Synthesis of asynchronous reactive programs from temporal specifications

Suguman Bansal, Rice University

Kedar S. Namjoshi, Nokia Bell Labs - Murray Hill

Yaniv Sa'ar, Nokia Bell Labs - Kfar Saba

CAV 2018 @ FLoC 2018, Oxford, UK

July 14, 2018

Hotel reservation

Hotel reservation

Designing asynchronous systems is hard

Partial/Outdated information

Unpredictable delays

Specifying intent of a program is easier

Can we automatically generate an asynchronous program from its specification?

Can we automatically generate an asynchronous program from its specification?

Asynchronous synthesis

Can we automatically generate an asynchronous program from its specification written in LTL?

Asynchronous LTL synthesis

2EXPTIME-complete [Pnueli, Rosner; ICALP 1989]

But no tools!

Contributions

Towards building tools/techniques for asynchronous LTL synthesis

Contributions

Towards building tools/techniques for asynchronous LTL synthesis

Simpler theoretical analysis

Contributions

Towards building tools/techniques for asynchronous LTL synthesis

Simpler theoretical analysis

First tool for asynchronous synthesis for all of LTL

Existing tools restricted to specific LTL formula [Klein, Piterman, Pnueli; VMCAI 2012]

Direct translation to SAT causes blowup [Schewe, Finkbeiner; STTT 2013]

Contributions

Towards building tools/techniques for asynchronous LTL synthesis

Simpler theoretical analysis

First tool for asynchronous synthesis for all of LTL

Existing tools restricted to specific LTL formula [Klein, Piterman, Pnueli; VMCAI 2012]

Direct translation to SAT causes blowup [Schewe, Finkbeiner; STTT 2013]

Identify LTL subclasses with lower complexity

Asynchronous systems

[Pnueli, Rosner; ICALP 1989]

Asynchronous system (denoted by f)

- Reactive system
- Receives input sequence **I**, observes a sub-sequence **X** only
- Generates output sequence **O** based on **X** only
- Outputs generated with (non-fixed) delay

Asynchronous systems

[Pnueli, Rosner; ICALP 1989]

Asynchronous system (denoted by f)

- Reactive system
- Receives input sequence **I**, observes a sub-sequence **X** only
- Generates output sequence **O** based on **X** only
- Outputs generated with (non-fixed) delay

Input (I)	T								
Output (O)	-								

Asynchronous systems

[Pnueli, Rosner; ICALP 1989]

Asynchronous system (denoted by f)

- Reactive system
- Receives input sequence **I**, observes a sub-sequence **X** only
- Generates output sequence **O** based on **X** only
- Outputs generated with (non-fixed) delay

Input (I)	T	T							
Output (O)	-	-							

Asynchronous systems

[Pnueli, Rosner; ICALP 1989]

Asynchronous system (denoted by f)

- Reactive system
- Receives input sequence **I**, observes a sub-sequence **X** only
- Generates output sequence **O** based on **X** only
- Outputs generated with (non-fixed) delay

Input (I)	T	T	F						
Output (O)	-	-	-						

Asynchronous systems

[Pnueli, Rosner; ICALP 1989]

Asynchronous system (denoted by f)

- Reactive system
- Receives input sequence **I**, observes a sub-sequence **X** only
- Generates output sequence **O** based on **X** only
- Outputs generated with (non-fixed) delay

Input (I)	T	T	F	T					
Output (O)	-	-	-	-					

Asynchronous systems

[Pnueli, Rosner; ICALP 1989]

Asynchronous system (denoted by f)

- Reactive system
- Receives input sequence **I**, observes a sub-sequence **X** only
- Generates output sequence **O** based on **X** only
- Outputs generated with (non-fixed) delay

Input (I)	T	T	F	T	T				
Output (O)	-	-	-	-	F				

Asynchronous systems

[Pnueli, Rosner; ICALP 1989]

Asynchronous system (denoted by f)

- Reactive system
- Receives input sequence **I**, observes a sub-sequence **X** only
- Generates output sequence **O** based on **X** only
- Outputs generated with (non-fixed) delay

Input (I)	T	T	F	T	T	T			
Output (O)	-	-	-	-	F	F			

Asynchronous systems

[Pnueli, Rosner; ICALP 1989]

Asynchronous system (denoted by f)

- Reactive system
- Receives input sequence **I**, observes a sub-sequence **X** only
- Generates output sequence **O** based on **X** only
- Outputs generated with (non-fixed) delay

Input (I)	T	T	F	T	T	T	T		
Output (O)	-	-	-	-	F	F	F		

Asynchronous systems

[Pnueli, Rosner; ICALP 1989]

Asynchronous system (denoted by f)

- Reactive system
- Receives input sequence **I**, observes a sub-sequence **X** only
- Generates output sequence **O** based on **X** only
- Outputs generated with (non-fixed) delay

Input (I)	T	T	F	T	T	T	T	F	…
Output (O)	-	-	-	-	F	F	F	T	…

Asynchronous LTL synthesis

[Pnueli, Rosner; ICALP 1989]

Given LTL specification **S,**

Generate asynch. system f s.t.

(I,O) satisfies **S** for all **I** and **X**

Challenge I: Partial information

Asych. system satisfies specification despite not reading all inputs

Challenge I: Partial information

Asych. system satisfies specification despite not reading all inputs

Example: Always (i → Eventually o)

✗ f returns the last observed value

Input (I)	T	T	F	T	F	F	T	…
Output (O)	-	-	-	F	F	F	F	…

Challenge I: Partial information

Asych. system satisfies specification despite not reading all inputs

Example: Always (i → Eventually o)

✗ f returns the last observed value

✓ f always returns True

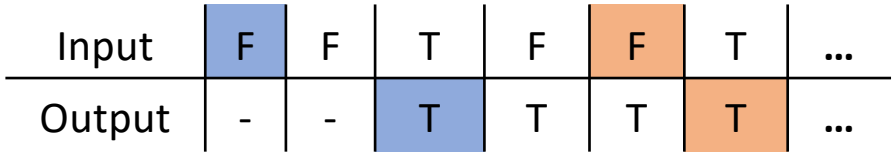
Input (I)	T	T	F	T	F	F	T	…
Output (O)	-	-	-	T	T	T	T	…

Challenge II: Non-deterministic delays

Between observing input and generating output

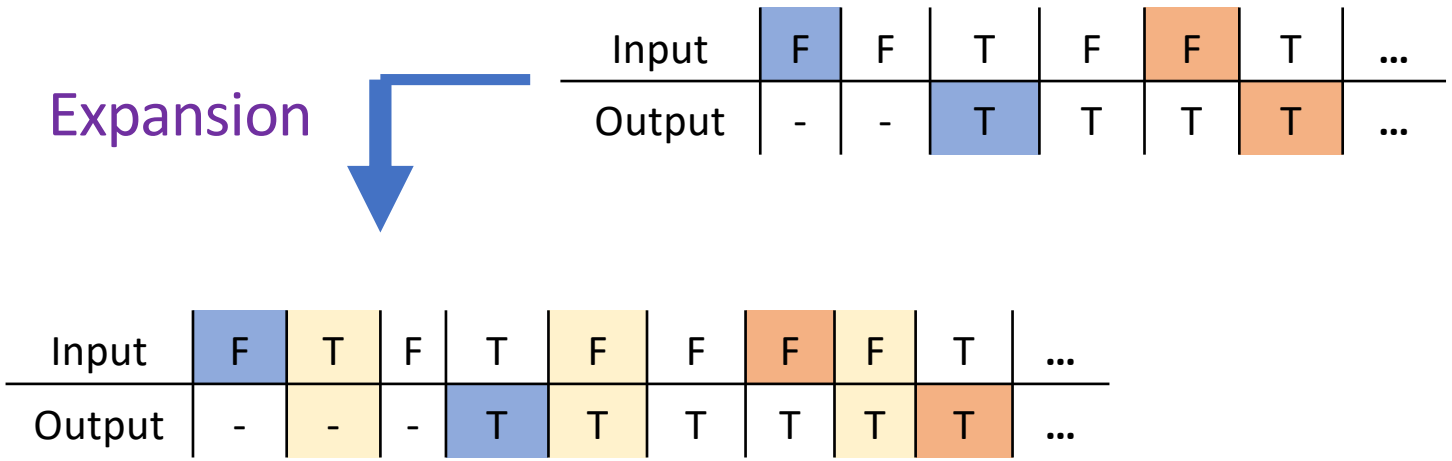
Challenge II: Non-deterministic delays

Between observing input and generating output



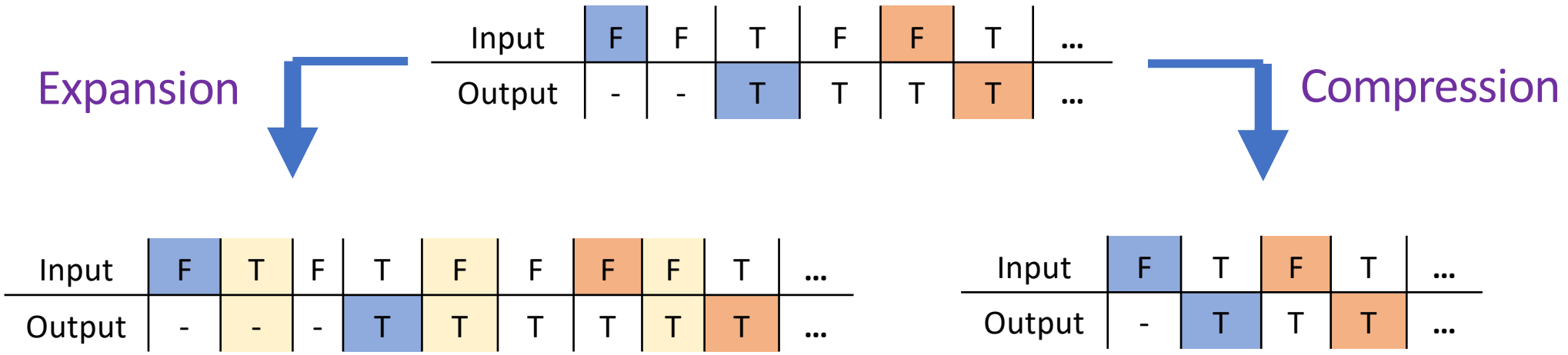
Challenge II: Non-deterministic delays

Between observing input and generating output



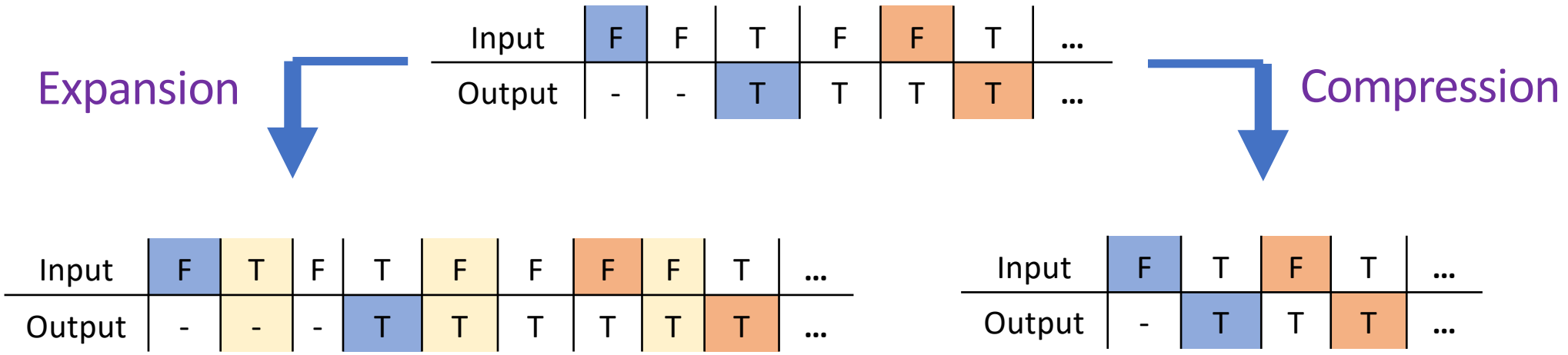
Challenge II: Non-deterministic delays

Between observing input and generating output



Challenge II: Non-deterministic delays

Between observing input and generating output



(I,O) satisfies **S** iff all expansions and compressions of (I,O) satisfy **S**

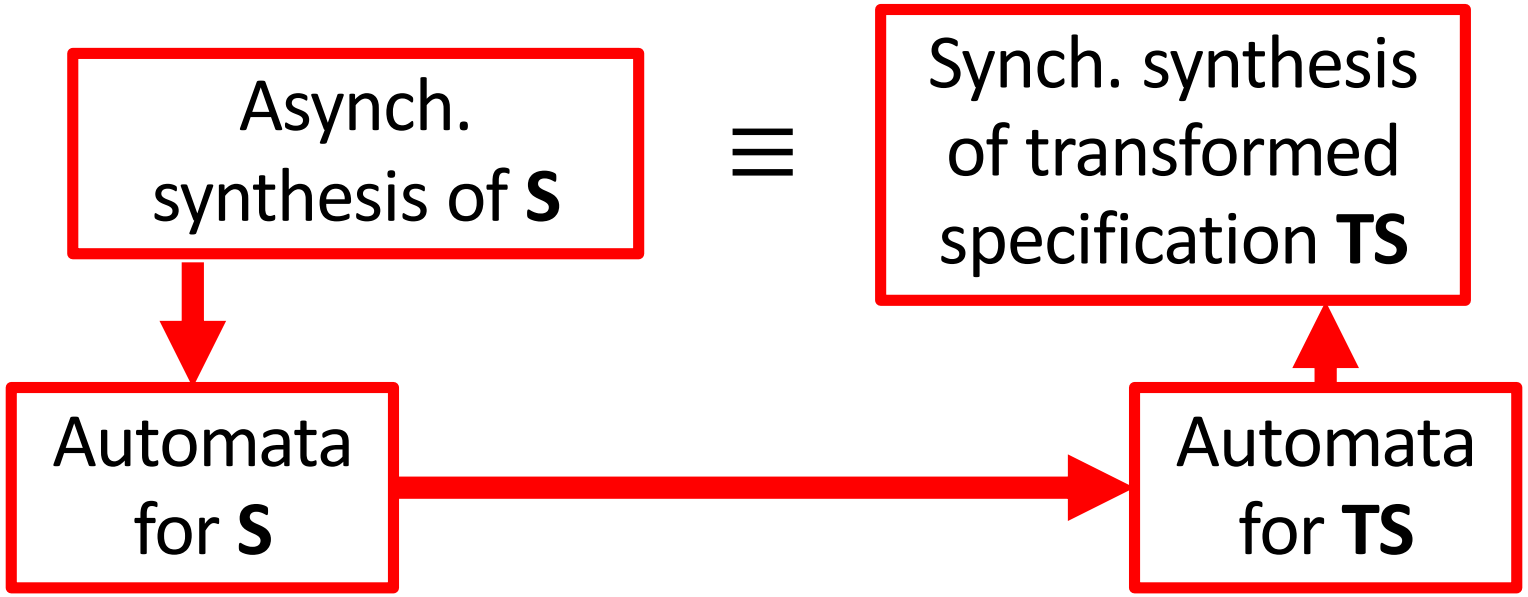
Previous work [Pnueli, Rosner; ICALP 1989]

A trace satisfies **TS** iff all expansion and compression of the trace satisfy **S**

Asynch. synthesis of **S** $\equiv$ Synch. synthesis of transformed specification **TS**

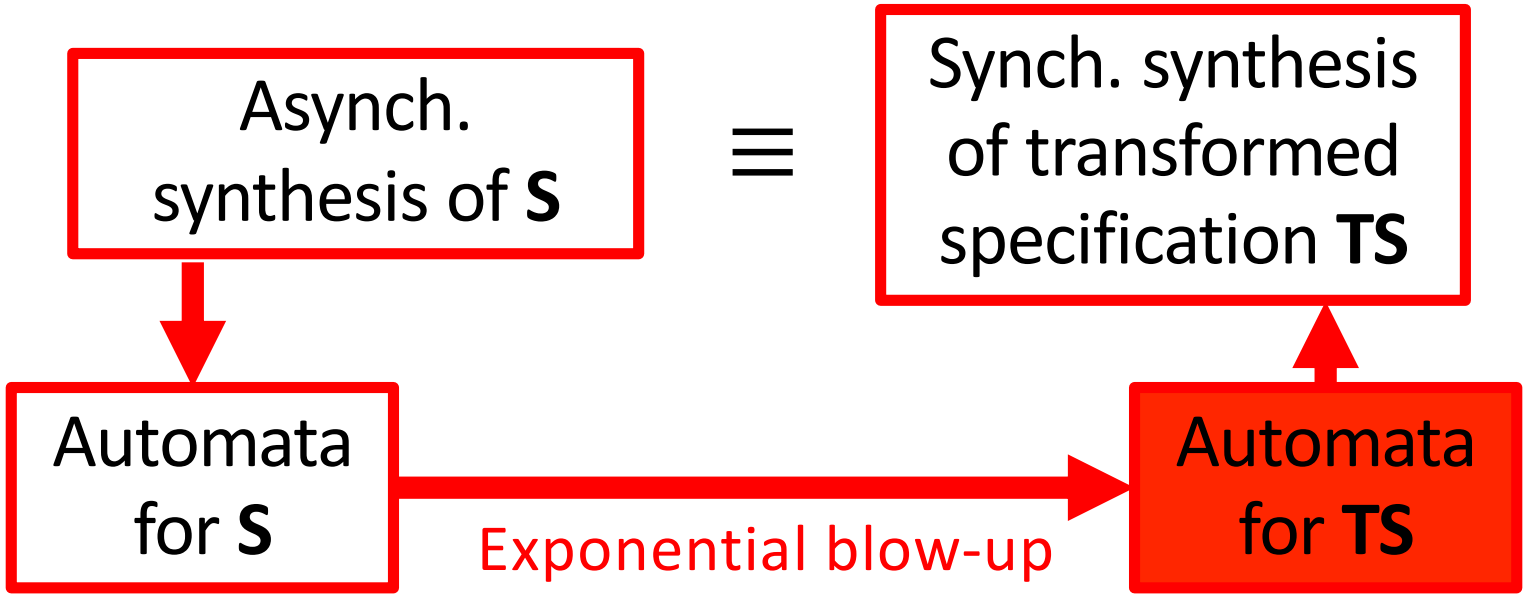
Previous work [Pnueli, Rosner; ICALP 1989]

A trace satisfies **TS** iff all expansion and compression of the trace satisfy **S**



Previous work [Pnueli, Rosner; ICALP 1989]

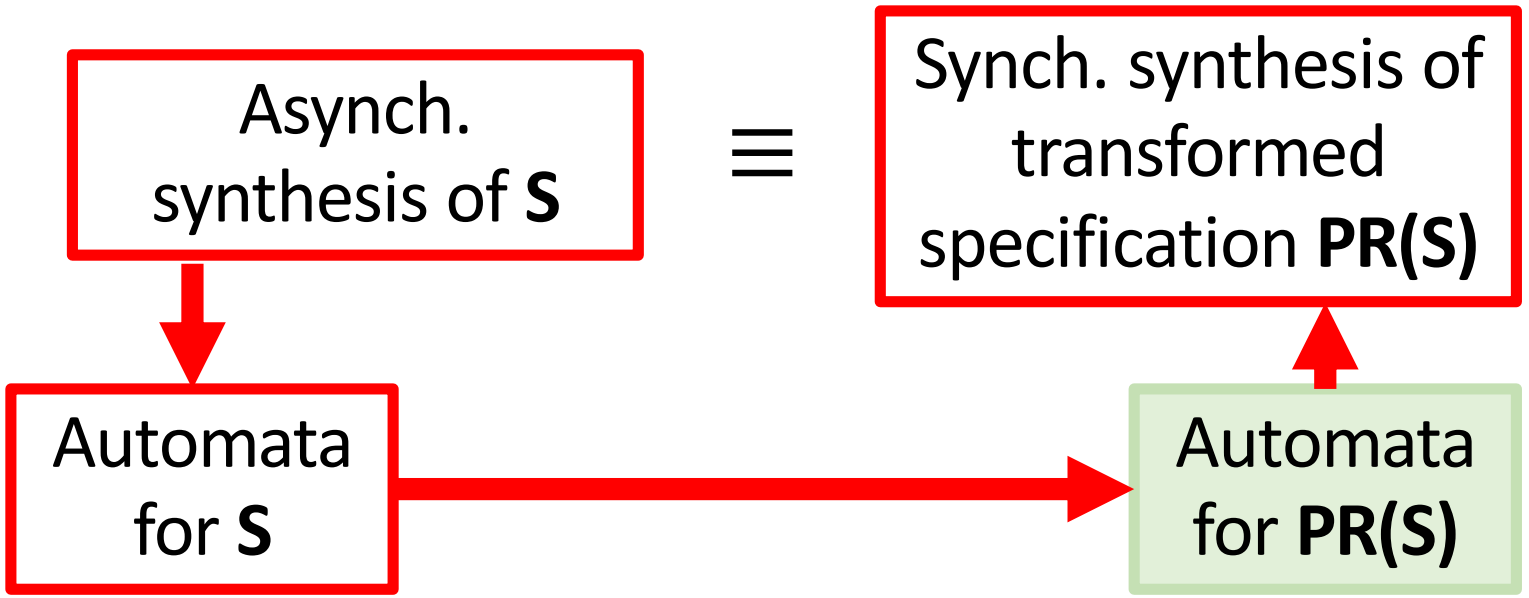
A trace satisfies **TS** iff all expansion and compression of the trace satisfy **S**



Caused by compression

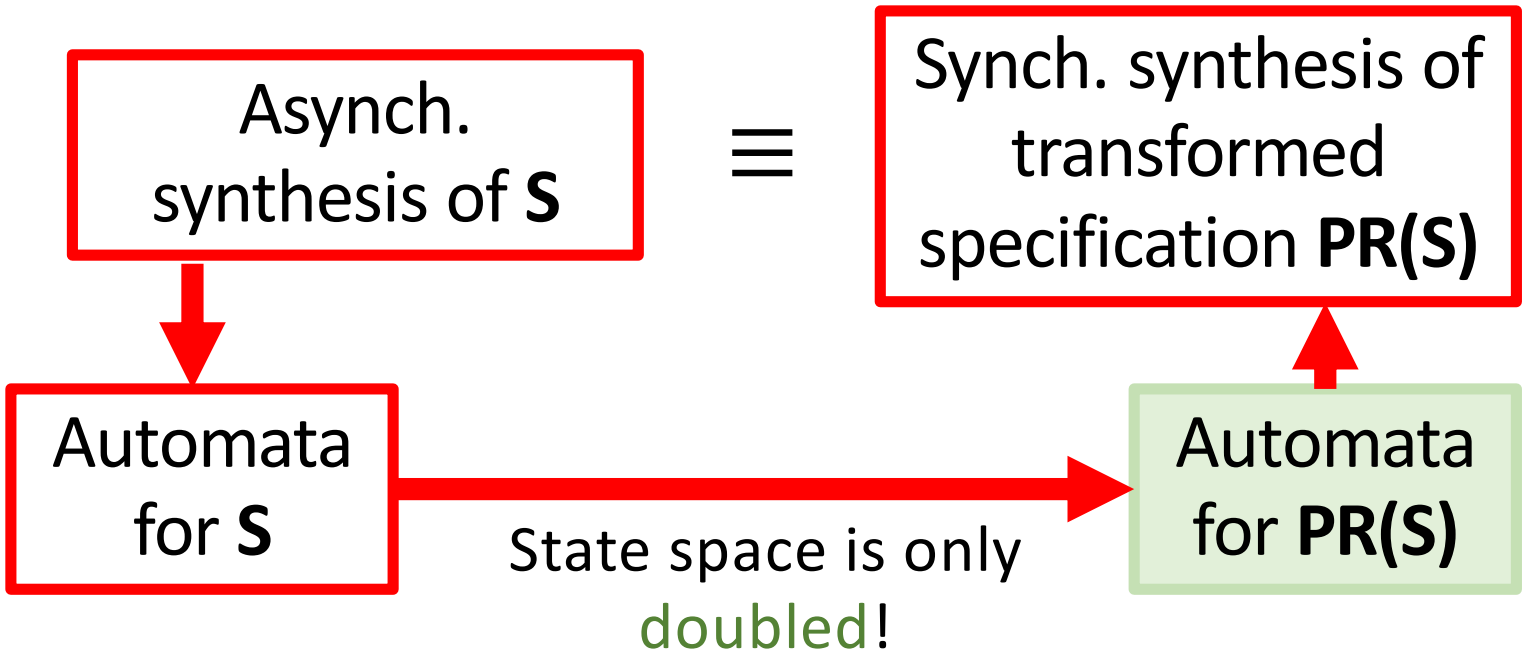
Our reduction: Pnueli-Rosner (PR) closure

A trace satisfies **PR(S)** iff all expansions ~~and compressions~~ of the trace satisfy **S**



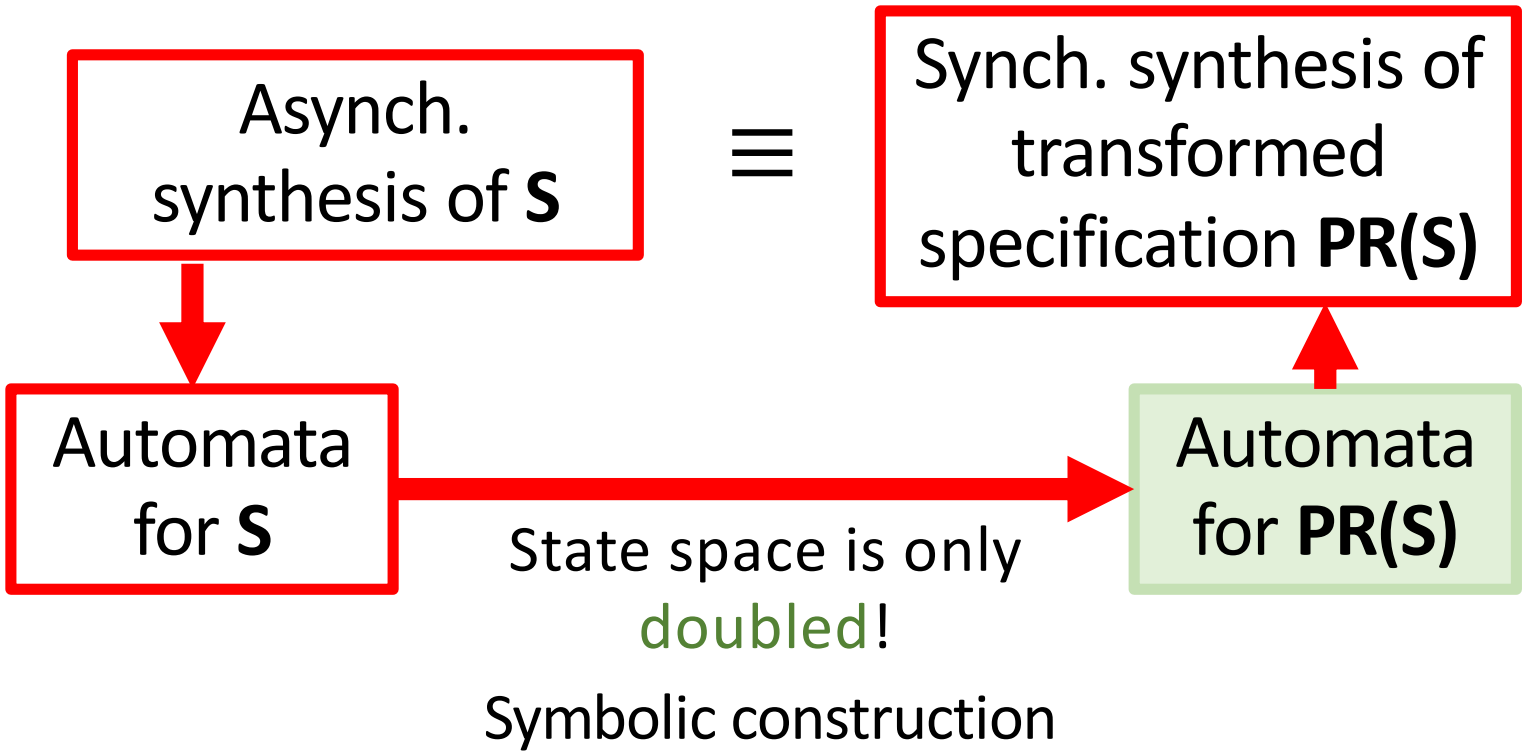
Our reduction: Pnueli-Rosner (PR) closure

A trace satisfies **PR(S)** iff all expansions ~~and compressions~~ of the trace satisfy **S**



Our reduction: Pnueli-Rosner (PR) closure

A trace satisfies **PR(S)** iff all expansions ~~and compressions~~ of the trace satisfy **S**



Implementation and Experiments

- Prototype tool BAS (**B**ounded **A**synchronous **S**ynthesis)
  - BDD-based symbolic construction of **PR(S)** from **S**
  - Bounded synch. synthesis tool
    - BoSy [Faymonville, Finkbeiner, Tentrup; CAV 2017]
    - Acacia+ [Bohy et. al.; CAV 2012]

Implementation and Experiments

- Prototype tool BAS (**B**ounded **A**synchronous **S**ynthesis)
  - BDD-based symbolic construction of **PR(S)** from **S**
  - Bounded synch. synthesis tool
    - BoSy [Faymonville, Finkbeiner, Tentrup; CAV 2017]
    - Acacia+ [Bohy et. al.; CAV 2012]
- Evaluate performance on ~15 formulae
  - Number of propositions in range of 2-16
  - Length of LTL-specification ranges from 5-80

Implementation and Experiments

- Prototype tool BAS (**B**ounded **A**synchronous **S**ynthesis)
  - BDD-based symbolic construction of **PR(S)** from **S**
  - Bounded synch. synthesis tool
    - BoSy [Faymonville, Finkbeiner, Tentrup; CAV 2017]
    - Acacia+ [Bohy et. al.; CAV 2012]

- Evaluate performance on ~15 formulae
  - Number of propositions in range of 2-16
  - Length of LTL-specification ranges from 5-80
  - Almost all formulae solved in less than 2 second
  - PR(φ) construction takes ~2% of total time

Asynchronous LTL synthesis, in a nutshell

- Efficient reduction of asynch. LTL synthesis to synch. synthesis
  - Modified spec. PR closure is *exponentially smaller* than previous reduction
  - Prototype implementation BAS

Asynchronous LTL synthesis, in a nutshell

- Efficient reduction of asynch. LTL synthesis to synch. synthesis
  - Modified spec. PR closure is *exponentially smaller* than previous reduction
  - Prototype implementation BAS

- Low-complexity subclasses of GR(1) spec.
  - Reduction to single-alternation QBF satisfiability
  - Much lower complexity than the general 2EXPTIME