Late Choice

Model Checking Asynchronous Systems with Queues using Constraints

Claus Traulsen
ctr@informatik.uni-kiel.de
joint work with Martin Steffen

Department of Computer Science and Applied Mathematics
Christian-Albrechts University of Kiel

October 19, 2005

Outline

1. Motivation and Definitions
2. A semantics with constraints
3. Use constraints for model checking
4. Experimental results
5. Conclusion

Problem Setting

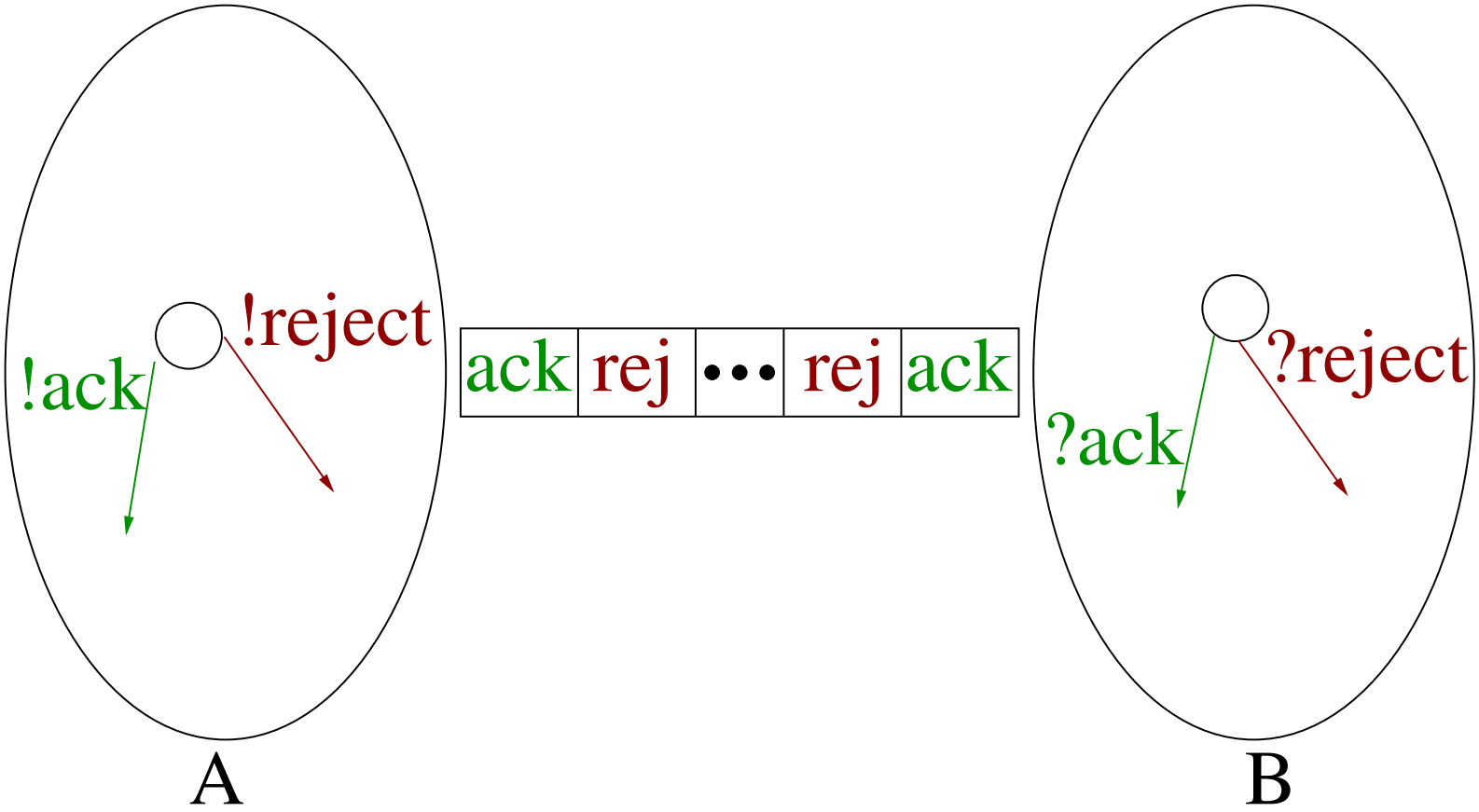
Enumerative Model Checking

- enumerate the reachable states
- show that a specification holds for all paths in the state-graph
- main problem: state explosion

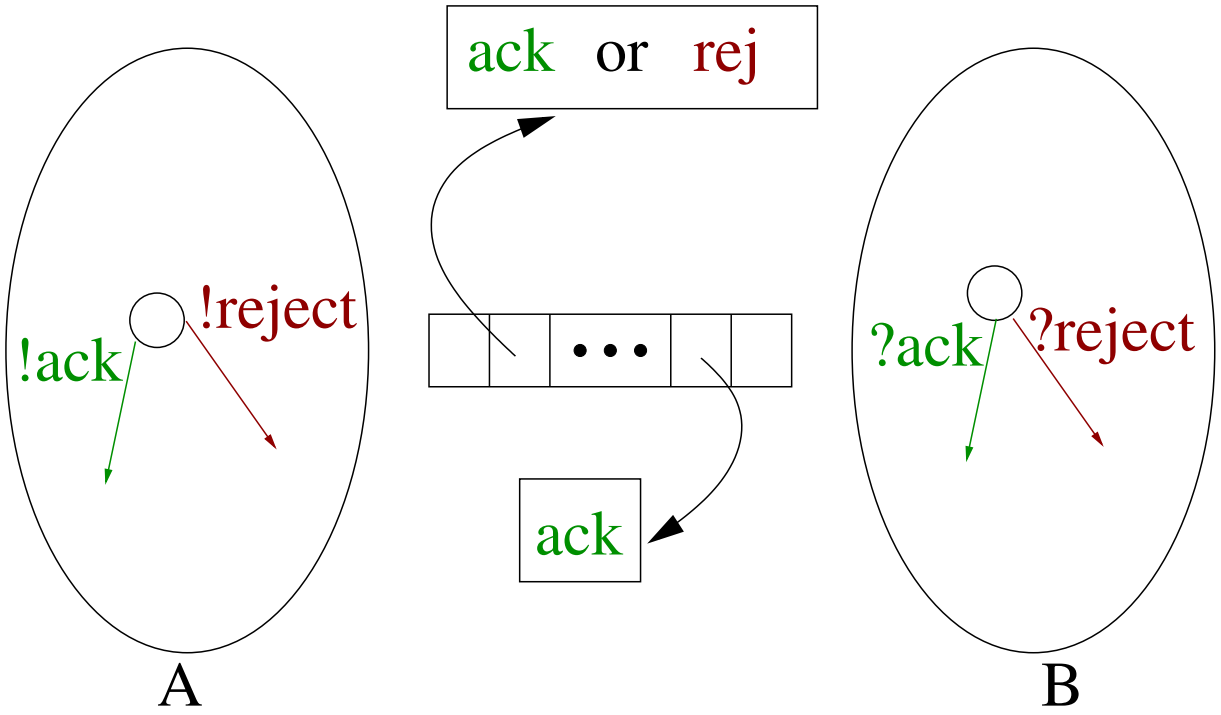
Queues

- are often needed to model protocols and distributed systems
- may lead to additional state explosion

State-Explosion caused by Queues



Late Choice



- the value is specified as soon as A or B uses it
- pointers are hard to formalize
  $\longrightarrow$ use constraints instead

Constraints

Definition: subset of $\mathrm{Dom}(x_1) \times \cdots \times \mathrm{Dom}(x_n)$

Sub-constraints: $C \subseteq_U D$, $U \subseteq \mathrm{var}(C) \cap \mathrm{var}(D)$

Representation: set of equations and in-equations

- set of variables is implicitly given
- constraint is set of valuations $\sigma : \mathit{Var} \mapsto \mathit{Dom}$ which solve this equations and in-equations

Syntax

- Parallel composition of LTS
- Communication via bounded queues

Labels

Input: $l \xrightarrow{a?x} l'$

Output: $l \xrightarrow{g \triangleright a!e} l'$

Assign: $l \xrightarrow{g \triangleright x:=e} l'$

External Input: $l \xrightarrow{ext?_{[y,z]}x} l'$

Skip: $l \xrightarrow{g} l'$

Defining the semantics: receiving data

- *StdInput*:

$$\frac{\vec{l} \xrightarrow{a?x} \vec{l}' \qquad len(a_Q) > 0 \qquad v = head(a_Q)}{(\vec{l}, \sigma, Q) \rightarrow (\vec{l}', \sigma[x \mapsto v], Q[tail(a)/a])}$$

- *ConInput*:

$$\frac{\vec{l} \xrightarrow{a?x} \vec{l}' \qquad len(a_Q) > 0 \qquad y = head(a_Q) \qquad \text{new } x'}{(\vec{l}, C, Q) \rightarrow (\vec{l}', C[x'/x] \cup \{x = y\}, Q[tail(a)/a])}$$

Defining the semantics: external inputs

- *StdExtInput*:

$$\frac{\vec{I} \xrightarrow{ext?_{[y,z]}x} \vec{I'} \qquad [\![y]\!]_\sigma \leq v \leq [\![z]\!]_\sigma}{(\vec{I}, \sigma, Q) \rightarrow (\vec{I'}, \sigma[x \mapsto v], Q)}$$

- *ConExtInput*:

$$\frac{\vec{I} \xrightarrow{ext?_{[y,z]}x} \vec{I'} \qquad C \cup \{y \leq z\} \not\models \bot \qquad \text{new } x'}{(\vec{I}, C, Q) \rightarrow (\vec{I'}, C[x'/x] \cup \{y \leq x, x \leq z\}, Q)}$$

How to deal with the new variables?

- *ConEquiv*

$$\frac{C \equiv_{Var_P \cup Var_Q} D}{(\vec{I}, C, Q) \rightsquigarrow (\vec{I}, D, Q)}$$

- *ConSpecialize*

$$\frac{x \in Var_H \qquad u \in [\![x]\!]_C}{(\vec{I}, C, Q) \rightsquigarrow (\vec{I}, C[u/x], Q)}$$

When shall we specialize to an exact value?

- Never $\longrightarrow$ minimal number of states.
- Always $\longrightarrow$ minimal size for every state.
- After receiving $\longrightarrow$ might be a good compromise.

Equivalence

Soundness

Every state represented by a reachable state of the semantics with constraints is reachable in the standard semantics.

Completeness

Every state reachable in the standard semantics is represented by a reachable state of the semantics with constraints.

Checking Arbitrary LTL Formulae

How to decide whether states are equal

1. reduce to same variables
2. test for sub-constraints

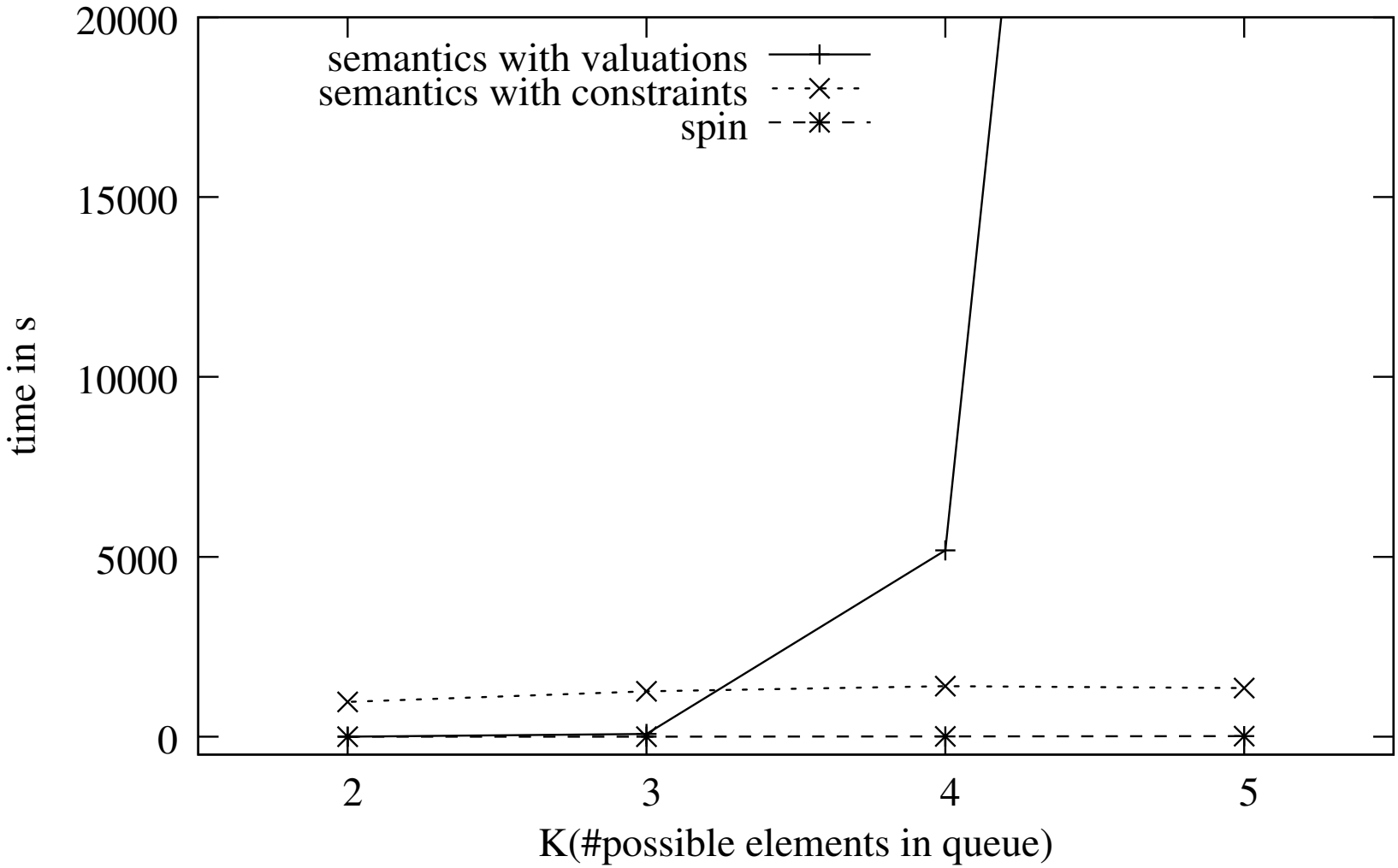
Problem

We enumerate the right states, but add spurious transitions.

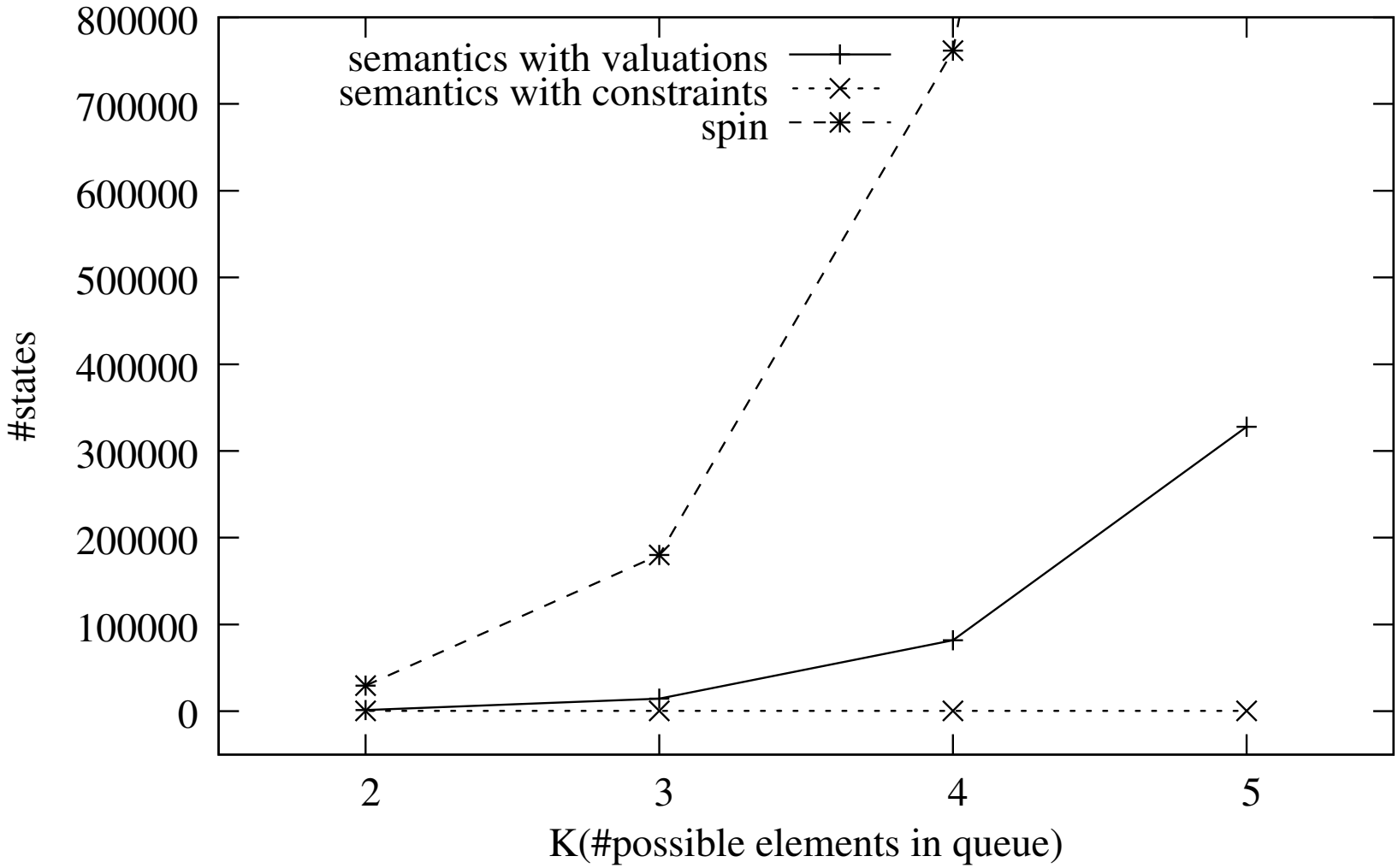
Solutions

- we do not have false positives, that is enough
- test for equality, not for sub-constraints
- use bounded model checking

Comparison: Time



Comparison: No. States



Conclusion

- Late choice helps to reduce the number of orderings in a queue
- Constraints can be used to encode different states by one
- Checking state properties works
- Checking arbitrary LTL-formulae is not so easy