An SMT-Based Approach to Coverability Analysis

Javier Esparza[1], Ruslán Ledesma-Garza[1], Rupak Majumdar[2], Philipp Meyer[1], **Filip Niksic**[2]

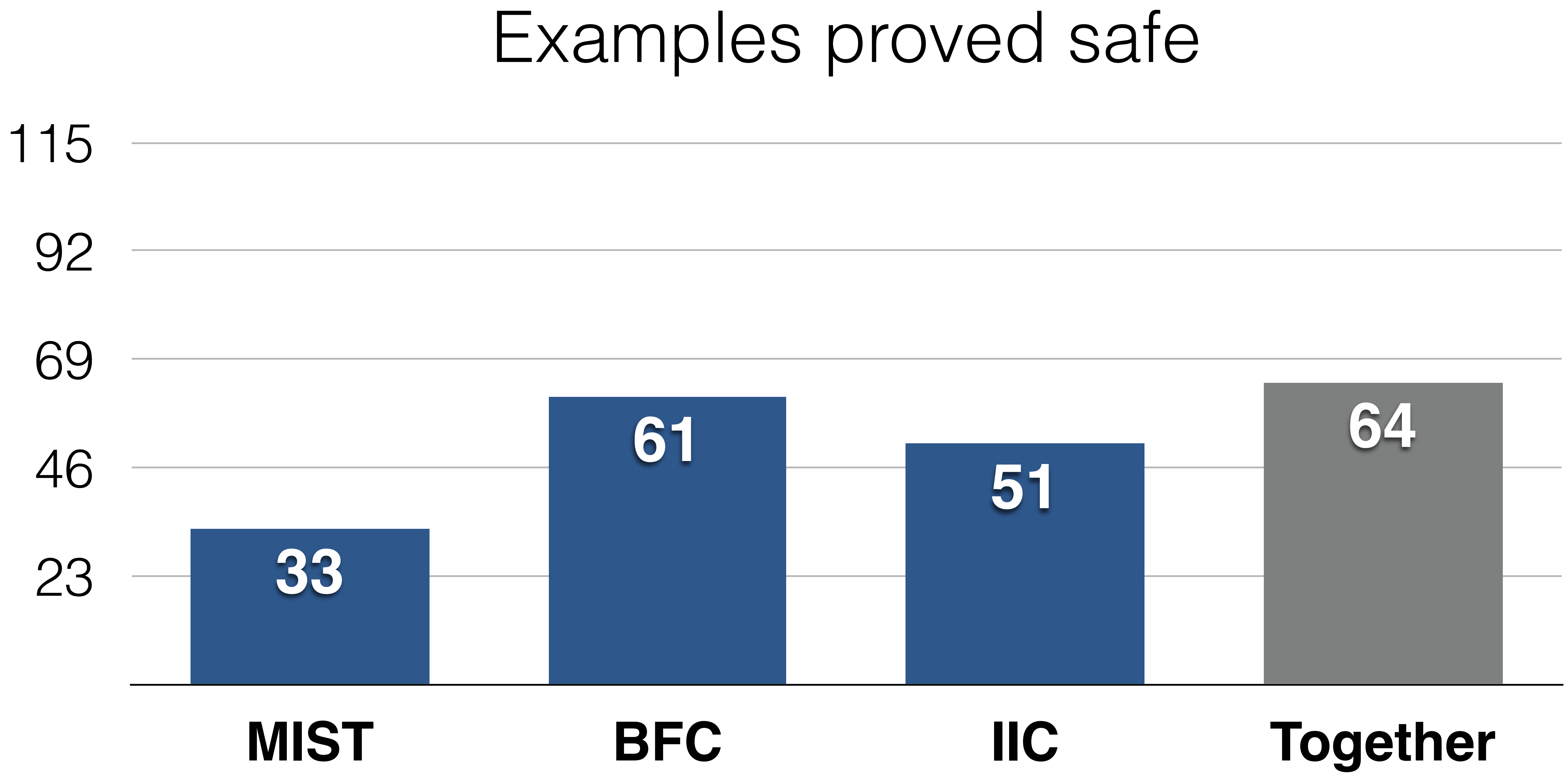
[1] Technische Universität München
[2] MPI-SWS

Petri net coverability is important, but difficult

- Many verification problems reduce to Petri net coverability problem
- Petri net coverability is **EXPSPACE-complete**
- Sophisticated tools and algorithms:

  MIST — Expand-enlarge-check [GRB '06]
  BFC — Minimal uncoverability proof [KKW '12]
  IIC — Incremental, inductive coverability [KMNP '13]

MIST, BFC and IIC don't scale well

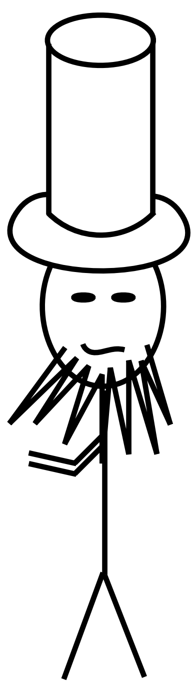


Reducing coverability to feasibility of linear constraints

Method **LinCon**:

- Based on **marking equation** [Murata '77]
  Incomplete

- Strengthened with **traps** [EM '00]
  Traps — essentially **Boolean constraints**
  Still incomplete

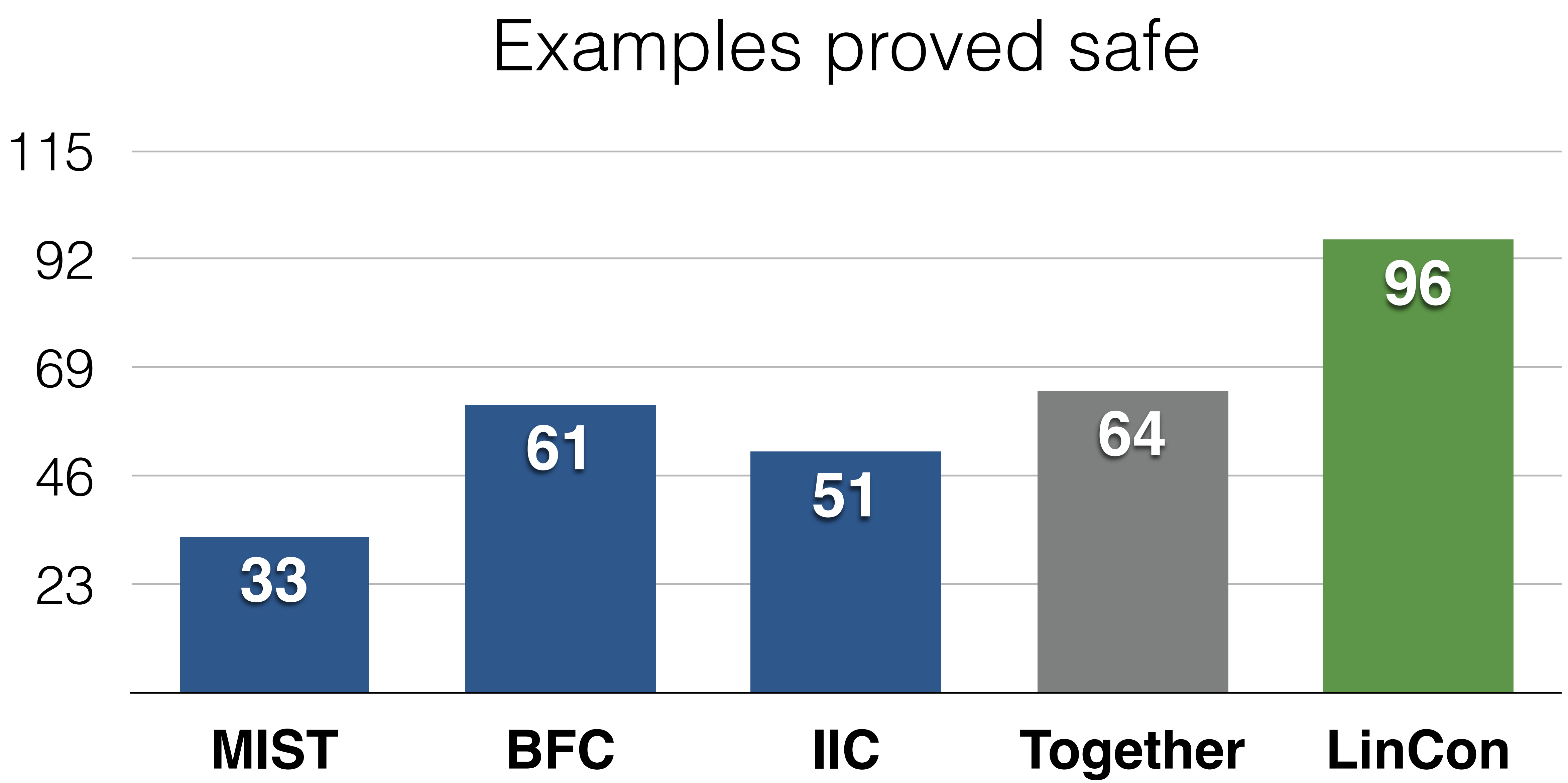


Use **SMT** for linear and Boolean constraints.

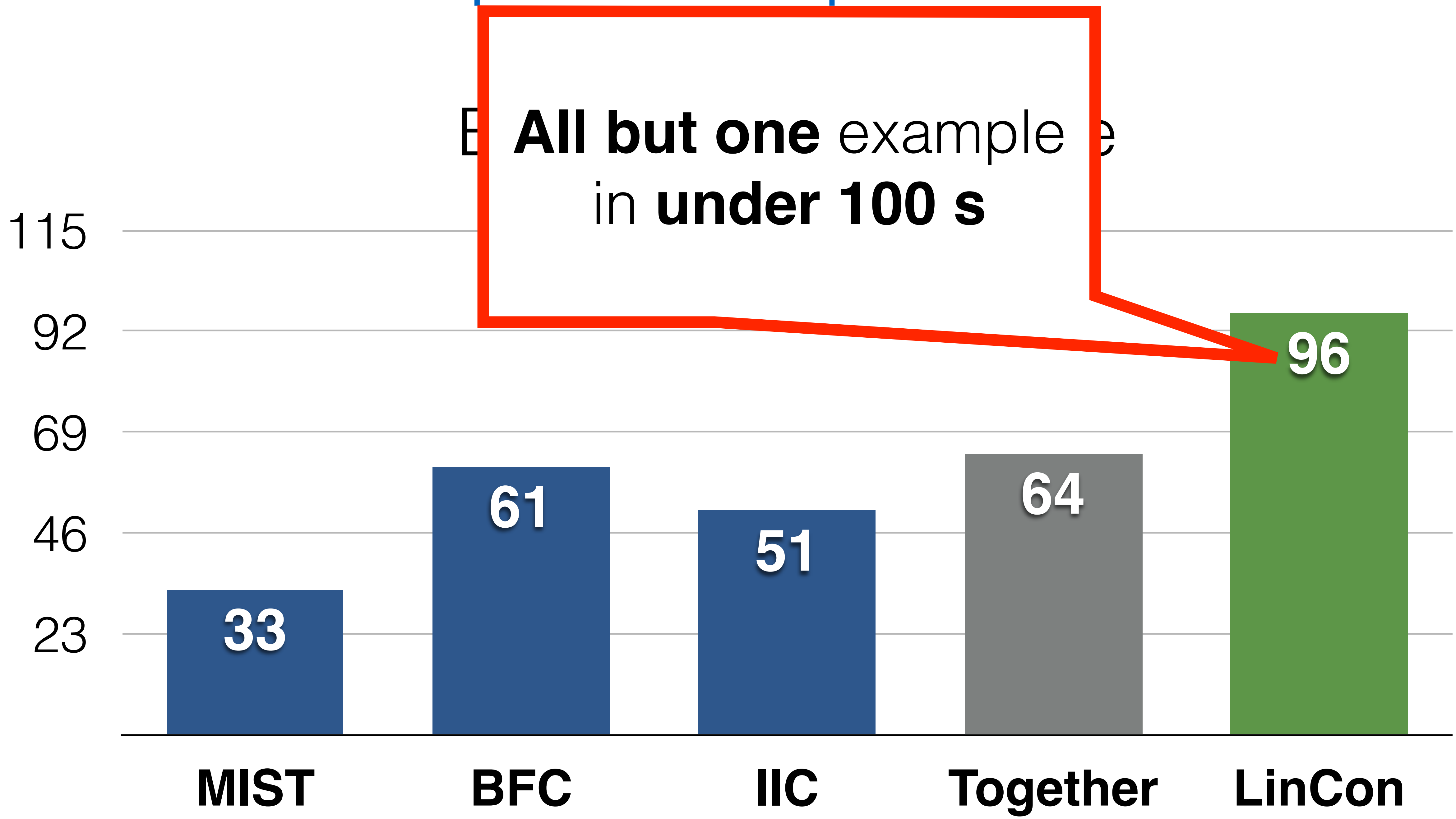
But **LinCon is incomplete.**

Does it make sense to use it?

Yes! For the right class of examples, LinCon is "quite complete"



Yes! For the right class of examples, LinCon is "quite complete"



Contributions

Main contribution:

- Extensive **experimental evaluation** showing that **LinCon works well**

Also:

- Using **duality** of linear programming to derive succinct **inductive invariants**

Contributions

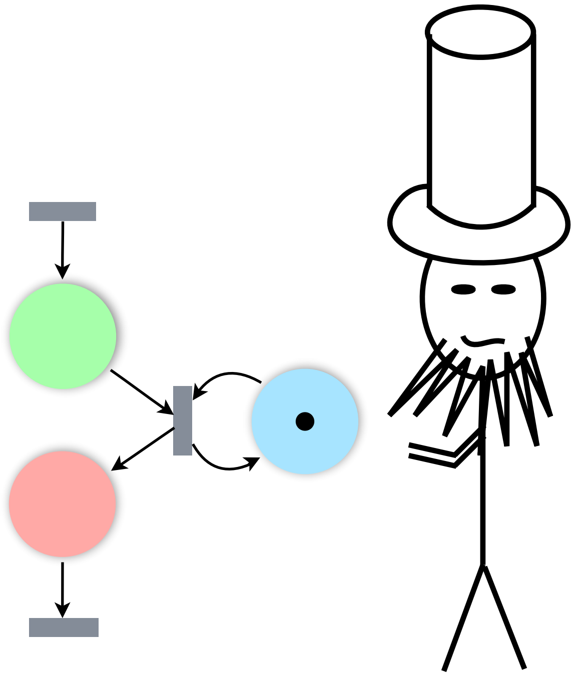
Main contribution:

- Extensive **experimental evaluation** showing that **LinCon works well**

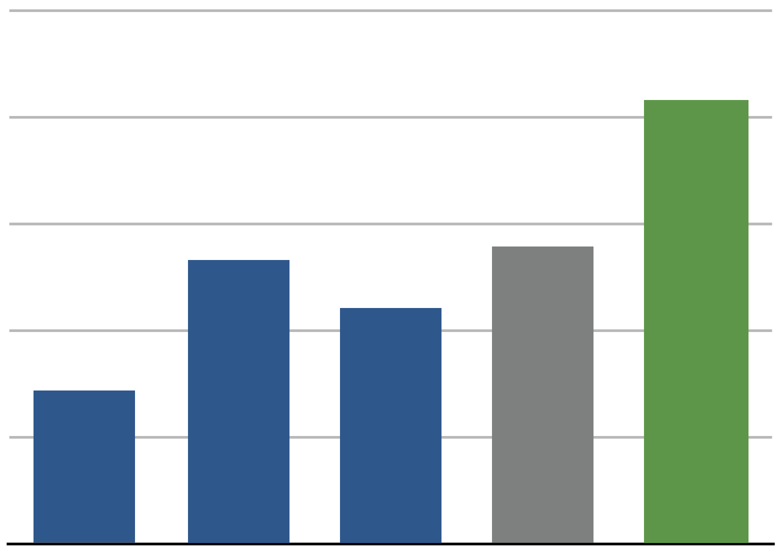
Also:

- Using **duality** of linear programming to derive succinct **inductive invariants**

In this talk

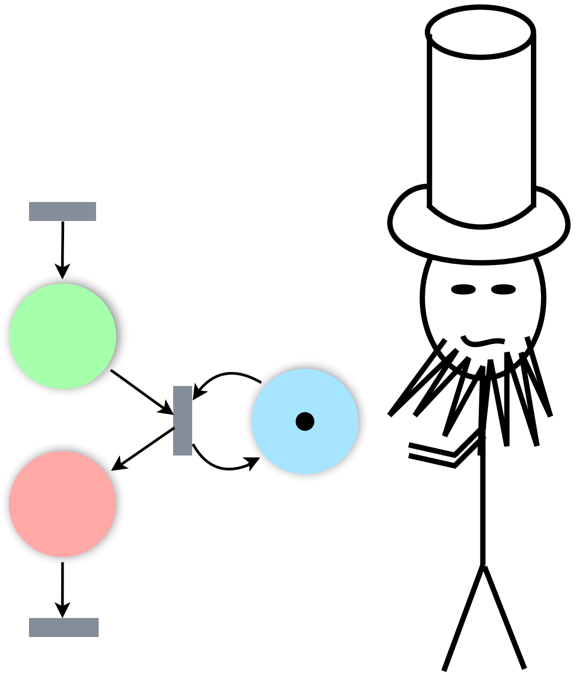


Petri nets
and LinCon



Experiments

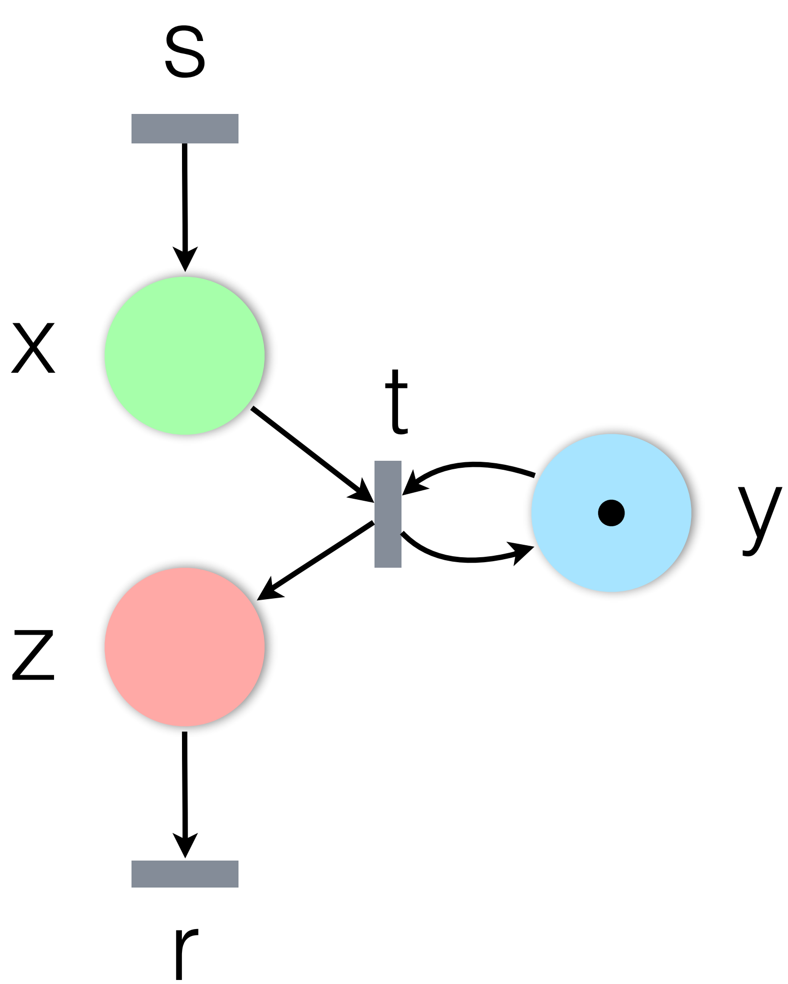
In this talk



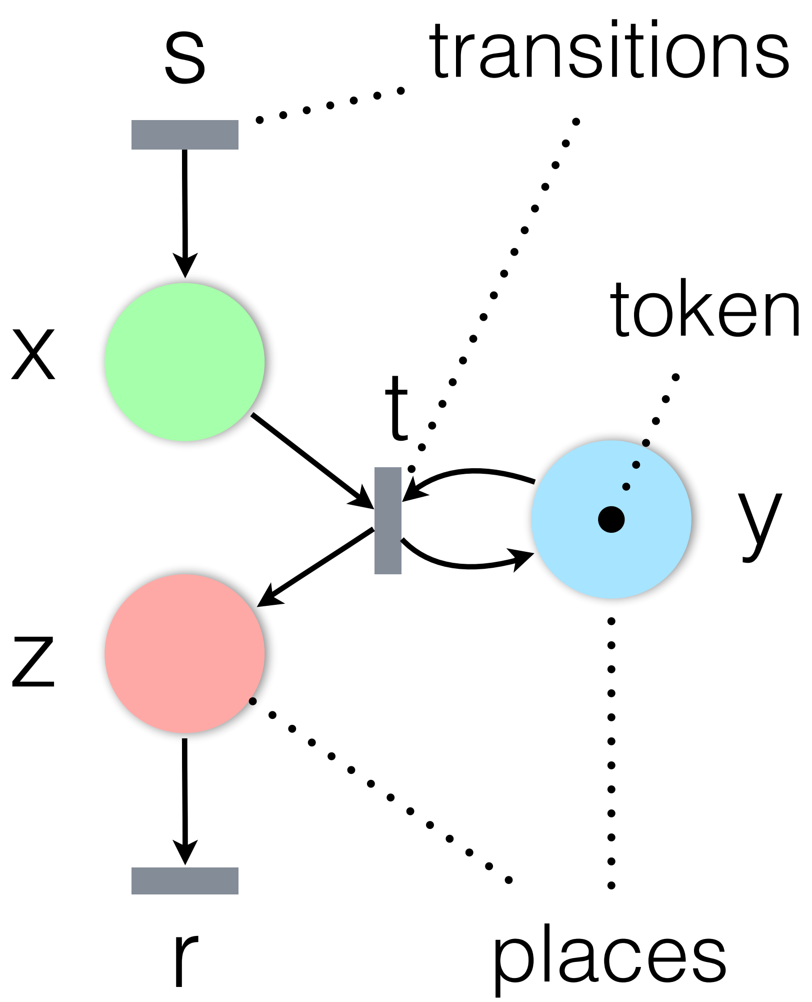
Petri nets
and LinCon

Experiments

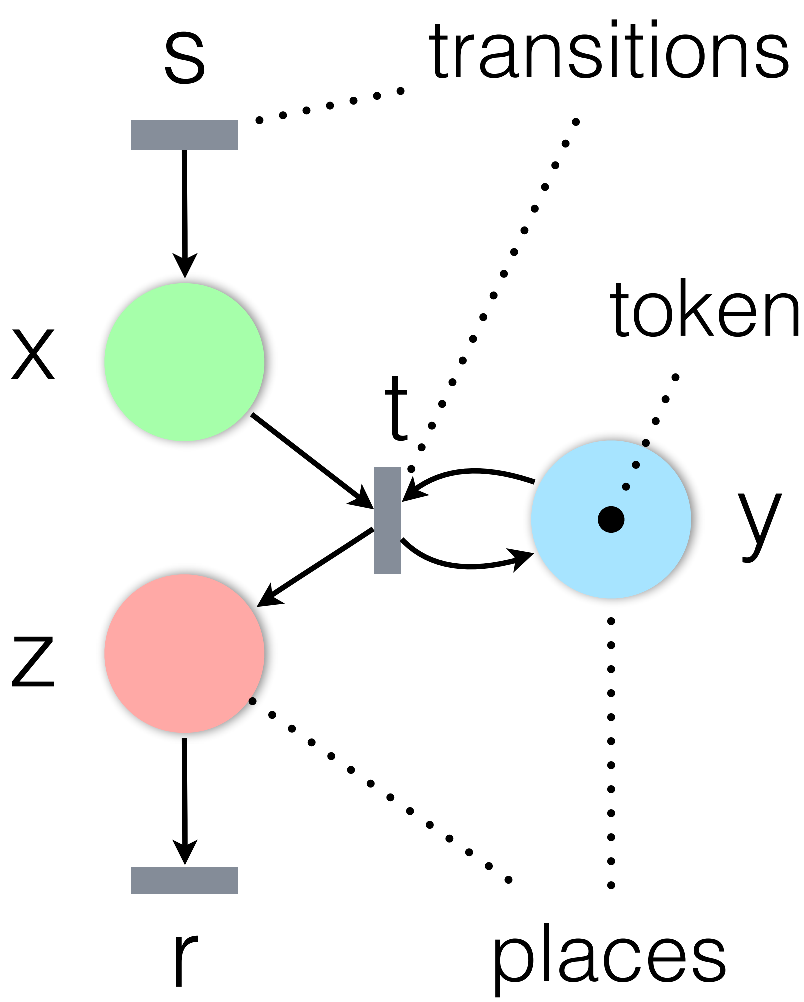
Petri nets are state transition systems



Petri nets are state transition systems



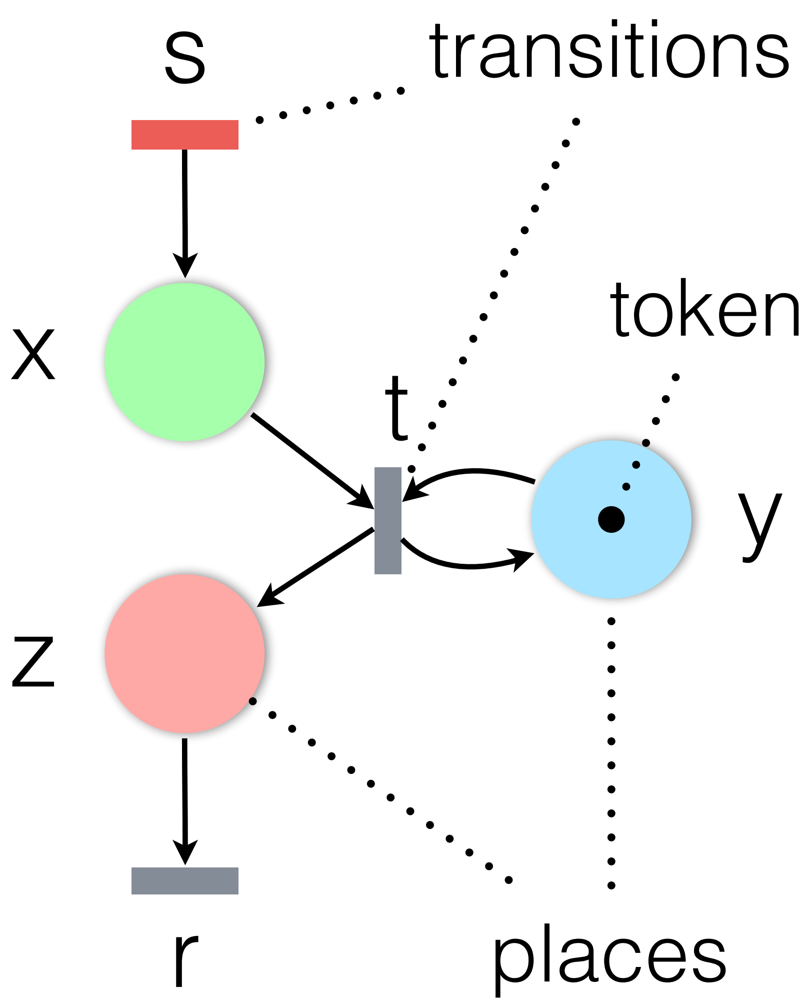
Petri nets are state transition systems



initial marking

(0, 1, 0)

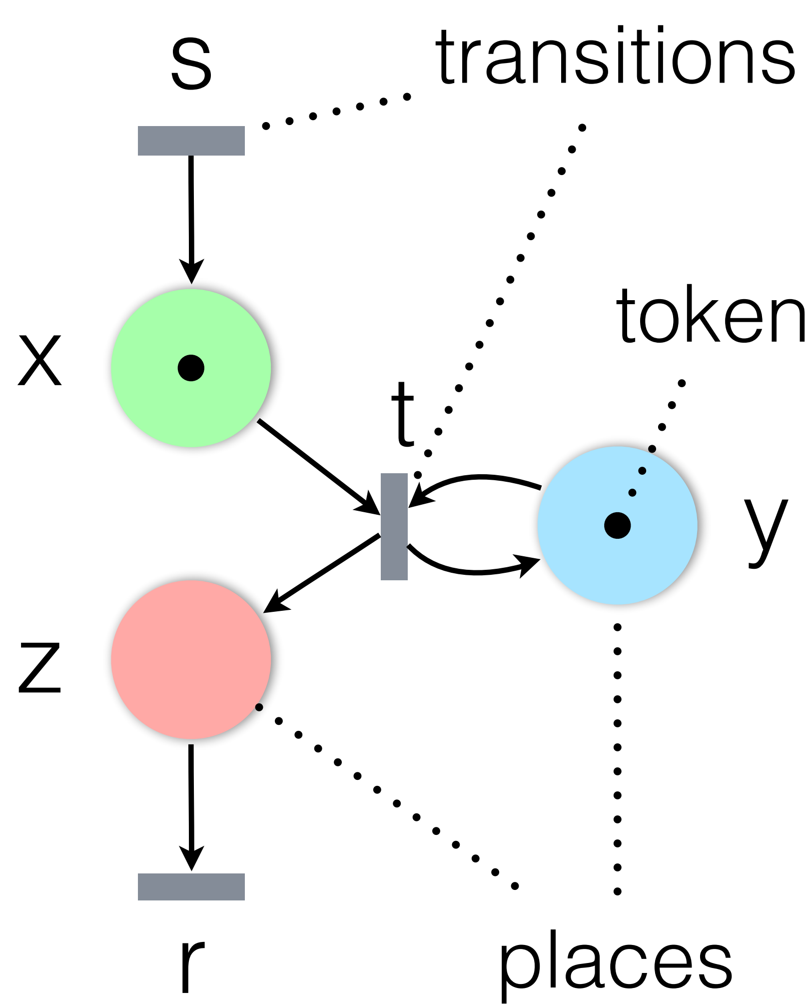
Petri nets are state transition systems



initial marking

(0, 1, 0)

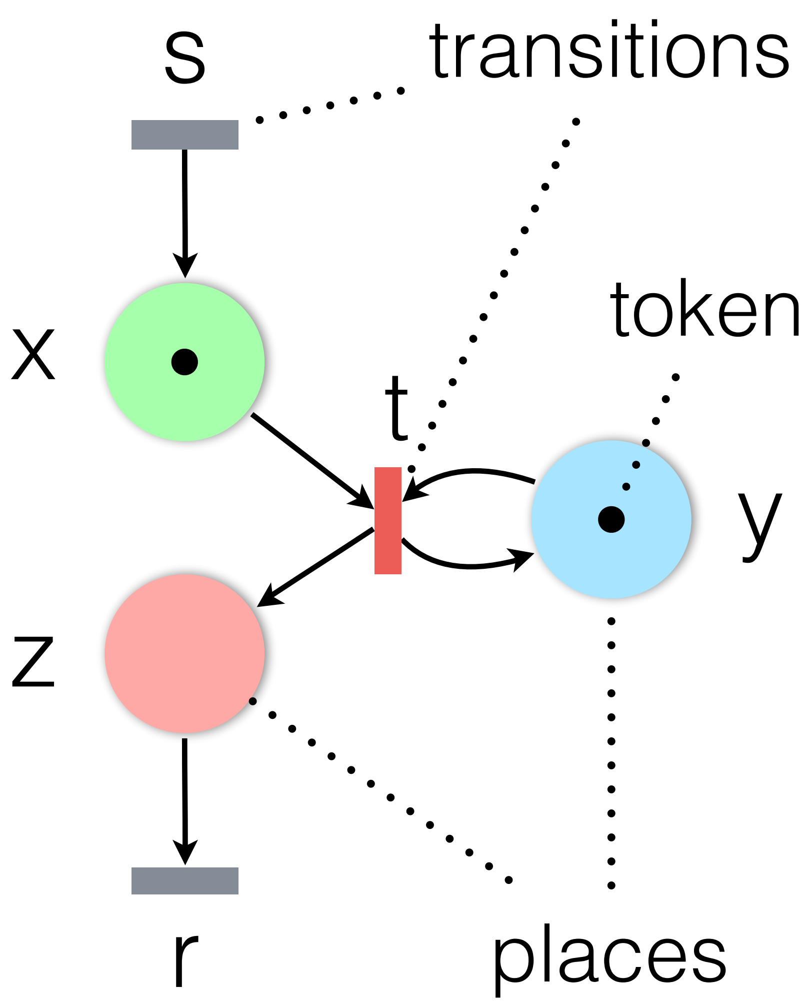
Petri nets are state transition systems



initial marking

$$\begin{array}{l} (0, 1, 0) \\ \quad\downarrow \cdots\cdots +(1, 0, 0) \\ (1, 1, 0) \end{array}$$

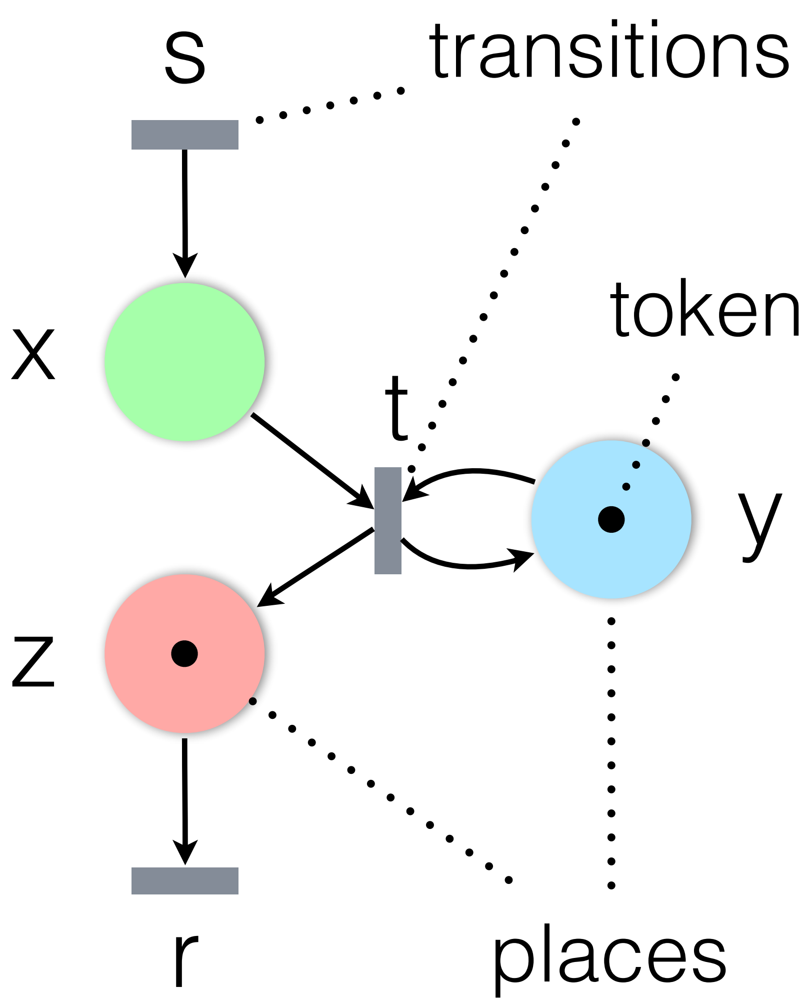
Petri nets are state transition systems



initial marking

$$(0, 1, 0) \xrightarrow{+(1, 0, 0)} (1, 1, 0)$$

Petri nets are state transition systems



initial marking

$$(0, 1, 0)$$

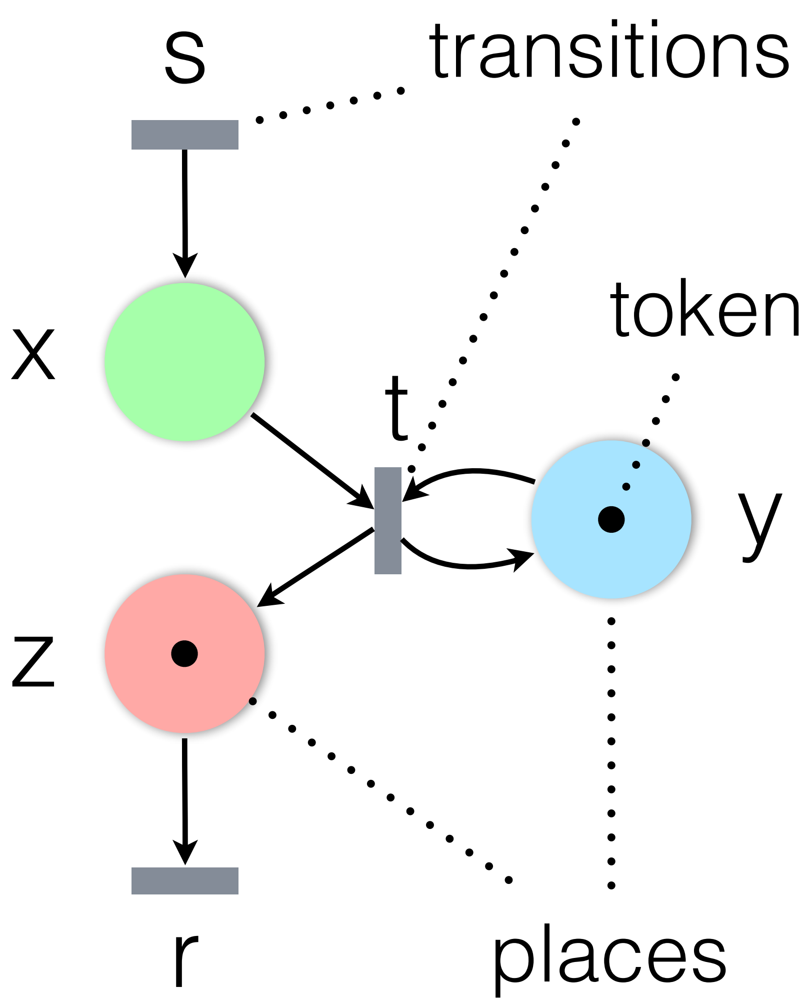
$$\big\downarrow \cdots\cdots +(1, 0, 0)$$

$$(1, 1, 0)$$

$$\big\downarrow \cdots\cdots +(-1, 0, 1)$$

$$(0, 1, 1)$$

Petri nets are state transition systems



initial marking

$(0, 1, 0)$

$\downarrow \; +(1, 0, 0)$

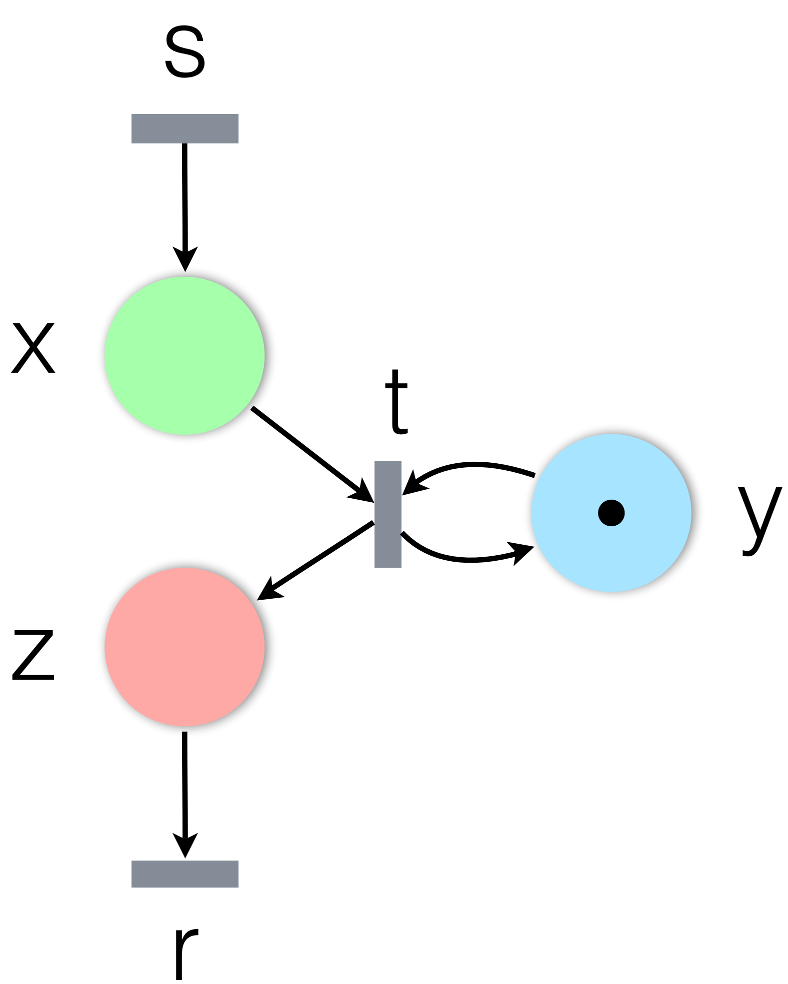
$(1, 1, 0)$

$\downarrow \; +(-1, 0, 1)$

$(0, 1, 1)$

reachable markings

Reachable markings satisfy marking equation

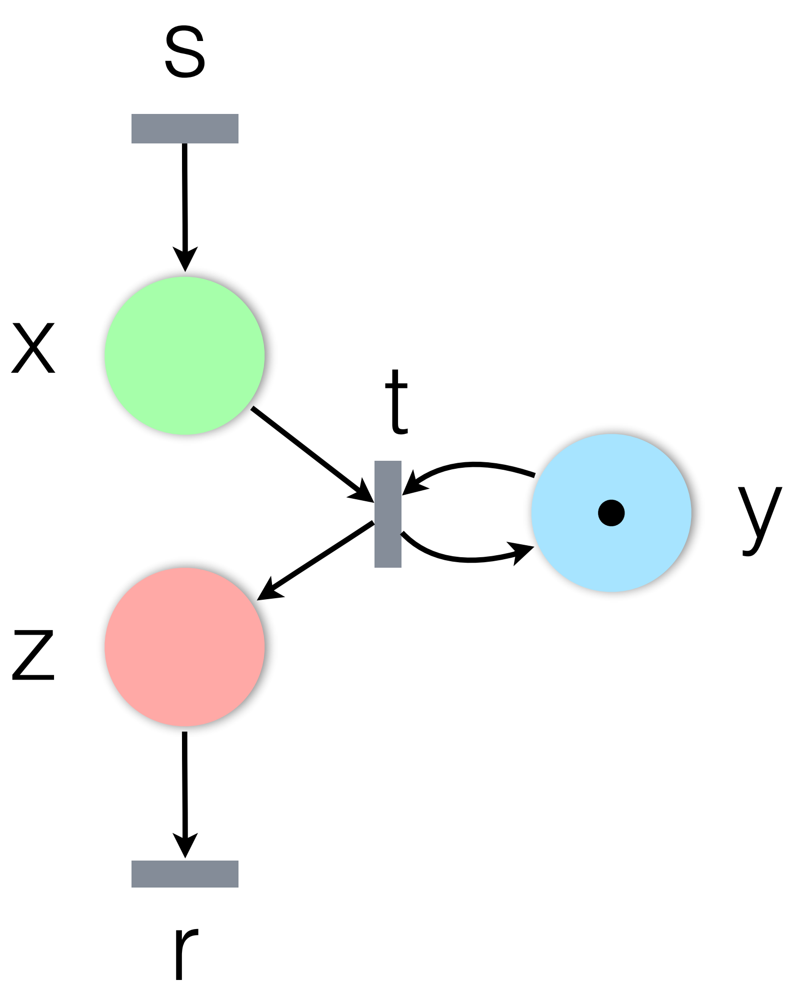


Ignore the order of transitions:

- **marking equation** [Murata '77]

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 1 & -1 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} s \\ t \\ r \end{bmatrix}$$

Reachable markings satisfy marking equation



Ignore the order of transitions:

- **marking equation** [Murata '77]

$$M = m_0 + C\,X$$

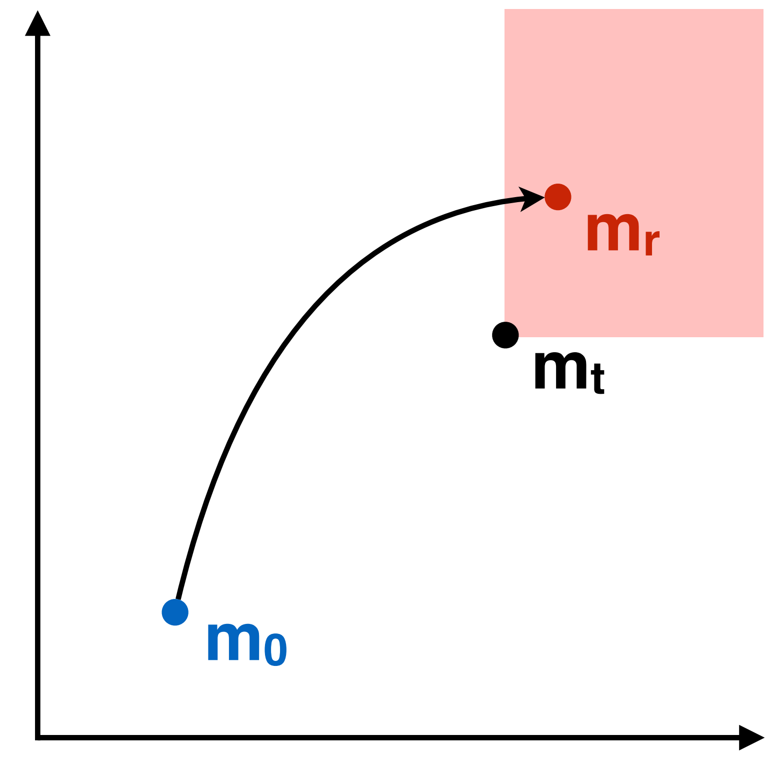
marking vector · initial marking · incidence matrix · transition vector

Coverability problem

Given a Petri net with:

- initial marking $\mathbf{m_0}$
- target marking $\mathbf{m_t}$

Is there a reachable marking $\mathbf{m_r}$ that **covers** $\mathbf{m_t}$?

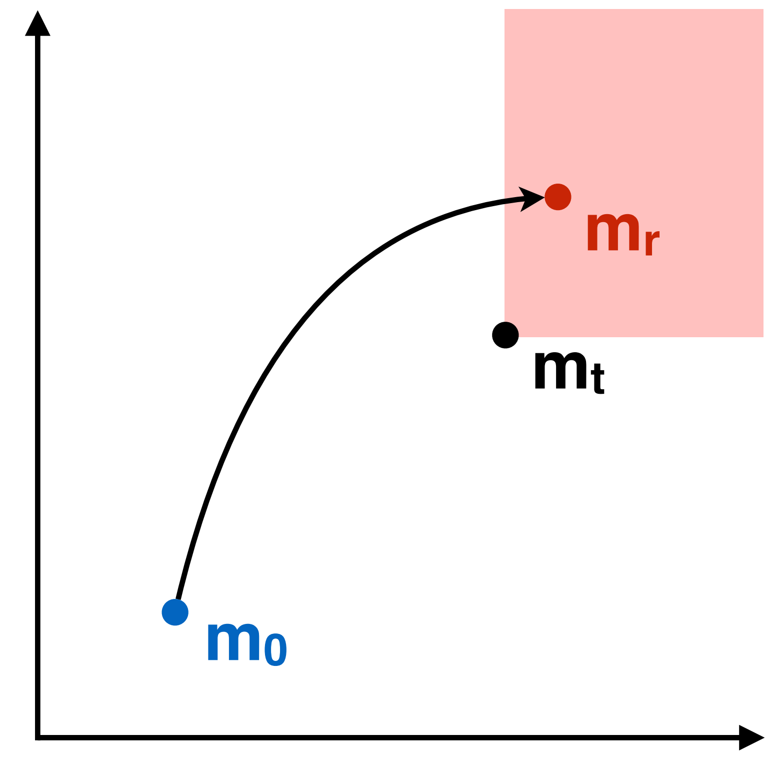


Coverability problem

Given a Petri net with:

- initial marking $\mathbf{m_0}$
- target marking $\mathbf{m_t}$

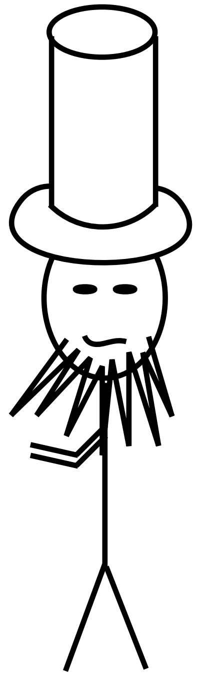
Is there a reachable marking $\mathbf{m_r}$ that **covers** $\mathbf{m_t}$?



If $\mathbf{m_t}$ is **not coverable**, Petri net is **safe**.

Adding coverability constraint to marking equation yields basic LinCon

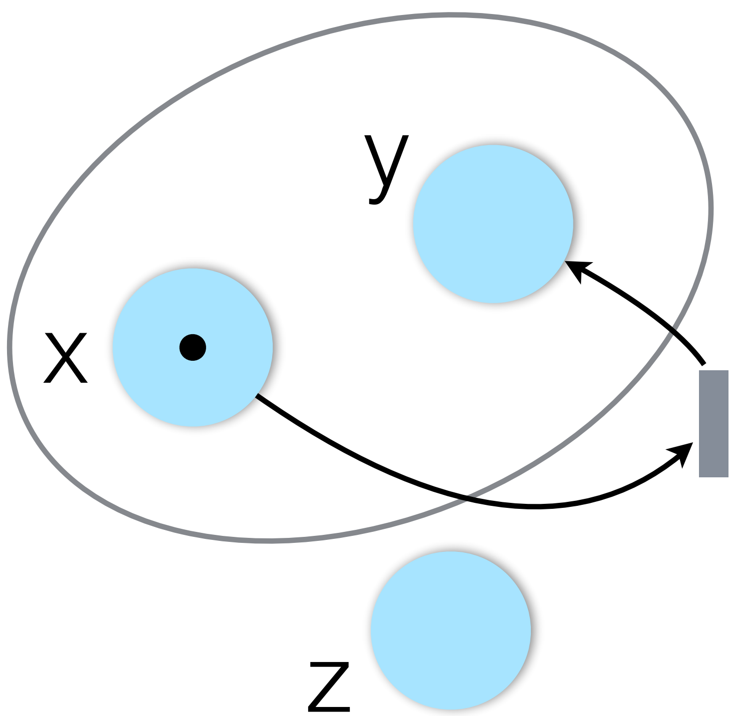
$$M = m_0 + CX$$
$$M \geq m_t$$
$$X \geq 0$$



If the **constraints are not feasible**, the **Petri net is safe**.

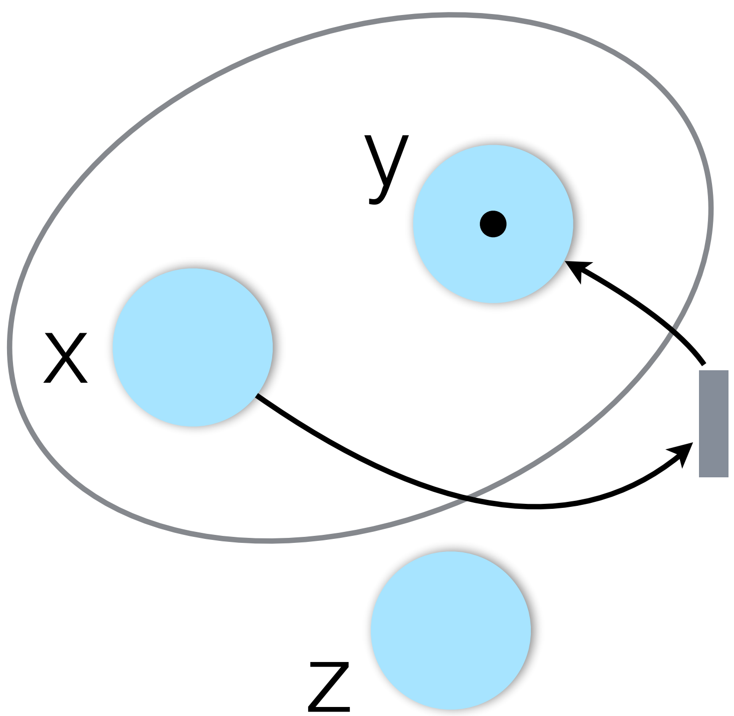
Strengthening LinCon using traps [EM '00]

Trap — **set of places** such that every **transition that consumes tokens** from it also **puts tokens** into it.



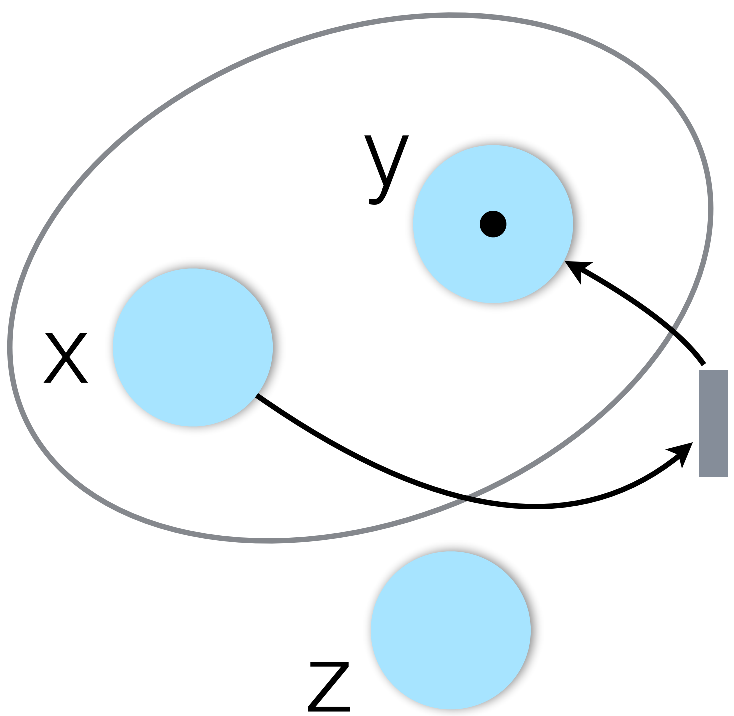
Strengthening LinCon using traps [EM ’00]

Trap — **set of places** such that every **transition that consumes tokens** from it also **puts tokens** into it.



Strengthening LinCon using traps [EM '00]

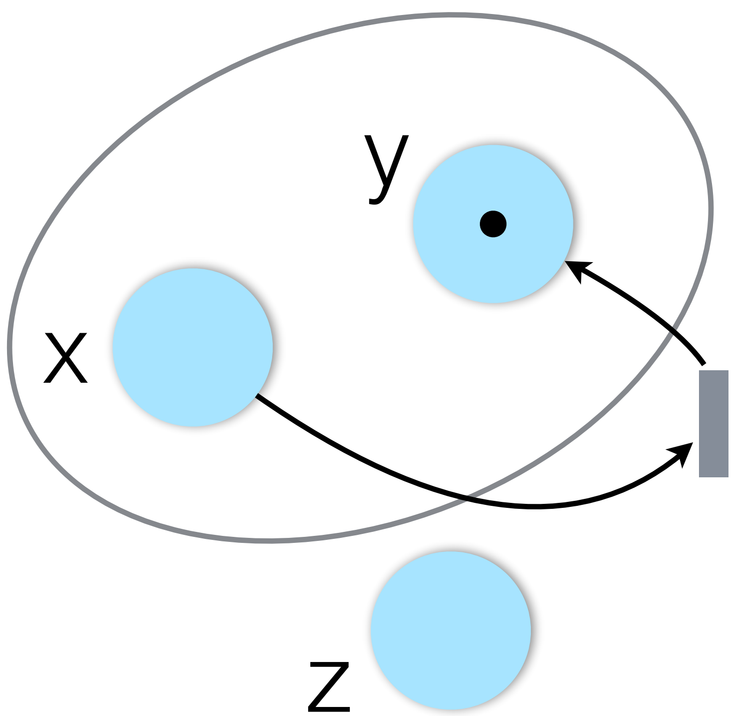
Trap — **set of places** such that every **transition that consumes tokens** from it also **puts tokens** into it.



If a trap is marked, it stays marked.

Strengthening LinCon using traps [EM '00]

Trap — **set of places** such that every **transition that consumes tokens** from it also **puts tokens** into it.



If a trap is marked, it stays marked.

$$x + y \geq 1$$

LinCon with traps [EM '00]

$$
\begin{aligned}
M &= m_0 + CX \\
M &\geq m_t \\
X &\geq 0
\end{aligned}
$$

LinCon with traps [EM '00]

$$M = m_0 + CX$$
$$M \geq m_t$$
$$X \geq 0$$

$\longrightarrow$ no solution

safe

LinCon with traps [EM '00]

$$M = m_0 + CX$$
$$M \geq m_t$$
$$X \geq 0$$

no solution
safe

$$M = m_r$$
$$X = x_r$$

LinCon with traps [EM '00]

$$M = m_0 + CX$$
$$M \geq m_t$$
$$X \geq 0$$

no solution

safe

$$M = m_r$$
$$X = x_r$$

Is there a trap

- initially marked
- empty at m_r

LinCon with traps [EM '00]

$$M = m_0 + CX$$
$$M \geq m_t$$
$$X \geq 0$$

no solution

safe

$$M = m_r$$
$$X = x_r$$

SAT query:

Is there a trap

- initially marked
- empty at m_r

LinCon with traps [EM '00]

$$M = m_0 + CX$$
$$M \geq m_t$$
$$X \geq 0$$

no solution
safe

$$M = m_r$$
$$X = x_r$$

SAT query:
Is there a trap
- initially marked
- empty at m_r

no solution
inconclusive

LinCon with traps [EM '00]

$$M = m_0 + CX$$
$$M \geq m_t$$
$$X \geq 0$$

no solution
safe

$$M = m_r$$
$$X = x_r$$

SAT query:
Is there a trap
- initially marked
- empty at m_r

no solution
inconclusive

$$T_p = \begin{cases} 1, & p \text{ in trap} \\ 0, & \text{otherwise} \end{cases}$$

LinCon with traps [EM ’00]

$$\begin{aligned} M &= m_0 + CX \\ M &\geq m_t \\ X &\geq 0 \\ T^{\tau} M &\geq 1 \end{aligned}$$

no solution
safe

$$M = m_r$$
$$X = x_r$$

SAT query:

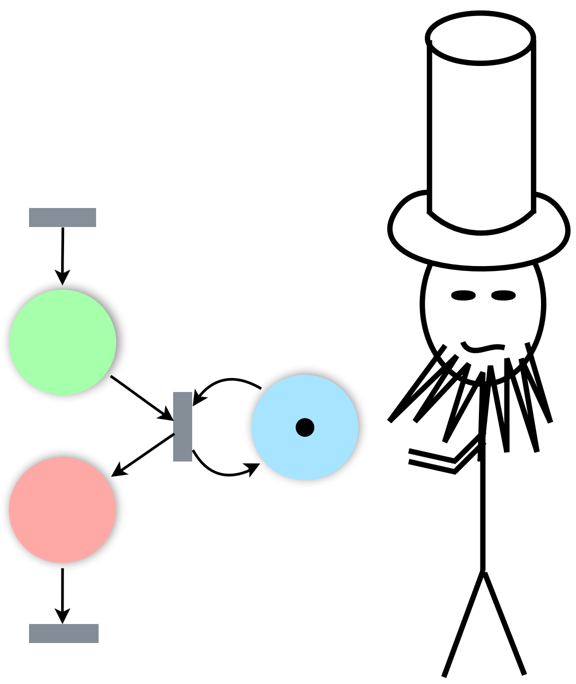
Is there a trap

- initially marked
- empty at m_r

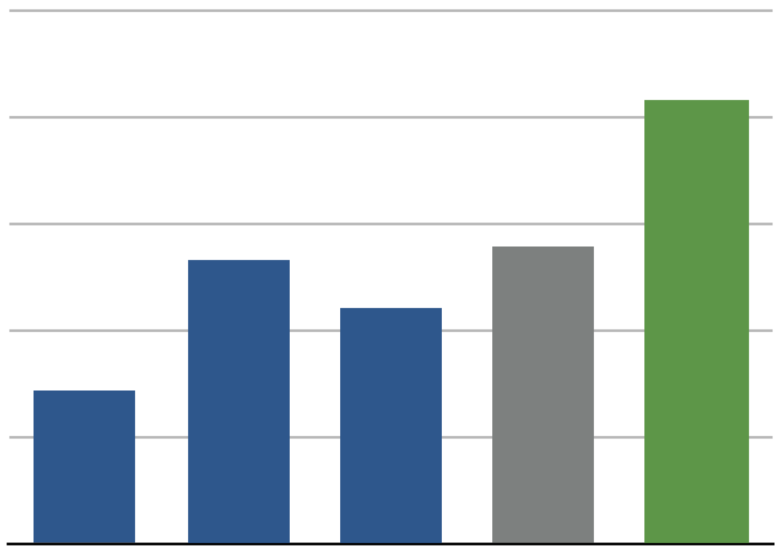
no solution
inconclusive

$$T_p = \begin{cases} 1, & p \text{ in trap} \\ 0, & \text{otherwise} \end{cases}$$

In this talk

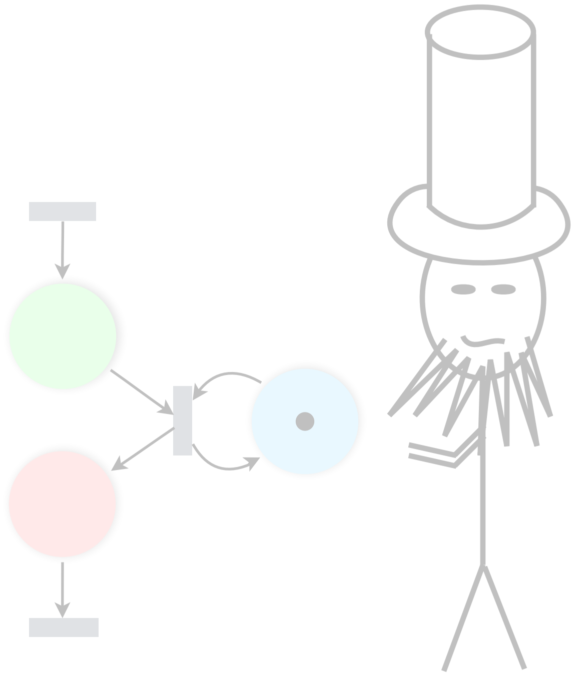


Petri nets
and LinCon

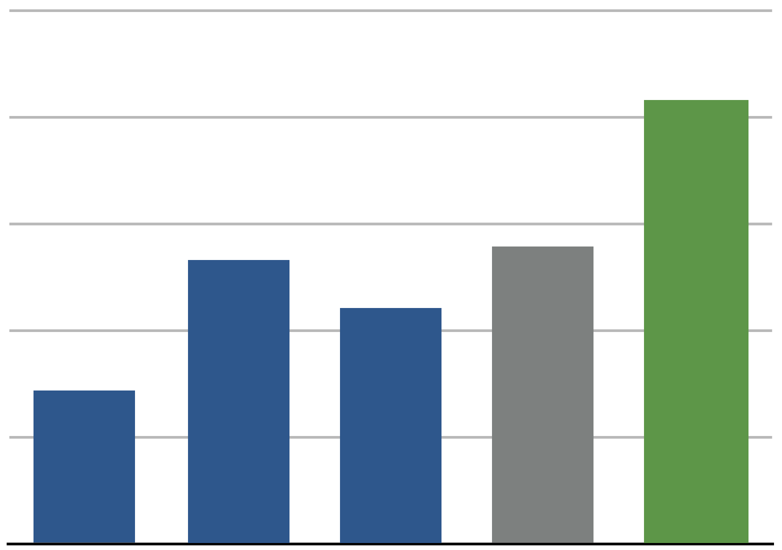


Experiments

In this talk



Petri nets
and LinCon



Experiments

The origin of examples

- MIST — https://github.com/pierreganty/mist

  Examples from the literature

- BFC — http://www.cprover.org/bfc/

  Examples from verification of concurrent C programs

- Provenance verification for message-passing programs [MMW '13]

  Examples modeling a medical system and a bug-tracking system

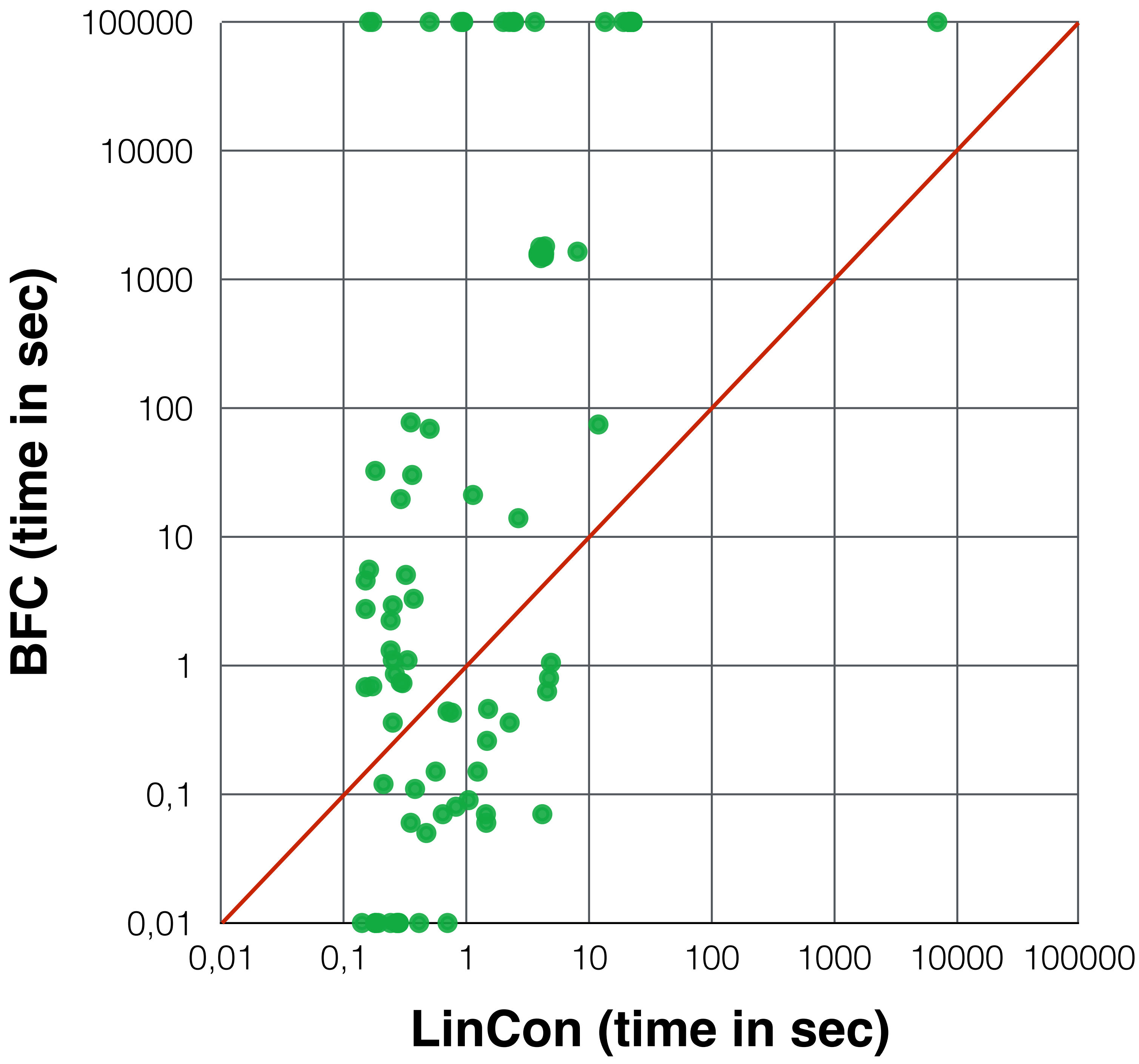
- SOTER — http://mjolnir.cs.ox.ac.uk/soter/ [DKO '13]

  Examples from verification of Erlang programs
  Contains a Petri net with **66,950 places** and **213,625 transitions**

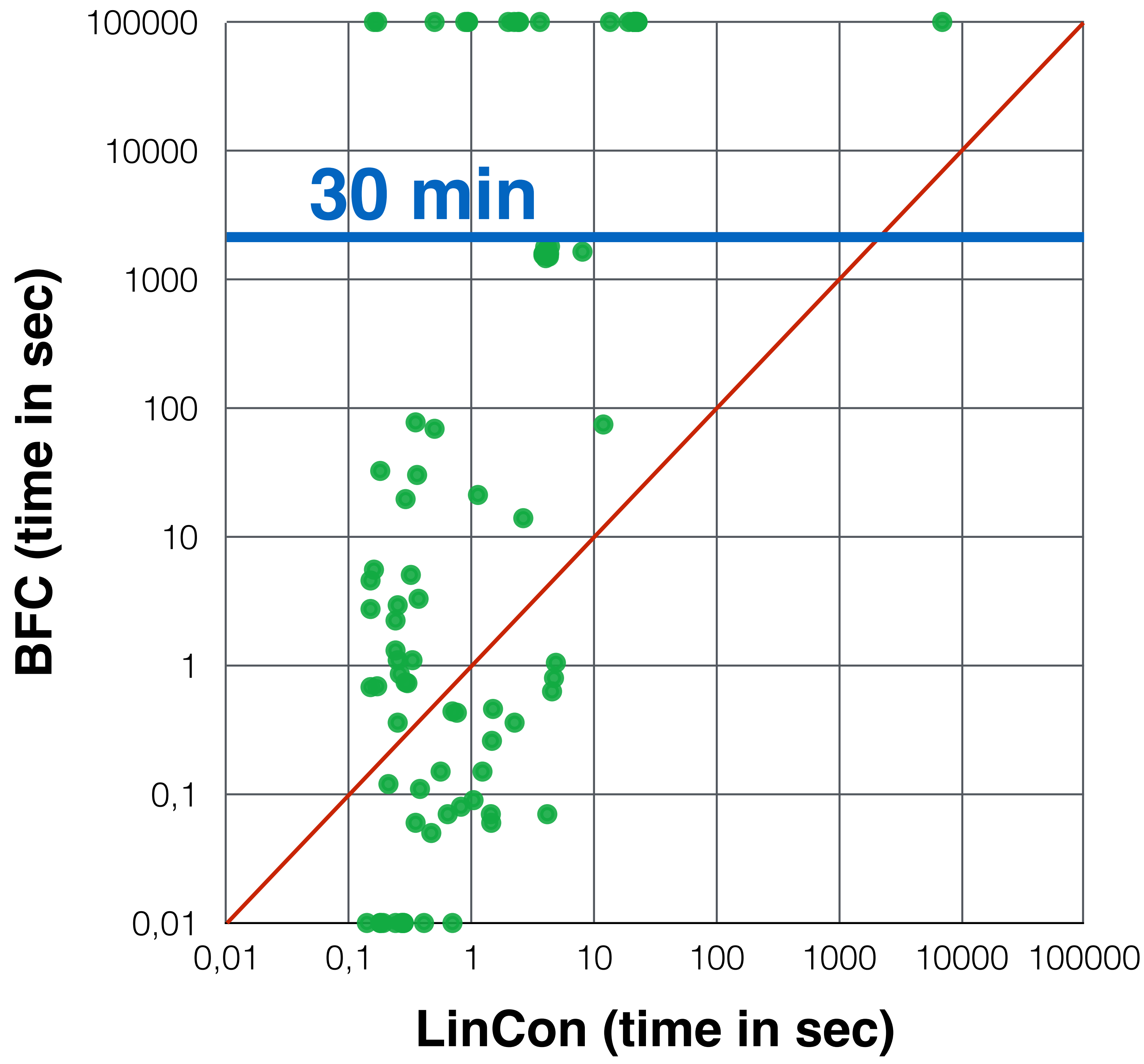
Main point here:

LinCon works well even without traps

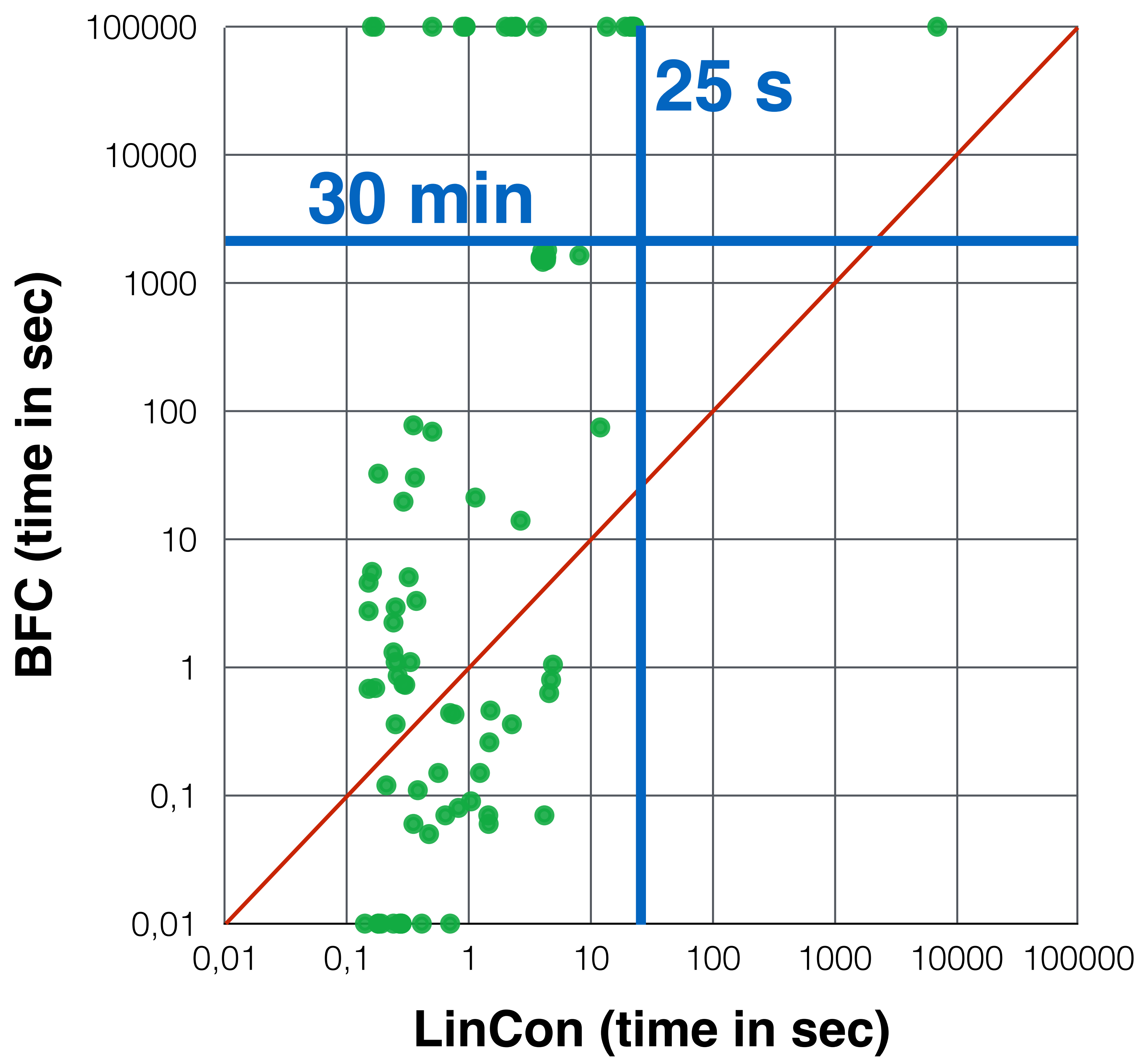
LinCon without traps is fast



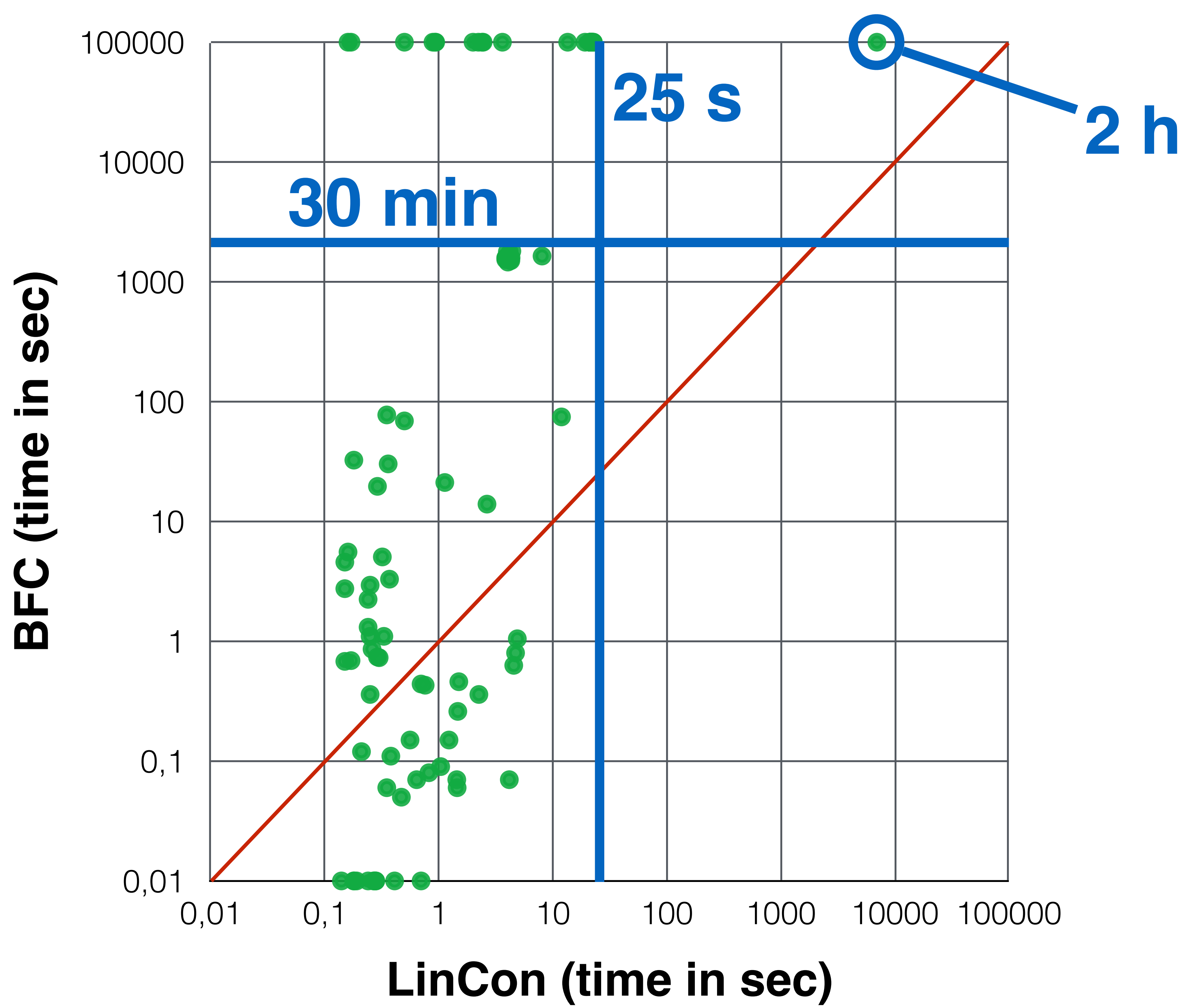
LinCon without traps is fast



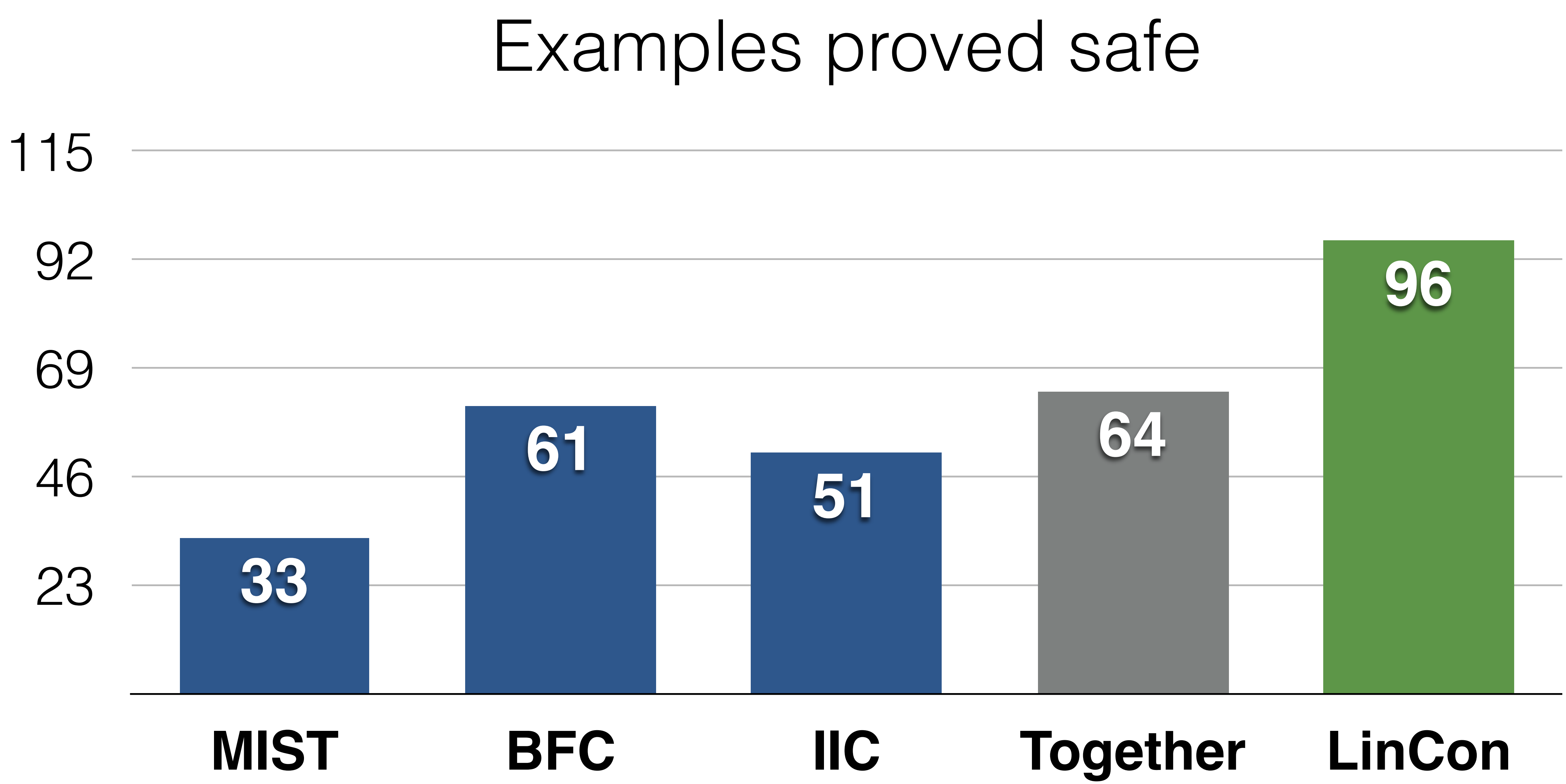
LinCon without traps is fast



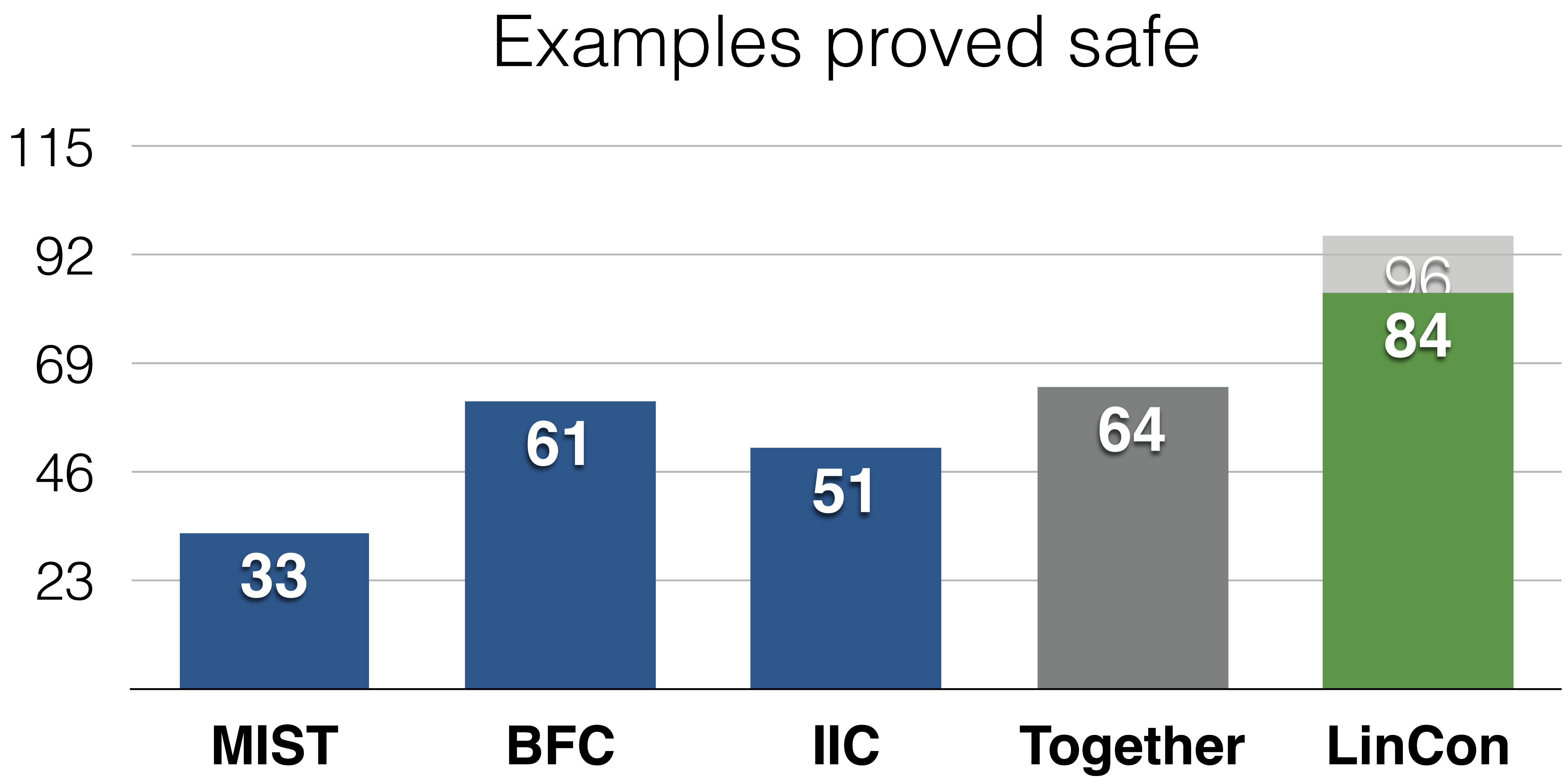
LinCon without traps is fast



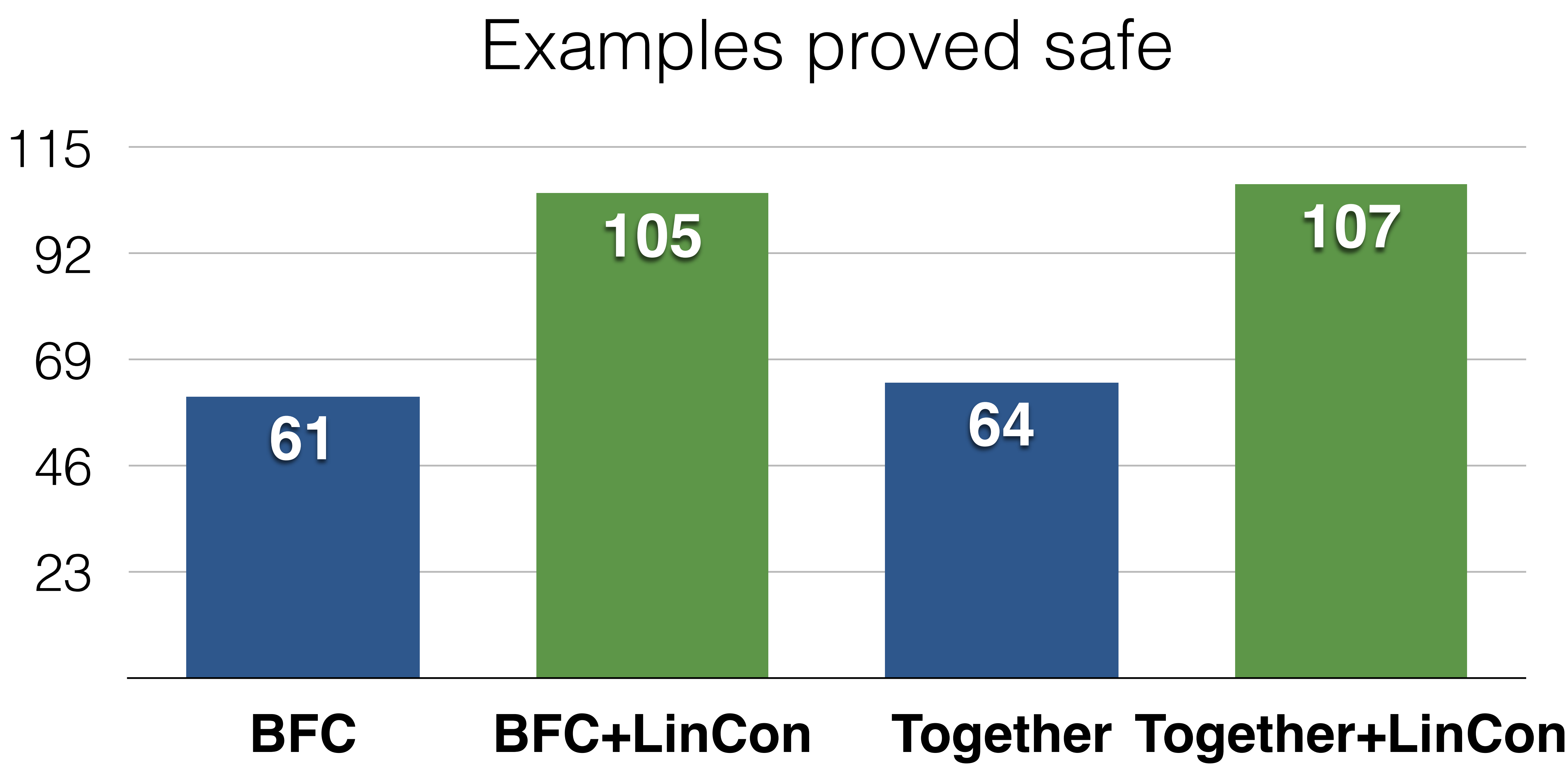
LinCon is “quite complete”



LinCon without traps is "quite complete"



If LinCon were combined with other tools



Summary

- We've revisited a **linear constraint approach** to **Petri net coverability**
- LinCon is **incomplete**, but **useful**

  … on its own

  … as a cheap preprocessing step in other tools