

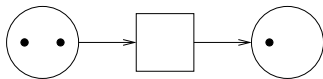
Regular Model Checking Made Simple and Efficient

Parosh Aziz Abdulla Bengt Jonsson Marcus Nilsson
Julien d'Orso

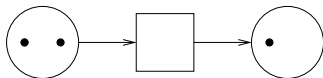
Presented October 8. 2007 by Joakim Byg

- 1 Introduction
 - Regular Model Checking
 - Abstractions and Approximations
- 2 Approach of Article
 - Computing T^*
 - Equivalence
 - Algorithm
- 3 Results
 - Time Tests
 - BDD Usage
- 4 Summary

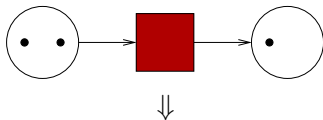
Regular Model Checking



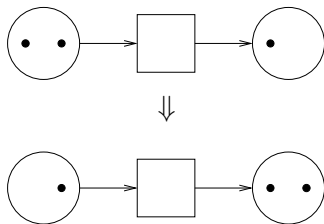
Regular Model Checking

 $oo \perp o$

Regular Model Checking

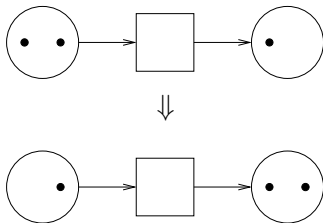

 $oo \perp o$


Regular Model Checking


 $oo \perp o$

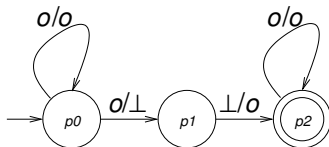
 $o \perp oo$

Regular Model Checking


 $oo \perp o$

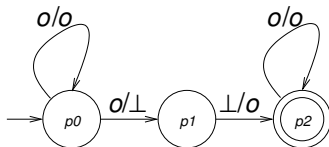
 $o \perp oo$

Transducer



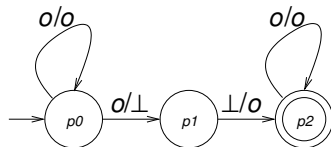
$oo\perp o \Rightarrow$

Transducer



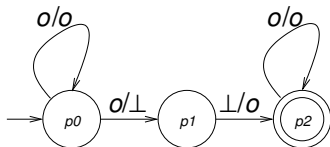
oo⊥o ⇒

Transducer



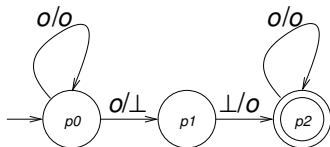
$oo\perp o \Rightarrow o$

Transducer



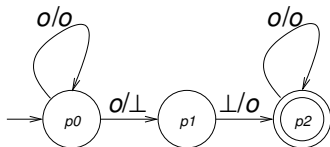
$oo\perp o \Rightarrow o$

Transducer



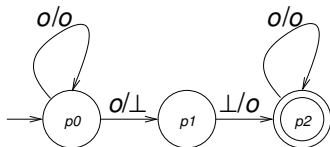
$oo\perp o \Rightarrow o\perp$

Transducer



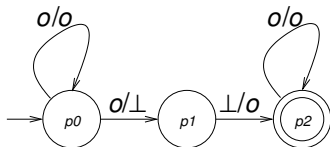
$oo\perp o \Rightarrow o\perp$

Transducer



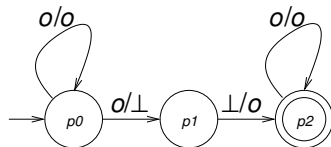
$oo\perp o \Rightarrow o\perp o$

Transducer



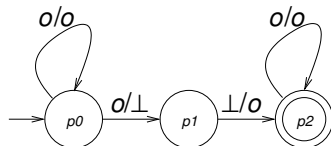
$oo\perp o \Rightarrow o\perp o$

Transducer



$oo\perp o \Rightarrow o\perp oo$

Transducer



$oo\perp o \Rightarrow o\perp oo \Rightarrow \perp ooo$

Power

Theorem

There is a regular set, I , and a length preserving finite-state transducer, T , such that the set $I \circ T^$ is not decidable.*

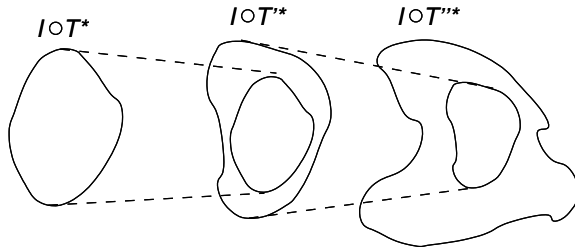
Power

Theorem

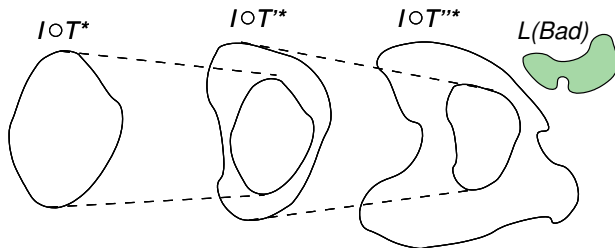
There is a regular set, I , and a length preserving finite-state transducer, T , such that the set $I \circ T^$ is not decidable.*

- But we would still like to do verifications

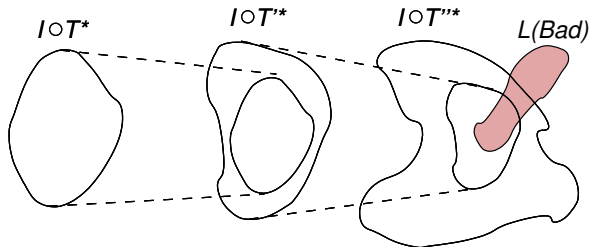
Abstractions and Approximations



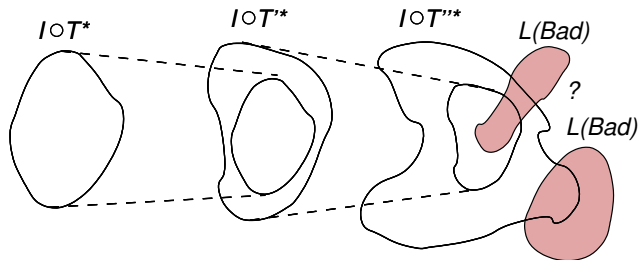
Abstractions and Approximations



Abstractions and Approximations



Abstractions and Approximations



- Uncertainties - Call for refinements

- 1 Introduction
 - Regular Model Checking
 - Abstractions and Approximations
- 2 Approach of Article
 - Computing T^*
 - Equivalence
 - Algorithm
- 3 Results
 - Time Tests
 - BDD Usage
- 4 Summary

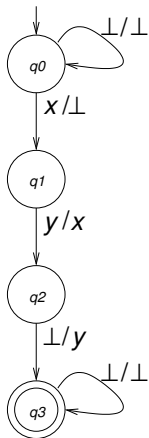
Compute T^*

- Find **one** transducer that represents T^*

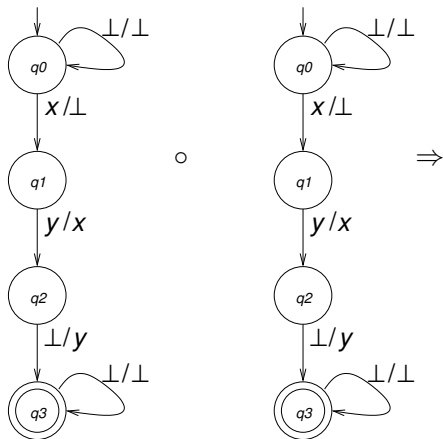
Compute T^*

- Find **one** transducer that represents T^*
- What is T^* ?

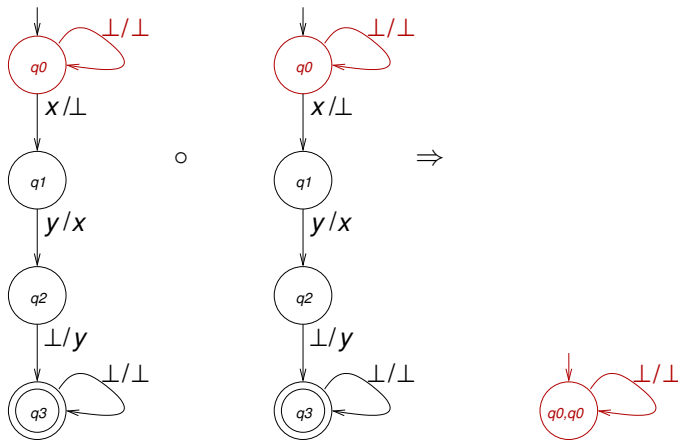
$T \circ T \circ T \circ T \dots$



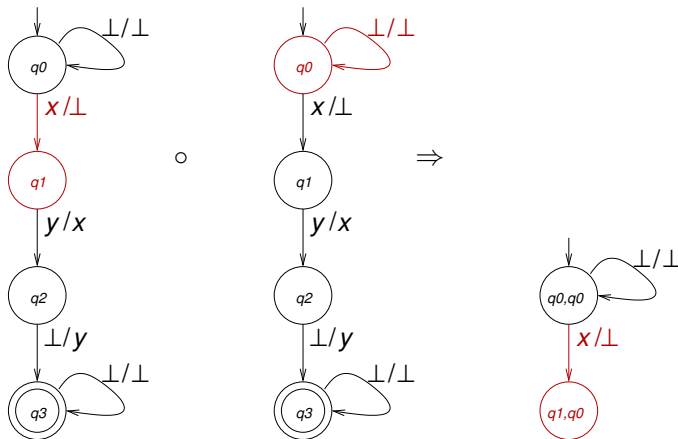
$T \circ T \circ T \circ T \dots$



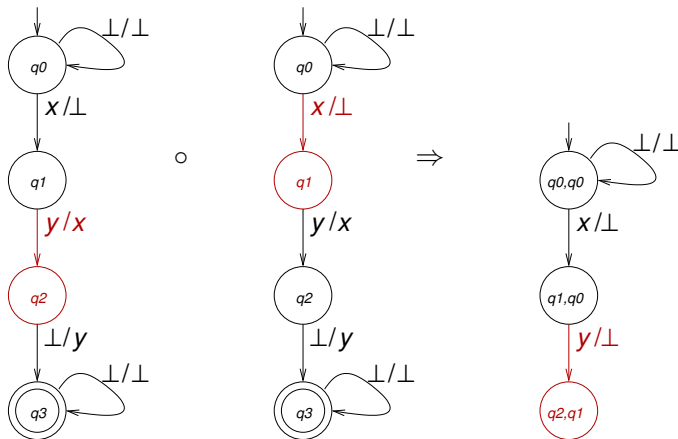
$T \circ T \circ T \circ T \dots$



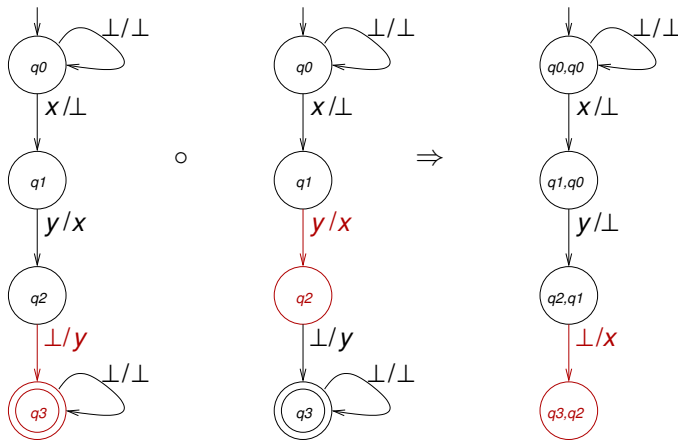
$T \circ T \circ T \circ T \dots$



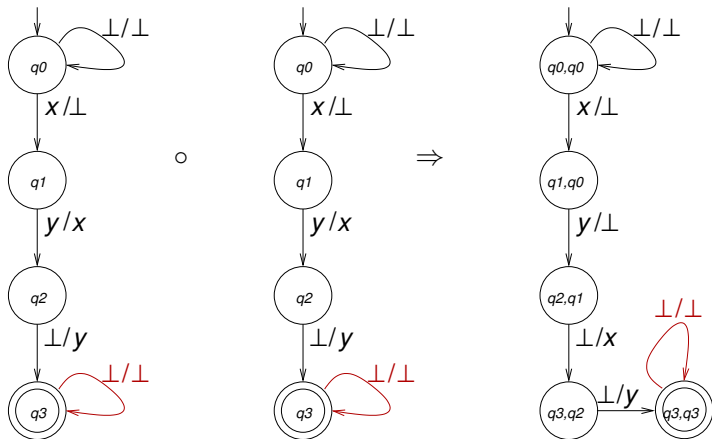
$T \circ T \circ T \circ T \dots$



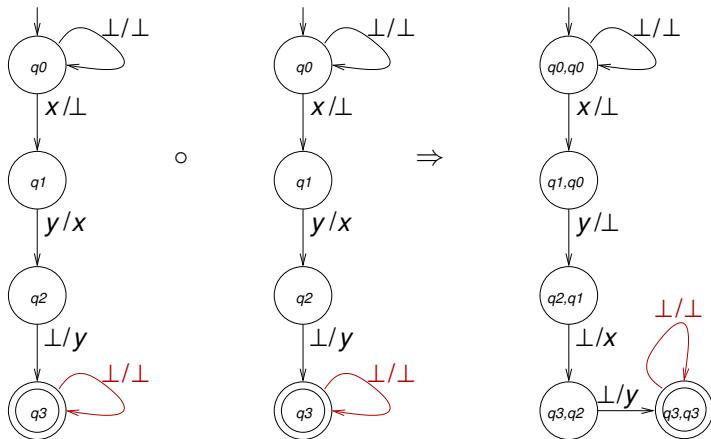
$T \circ T \circ T \circ T \dots$



$T \circ T \circ T \circ T \dots$

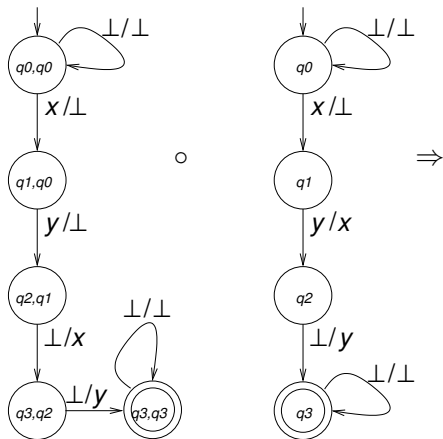


$T \circ T \circ T \circ T \dots$



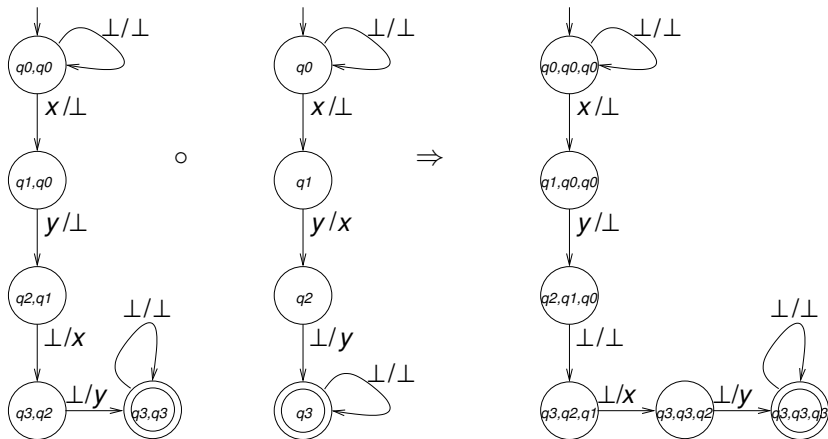
$\bullet T^* = \bigcup_{i \geq 1} T^i$

$T \circ T \circ T \circ T \dots$



$T^* = \bigcup_{i \geq 1} T^i$

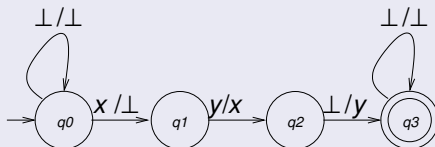
$T \circ T \circ T \circ T \dots$



$\bullet T^* = \bigcup_{i \geq 1} T^i$

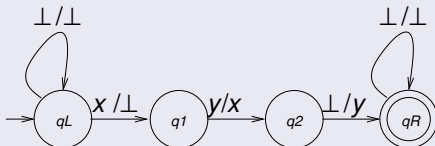
Equivalence

Pre- and Suffix Copying



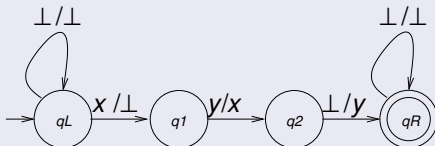
Equivalence

Pre- and Suffix Copying



Equivalence

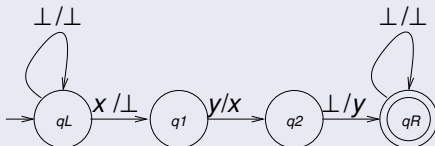
Pre- and Suffix Copying



Column Equivalence, \approx

Equivalence

Pre- and Suffix Copying

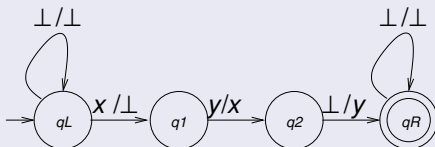


Column Equivalence, \simeq

- $qL_i, qL_j, q, qR_k \simeq qL_i, q, qR_k$, where $i = j \in \mathbb{N}$

Equivalence

Pre- and Suffix Copying

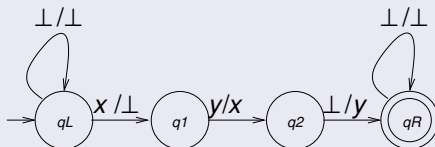


Column Equivalence, \simeq

- $qL_i, qL_j, q, qR_k \simeq qL_i, q, qR_k$, where $i = j \in \mathbb{N}$
- $L_i, q, R_k, R_\ell \simeq qL_i, q, qR_k$, where $k = \ell \in \mathbb{N}$

Equivalence

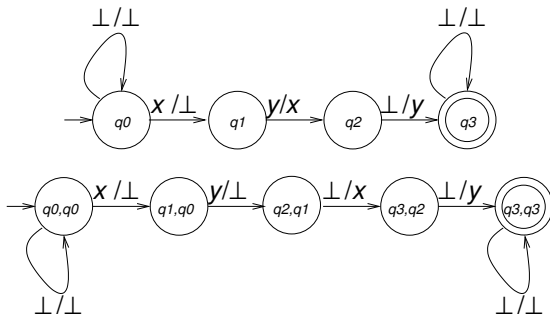
Pre- and Suffix Copying



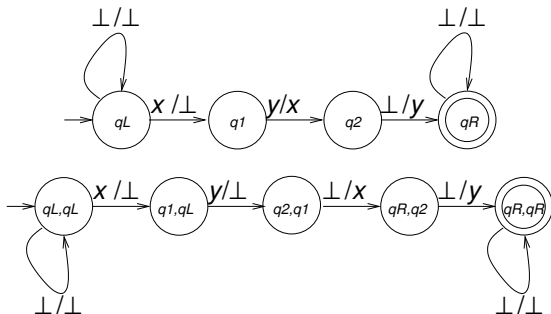
Column Equivalence, \simeq

- $qL_i, qL_j, q, qR_k \simeq qL_i, q, qR_k$, where $i = j \in \mathbb{N}$
- $L_i, q, R_k, R_\ell \simeq qL_i, q, qR_k$, where $k = \ell \in \mathbb{N}$
- hence $qL_i, qL_j, q, qR_k \simeq L_i, q, qR_k, qR_\ell$, where $i = j \in \mathbb{N}$ and $k = \ell \in \mathbb{N}$

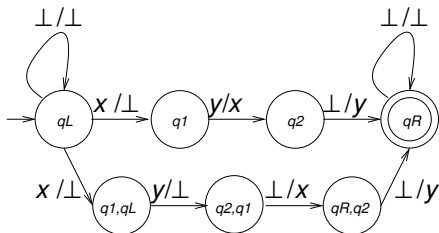
Example



Example



Example

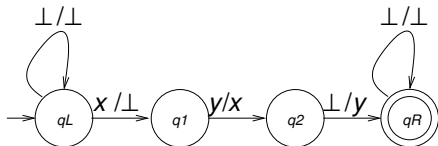
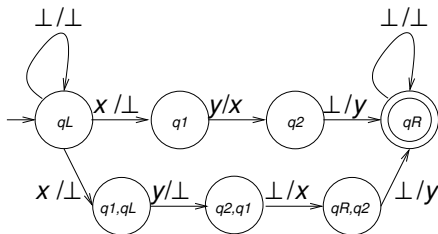


- 1 Introduction
 - Regular Model Checking
 - Abstractions and Approximations
- 2 Approach of Article
 - Computing T^*
 - Equivalence
 - Algorithm
- 3 Results
 - Time Tests
 - BDD Usage
- 4 Summary

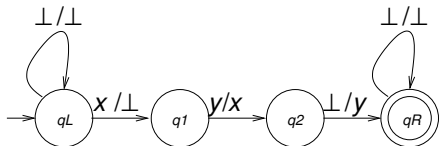
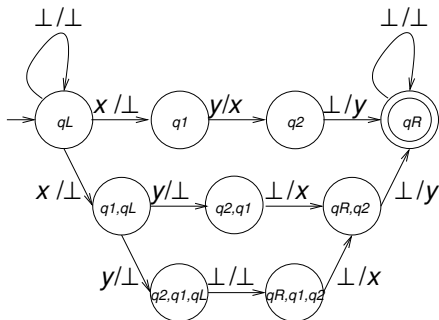
Algorithm

- 1 Initialize: Start from the transducer T and name left- and right copying states with L_i 's and R_k 's, where $i, k \in \mathbb{N}$.
- 2 Until Converging: Add new transitions and columns:
 $C_i \circ C_{i+1} \Longrightarrow^{A/C} C'_i \circ C'_{i+1}$, where
 $C_i \Longrightarrow^{A/B} C'_i$ and $C_{i+1} \Longrightarrow^{B/C} C'_{i+1}$

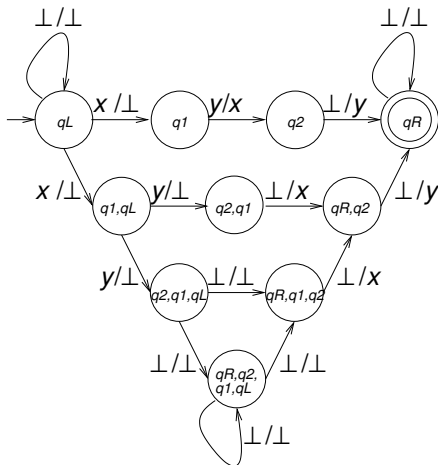
Example Finished



Example Finished

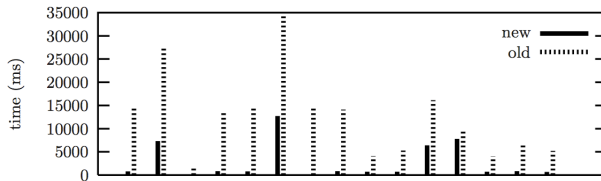


Example Finished

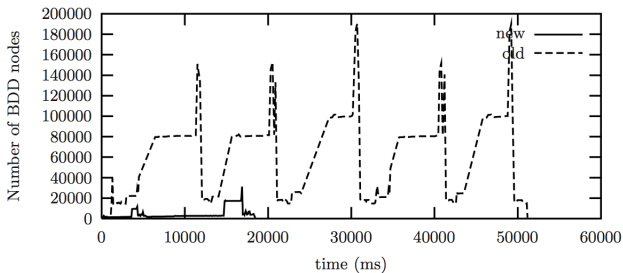


- 1 Introduction
 - Regular Model Checking
 - Abstractions and Approximations
- 2 Approach of Article
 - Computing T^*
 - Equivalence
 - Algorithm
- 3 Results
 - Time Tests
 - BDD Usage
- 4 Summary

Time Tests



BDD Usage



- 1 Introduction
 - Regular Model Checking
 - Abstractions and Approximations
- 2 Approach of Article
 - Computing T^*
 - Equivalence
 - Algorithm
- 3 Results
 - Time Tests
 - BDD Usage
- 4 Summary