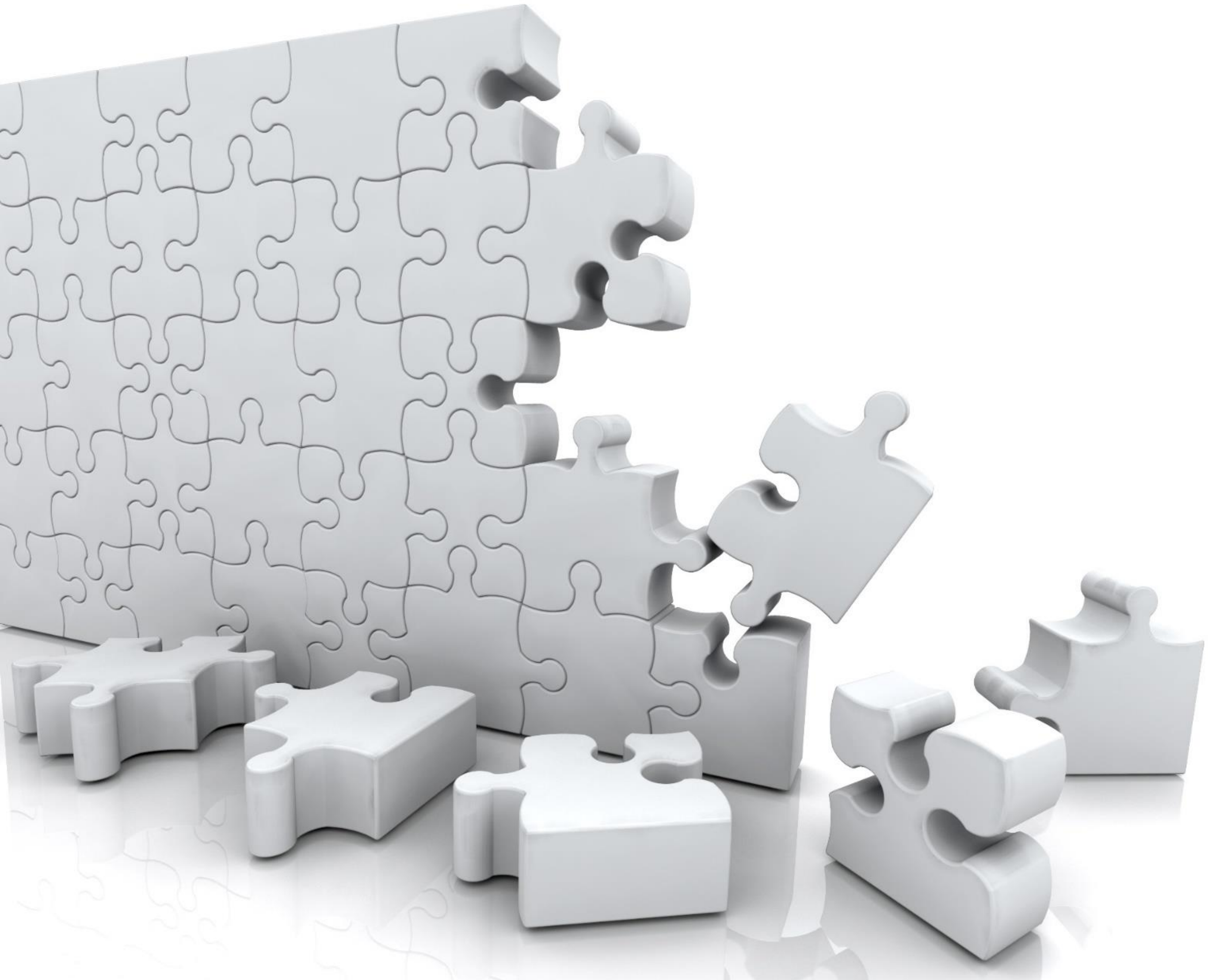


Image created by Kjpargeter - Freepik.com

Parameterized Verification and Synthesis

Swen Jacobs

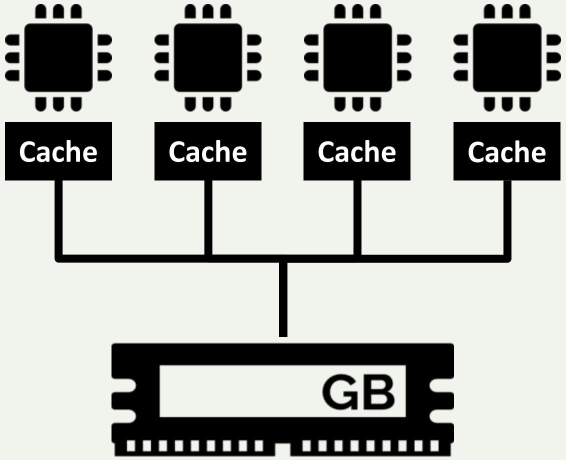
February 25th 2019

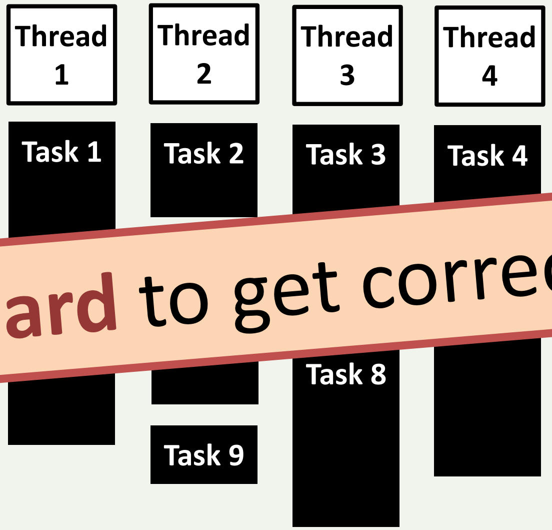
Scientific Talk
in the Habilitation Process

Problem: Correct Design of Parameterized Systems

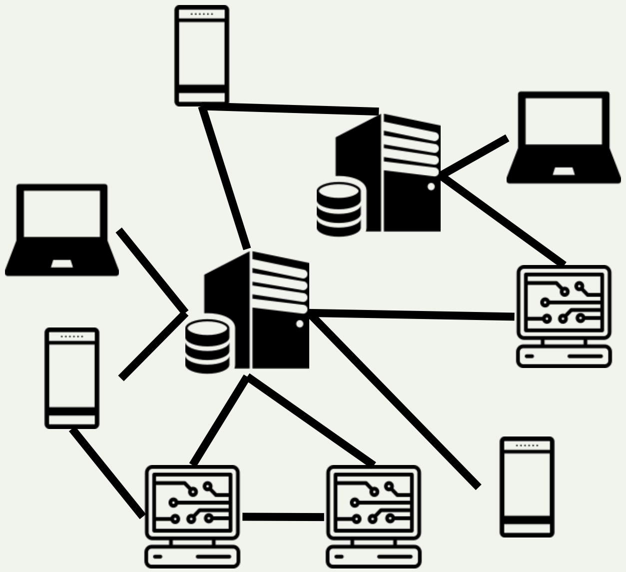
Concurrent systems are **everywhere**

Often: **parametric** number of components





Hard to get correct



Icons made by Freepik from www.flaticon.com

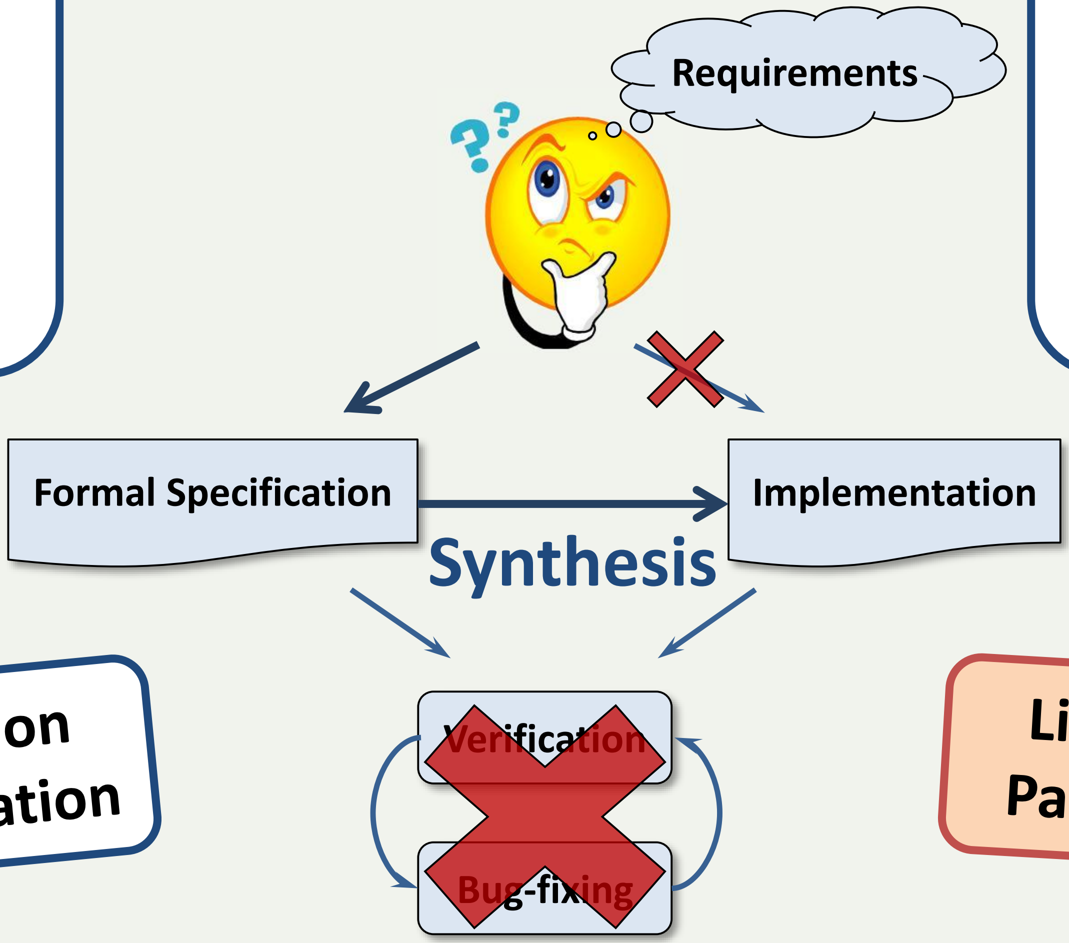
How to Get (Parameterized) Systems Right

Verification Workflow:
1. manual implementation
2. formalize requirements
3. check implementation against specification
4. manual bug-fixing

standard practice for many systems

Synthesis Workflow:
1. formalize requirements
2. automatically obtain correct implementation

computationally expensive (or undecidable)



Very active research on Parameterized Verification

Little previous work on Parameterized Synthesis

Outline

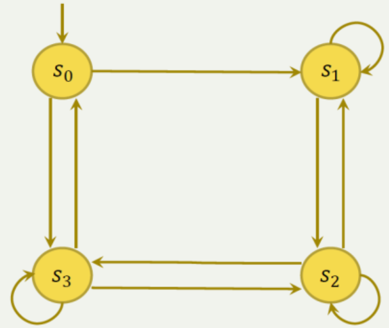
I. (Parameterized) Verification and Synthesis: State of the Art
II. Parameterized Synthesis based on Cutoffs
III. Cutoff Results for Verification and Synthesis

PARAMETERIZED VERIFICATION AND SYNTHESIS: STATE OF THE ART

(Parameterized) Verification: State of the Art (I)

For finite-state systems, we can decide verification problems:

- Model Checking [EC80,QS82]

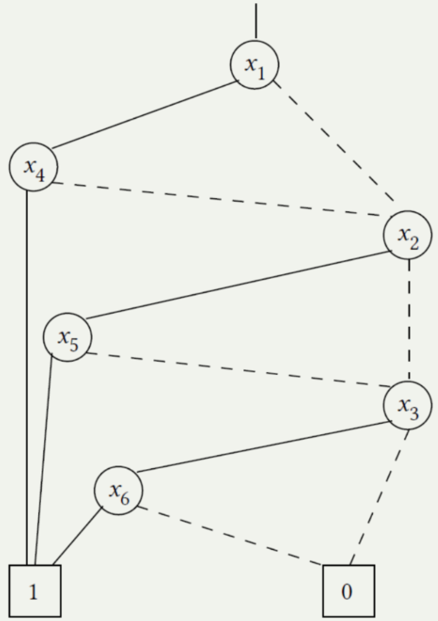


state space explosion:
explicit-state model checking fails for large state spaces

- Symbolic Model Checking [B+92,CVWY92]

symbolic representations allow us to handle systems with “10^{20} states and beyond“

$$(x_1 \wedge x_4) \vee (x_2 \wedge x_5) \vee (x_3 \wedge x_6)$$



(Parameterized) Verification: State of the Art (II)

For many applications, state space is not fixed, but depends on **parameters** such as

- the number of users/participants
- the size of data structures

Expectation: most errors manifest already in systems of „small“ size

Counterexample: cache coherence protocols correct with small number of participants [C+92], but exhibit errors for larger number [K+97]

need **formal argument** why correctness extends to systems of arbitrary size

(Parameterized) Verification: State of the Art (III)

Parameterized Verification is difficult:

Even if systems can be represented as compositions of finite-state components, simple safety properties can be undecidable [S88].

This led to research into

- restrictions that yield **decidable cases**
- **decidable approximations**
- **semi-decision procedures**


Reasoning about Systems with Many Processes

STEVEN M. GERMAN

GTE Laboratories, Inc.,

AND

A. PRASAD SISTLA

Reasoning about Rings*

E. Allen Emerson Kedar S. Namjoshi

Department of Computer Sciences

Regular Model Checking

Ahmed Bouajjani[1], Bengt Jonsson[2], Marcus Nilsson*[2], and Tayssir Touili[1]

Dynamic Cutoff Detection in Parameterized Concurrent Programs*

Alexander Kaiser, Daniel Kroening, and Thomas Wahl

Oxford University Computing Laboratory, United Kingdom

Environment Abstraction for Parameterized Verification*

Edmund Clarke[1], Muralidhar Talupur[1], and Helmut Veith[2]

(Parameterized) Synthesis: State of the Art

For finite-state systems, similar situation as in verification:

- finite-state **two-player game** yields implementations [BL69,PR89]
- **symbolic implementations** can solve problems of significant complexity [JB06]

Major difference:

Synthesis of distributed systems is **undecidable** in general [PR90,FS05]

Parameterized synthesis:

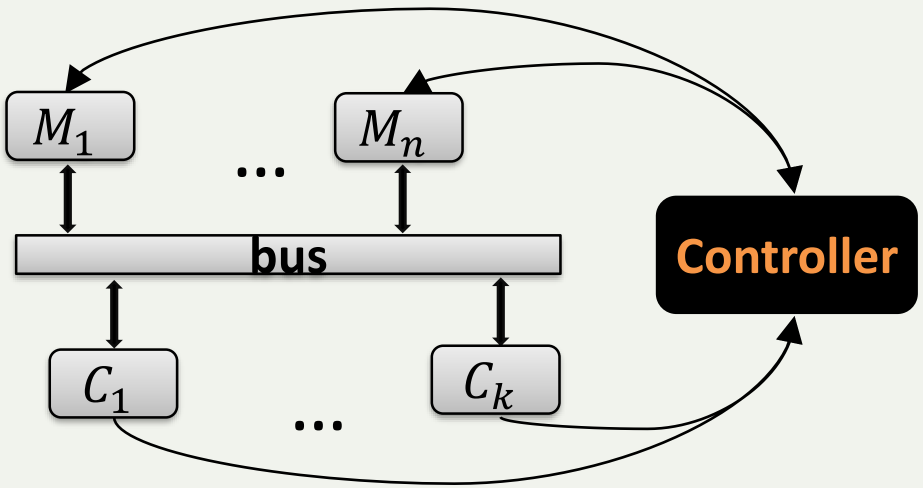
Emerson and Attie synthesize pairs of processes that can be generalized to parameterized systems [EA98] – restrictions on specifications, process implementations & system model make the problem decidable, but limit generality

PARAMETERIZED SYNTHESIS BASED ON CUTOFFS

Synthesis of AMBA Bus Controller

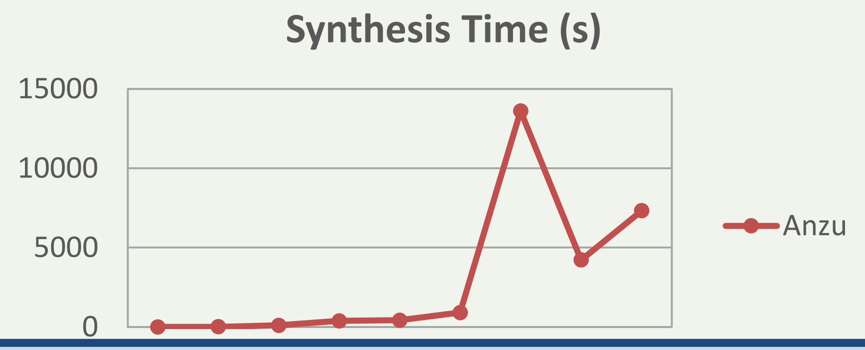
Industrial synthesis benchmark:

Synthesize bus controller with locked accesses, bursts, and other features from temporal logic specification

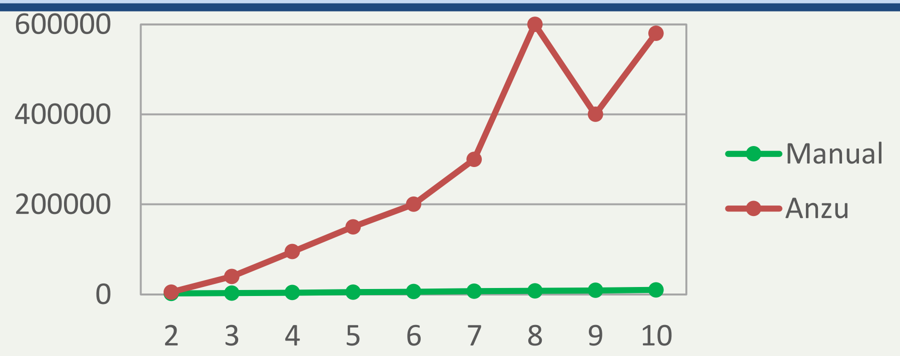


Parameterized in # of masters accessing the bus

Synthesis for increasing # of masters:



Can we avoid this explosion and solve the general case instead?



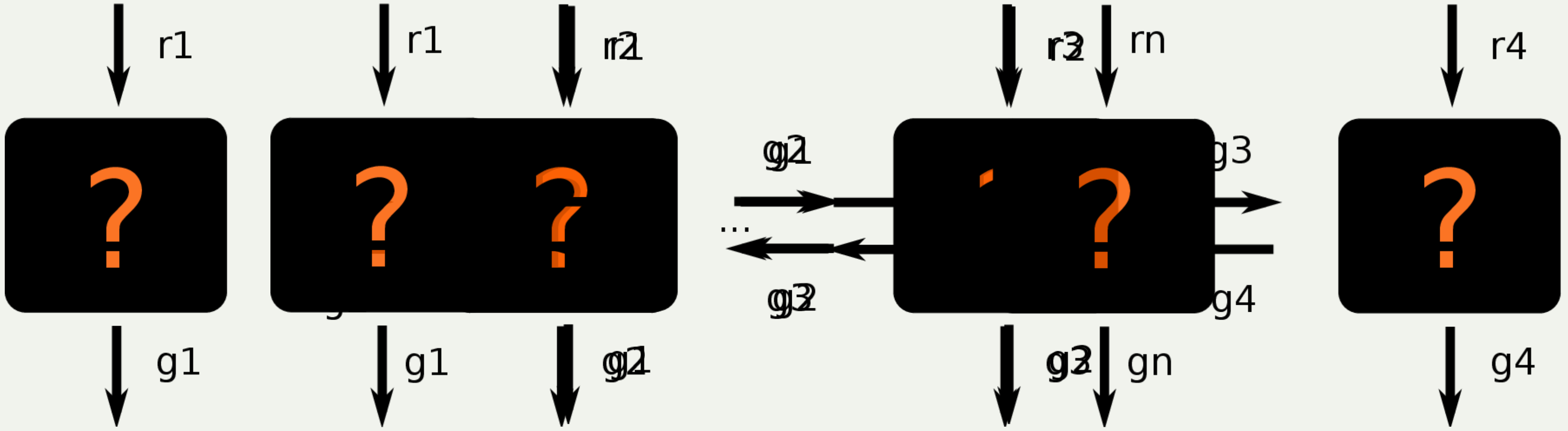
Idea: Synthesis of Replicable Building Blocks

Consider distributed arbiter with specification

Request-response: $\bigwedge_{i \in \{1,\dots,n\}} \mathbf{G}(r_i \rightarrow \mathbf{F} g_i)$

Mutual exclusion: $\bigwedge_{i \neq j \in \{1,\dots,n\}} \mathbf{G}\, \neg(g_i \wedge g_j)$

To obtain systems that work for any n, we synthesize components that **act only on local information** and therefore **can be replicated**



How to ensure correctness for all sizes?

Synthesis Problems

Specification Language:

LTL

temporal operators $\mathbf{G}, \mathbf{F}, \mathbf{U}$

propositional variables

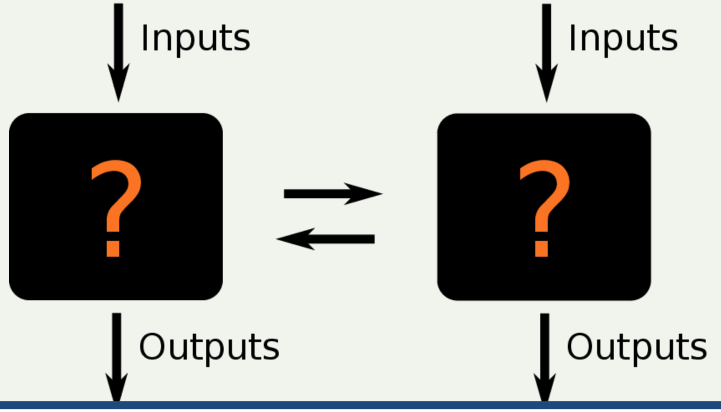
& connectives

Parameterized case: indexed LTL\X
indexed variables p_i, index quantifiers $\forall i, \exists i$.

Implementation:

given as **labelled transition system (LTS)**

Architecture:



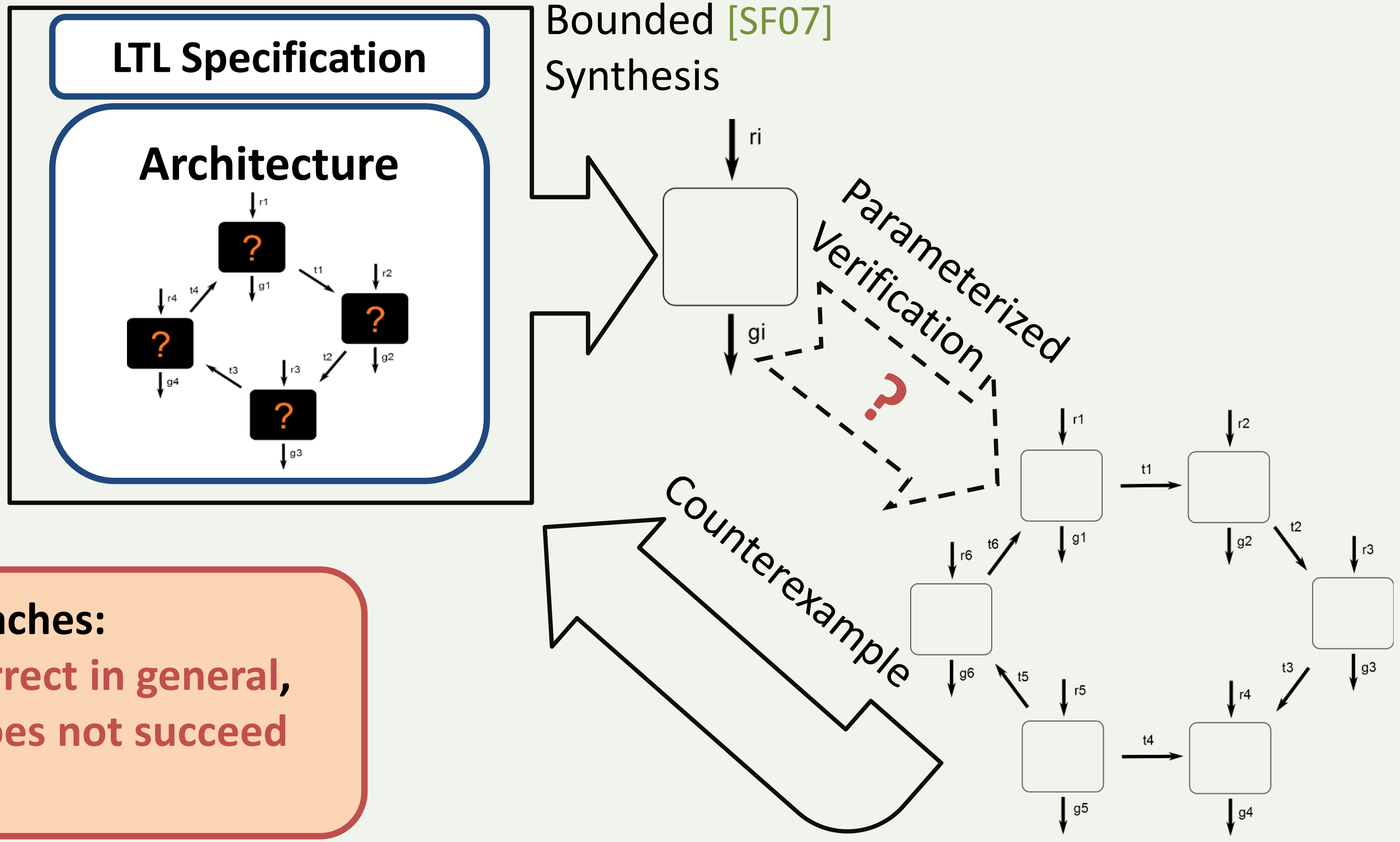
Parameterized architecture:
Sequence of architectures (of increasing size).

Parameterized Synthesis Problem:

Given: 1. indexed LTL Specification φ
2. parameterized Architecture $\boldsymbol{A}$

Find: Implementation S with
$S, \boldsymbol{A}_n \vDash \varphi$ for all n

Parameterized Synthesis



Existing Approaches:

- **Implementations not correct in general, or param. verification does not succeed**
- **Synthesis does not scale**

Cutoffs for Parameterized Systems

Consider

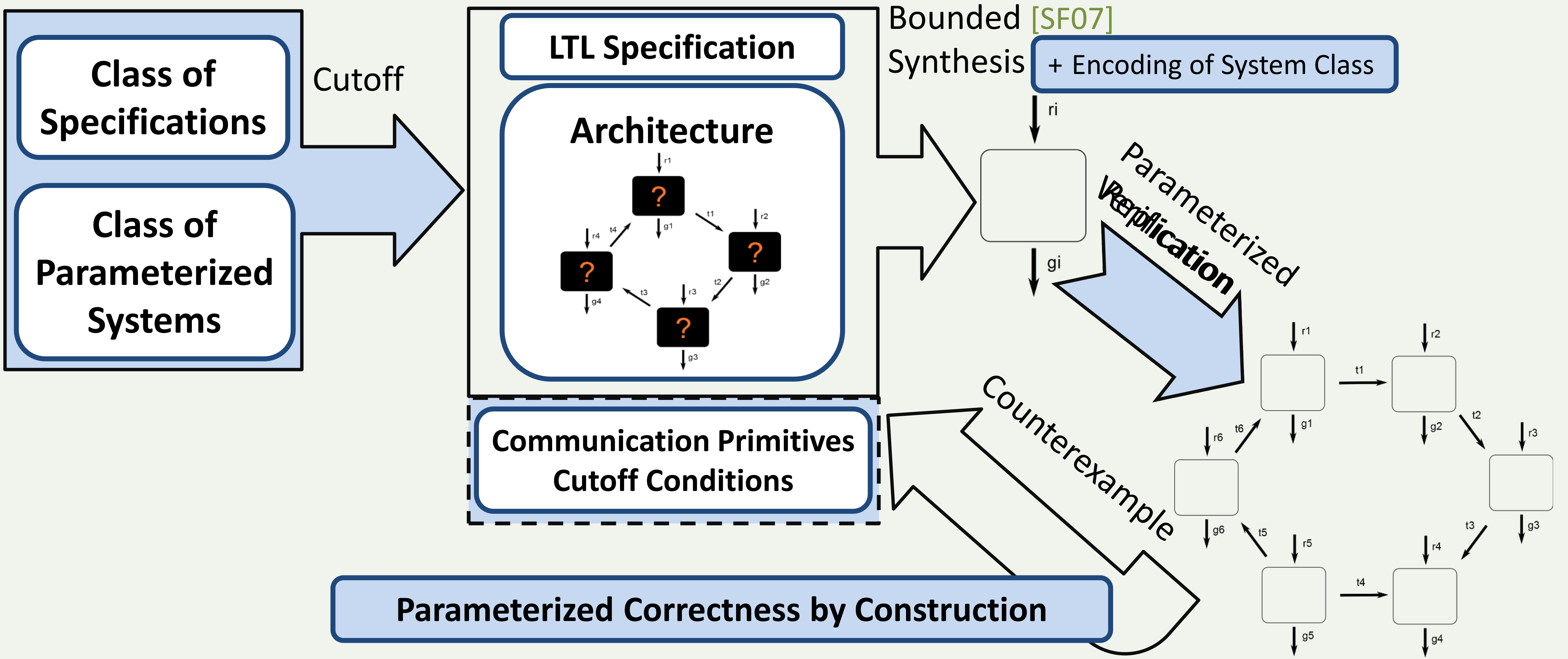
- a class $\boldsymbol{P}$ of parameterized systems, defined by a parameterized architecture $\boldsymbol{A}$ and additional restrictions on the process implementations,
- a class Φ of specifications, e.g. indexed LTL with fixed # of indices.

Cutoff:

A number $c \in \mathbb{N}$ is a **cutoff** for $\boldsymbol{P}$ and Φ if for every specification $\varphi \in \Phi$ and every S from $\boldsymbol{P}$, the following holds:

$$\forall n \geq c: (S, \boldsymbol{A}_c \vDash \varphi \quad \Longleftrightarrow \quad S, \boldsymbol{A}_n \vDash \varphi)$$

Parameterized Synthesis



Parameterized Correctness by Construction

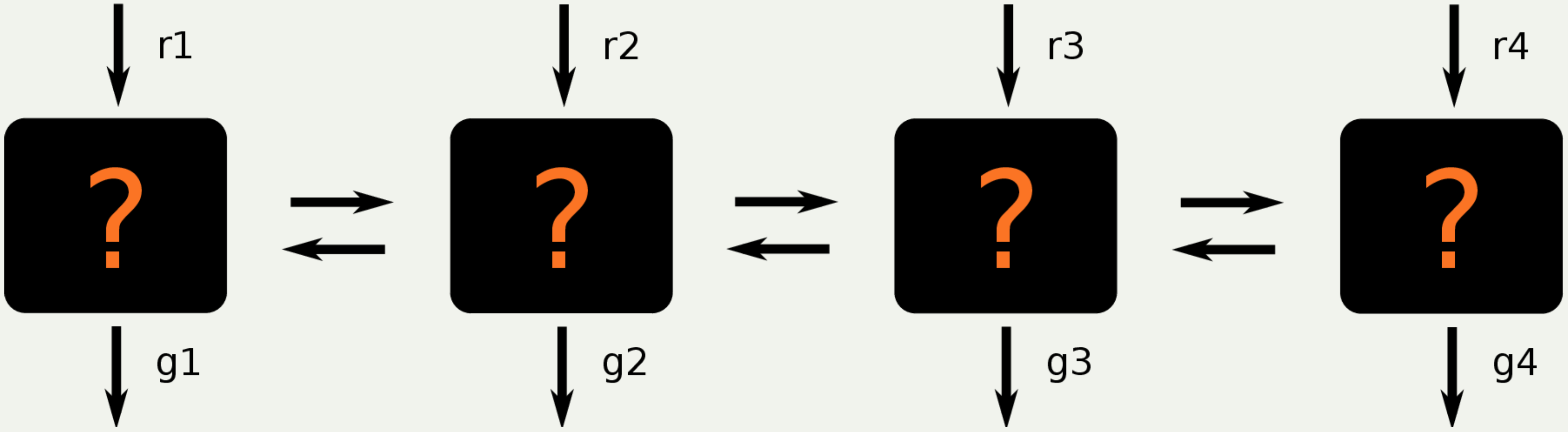
Example: Simple Distributed Arbiter

Distributed arbiter with specification

Request-response: $\forall i: \mathbf{G}(r_i \rightarrow \mathbf{F} g_i)$

Mutual exclusion: $\forall i \neq j: \mathbf{G}\neg\left(g_i \wedge g_j\right)$

What is a suitable class of parameterized systems?



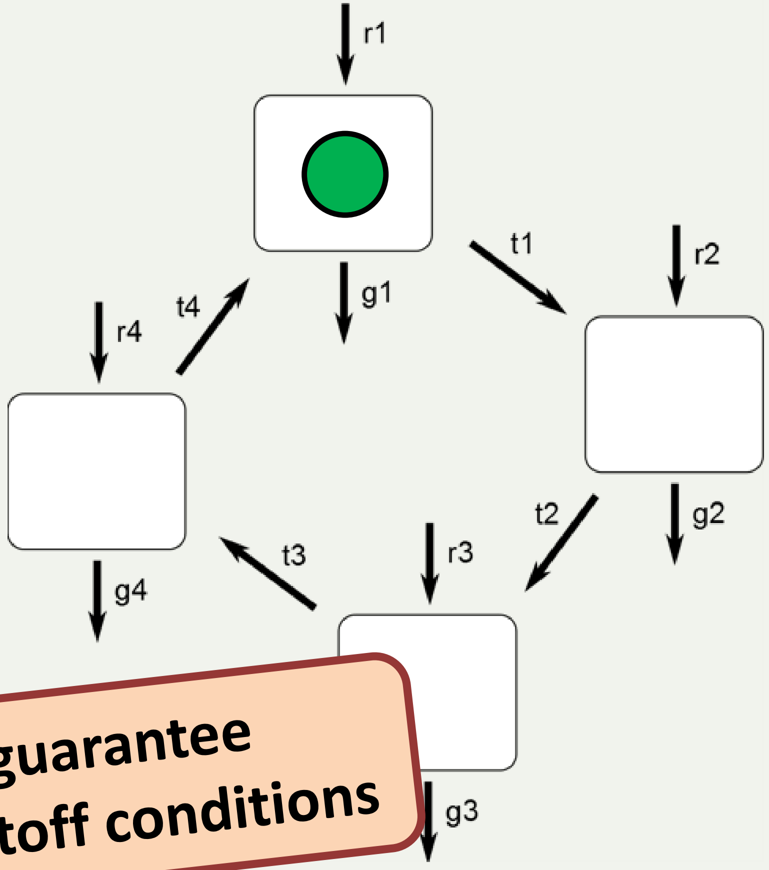
Cutoff Results for Token Rings

Theorem [EN95]:

In **token rings** with interleaving semantics and fair token passing,
a given process implementation satisfies a specification φ ∈ indexed CTL*\X in all rings **iff it satisfies φ in rings of small size.**

For $\forall i. \varphi(i)$, cutoff is 2.

For $\forall i, j. \varphi(i, j)$, cutoff is 4.



Corollary: For **parameterized synthesis**, it is sufficient to synthesize a process implementation satisfying φ (and Thm. conditions) in a small ring.

Need to guarantee additional cutoff conditions

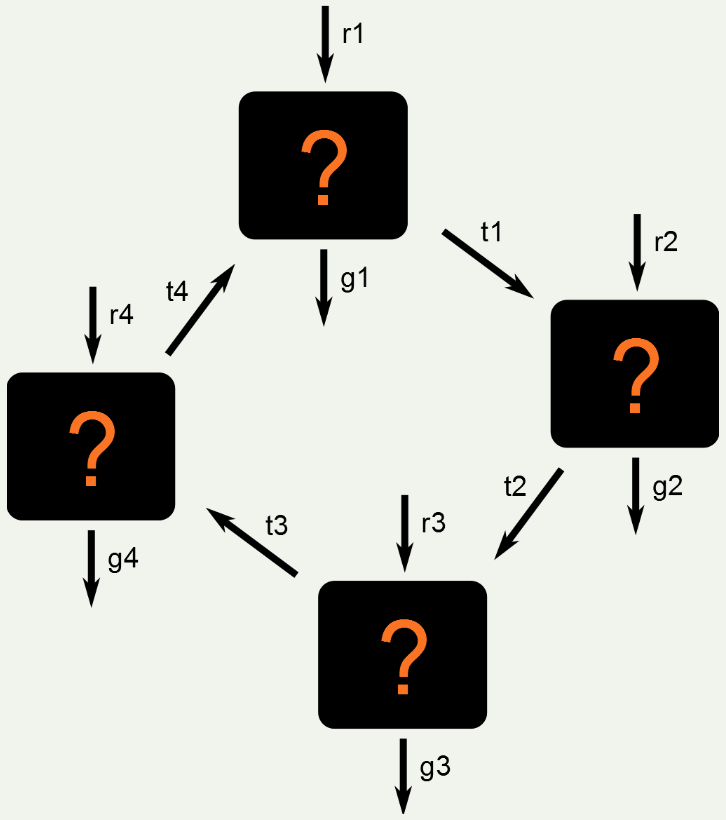
Parameterized Synthesis based on Cutoff Results

Distributed arbiter in token ring
of 4 processes with specification

Request-response: $\forall i \colon \mathbf{G}(r_i \rightarrow \mathbf{F} g_i)$

Mutual exclusion: $\forall i \neq j \colon \mathbf{G}\neg\left(g_i \wedge g_j\right)$

synthesized in $\sim$ 10 sec.
Cutoff results **guarantee correctness**
in rings of arbitrary size.

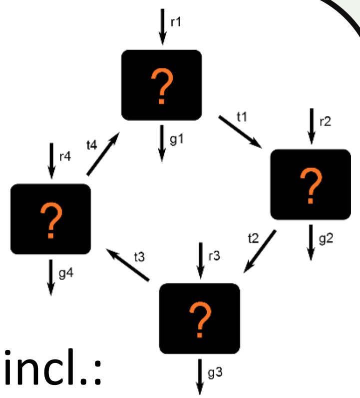


Challenges:
- **Scalability (in size of specification)**
- **Reduction only possible for limited class of systems and specifications**

Parameterized Synthesis of an AMBA Bus Controller

Reduction:

Existing cutoffs for CTL*\X in token-rings (interleaving semantics)



Extended in several dimensions, incl.:

- synchronous systems [SYNT14]
- local X operator [VMCAI13]
- global assumptions [VMCAI14,SYNT14]

Bounded Synthesis:

Existing approach for fixed-size systems

Adapted to token-rings [TACAS12]
Optimized (for token-rings) [VMCAI13,SYNT14]

Assumptions :

	G	$(\mathrm{HMASTLOCK} \wedge \mathrm{HBURST} = \mathrm{INCR}) \rightarrow \mathsf{X}\,\mathsf{F}\,\neg\mathrm{HBUSREQ}[\mathrm{HMASTER}]$	$(A1)$
	GF	HREADY	$(A2)$
$\forall i:$	G	$\mathrm{HLOCK}[i] \rightarrow \mathrm{HBUSREQ}[i]$	$(A3)$
$\forall i:$		$\neg\mathrm{HBUSREQ}[i] \wedge \neg\mathrm{HLOCK}[i] \wedge \neg\mathrm{HREADY}$	$(A4)$

Guarantees :

	G	$\neg\mathrm{HREADY} \rightarrow \mathsf{X}\,\neg\mathrm{START}$	$(G1)$
	G	$(\mathrm{HMASTLOCK} \wedge \mathrm{HBURST} = \mathrm{INCR} \wedge \mathrm{START}) \rightarrow \mathsf{X}(\neg\mathrm{START}\,\mathsf{W}\,(\neg\mathrm{START} \wedge \mathrm{HBUSREQ}[\mathrm{HMASTER}]))$	$(G2)$
	G	$(\mathrm{HMASTLOCK} \wedge \mathrm{HBURST} = \mathrm{BURST4} \wedge \mathrm{START} \wedge \mathrm{HREADY}) \rightarrow \mathsf{X}(\neg\mathrm{START}\,\mathsf{W}\,[3](\neg\mathrm{START} \wedge \mathrm{HREADY}))$	$(G3.1)$
	G	$(\mathrm{HMASTLOCK} \wedge \mathrm{HBURST} = \mathrm{BURST4} \wedge \mathrm{START} \wedge \neg\mathrm{HREADY}) \rightarrow \mathsf{X}(\neg\mathrm{START}\,\mathsf{W}\,[4](\neg\mathrm{START} \wedge \mathrm{HREADY}))$	$(G3.2)$
$\forall i:$	G	$\mathrm{HREADY} \rightarrow (\mathrm{HGRANT}[i] \leftrightarrow \mathsf{X}(\mathrm{HMASTER} = i))$	$(G4)$
		$\mathrm{HGRANT}[0] \wedge (\forall i \neq 0 : \neg\mathrm{HGRANT}[i]) \wedge \mathrm{HMASTER} = 0 \wedge \neg\mathrm{HMASTLOCK} \wedge \mathrm{DECIDE} \wedge \mathrm{START}$	$(G11)$

We **synthesized a solution** for the AMBA Bus Controller with **parameterized correctness guarantee**

Figure 1: Formal specification of the AMBA AHB [12], in the GR(1) fragment of LTL.

CUTOFF RESULTS FOR VERIFICATION AND SYNTHESIS

Decidability Results for Parameterized Verification

Existing decidability and undecidability results:

- Many separate results
- Many **different system models,** sometimes with **implicit assumptions**

⇒ **Hard** to get an overview

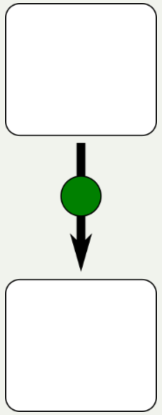
Goal: collect, compare and unify
decidability results in parameterized verification
(for systems with uniform finite-state components)

Parameterized Verification Survey

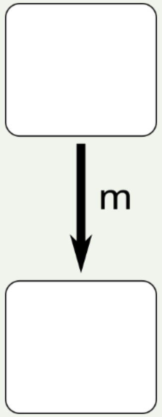
We

- systematically compared existing **models and decidability results**,
- reviewed **proof methods** to obtain them, and
- introduced **common computational model** that captures existing models of systems communicating via different forms of synchronization:

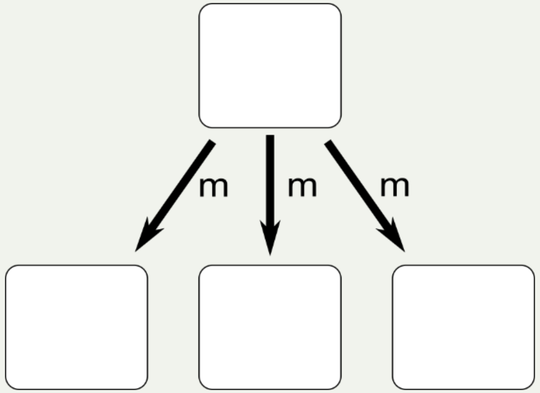
token-passing



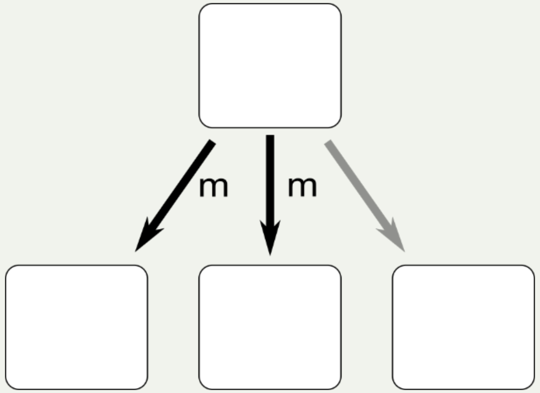
pairwise rendezvous



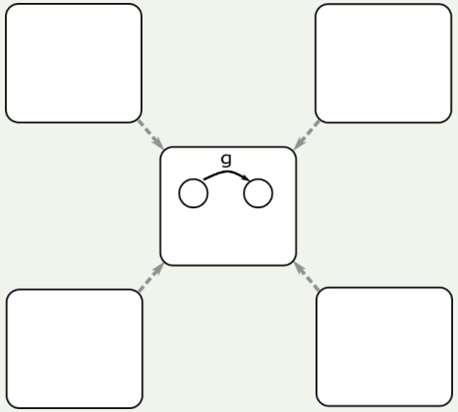
broadcasts



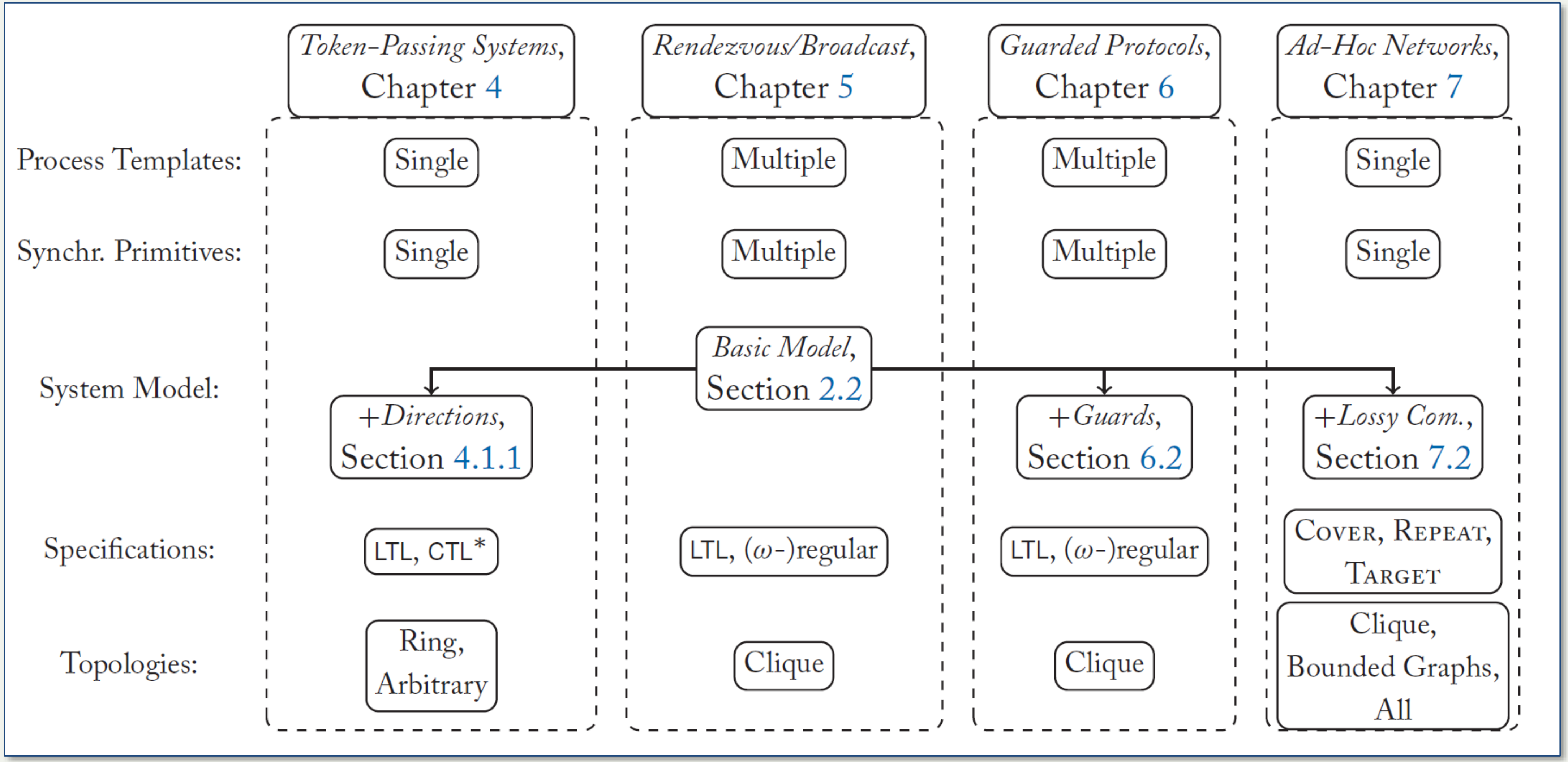
lossy synchronization



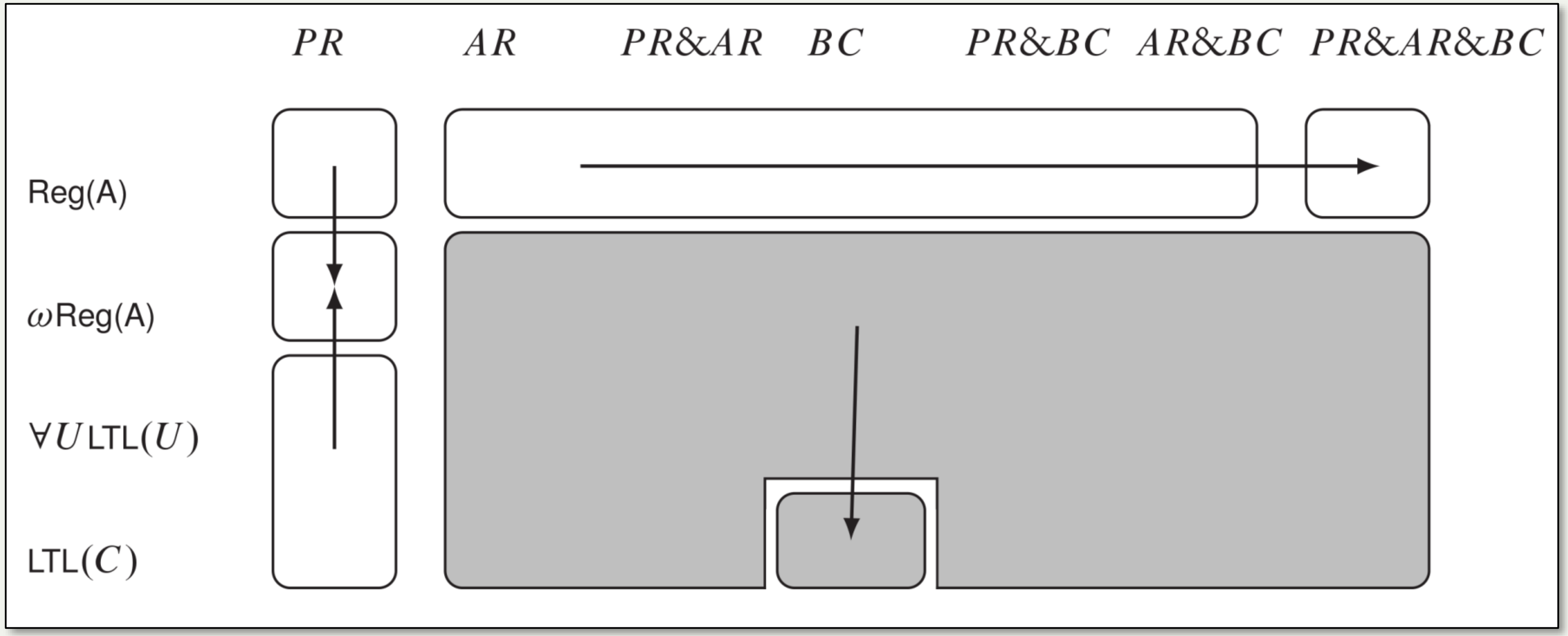
guarded transitions

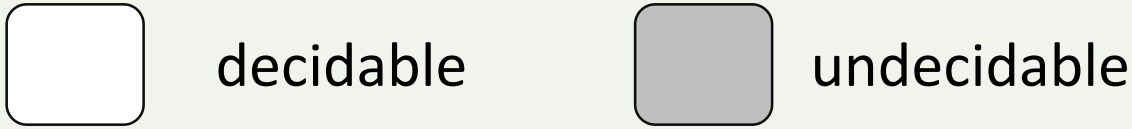


Parameterized Verification Survey

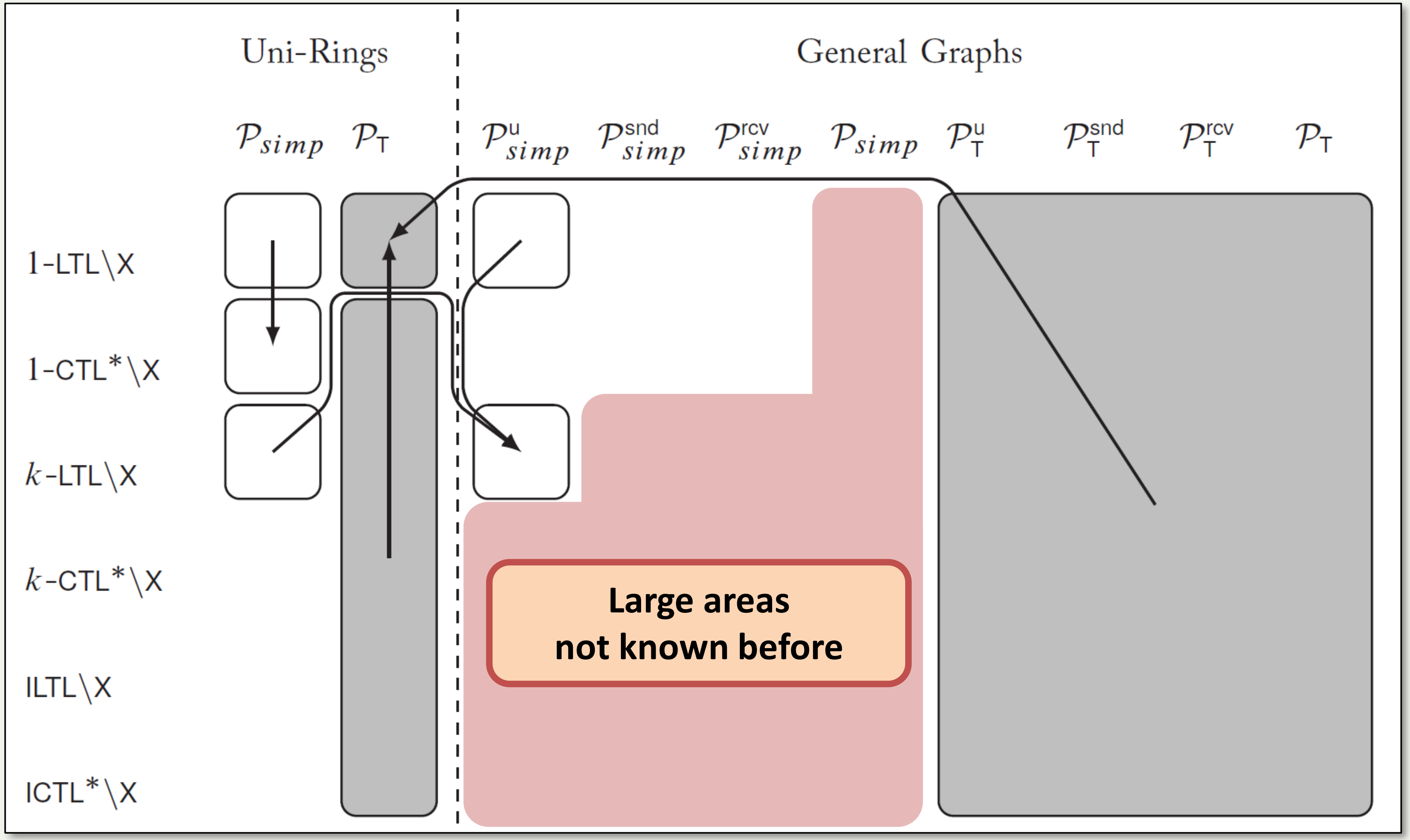


Decidability Results for Rendezvous and Broadcast





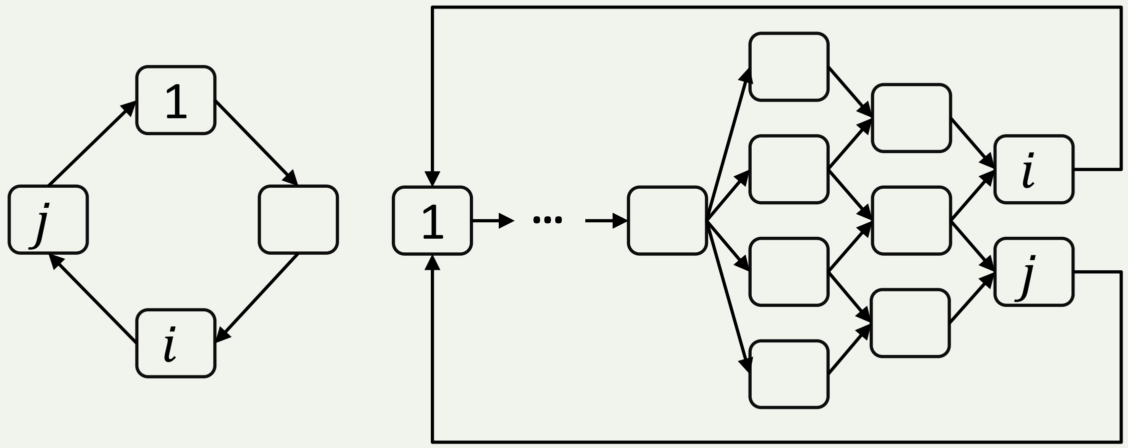
Decidability Results for Token-Passing Systems



(Parameterized) Token-Passing Systems

Consider general token-passing systems with

- **arbitrary topologies**
- **branching-time specifications, i.e.,** fragments of indexed CTL*



Observation: existing cutoff results are **limited**

- either to **non-branching topology** (token rings) or to **non-branching specifications** (fragments of LTL)
- to processes that **do not control or observe different directions** in branching topologies

Insight: effect of token communication is captured by its movement through the network

New Results:

- **undecidable** if **processes control directions** or **specifications can branch unboundedly**
- **cutoffs** exist for **arbitrary topologies** and **specifications with bounded branching**

Additional Results

Cutoffs for Guarded Protocols:

- added support for **fairness assumptions** [VMCAI16]
- showed how to obtain **smaller cutoffs** depending on additional parameters [VMCAI18]

Synthesis of Fault-Tolerant Parameterized Systems:

- **Self-stabilization** (against transient, global faults) [CAV16,OPODIS18]
- **Byzantine fault-tolerance** (against permanent, local faults) [CAV16]

Recent Related Work

- **Lots** of new contributions to parameterized verification literature
- On the border to adjacent fields:
  - „computational algorithm design“ [D+16]
  - verification of multi-agent systems [KL16]
  - parameterized planning [GMRS16]
  - synthesis of distributed algorithms [LKW17]
- Synthesis with identifiers [ESKG14]
- Synthesis of self-stabilizing rings [EK17]
- Control of parameterized systems [BLS18]

SUMMARY

Parameterized Verification and Synthesis

I. **(Parameterized) Verification and Synthesis: State of the Art**
   Parameterized verification is important and difficult, lots of different approaches
   Parameterized synthesis has rarely been considered

II. **Parameterized Synthesis based on Cutoffs**
   First general approach for parameterized synthesis
   Scales to long-standing industrial benchmark

III. **Cutoff Results for Verification and Synthesis**
   Surveyed existing decidability & cutoff results
   Generalized these results to close gaps and make them useful in synthesis

Thank You!

Publications Constituting this Thesis

[1] Roderick Bloem, Swen Jacobs, Ayrat Khalimov, Igor Konnov, Sasha Rubin, Helmut Veith, and Josef Widder. Decidability of Parameterized Verification. Synthesis Lectures on Distributed Computing Theory. Morgan & Claypool Publishers, 2015.

[2] Benjamin Aminof, Swen Jacobs, Ayrat Khalimov, and Sasha Rubin. Parameterized model checking of token-passing systems. In VMCAI, volume 8318 of LNCS, pages 262–281, January 2014.

[3] Simon Außerlechner, Swen Jacobs, and Ayrat Khalimov. Tight cutoffs for guarded protocols with fairness. In VMCAI, volume 9583 of LNCS, pages 476–494. Springer, 2016.

[4] Bernd Finkbeiner and Swen Jacobs. Lazy synthesis. In VMCAI, volume 7148 of LNCS, pages 219–234. Springer, 2012.

[5] Roderick Bloem, Krishnendu Chatterjee, Swen Jacobs, and Robert Könighofer. Assume-guarantee synthesis for concurrent reactive programs with partial information. In TACAS, volume 9035 of LNCS, pages 517–532. Springer, 2015.

[6] Roderick Bloem, Nicolas Braud-Santoni, and Swen Jacobs. Synthesis of self-stabilising and byzantine-resilient distributed systems. In CAV (1), volume 9779 of LNCS, pages 157–176. Springer, 2016.

[7] Swen Jacobs, Leander Tentrup, and Martin Zimmermann. Distributed PROMPT-LTL synthesis. In GandALF, volume 226 of EPTCS, pages 228–241, 2016.

[8] Swen Jacobs and Roderick Bloem. Parameterized synthesis. Logical Methods in Computer Science, 10:1–29, 2014.

[9] Ayrat Khalimov, Swen Jacobs, and Roderick Bloem. Towards efficient parameterized synthesis. In VMCAI, volume 7737 of LNCS, pages 108–127. Springer, 2013.

[10] Ayrat Khalimov, Swen Jacobs, and Roderick Bloem. PARTY parameterized synthesis of token rings. In CAV, volume 8044 of LNCS, pages 928–933. Springer, 2013.

[11] Roderick Bloem, Swen Jacobs, and Ayrat Khalimov. Parameterized synthesis case study: AMBA AHB. In SYNT, volume 157 of EPTCS, pages 68–83, 2014.

Bibliography (in order of appearance)

[EC80] E. Allen Emerson and Edmund M. Clarke. Characterizing correctness properties of parallel programs using fixpoints. In ICALP, volume 85 of LNCS, pages 169–181. Springer, 1980.

[QS82] Jean-Pierre Queille and Joseph Sifakis. Specification and verification of concurrent systems in CESAR. In International Symposium on Programming, volume 137 of LNCS, pages 337–351. Springer, 1982.

[B+92] Jerry R. Burch, Edmund M. Clarke, Kenneth L. McMillan, David L. Dill, and L. J. Hwang. Symbolic model checking: 10ˆ20 states and beyond. Inf. Comput., 98(2):142–170, 1992.

[CVWY92] Costas Courcoubetis, Moshe Y. Vardi, Pierre Wolper, and Mihalis Yannakakis. Memory-efficient algorithms for the verification of temporal properties. Formal Methods in System Design, 1(2/3):275–288, 1992.

[C+92] Edmund Clarke, Orna Grumberg, Hiromi Hiraishi, Somesh Jha, David Long, Kenneth McMillan, and Linda Ness. Verification of the futurebus+ cache coherence protocol. Technical report, DTIC Document, 1992.

[K+97] Yonit Kesten, Oded Maler, Monica Marcus, Amir Pnueli, and Elad Shahar. Symbolic model checking with rich assertional languages. In CAV, pages 424–435. Springer, 1997.

[S88] Ichiro Suzuki. Proving properties of a ring of finite-state machines. Inf. Process. Lett., 28(4):213–214, July 1988.

[BL69] J. Richard Büchi and Lawrence H. Landweber. Definability in the monadic second-order theory of successor. J. Symb. Log., 34(2):166–170, 1969.

[PR89] Amir Pnueli and Roni Rosner. On the synthesis of a reactive module. In POPL, pages 179–190. ACM Press, 1989.

[JB06] Barbara Jobstmann and Roderick Bloem. Optimizations for LTL synthesis. In FMCAD, pages 117–124. IEEE Computer Society, 2006.

[PR90] Amir Pnueli and Roni Rosner. Distributed reactive systems are hard to synthesize. In FOCS, pages 746–757. IEEE Computer Society, 1990.

[FS05] Bernd Finkbeiner and Sven Schewe. Uniform distributed synthesis. In (LICS 2005), pages 321–330. IEEE Computer Society, 2005.

[EA98] Paul C. Attie, E. Allen Emerson. Synthesis of Concurrent Systems with Many Similar Processes. In ACM Trans. Program. Lang. Syst. 20(1), pages 51-115. 1998.

Bibliography (in order of appearance)

[SF07] Sven Schewe, Bernd Finkbeiner. Bounded Synthesis. In ATVA 2007, pages 474-488. 2007.

[EN95] E. Allen Emerson, Kedar S. Namjoshi. Reasoning about Rings. In POPL 1995, pages 85-94. 1995.

[VMCAI18] Swen Jacobs, Mouhammad Sakr. Analyzing Guarded Protocols. Better Cutoffs, More Systems, More Expressivity. In VMCAI 2018, pages 247-268.

[OPODIS18] Nahal Mirzaie, Fathiyeh Faghih, Swen Jacobs, Borzoo Bonakdarpour. Parameterized Synthesis of Self-Stabilizing Protocols in Symmetric Rings. In OPODIS 2018, pages 29:1-29:17. 2018.

[D+16] Danny Dolev, Keijo Heljanko, Matti Järvisalo, Janne H. Korhonen, Christoph Lenzen, Joel Rybicki, Jukka Suomela, Siert Wieringa. Synchronous counting and computational algorithm design. In J. Comput. Syst. Sci. 82(2), pages 310-332. 2016.

[KL16] Panagiotis Kouvaros, Alessio Lomuscio. Parameterised verification for multi-agent systems. In Artif. Intell. 234, pages 152-189. 2016.

[GMRS16] Giuseppe De Giacomo, Aniello Murano, Sasha Rubin, Antonio Di Stasio. Imperfect-Information Games and Generalized Planning. In IJCAI 2016, pages 1037-1043. 2016.

[LKW17] Marijana Lazic, Igor Konnov, Josef Widder, Roderick Bloem. Synthesis of Distributed Algorithms with Parameterized Threshold Guards. In OPODIS 2017, pages 32:1-32:20. 2017.

[ESKG14] Rüdiger Ehlers, Sanjit A. Seshia, Hadas Kress-Gazit. Synthesis with Identifiers. In VMCAI 2014, pages 415-433. 2014.

[EK17] Alex Klinkhamer, Ali Ebnenasir. Synthesizing Parameterized Self-stabilizing Rings with Constant-Space Processes. In FSEN 2017, pages 100-115. 2017.

[BLS18] Benedikt Bollig, Mathieu Lehaut, Nathalie Sznajder. Round-Bounded Control of Parameterized Systems. In ATVA 2018, pages 370-386. 2018.