



HAL
open science

Polyhedral Reductions for Petri nets

Nicolas Amat, Bernard Berthomieu, Silvano Dal Zilio, Didier Le Botlan

► **To cite this version:**

Nicolas Amat, Bernard Berthomieu, Silvano Dal Zilio, Didier Le Botlan. Polyhedral Reductions for Petri nets. Modélisation des Systèmes Réactifs (MSR'23), Nov 2023, Toulouse, France. , 2023. hal-04355257

HAL Id: hal-04355257

<https://hal.science/hal-04355257v1>

Submitted on 20 Dec 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution 4.0 International License

Polyhedral equivalence

$$(N_1, m_1) \equiv_E (N_2, m_2)$$

Polyhedral equivalence defines a *linear dependence relation* (E) between the reachable markings of two nets; where E is a Presburger formula, or a system of linear Diophantine equations. This equivalence, denoted $(N_1, m_1) \equiv_E (N_2, m_2)$, preserves enough information in E so that we can rebuild the state space of N_1 knowing only the one of N_2 .

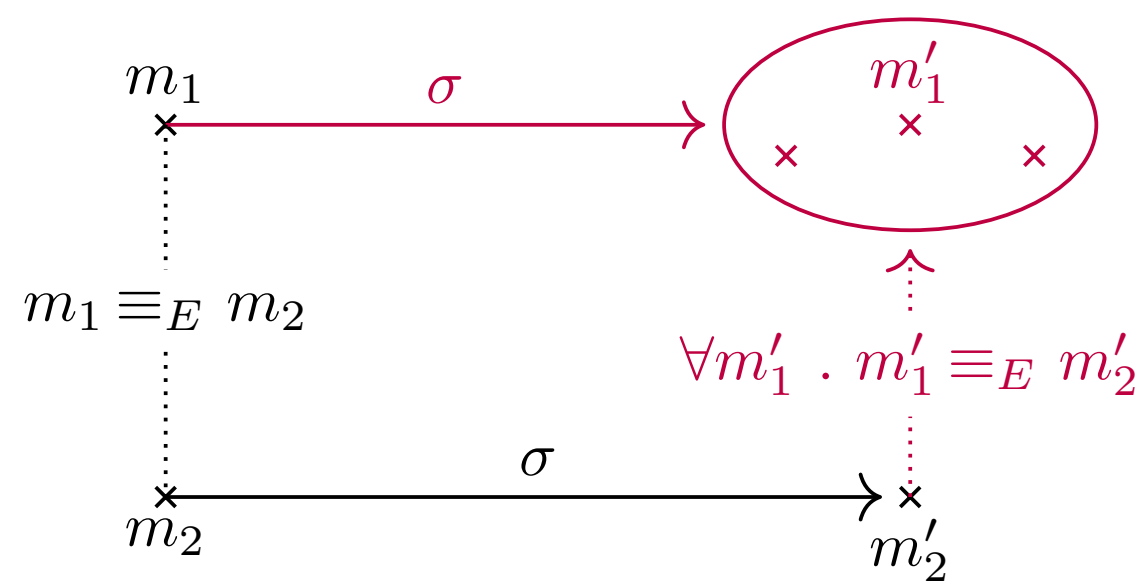
We say that two markings m_1, m_2 are compatible up-to E , denoted $m_1 \equiv_E m_2$, when formula $E \wedge m_1 \wedge m_2$ is satisfiable, with m the conjunction $\bigwedge_{p \in P} p = m(p)$ (e.g., $(p_0 = 5) \wedge (p_1 = 0) \wedge \dots$).

Nets (N_1, m_1) and (N_2, m_2) are E -equivalent when:

- (A1) $E \wedge m$ is satisfiable for all markings m in $R(N_1, m_1)$ or $R(N_2, m_2)$;
- (A2) initial markings are *compatible*: $m_1 \equiv_E m_2$;
- (A3) assume m'_1, m'_2 are markings of N_1, N_2 , such that $m'_1 \equiv_E m'_2$, then m'_1 is reachable iff m'_2 is reachable.

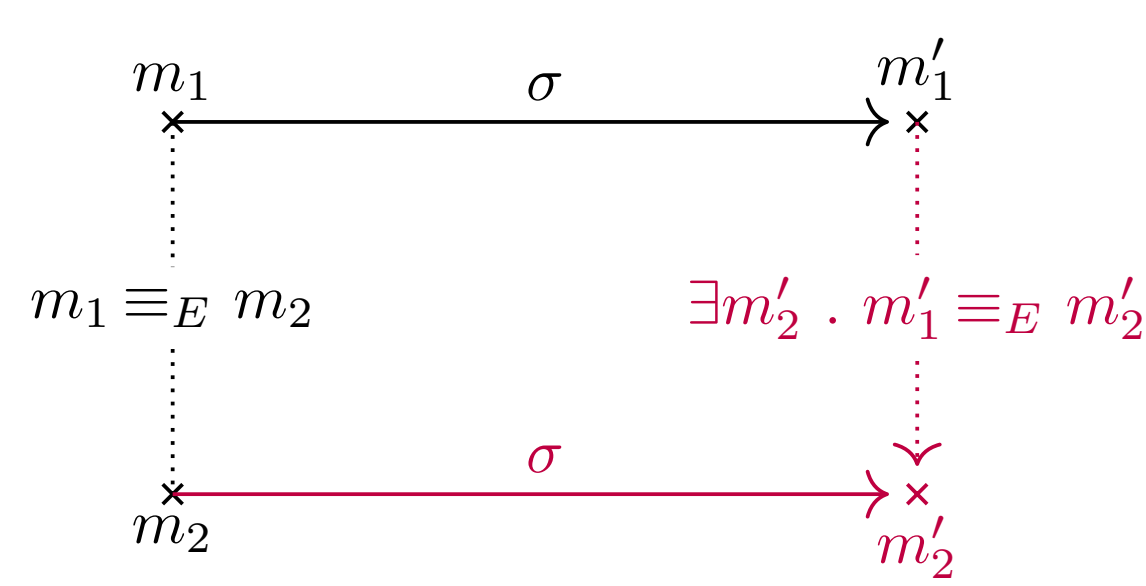
Lemma (Reachability Checking).

Given m'_2 reachable in N_2 then $m'_1 \equiv_E m'_2$ implies $m'_1 \in R(N_1, m_1)$.



Lemma (Invariance Checking).

For all m'_1 in $R(N_1, m_1)$ there is m'_2 in $R(N_2, m_2)$ such that $m'_1 \equiv_E m'_2$.



Counting using SDD

Tedd

```
$ sift AirplaneLD-PT-0050.pnml
4471223 marking(s), 19756224 transitions(s)
67.569s

$ tedd -q AirplaneLD-PT-0050.pnml
4471223 state(s)
2.691s

$ tedd -q AirplaneLD-PT-4000.pnml
2.1773922960e12 state(s)
33.574s

$ tedd -rg,redundant,compact AirplaneLD-PT-4000.pnml
2.1773922960e12 state(s)
0.000s
```

Reachability using SMT

SMP/T

To check if property F_1 is reachable in N_1 , it is enough to check its E -transform, F_2 , on N_2 .

$$F_2(y) \triangleq \exists x. \tilde{E}(x, y) \wedge F_1(x)$$

We use this property in a SMT-based model-checker with structural reductions. It supports generalized reachability formulas of the form: $AG((p_1 + p_3 \geq 5) \vee (p_0 < p_4))$

```
$ python3 -m smpt -n SmallOpSys-MT0016.pnml
--methods BMC K-INDUCTION PDR-REACH STATE-EQUATION
--show-time --show-techniques ...
FORMULA RC4 FALSE TECHNIQUES ... BMC TIME 0.04s

$ python3 -m smpt -n SmallOpSys-MT2048.pnml ...
timeout

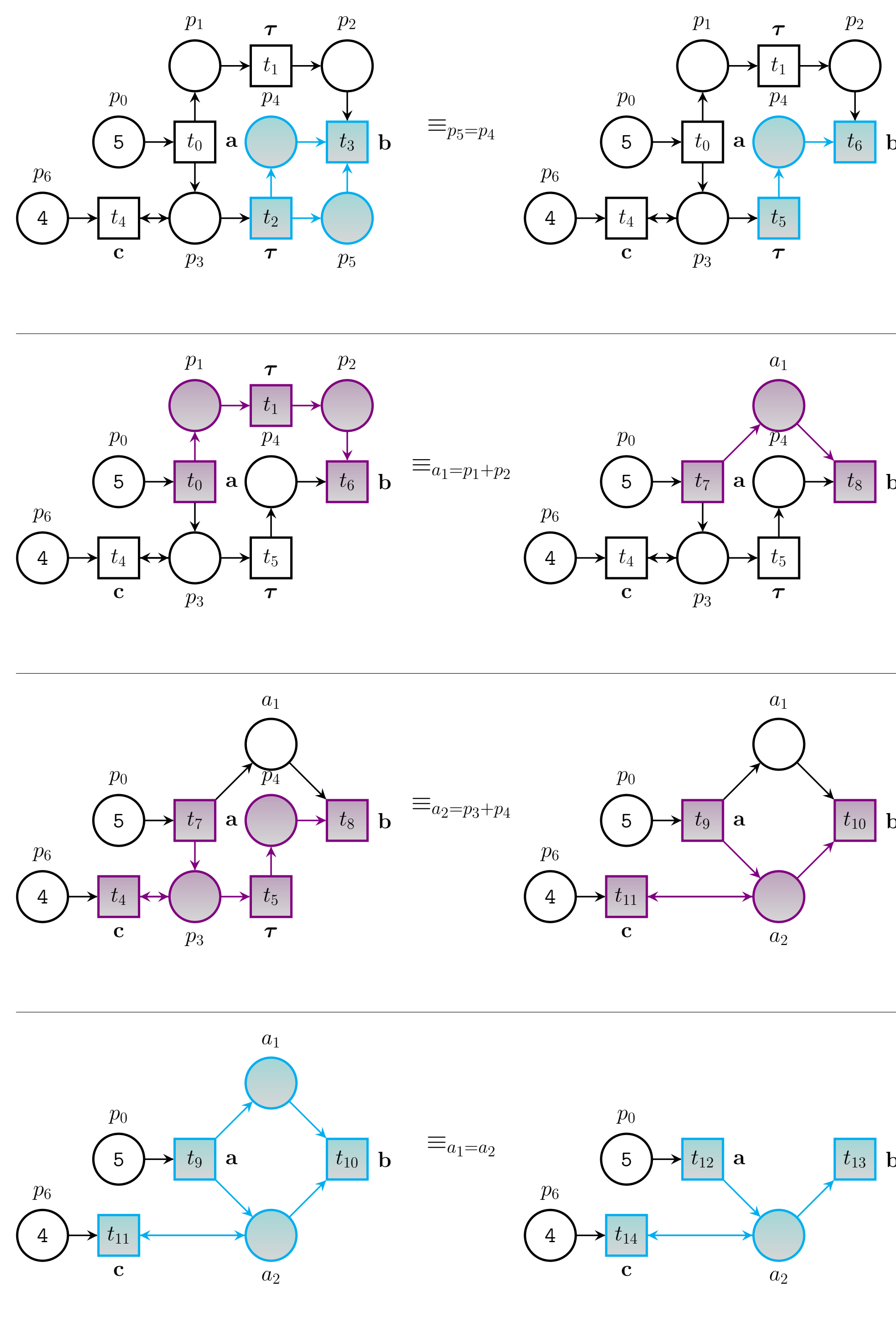
$ same with option --auto-reduce or --mcc
# Reduction time (Full): 0.11s
FORMULA RC4 FALSE TIME 0.05s

$ or with options --project --show-projection
# Reduction time (TFG): 0.09s
# Reduction ratio (TFG) 44%
# Projection of RC4: [] - (1 ≤ p0 + p2 + a2) ∧ ...
```

Polyhedral reductions

Reduce

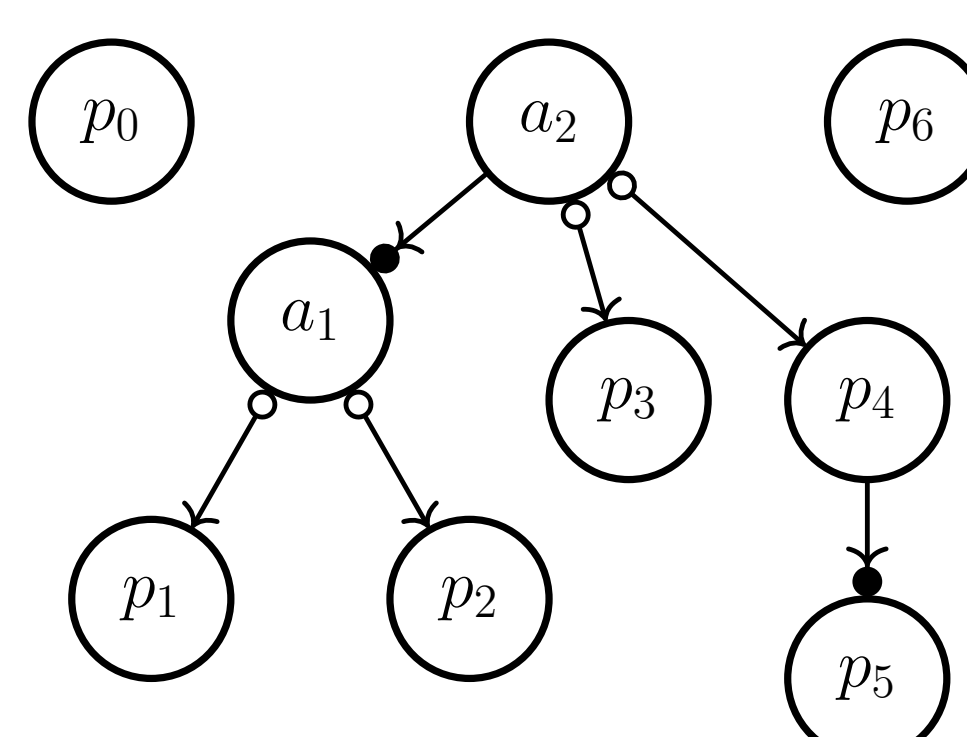
We can compute instances of polyhedral equivalences using structural reductions.



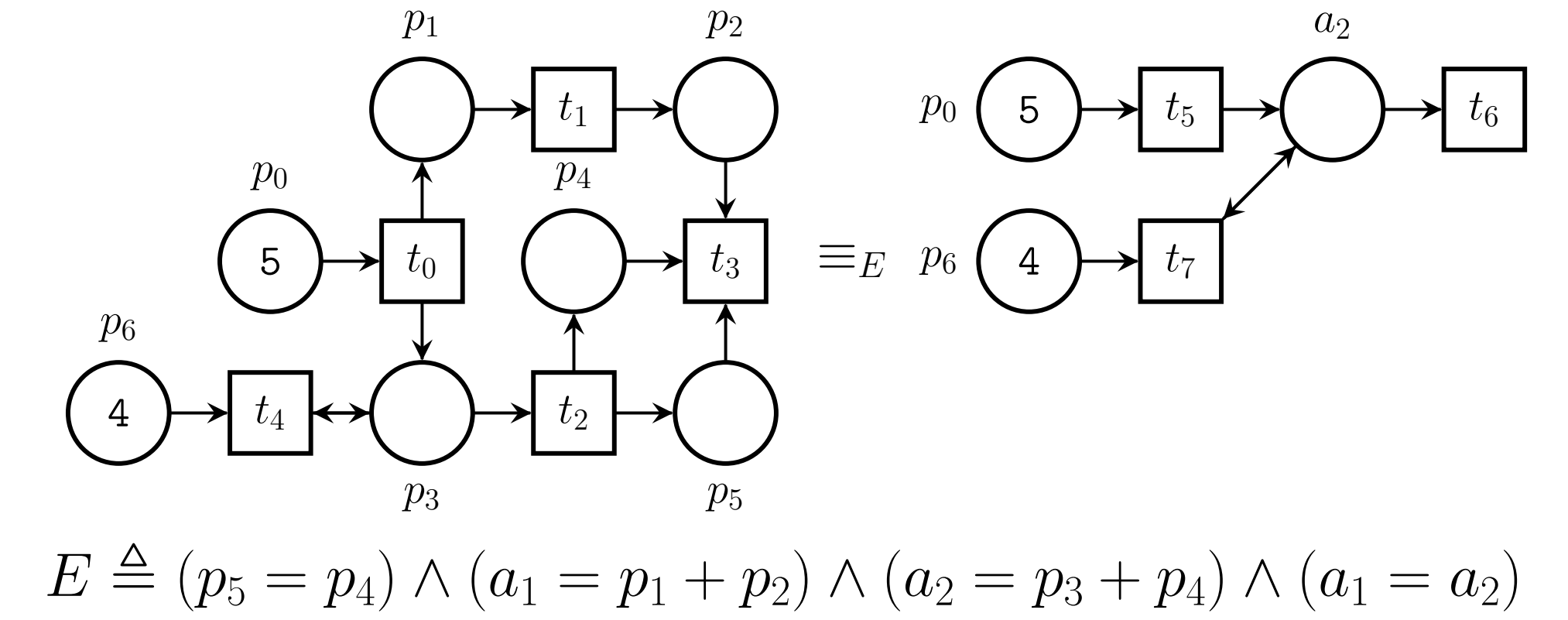
Token Flow Graph

Octant
Kong
Polycount

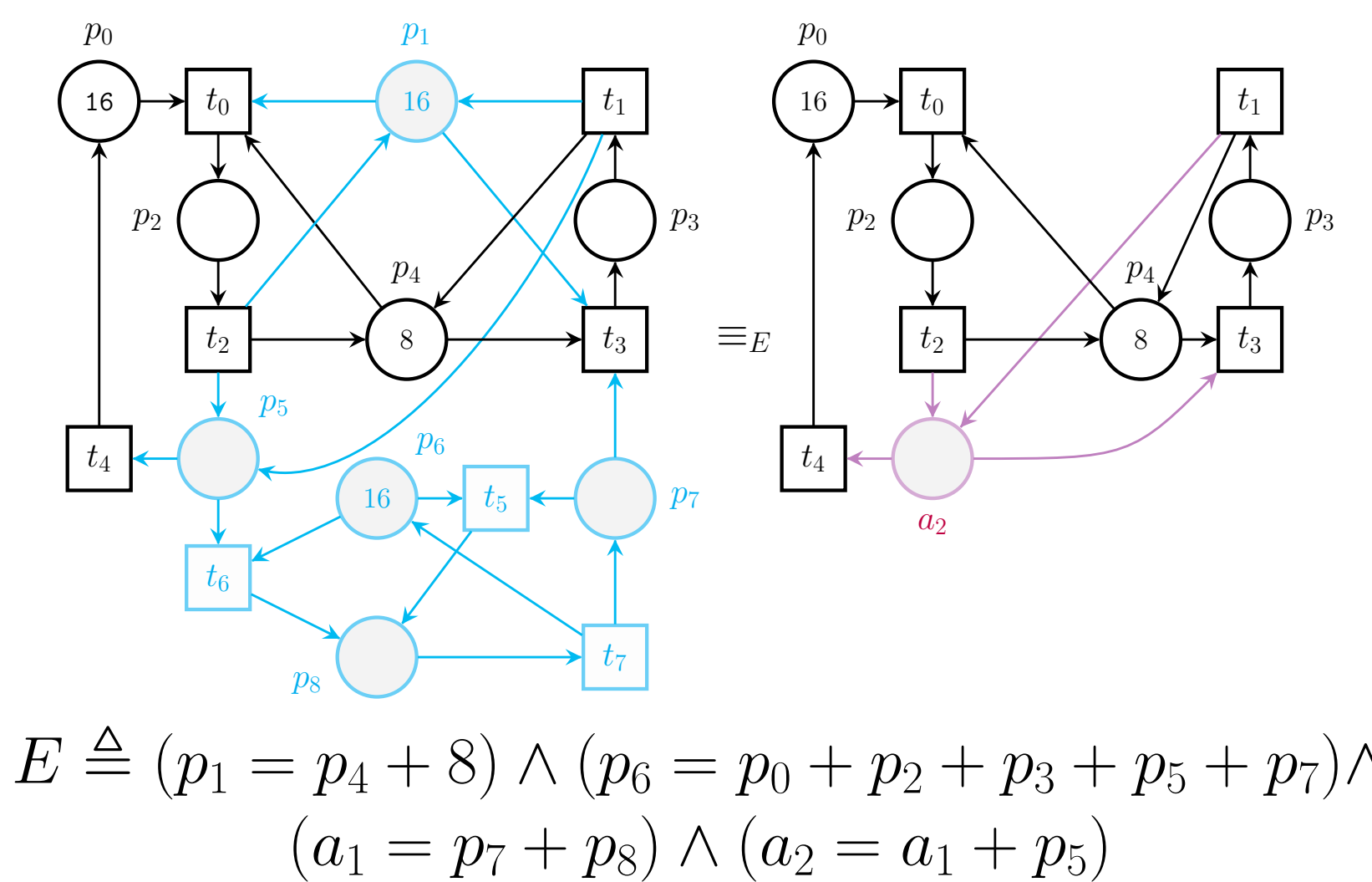
- # $R \vdash p_5 = p_4$
- # $A \vdash a_1 = p_2 + p_1$
- # $A \vdash a_2 = p_4 + p_3$
- # $R \vdash a_1 = a_2$



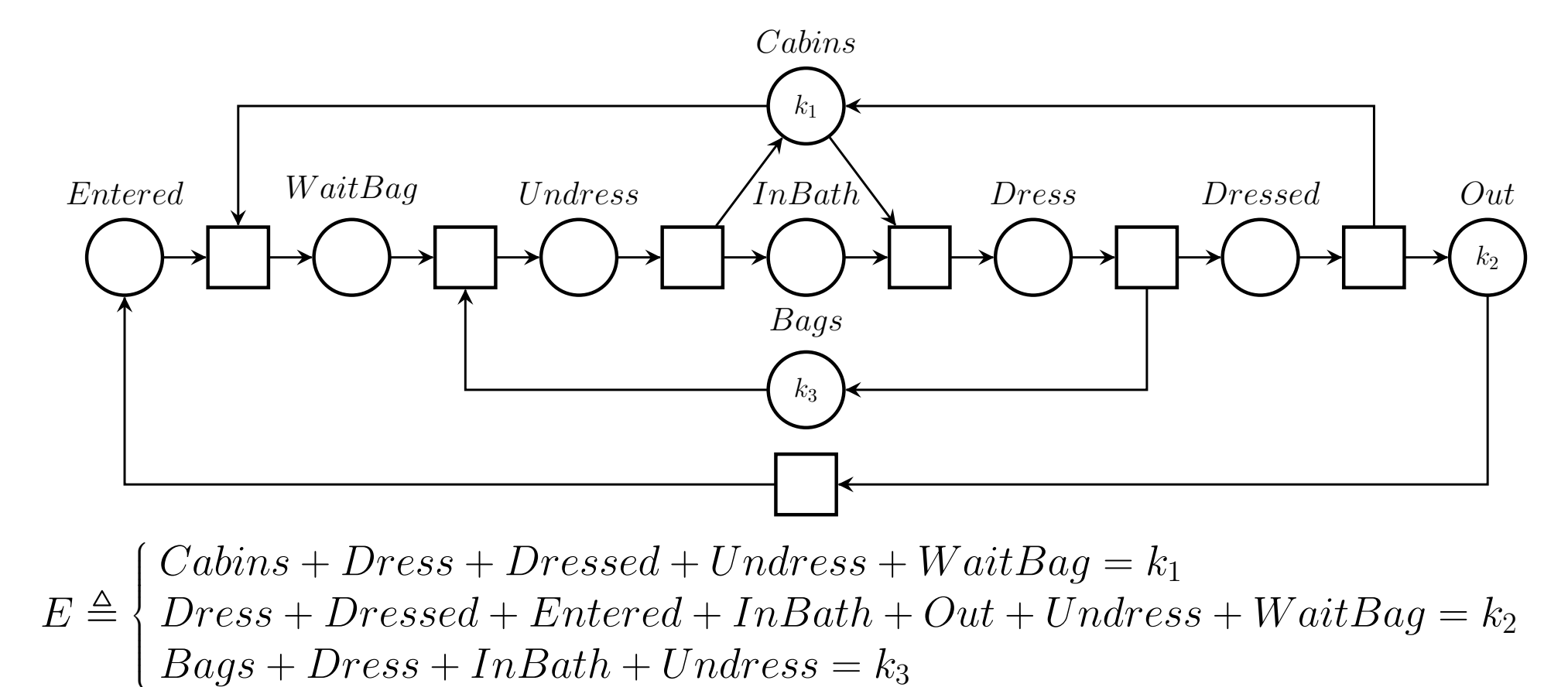
Some examples



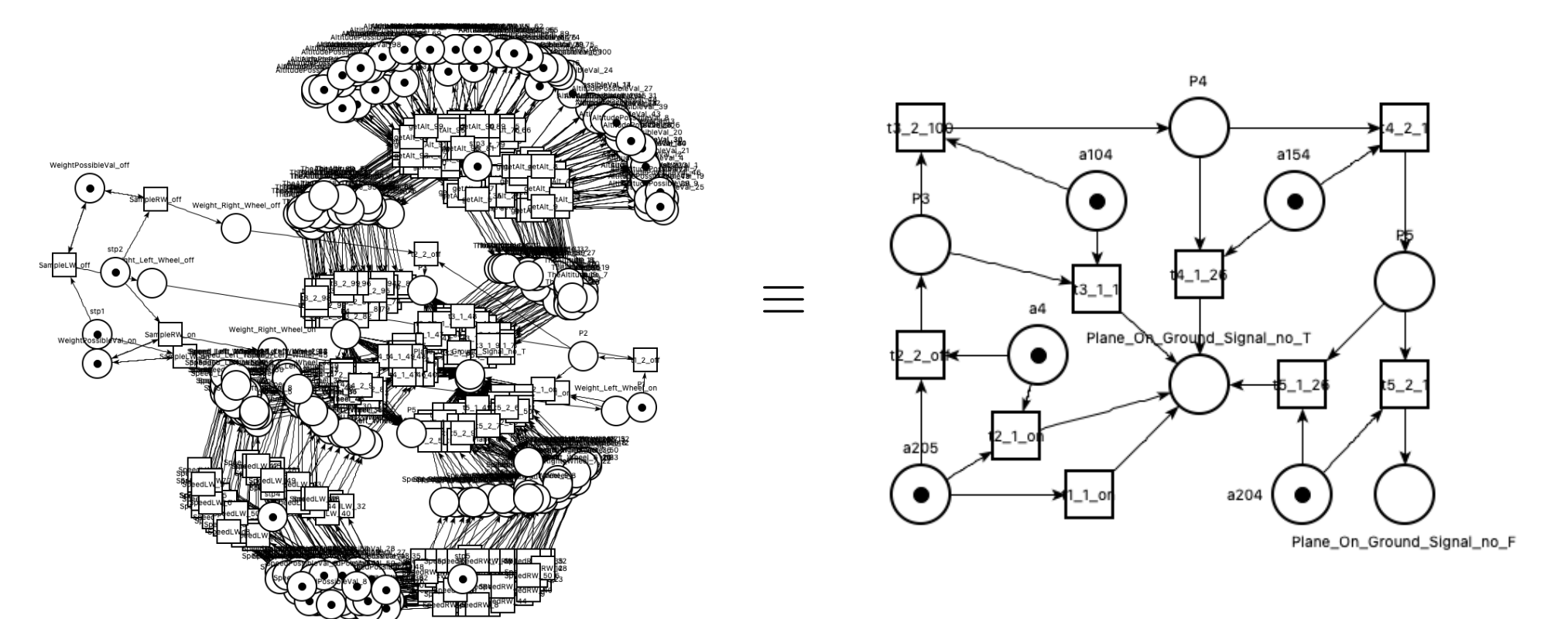
SmallOperatingSystem-PT-MT0016DC0008



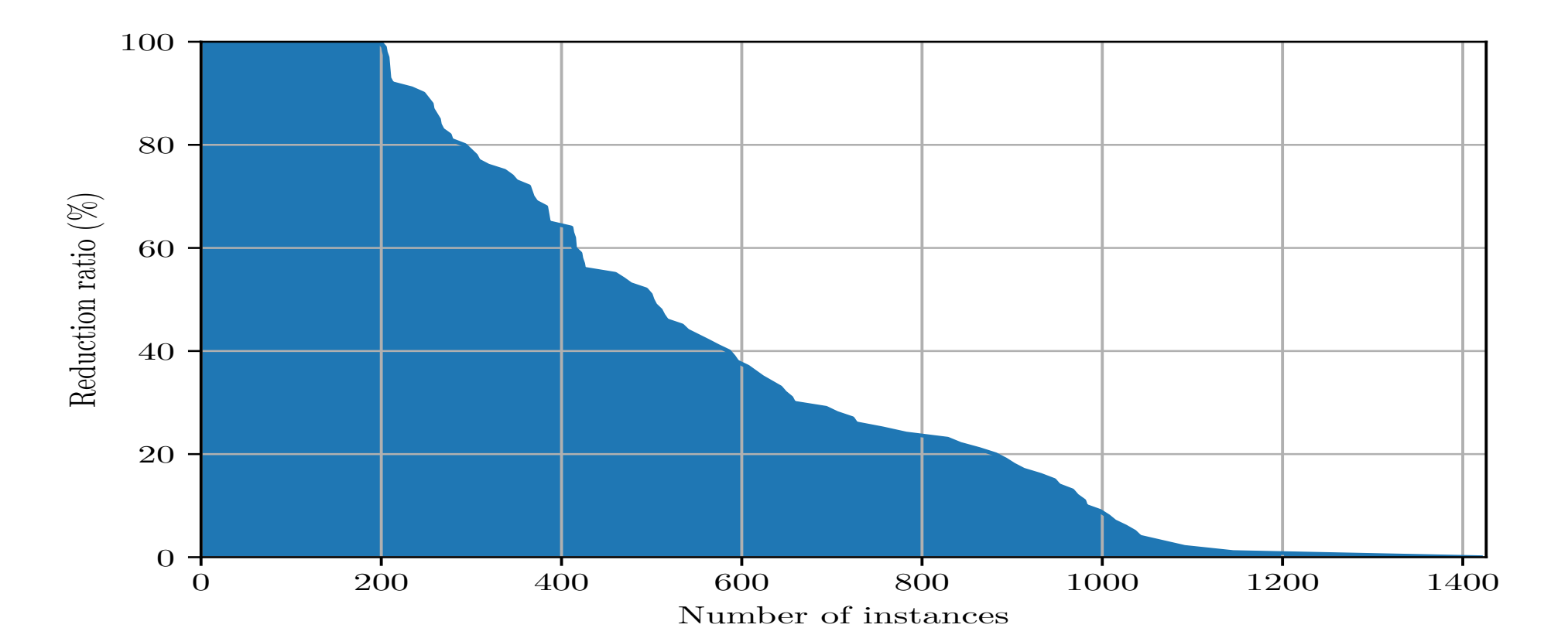
SwimmingPool (fully reduced)



AirplaneLD-PT-0050



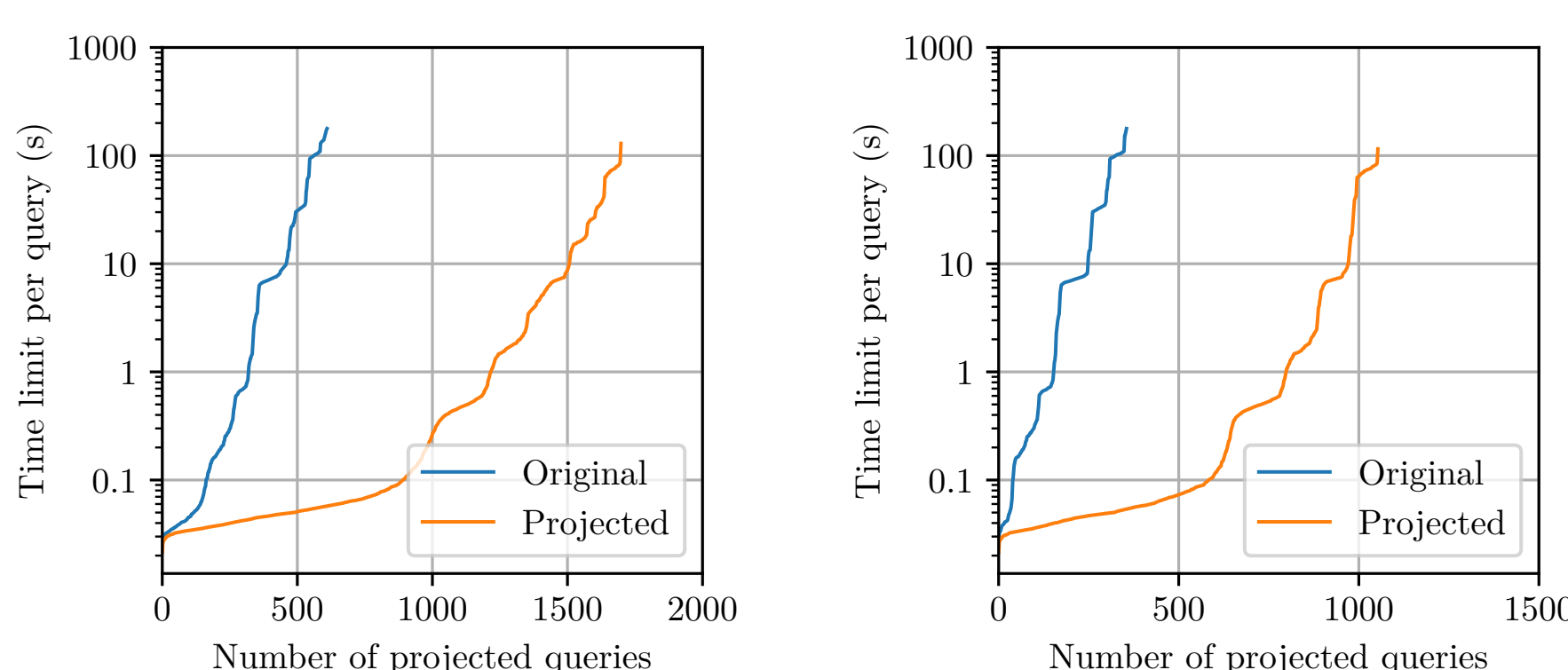
Prevalence of reductions in the MCC



Automatic equivalence checking

The problem of checking whether two nets are equivalent is undecidable. But we have implemented a sound, complete procedure in the case of equivalence between *parametric nets*: $(N_1, C_1) \approx_E (N_2, C_2)$.

Reductron



$r \in [0.01, 0.5[$ $r \in [0.5, 1]$

k -induction with and without reductions

