2024
DESIGN AND VERIFICATION™
DVCON
CONFERENCE AND EXHIBITION
UNITED STATES
SAN JOSE, CA, USA
MARCH 4-7, 2024

Automated Generation of Interval Properties From Trace-Based Function Models

Robert Kunzelmann, Aishwarya Sridhar, Daniel Gerl, Lakshmi Vidhath Boga, Wolfgang Ecker

Agenda

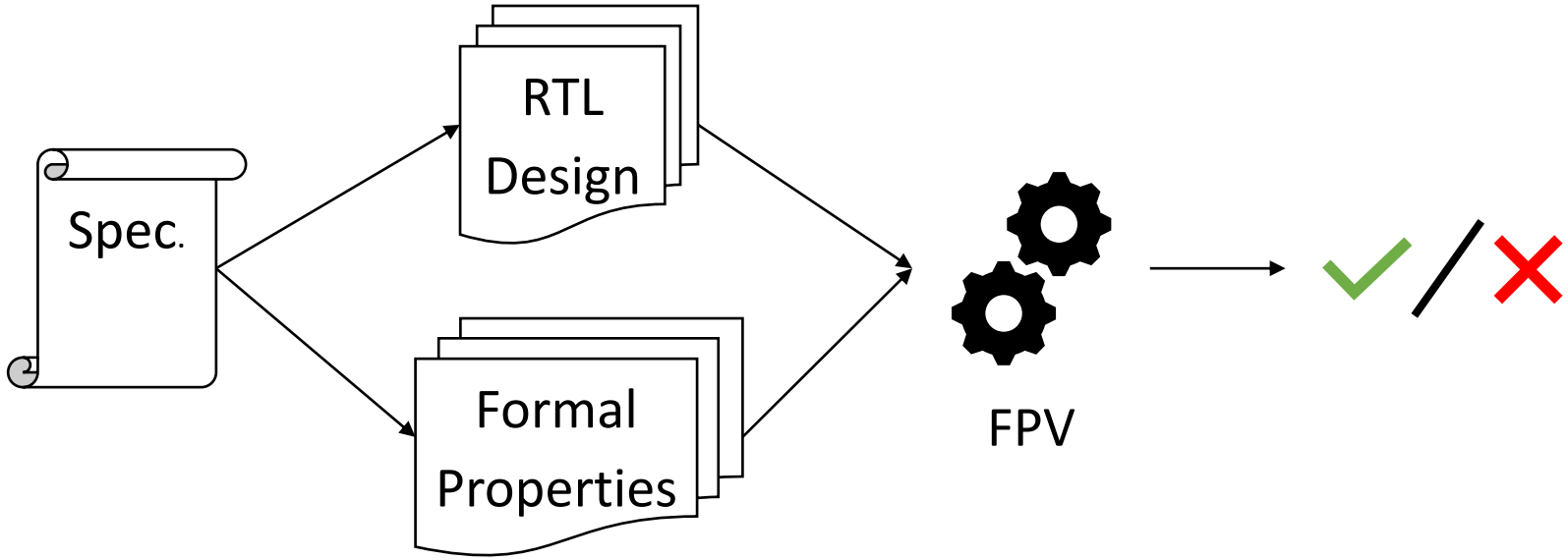
- Motivation and Background
- Trace Models
  - Concept
  - Workflow
- Property Generation From Traces
- Results and Conclusion

Motivation and Background

accellera
SYSTEMS INITIATIVE

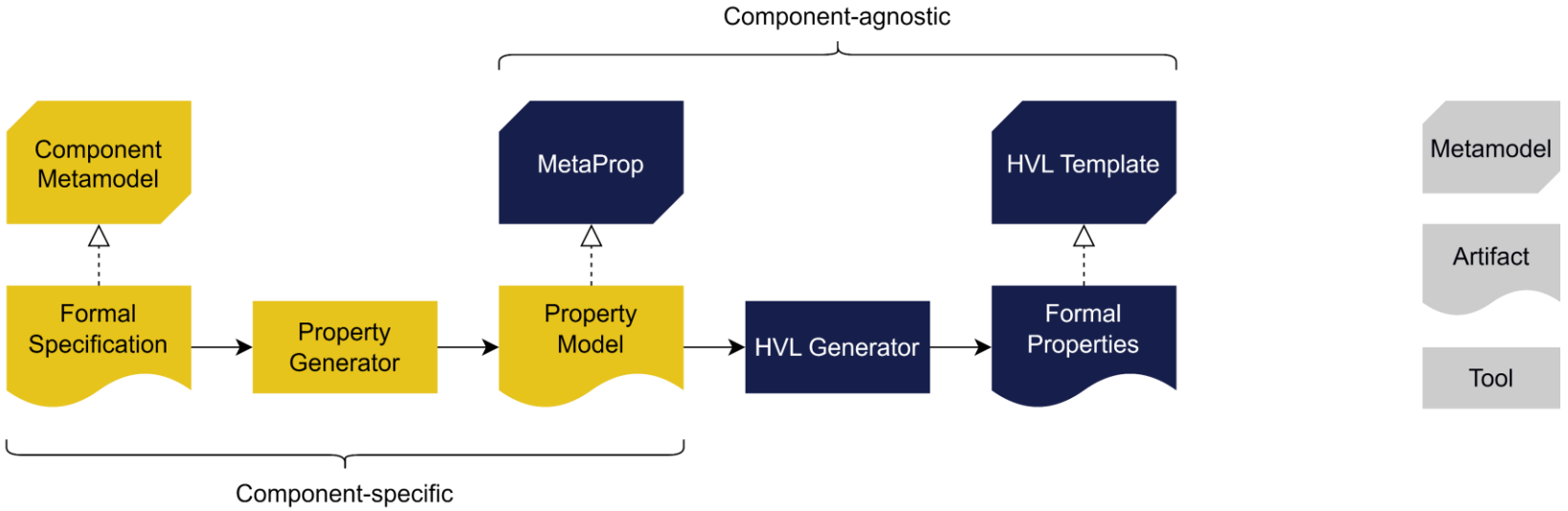
Motivation of (Formal) Verification

- Process of proving that a design adheres to its specification
- Addressed challenge in this work: specification capture



Background on MetaProp[1]

- MetaProp is a code generation framework for formal properties
- Follows metamodeling and model-driven architecture principles



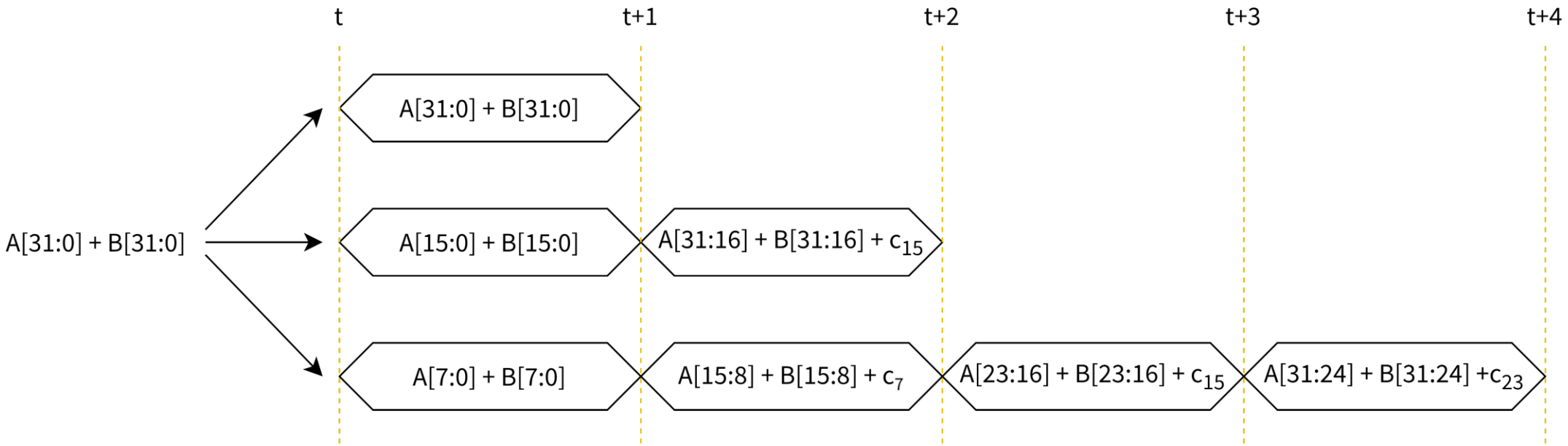
[1]K. Devarajegowda and W. Ecker, "Meta-model Based Automation of Properties for Pre-Silicon Verification," 2018 IFIP/IEEE International Conference on Very Large Scale Integration (VLSI-SoC)

Trace Concept

accellera
SYSTEMS INITIATIVE

Trace Definition

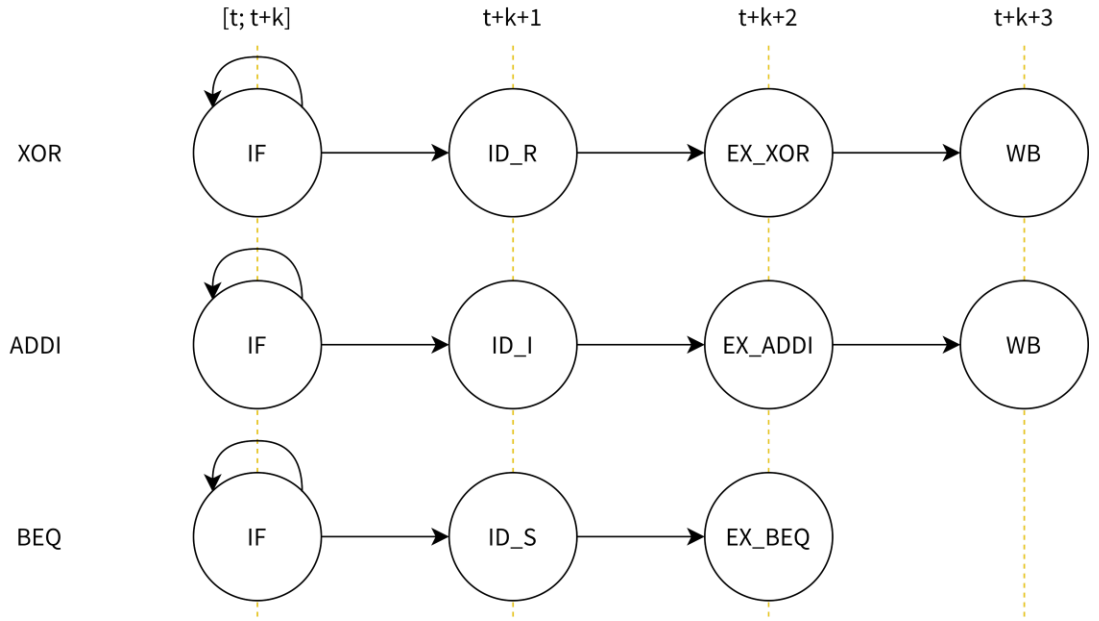
- Traces specify the temporal and functional behavior on the RT level[1]



[1]https://kluedo.ub.rptu.de/frontdoor/index/index/docId/6640

Modeling Traces

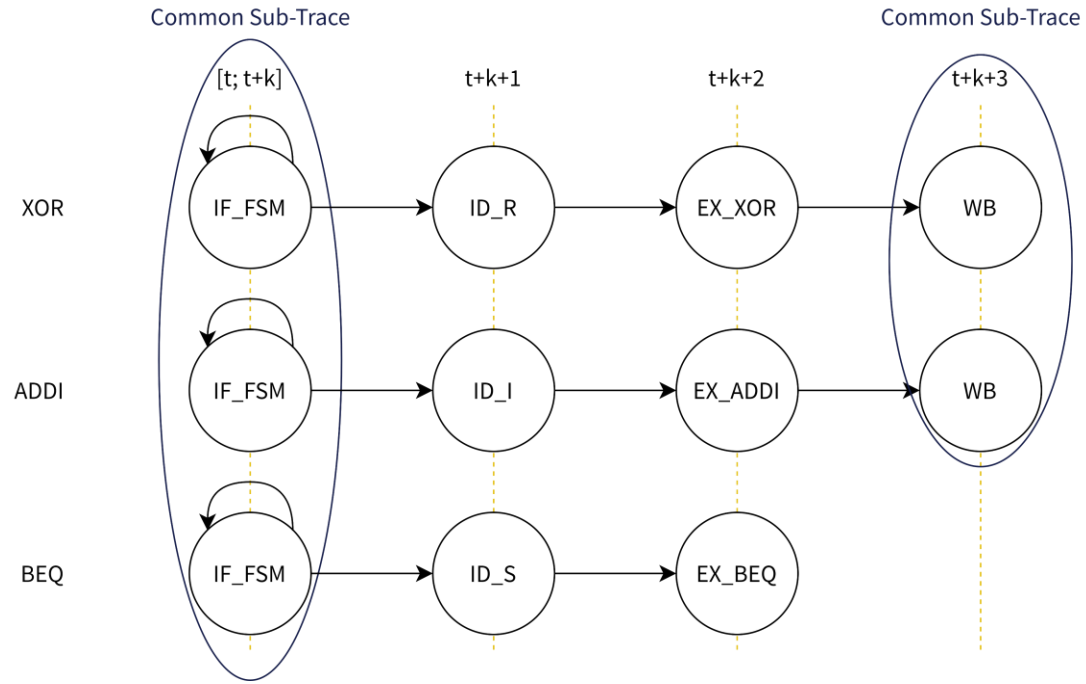
- Traces use an FSM notation
- States are aligned to a time-grid (clock)
- Transitions model conditional data propagation



Working With Traces

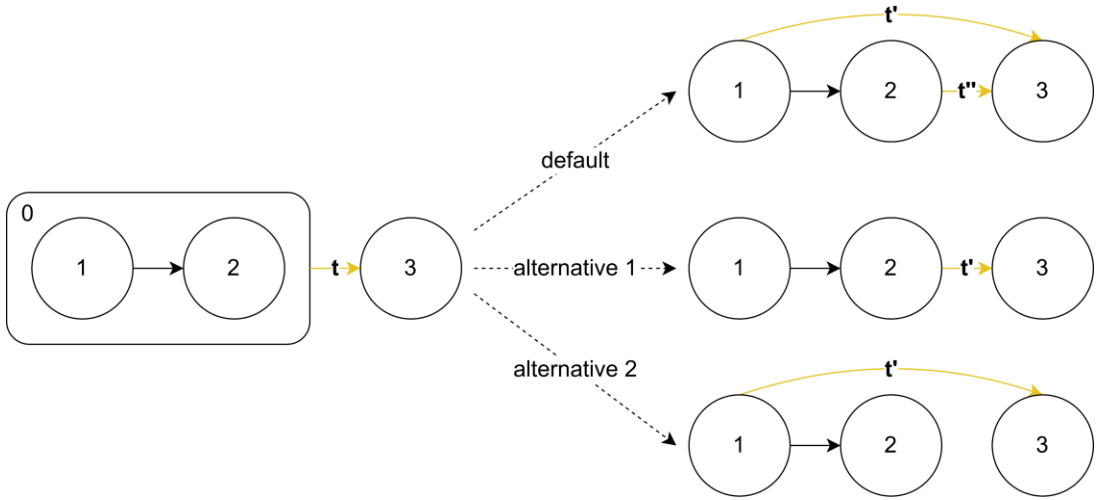
Trace Notation Using Hierarchical State Machines (HSMs)

- HSMs allow reusing sub-traces
- Clearer modeling of complex systems



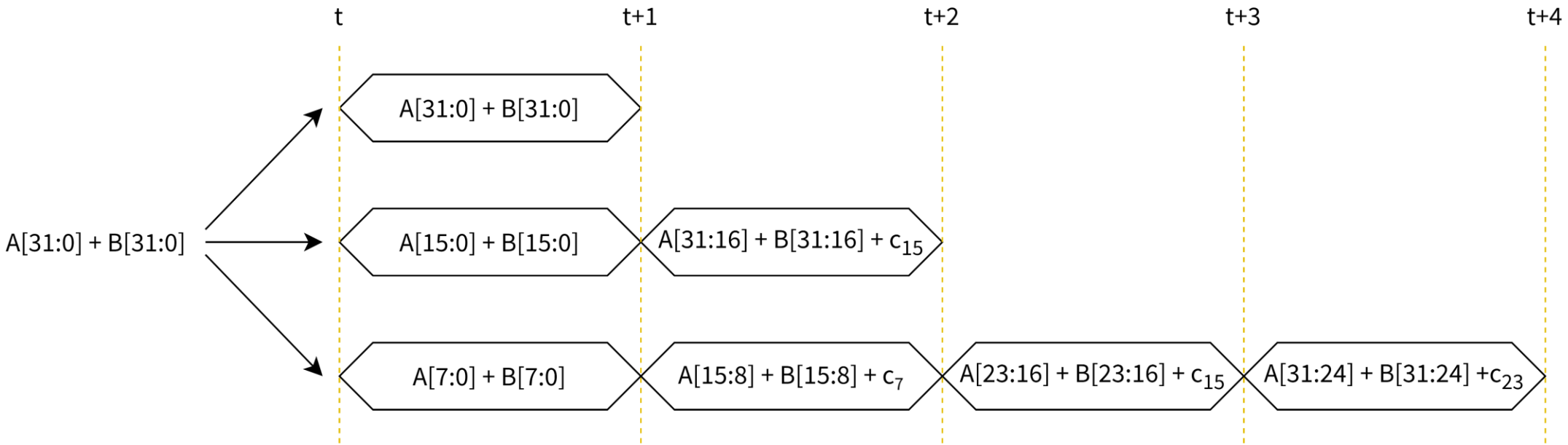
Transition Mapping and HSM Flattening

- Default flattening
  - Hierarchical transitions are duplicated to each state
- Manual transition map
  - Maps each hierarchical transition to a set of flat transitions



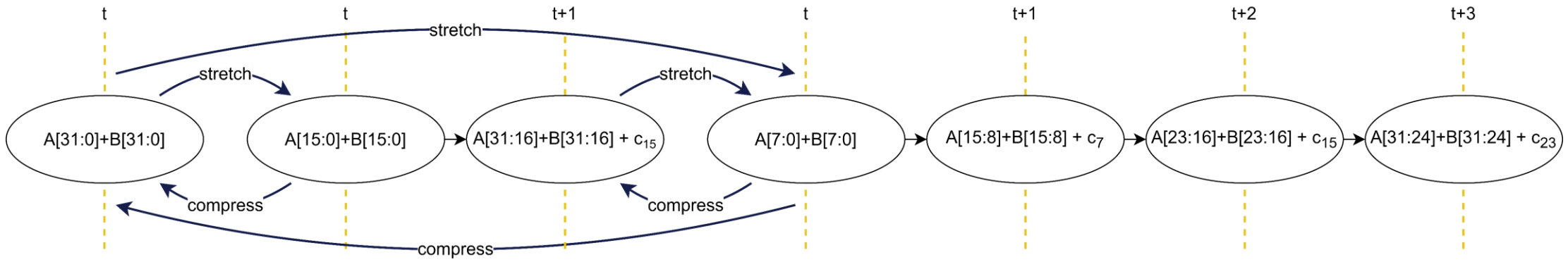
Time-Annotation by Stretching and Compressing (I)

- Different temporal implementations of one function



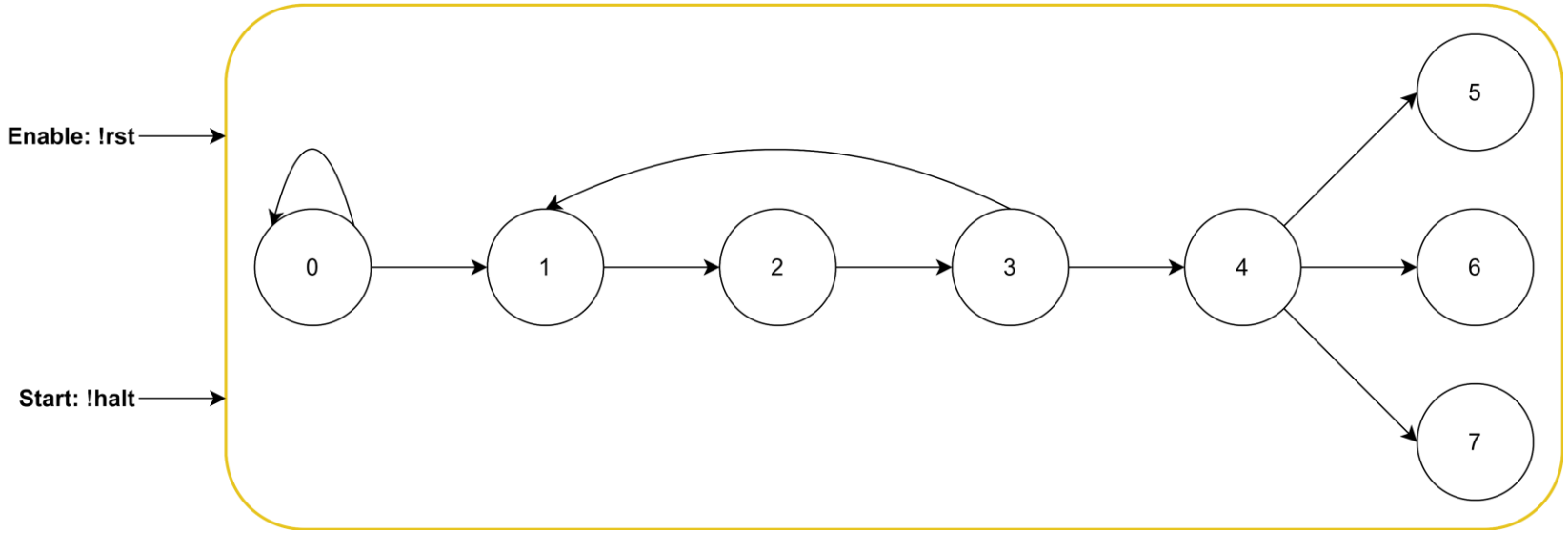
Time-Annotation by Stretching and Compressing (II)

- Reuse traces by stretching and compressing their modeled temporal behavior



Separating Control and Data Flow

- Traces model data flow
- Control flow is separated as much as possible

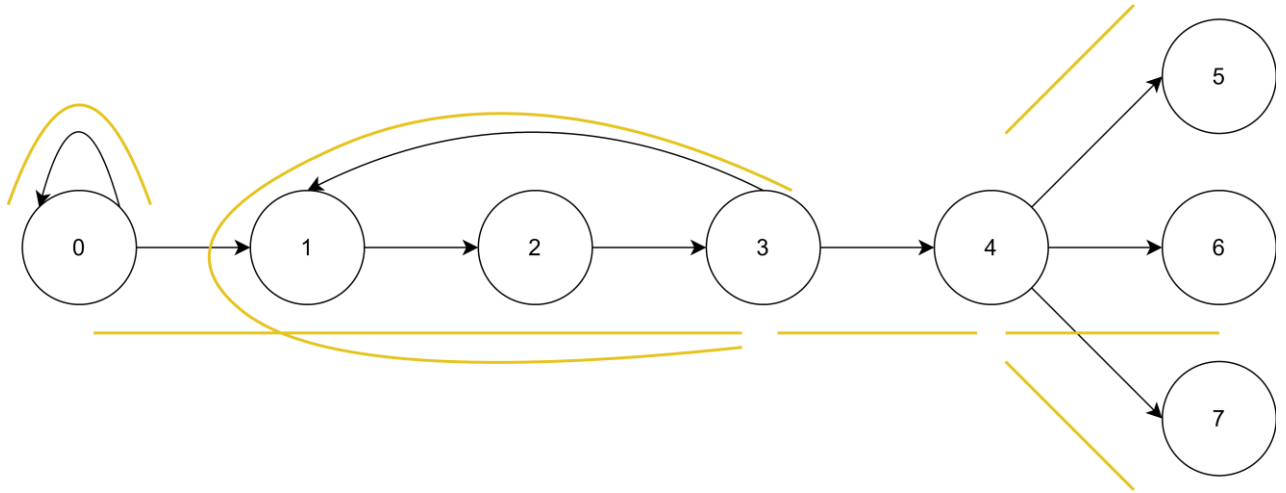


Formal Property Generation

accellera
SYSTEMS INITIATIVE

Complete Formal Verification of State Automata

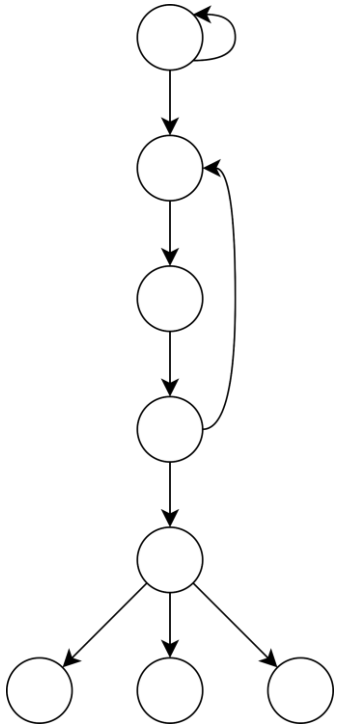
- There exists a finite set of fixed-length intervals from which any execution sequence of an automaton can be constructed[1]
- Extracting this set of intervals follows a set of rules
  - Intervals must end at branches
  - One interval per branch
  - One interval per loop



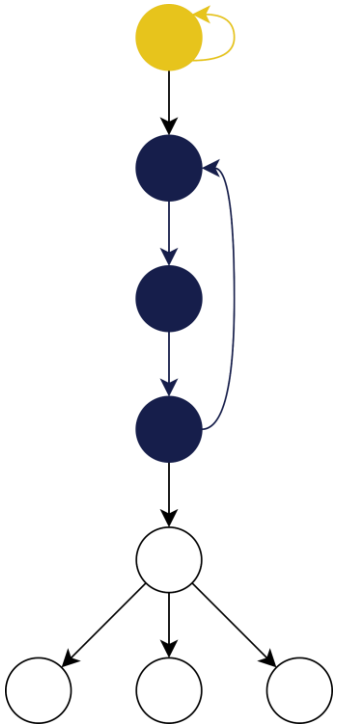
[1]https://kluedo.ub.rptu.de/frontdoor/index/index/year/2017/docId/4680

Automated Interval Extraction

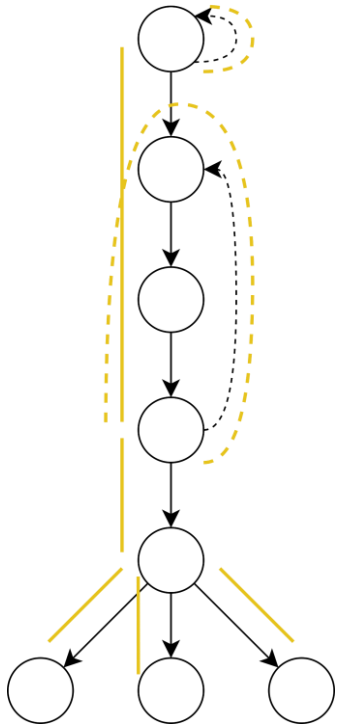
Directed Cyclic Graph



Loop Detection

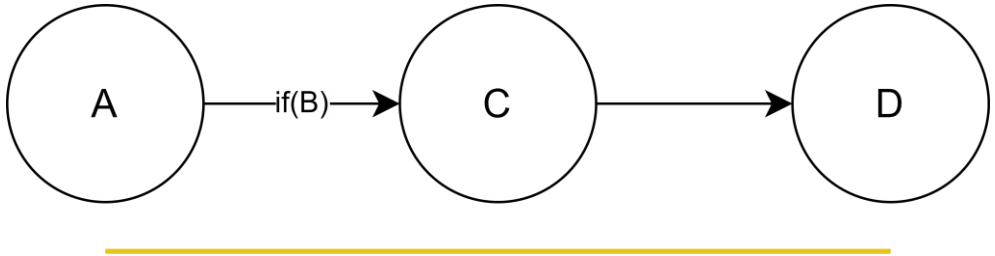


Interval Extraction



Property Generation

- Per Interval
  - All transition triggers -> precondition
  - All state actions -> postcondition




```
property ABCD;
@(posedge clk)
     A && B
|-> ##1
     C ##1 D;
endproperty
```

Application and Results

accellera
SYSTEMS INITIATIVE

Traces Save Modeling Effort

Source Files	Lines-of-Code					
	AHB (S)	AHB (M)	Register File (32x32)	FIFO (8x32)	DMA	CPU (RV32IMC)
Trace Model	81	131	123	213	214	213
Properties Generator	472					
Formal Properties	144	229	500	341	994	1356

Efficiently Reusing and Adapting Traces

CPU (RV32IMC)	Lines-of-Code		
	1-stage (naïve)	2-stage	3-stage
Trace Model	134	-	-
Trace Refinement	-	+79	+14
Properties Generator	472		
Formal Properties	1230	1356	1382

Conclusion

accellera
SYSTEMS INITIATIVE

Traces vs. SVA Sequences

Traces

- Custom data structure
- Extensive refinement possible
- Generates correct sequences

Sequences

- Standard SVA data structure
- Limited adaptability
- Engineer responsible for correctness

The key difference between traces and SVA sequences is the extraction of a complete set of interval properties from traces

Conclusion

- Traces formalize specification
  - Capture **functional and temporal behavior**
- Trace format is built with reusability in mind
  - Hierarchical structure to **reuse sub-traces**
  - Refinement **saves modeling time**
- Highly reusable code generators **save verification effort**

Questions?

accellera
SYSTEMS INITIATIVE