_{p \in P} w_p.m(p) \leqslant k$, with $w_1, \ldots, w_n, k$ constants in $\mathbb{Z}$

- Explore new reduction rules

Perspectives

- Generalized Mutual Exclusion Constraints

  $\sum_{p \in P} w_p.m(p) \leqslant k$, with $w_1, \ldots, w_n, k$ constants in $\mathbb{Z}$

- Explore new reduction rules

Still a lot of work to be done to compute polyhedral reductions, and to apply them on useful and complex problems!