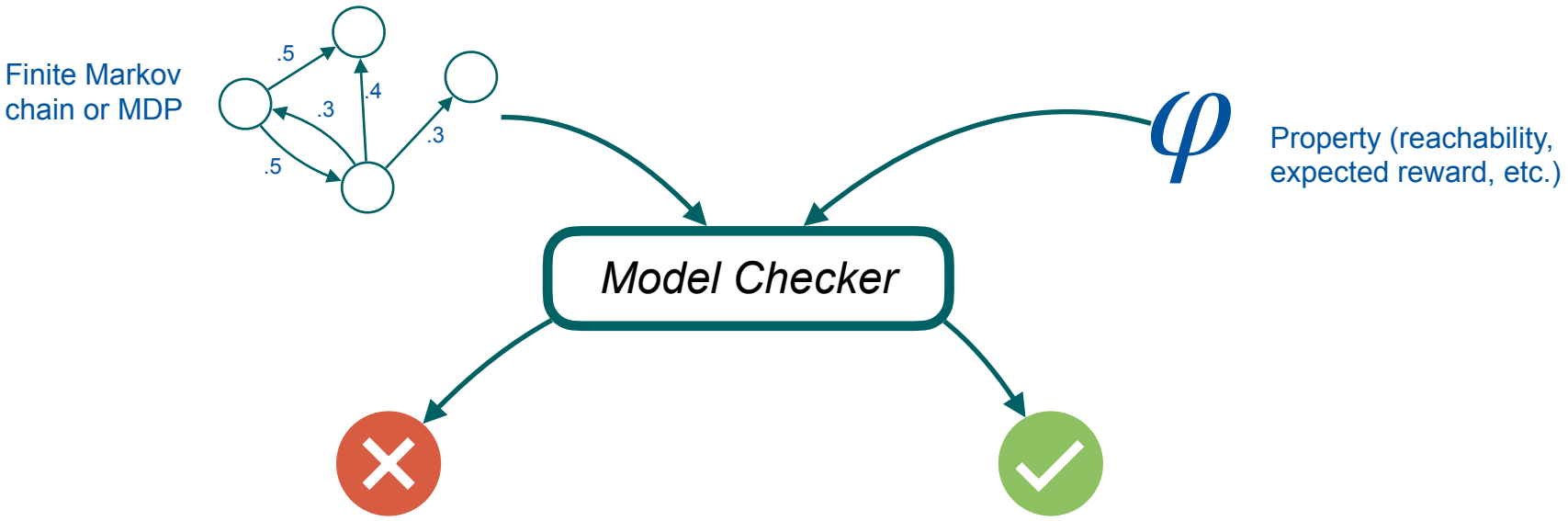
Out of Control: Reducing Probabilistic Models by Control-State Elimination

Tobias Winkler, Johannes Lehmann, Joost-Pieter Katoen

Software Modeling and Verification Chair

RWTH AACHEN UNIVERSITY

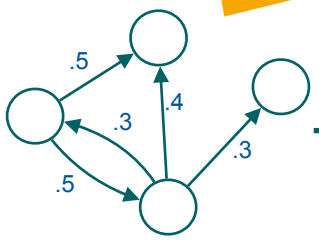
Probabilistic Model Checking in Theory



Probabilistic Model Checking in Practice

High level program description (finite variable domains)

```
1   stmc
2
3   const double p = 0.5;
4
5   module process1
6
7       x1 : [0..1];
8
9       [step]  (x1=x3) ->
10      [step] !(x1=x3) ->
11
12  endmodule
```



Finite Markov chain or MDP

φ

Property (reachability, expected reward, etc.)

Model Checker

PRISM model checker

Storm

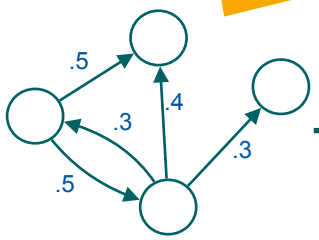
www.stormchecker.org

Probabilistic Model Checking in Practice

High level program description (finite variable domains)

```
1   stmc
2
3   const double p = 0.5;
4
5   module process1
6
7       x1 : [0..1];
8
9       [step]  (x1=x3) ->
10      [step] !(x1=x3) ->
11
12  endmodule
```

State explosion!



Finite Markov chain or MDP

φ

Property (reachability, expected reward, etc.)

Model Checker

PRISM model checker

Storm

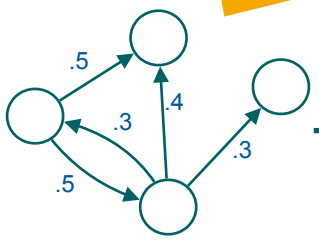
www.stormchecker.org

Probabilistic Model Checking in Practice

High level program description (finite variable domains)

```
1   stmc
2
3   const double p = 0.5;
4
5   module process1
6
7       x1 : [0..1];
8
9       [step]  (x1=x3) ->
10      [step] !(x1=x3) ->
11
12  endmodule
```

State explosion!



Finite Markov chain or MDP

φ

Property (reachability, expected reward, etc.)

Model Checker

? MO/TO

PRISM model checker

Storm
www.stormchecker.org

Probabilistic Model Checking in Practice

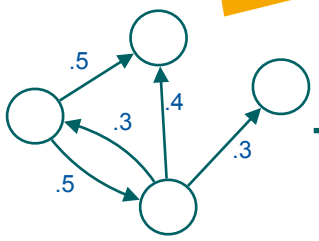
High level program description
(finite variable domains)

```
1   stmc
2
3   const double p = 0.5;
4
5   module process1
6
7       x1 : [0..1];
8
9       [step]  (x1=x3) ->
10      [step] !(x1=x3) ->
11
12  endmodule
```

State explosion!

Goal:
Mitigate state explosion through program transformations

Finite Markov chain or MDP



Property (reachability, expected reward, etc.)

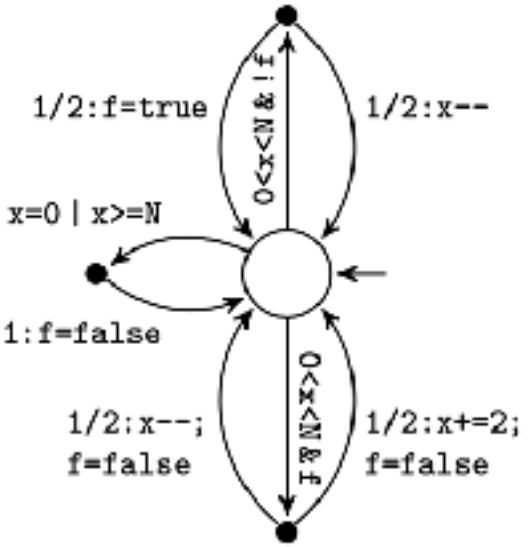
Model Checker

MO/TO

PRISM model checker

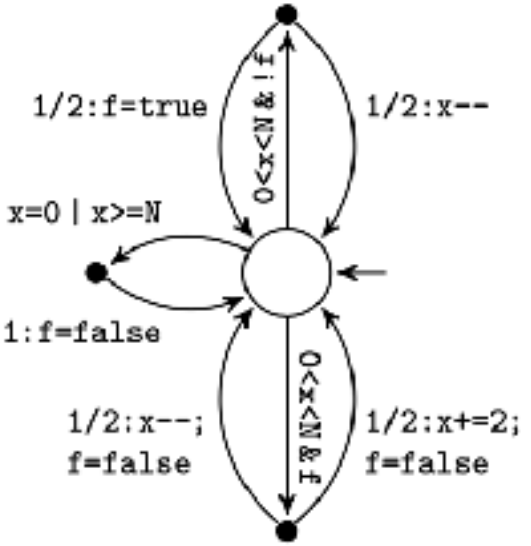
The PRISM Modelling Language and Control-Flow Graphs

```
1   dtmc
2
3   const int N = 5;
4
5   module coingame
6
7       x : [0..N+1] init N/2;
8
9       f : bool init false;
10
11      [] 0<x & x<N & !f -> 0.5 : (x'=x-1) + 0.5 : (f'=true);
12      [] 0<x & x<N & f -> 0.5 : (x'=x-1) & (f'=false) + 0.5 : (x'=x+2)  & (f'=false);
13      [] x=0 | x>=N -> true;
14
15  endmodule
```

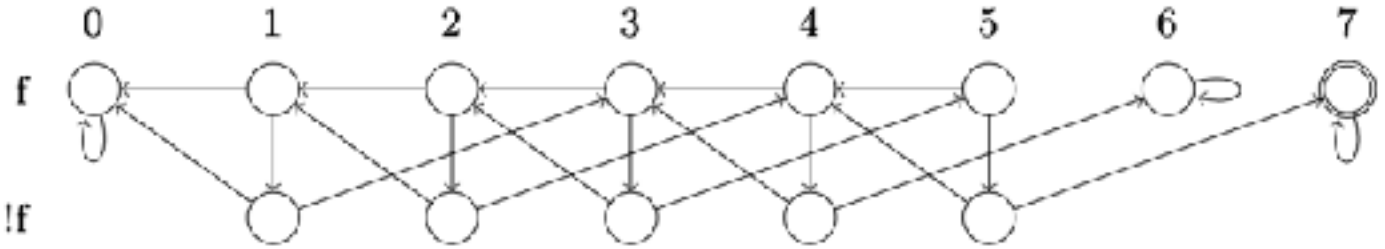


The PRISM Modelling Language and Control-Flow Graphs

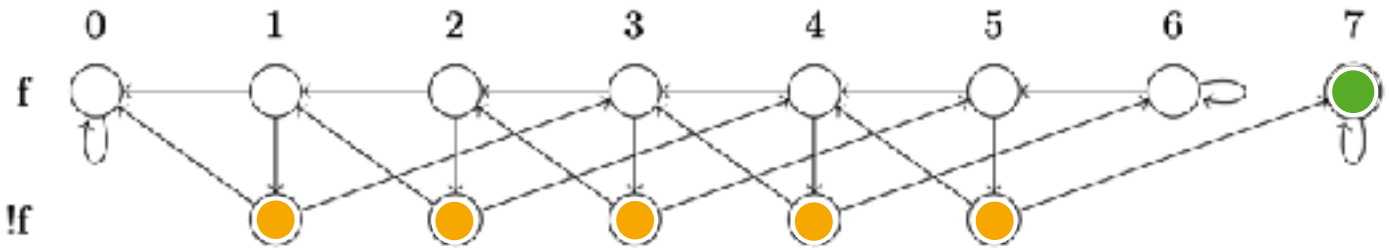
```
1   dtmc
2
3   const int N = 5;
4
5   module coingame
6
7       x : [0..N+1] init N/2;
8
9       f : bool init false;
10
11      [] 0<x & x<N & !f -> 0.5 : (x'=x-1) + 0.5 : (f'=true);
12      [] 0<x & x<N & f -> 0.5 : (x'=x-1) & (f'=false) + 0.5 : (x'=x+2)  & (f'=false);
13      [] x=0 | x>=N -> true;
14
15  endmodule
```



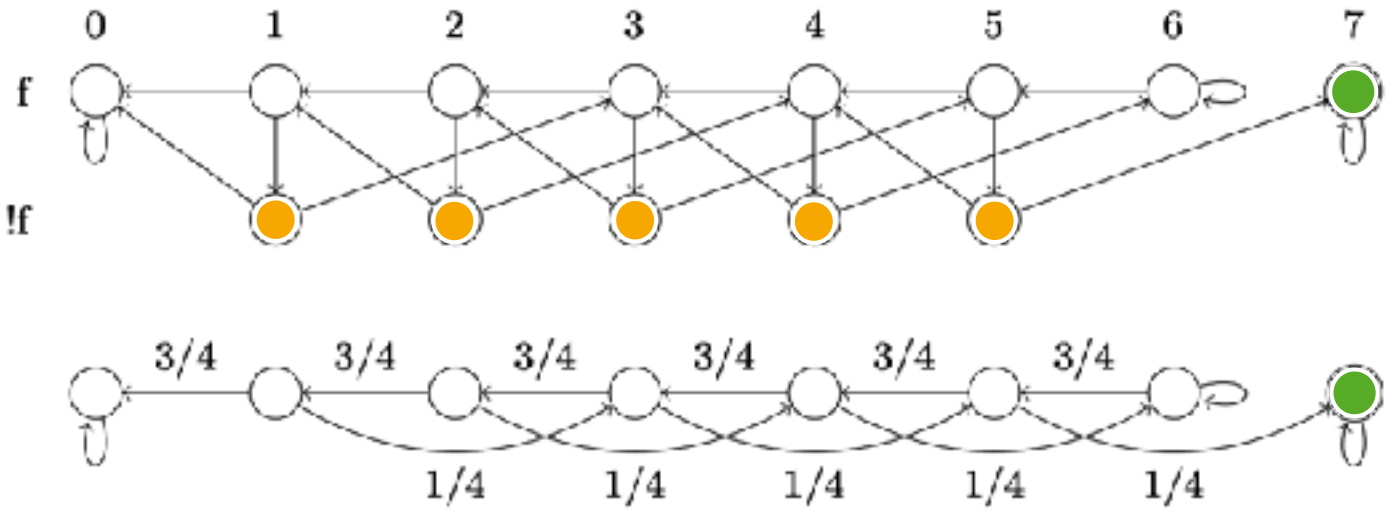
Resulting Markov chain for N=6:



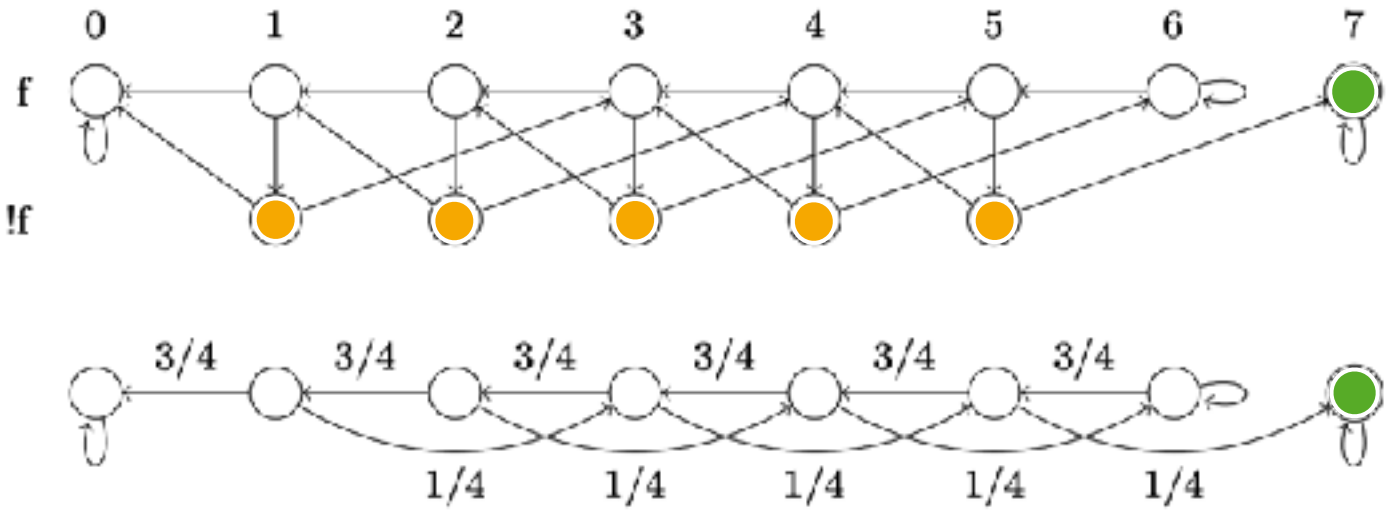
State Elimination



State Elimination

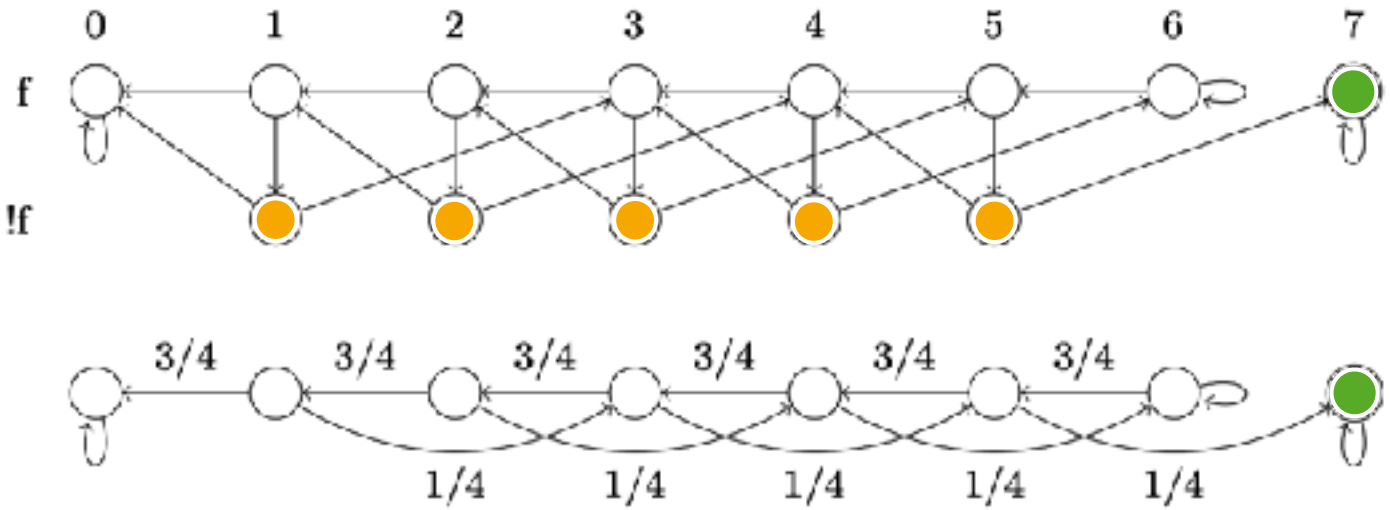


State Elimination



These Markov chains are equivalent wrt. reaching ◉ !

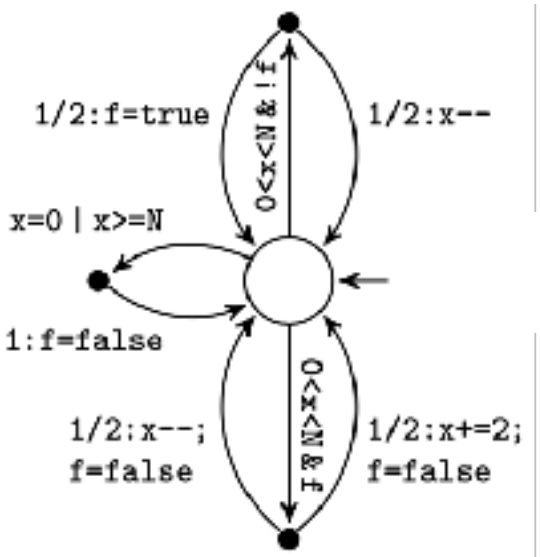
State Elimination



These Markov chains are equivalent wrt. reaching ◉ !

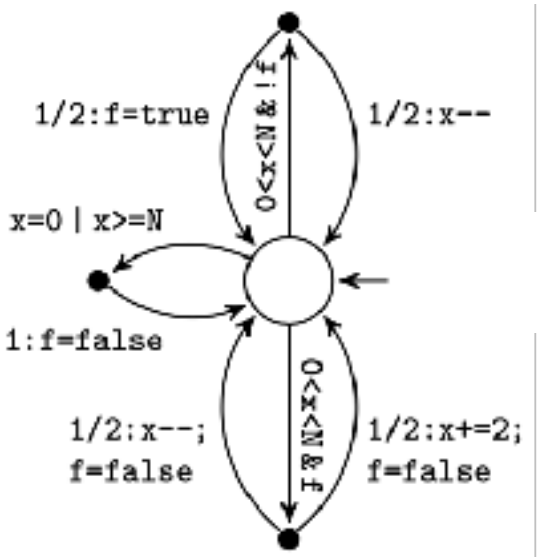
*Can we (automatically) achieve such simplifications by **manipulating the program**?*

Step 1: Unfold Variable into Control-flow Graph

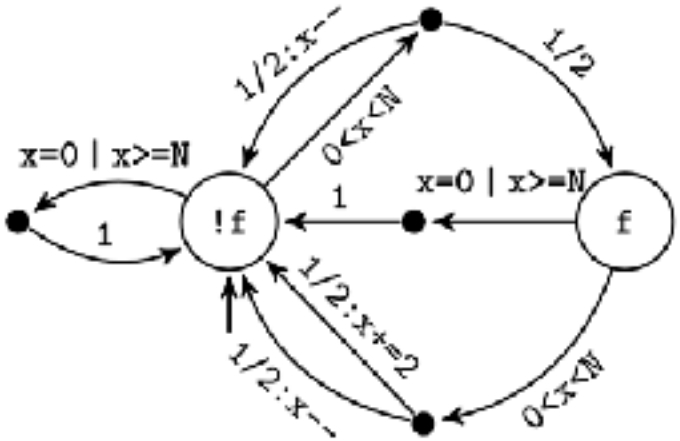


unfold `f`

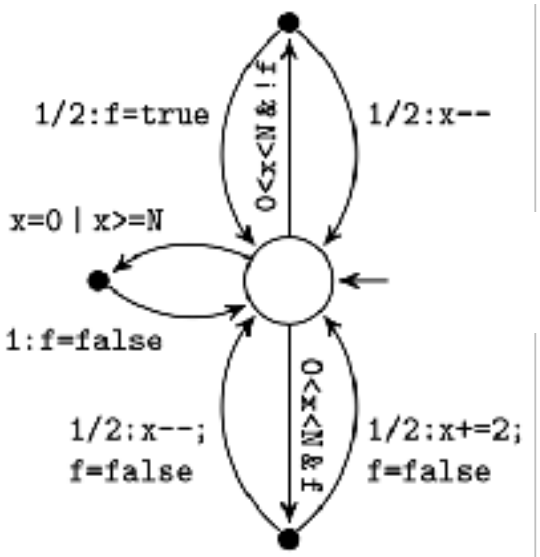
Step 1: Unfold Variable into Control-flow Graph

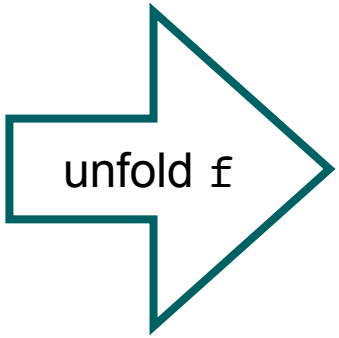


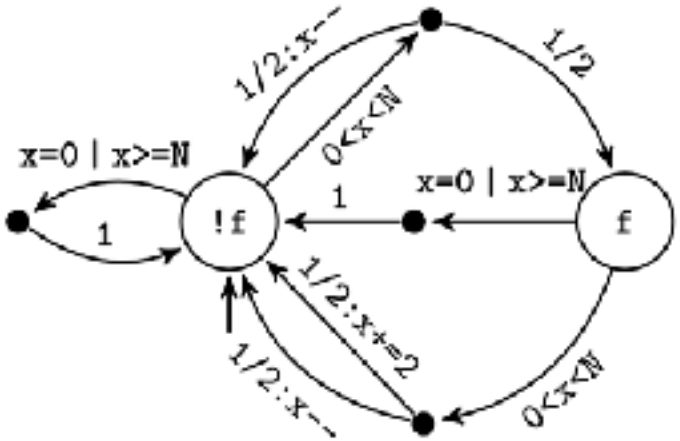
unfold `f`



Step 1: Unfold Variable into Control-flow Graph



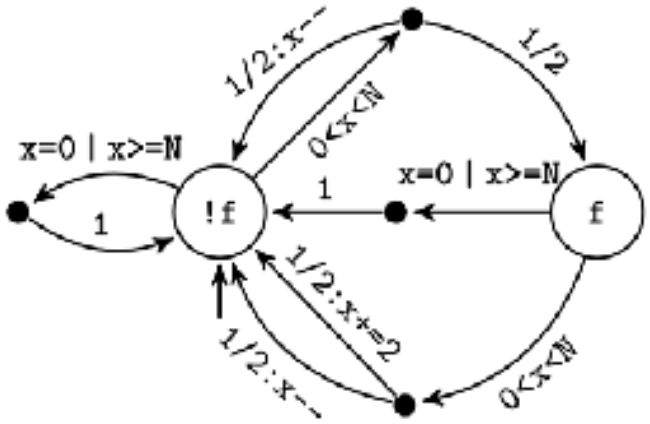


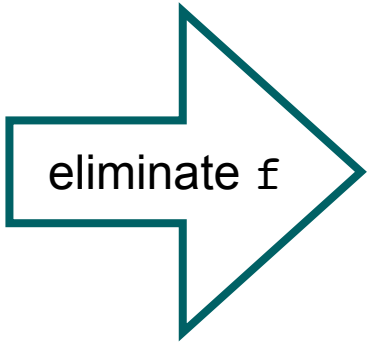


Variables cannot always be unfolded so easily

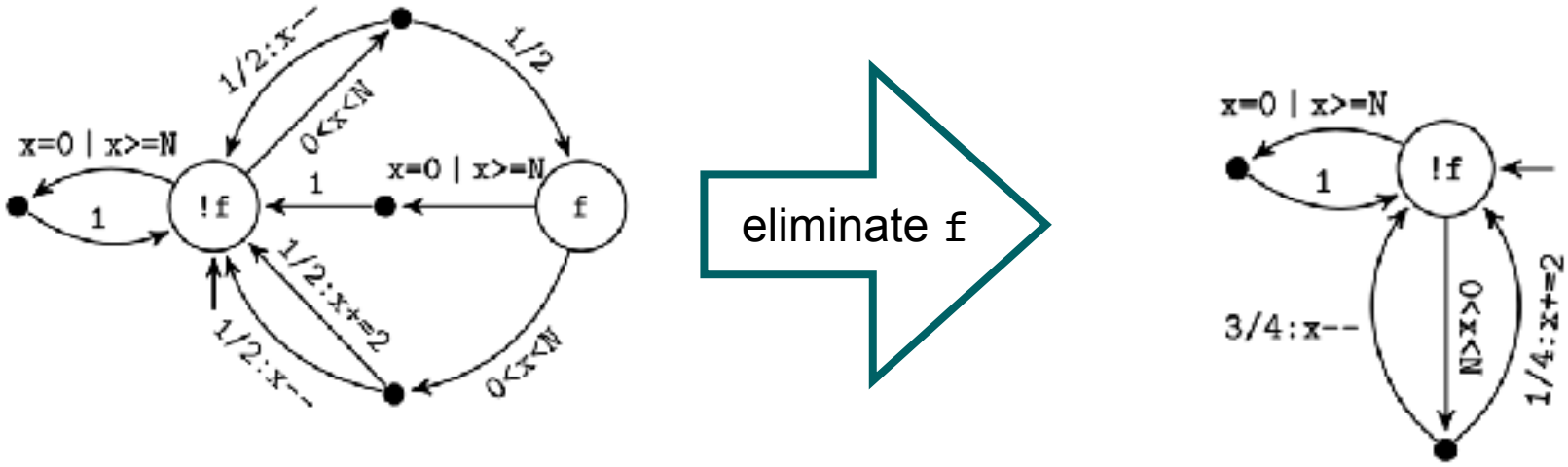
- E.g. cannot unfold `f` if assignment `f = x` occurs
  - `x` must be unfolded first
- Most real-world instances have some unfoldable variables

se

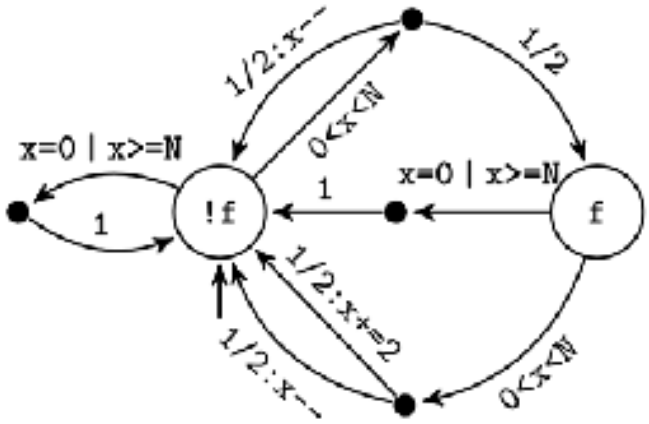




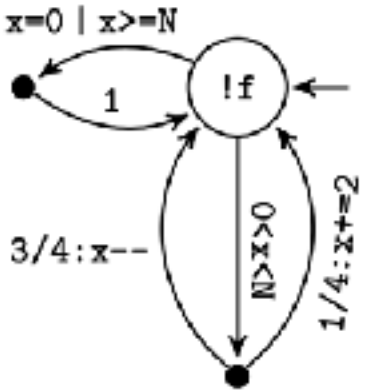
se

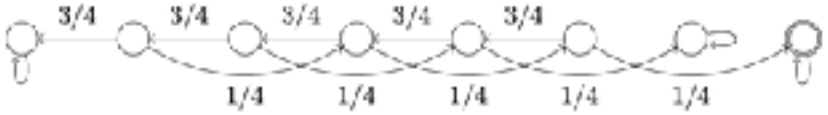

x=0 | x>=N
1
!f
1/2:x--
0<x<N
1/2
x=0 | x>=N
1
f
1/2:x+=2
1/2:x--
0<x<N
eliminate f
x=0 | x>=N
1
!f
3/4:x--
0<x<N
1/4:x+=2

se

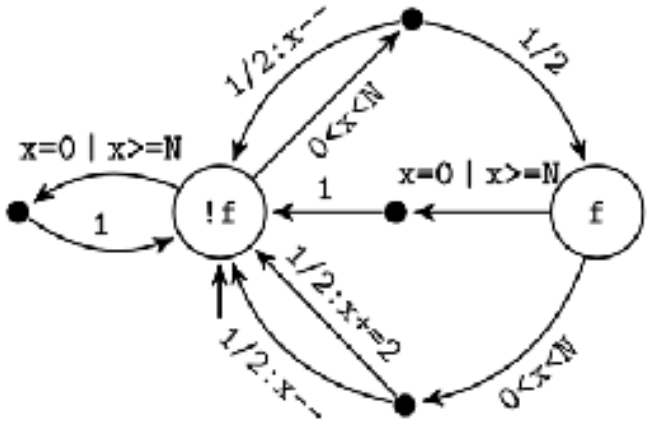


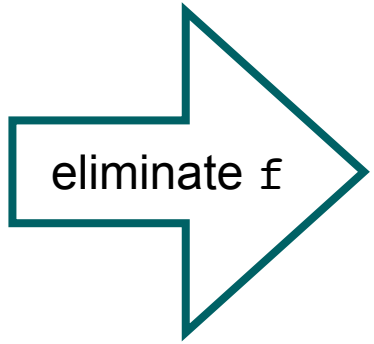
eliminate `f`

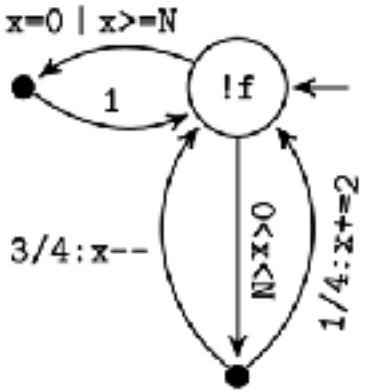


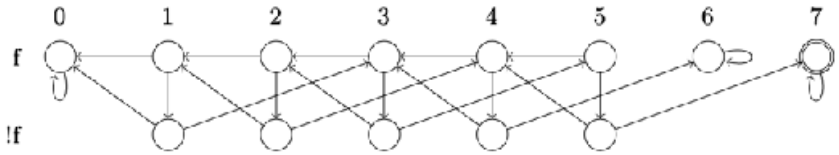


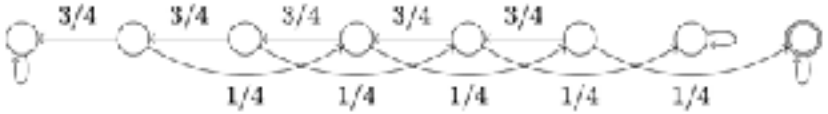
se


x=0 | x>=N
1
!f
1/2:x--
0<x<N
1/2
x=0 | x>=N
f
1
1/2:x+=2
1/2:x--
0<x<N


eliminate f

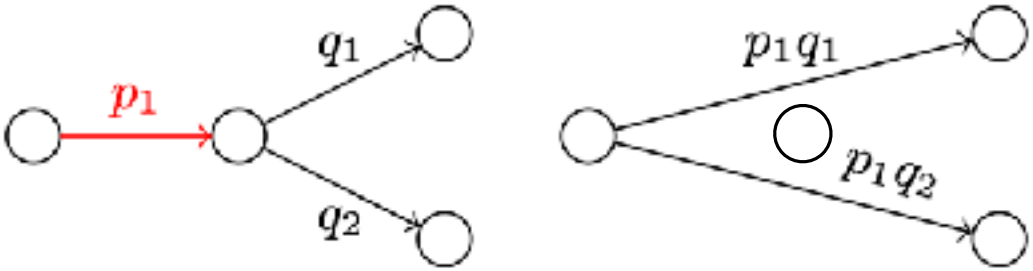

x=0 | x>=N
1
!f
3/4:x--
0<x<N
1/4:x+=2


0
1
2
3
4
5
6
7
f
!f


3/4
3/4
3/4
3/4
3/4
1/4
1/4
1/4
1/4
1/4

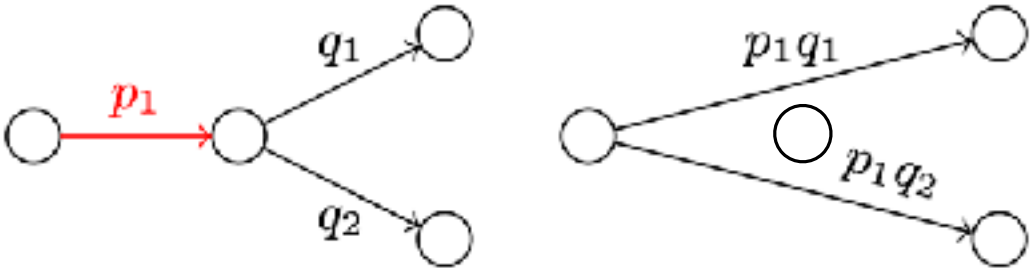
Elimination Rule

In plain Markov chains:

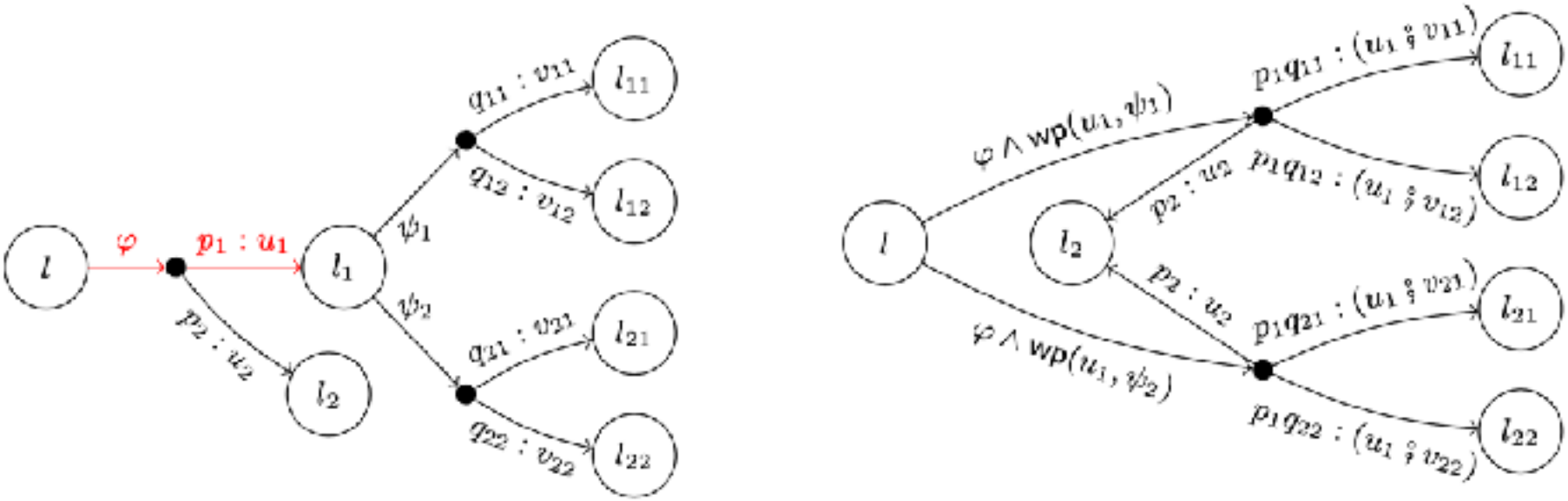


Elimination Rule

In plain Markov chains:

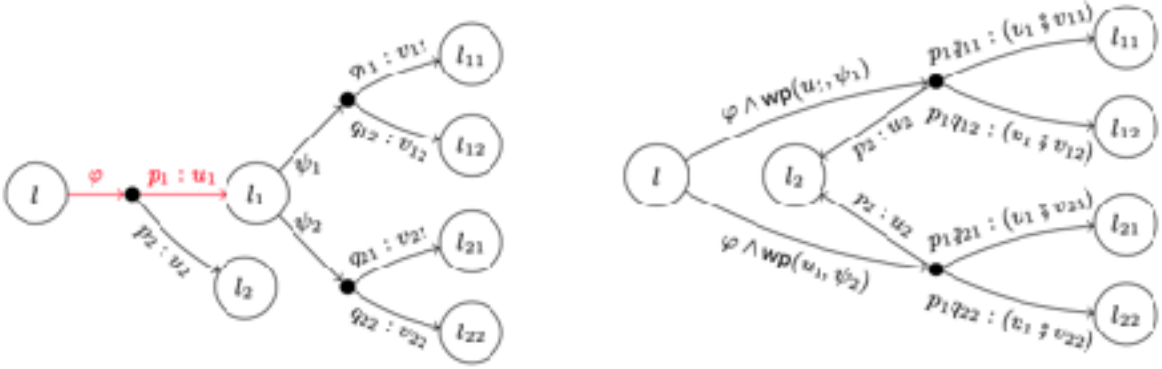


In the control-flow graph (eliminating a single transition):



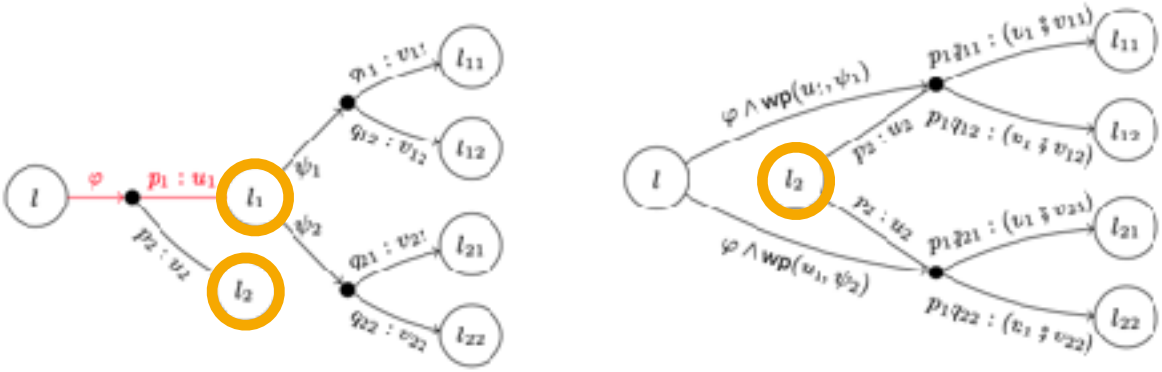
Location Elimination in general

Each location (without self-loops) can be eliminated by successively applying the transition-elimination rule to all its incoming transitions.



Location Elimination in general

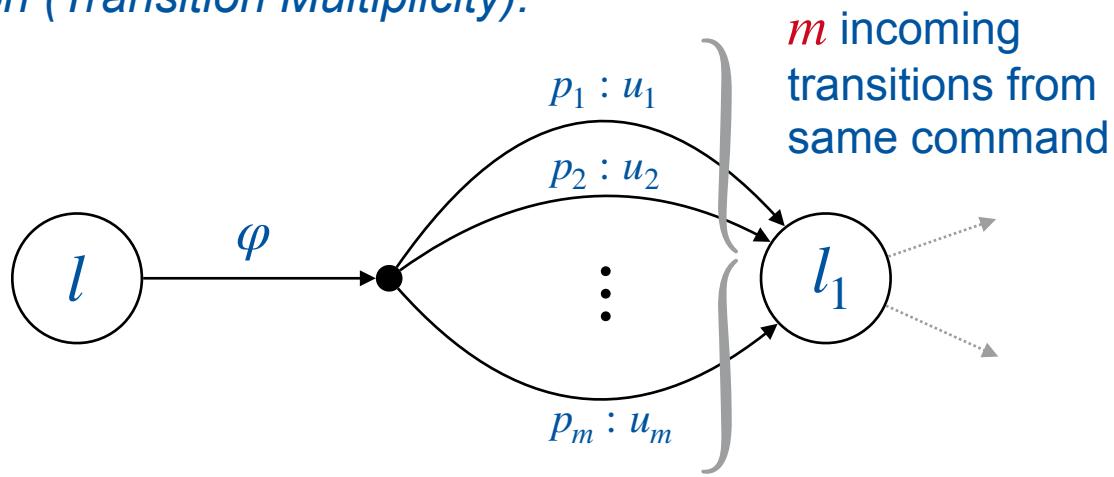
Each location (without self-loops) can be eliminated by successively applying the transition-elimination rule to all its incoming transitions.



Treat case $l_1 = l_2$ with extra care

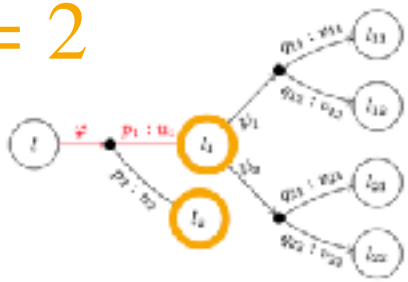
Complexity of Location Elimination

Definition (Transition Multiplicity):

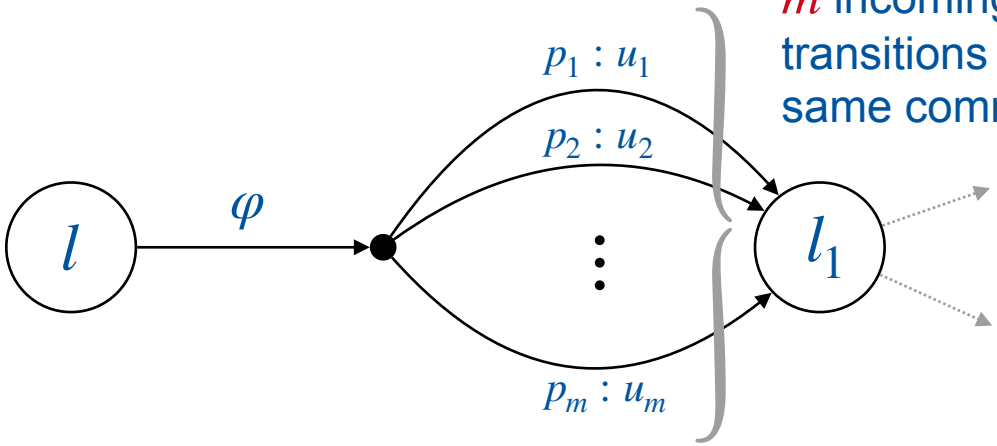


Complexity of Location Elimination

$m = 2$

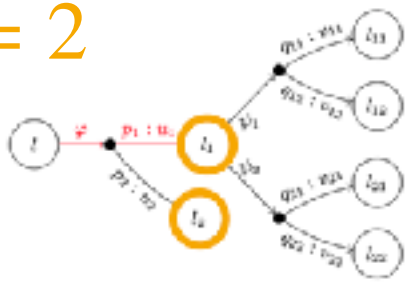


Definition (Transition Multiplicity):

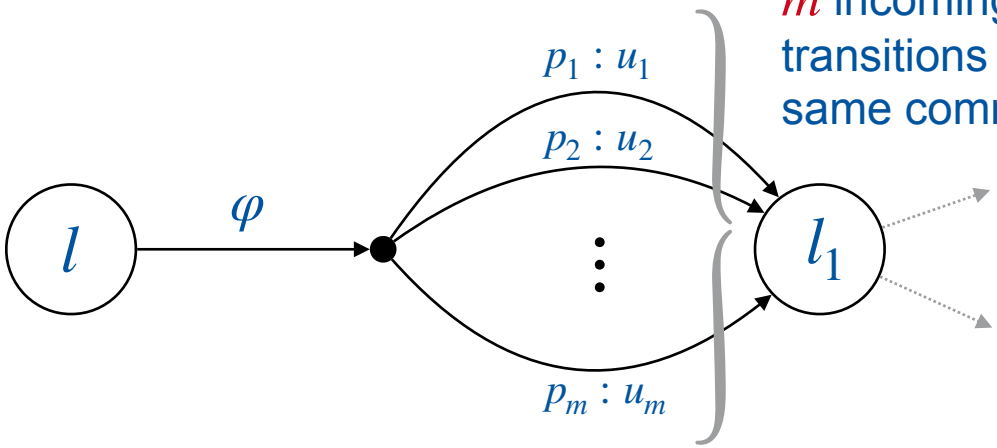


Complexity of Location Elimination

$m = 2$



Definition (Transition Multiplicity):

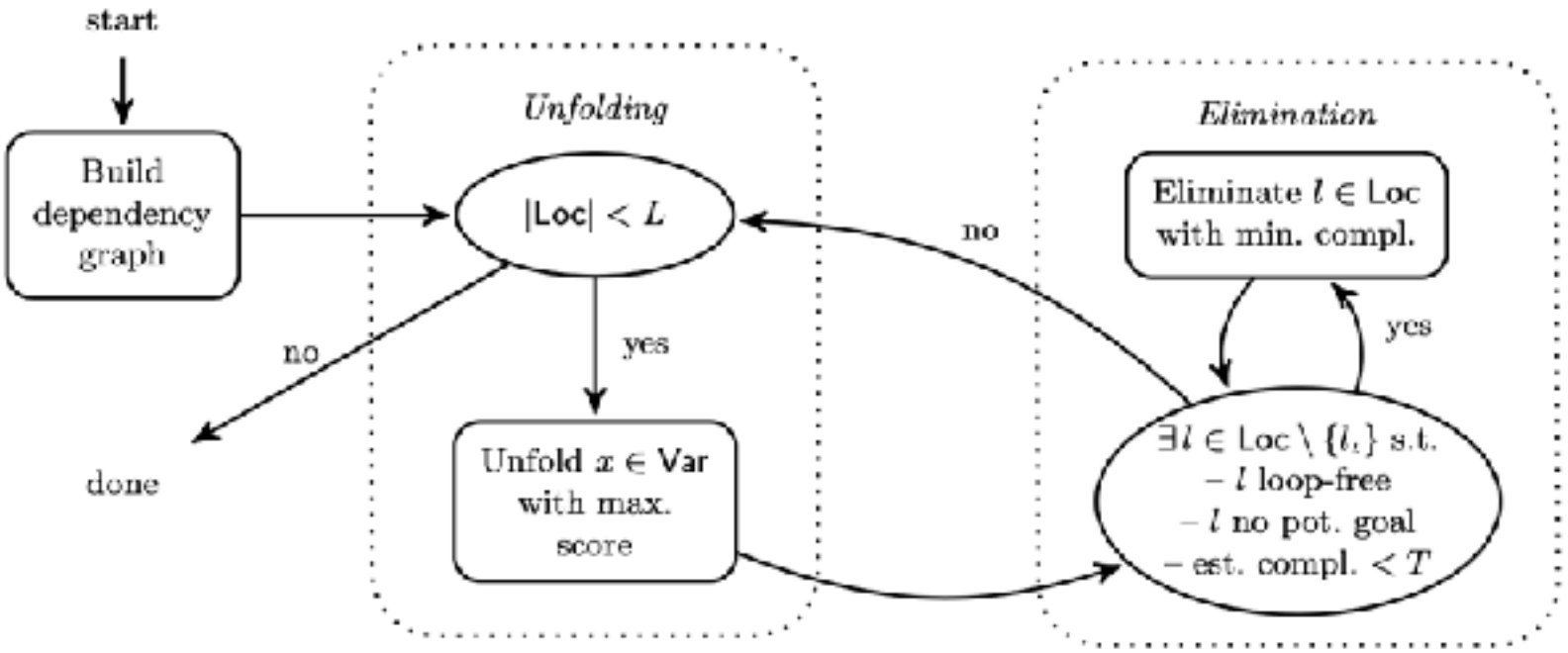


Theorem:

Exponentially many (in m) applications of transition elimination are sufficient ***and necessary*** to eliminate location l_1

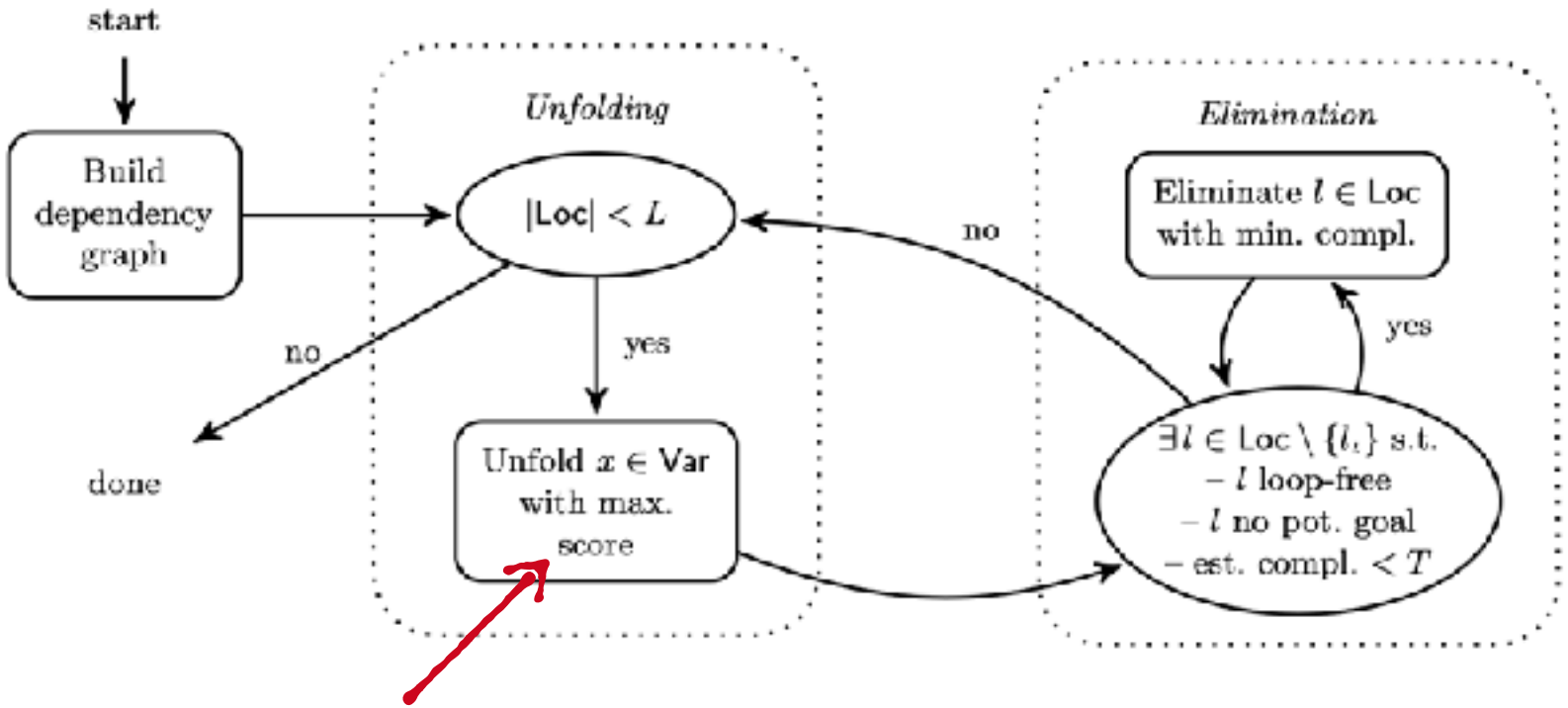
Automization

Heuristics: *Unfold a bit, eliminate reasonably*



Automization

Heuristics: ***Unfold a bit, eliminate reasonably***

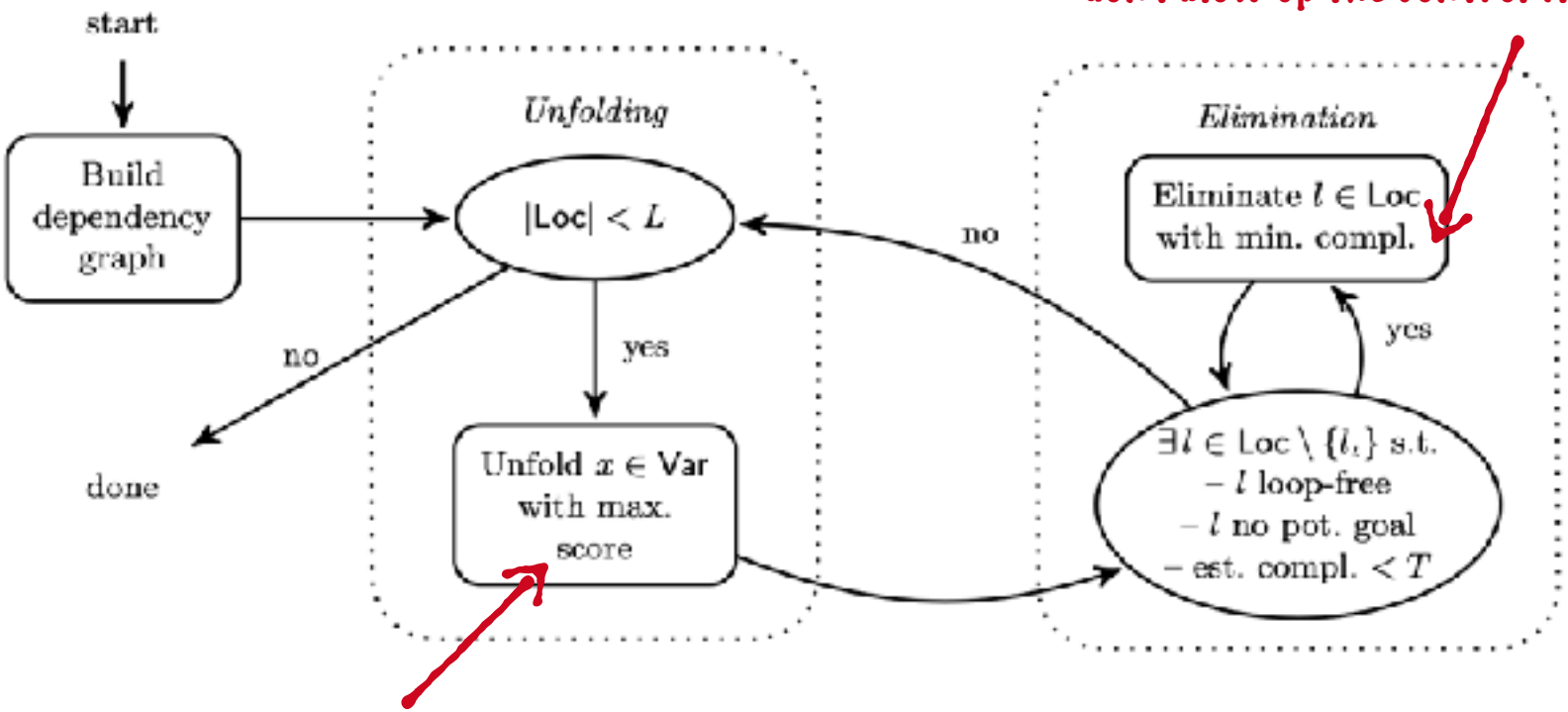


higher score = generates more self-loop free locations

Automization

Heuristics: ***Unfold a bit, eliminate reasonably***

don't blow up the control flow graph

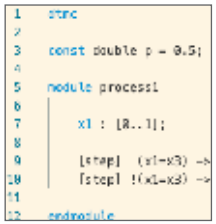


higher score = generates more self-loop free locations

Implementation

- Extension to the probabilistic model checker Storm
- Used as a simplification front end:

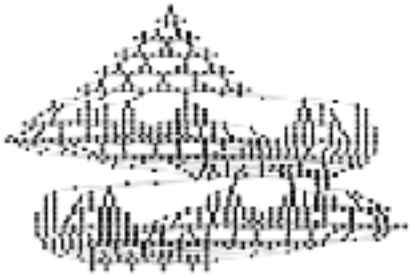
Input Program

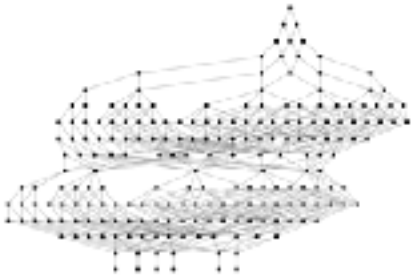


Input property

Unfolding + Elimination *until too many control-flow locations*

Full state space





Reduced State Space

Experimental Results

Name	Type	Prop. type	Red. time	Params.	States orig.	States red.	Transitions orig.	Transitions red.	Build time orig.	Build time red.	Check time orig.	Check time red.	Total time orig.	Total time red.
BRP	dtmc	P	134	$2^{10}/5$	78.9K	-44%	106K	-33%	261	-33%	22	-38%	16,418	-46%
				$2^{11}/10$	291K	-45%	397K	-33%	1,027	-39%	101	-46%		
				$2^{12}/20$	1.11M	-46%	1.53M	-33%	3,945	-48%	462	-48%		
				$2^{13}/25$	2.76M	-46%	3.8M	-33%	9,413	-47%	1,187	-47%		
COINGAME	dtmc	P	35	10^4	20K	-50%	40K	-50%	53	-24%	18,500	-79%	18,553	-78%
DICE5	mdp	P	671	n/a	371K	-84%	2.01M	-83%	1,709	-82%	9,538	-99%	11,247	-91%
EAJS	mdp	R	223	10^3	194K	-28%	326K	-1%	1,242	-43%	220	-32%	18,397	-42%
				10^4	2M	-28%	3.38M	-1%	13,154	-46%	3,780	-31%		
GRID	dtmc	P	117	10^4	300K	-47%	410K	-34%	1,062	-57%	17	-52%	11,716	-52%
				10^5	3M	-47%	4.1M	-34%	10,430	-53%	207	-54%		
HOSPITAL	mdp	P	57	n/a	160K	-66%	396K	-27%	502	-50%	19	-56%	521	-39%
NAND	dtmc	P	80	20/4	308K	-79%	476K	-52%	589	-45%	108	-75%	86,060	-56%
				40/4	4M	-80%	6.29M	-51%	8,248	-50%	1,859	-77%		
				60/2	9.42M	-80%	14.9M	-50%	19,701	-49%	4,685	-76%		
				60/4	18.8M	-80%	29.8M	-50%	40,168	-53%	10,703	-77%		
ND-NAND	mdp	P	106	20/4	308K	-79%	476K	-52%	618	-36%	127	-74%	96,956	-52%
				40/4	4M	-80%	6.29M	-51%	8,783	-42%	2,270	-77%		
				60/2	9.42M	-80%	14.9M	-50%	21,792	-47%	5,646	-75%		
				60/4	18.8M	-80%	29.8M	-50%	44,409	-46%	13,312	-76%		
NEGOTIATION	dtmc	P	148	10^4	129K	-32%	184K	-26%	481	-39%	22	-49%	5,631	-39%
				10^5	1.29M	-32%	1.84M	-26%	4,930	-43%	197	-30%		
POLE	dtmc	R	208	10^2	315K	-46%	790K	-4%	1,496	-46%	26	-42%	17,431	-45%
				10^3	3.16M	-46%	7.9M	-4%	15,503	-47%	406	-33%		

Experimental Results

Name	Type	Prop. type	Red. time	Params.	States orig.	States red.	Transitions orig.	Transitions red.	Build time orig.	Build time red.	Check time orig.	Check time red.	Total time orig.	Total time red.
BRP	dtmc	P	134	$2^{10}/5$	78.9K	-44%	106K	-33%	261	-33%	22	-38%	16,418	-46%
				$2^{11}/10$	291K	-45%	397K	-33%	1,027	-39%	101	-46%		
				$2^{12}/20$	1.11M	-46%	1.53M	-33%	3,945	-48%	462	-48%		
				$2^{13}/25$	2.76M	-46%	3.8M	-33%	9,413	-47%	1,187	-47%		
COINGAME	dtmc	P	35	10^4	20K	-50%	40K	-50%	53	-24%	18,500	-79%	18,553	-78%
DICE5	mdp	P	671	n/a	371K	-84%	2.01M	-83%	1,709	-82%	9,538	-99%	11,247	-91%
BAJS	mdp	R	223	10^3	194K	-28%	326K	-1%	1,242	-43%	220	-32%	18,397	-42%
				10^4	2M	-28%	3.38M	-1%	13,154	-46%	3,780	-31%		
GRID	dtmc	P	117	10^4	300K	-47%	410K	-34%	1,062	-57%	17	-52%	11,716	-52%
				10^5	3M	-47%	4.1M	-34%	10,430	-53%	207	-54%		
HOSPITAL	mdp	P	57	n/a	160K	-66%	396K	-27%	502	-50%	19	-56%	521	-39%
NAND	dtmc	P	80	20/4	308K	-79%	476K	-52%	589	-45%	108	-75%	86,060	-56%
				40/4	4M	-80%	6.29M	-51%	8,248	-50%	1,859	-77%		
				60/2	9.42M	-80%	14.9M	-50%	19,701	-49%	4,685	-76%		
				60/4	18.8M	-80%	29.8M	-50%	40,168	-53%	10,703	-77%		
ND-NAND	mdp	P	106	20/4	308K	-79%	476K	-52%	618	-36%	127	-74%	96,956	-52%
				40/4	4M	-80%	6.29M	-51%	8,783	-42%	2,270	-77%		
				60/2	9.42M	-80%	14.9M	-50%	21,792	-47%	5,646	-75%		
				60/4	18.8M	-80%	29.8M	-50%	44,409	-46%	13,312	-76%		
NEGOTIATION	dtmc	P	148	10^4	129K	-32%	184K	-26%	481	-39%	22	-49%	5,631	-39%
				10^5	1.29M	-32%	1.84M	-26%	4,930	-43%	197	-30%		
POLE	dtmc	R	208	10^2	315K	-46%	790K	-4%	1,496	-46%	26	-42%	17,431	-45%
				10^3	3.16M	-46%	7.9M	-4%	15,503	-47%	406	-33%		

Experimental Results

Name	Type	Prop. type	Red. time	Params.	States orig.	States red.	Transitions orig.	Transitions red.	Build time orig.	Build time red.	Check time orig.	Check time red.	Total time orig.	Total time red.
BRP	dtmc	P	134	$2^{10}/5$	78.9K	-44%	106K	-33%	261	-33%	22	-38%	16,418	-46%
				$2^{11}/10$	291K	-45%	397K	-33%	1,027	-39%	101	-46%		
				$2^{12}/20$	1.11M	-46%	1.53M	-33%	3,945	-48%	462	-48%		
				$2^{13}/25$	2.76M	-46%	3.8M	-33%	9,413	-47%	1,187	-47%		
COINGAME	dtmc	P	35	10^4	20K	-50%	40K	-50%	53	-24%	18,500	-79%	18,553	-78%
DICE5	mdp	P	671	n/a	371K	-84%	2.01M	-83%	1,709	-82%	9,538	-99%	11,247	-91%
BAJS	mdp	R	223	10^3	194K	-28%	326K	-1%	1,242	-43%	220	-32%	18,397	-42%
				10^4	2M	-28%	3.38M	-1%	13,154	-46%	3,780	-31%		
GRID	dtmc	P	117	10^4	300K	-47%	410K	-34%	1,062	-57%	17	-52%	11,716	-52%
				10^5	3M	-47%	4.1M	-34%	10,430	-53%	207	-54%		
HOSPITAL	mdp	P	57	n/a	160K	-66%	396K	-27%	502	-50%	19	-56%	521	-39%
NAND	dtmc	P	80	20/4	308K	-79%	476K	-52%	589	-45%	108	-75%	86,060	-56%
				40/4	4M	-80%	6.29M	-51%	8,248	-50%	1,859	-77%		
				60/2	9.42M	-80%	14.9M	-50%	19,701	-49%	4,685	-76%		
				60/4	18.8M	-80%	29.8M	-50%	40,168	-53%	10,703	-77%		
ND-NAND	mdp	P	106	20/4	308K	-79%	476K	-52%	618	-36%	127	-74%	96,956	-52%
				40/4	4M	-80%	6.29M	-51%	8,783	-42%	2,270	-77%		
				60/2	9.42M	-80%	14.9M	-50%	21,792	-47%	5,646	-75%		
				60/4	18.8M	-80%	29.8M	-50%	44,409	-46%	13,312	-76%		
NEGOTIATION	dtmc	P	148	10^4	129K	-32%	184K	-26%	481	-39%	22	-49%	5,631	-39%
				10^5	1.29M	-32%	1.84M	-26%	4,930	-43%	197	-30%		
POLE	dtmc	R	208	10^2	315K	-46%	790K	-4%	1,496	-46%	26	-42%	17,431	-45%
				10^3	3.16M	-46%	7.9M	-4%	15,503	-47%	406	-33%		

Control-Flow Reduction and Bisimulation Minimization

Name	Params.	States		
		Bisim.	CFR	both
BRP	$2^{12}/20$	598K	606K	344K
NAND	40/4	3.21M	816K	678K
POLE	10^3	4.06K	1.72M	1.2K

Control-Flow Reduction and Bisimulation Minimization

Name	Params.	States		
		Bisim.	CFR	both
BRP	$2^{12}/20$	598K	606K	344K
NAND	40/4	3.21M	816K	678K
POLE	10^3	4.06K	1.72M	1.2K

Are orthogonal and can be combined!

Take-Home Messages

In probabilistic model checking …

Take-Home Messages

In probabilistic model checking …

1) Mechanizable program transformations can reduce the state space

Take-Home Messages

In probabilistic model checking …

1) Mechanizable program transformations can reduce the state space

2) There are "symmetries" beyond bisimulation

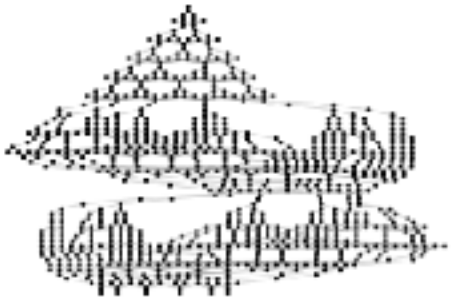
Take-Home Messages

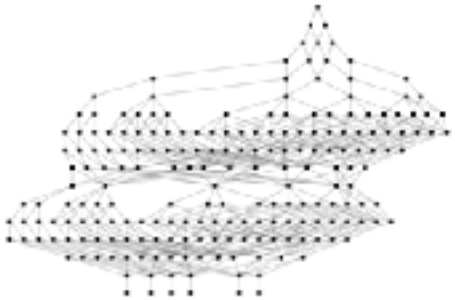
In probabilistic model checking …

1) Mechanizable program transformations can reduce the state space

2) There are "symmetries" beyond bisimulation

Thank you!





Take-Home Messages

In probabilistic model checking …

1) Mechanizable program transformations can reduce the state space

2) There are "symmetries" beyond bisimulation

Thank you!