FSMT

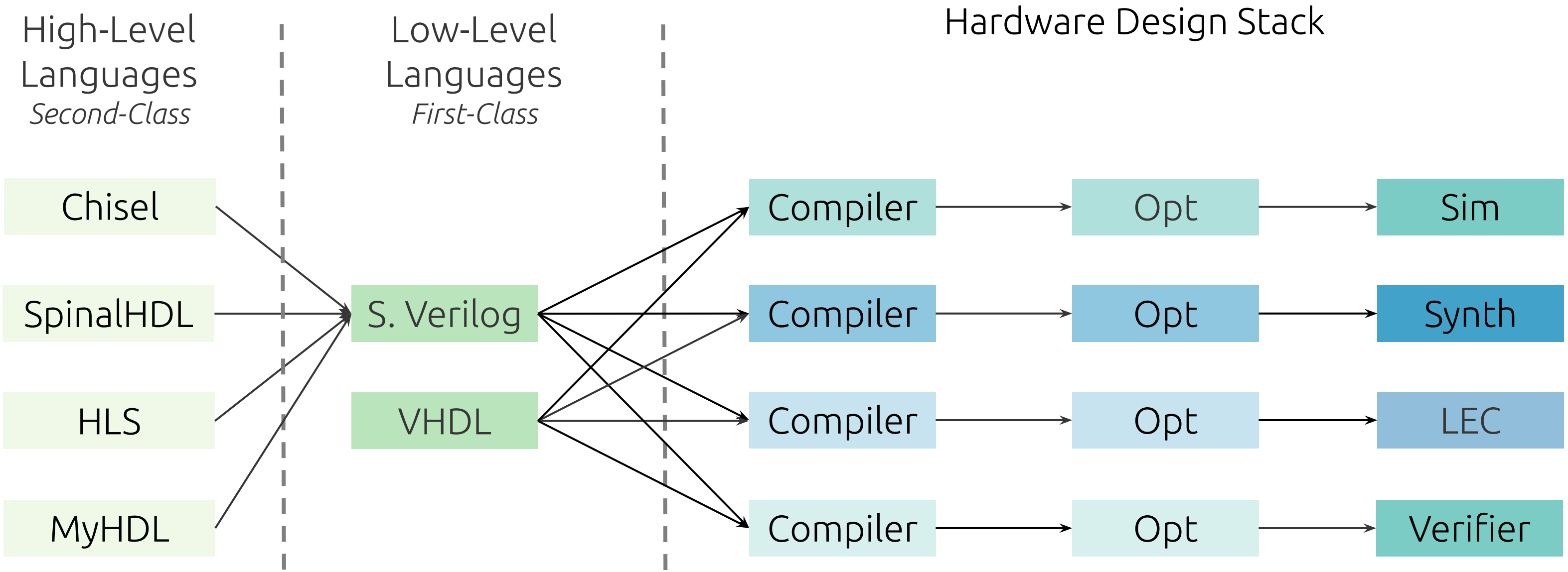
Exploiting Wasted Hardware Abstractions For Efficient Model Checking

LLVM Developers’ Meeting 2024 - Santa Clara, 23 October 2024

Luisa Cicolini, Bea Healy, Tobias Grosser

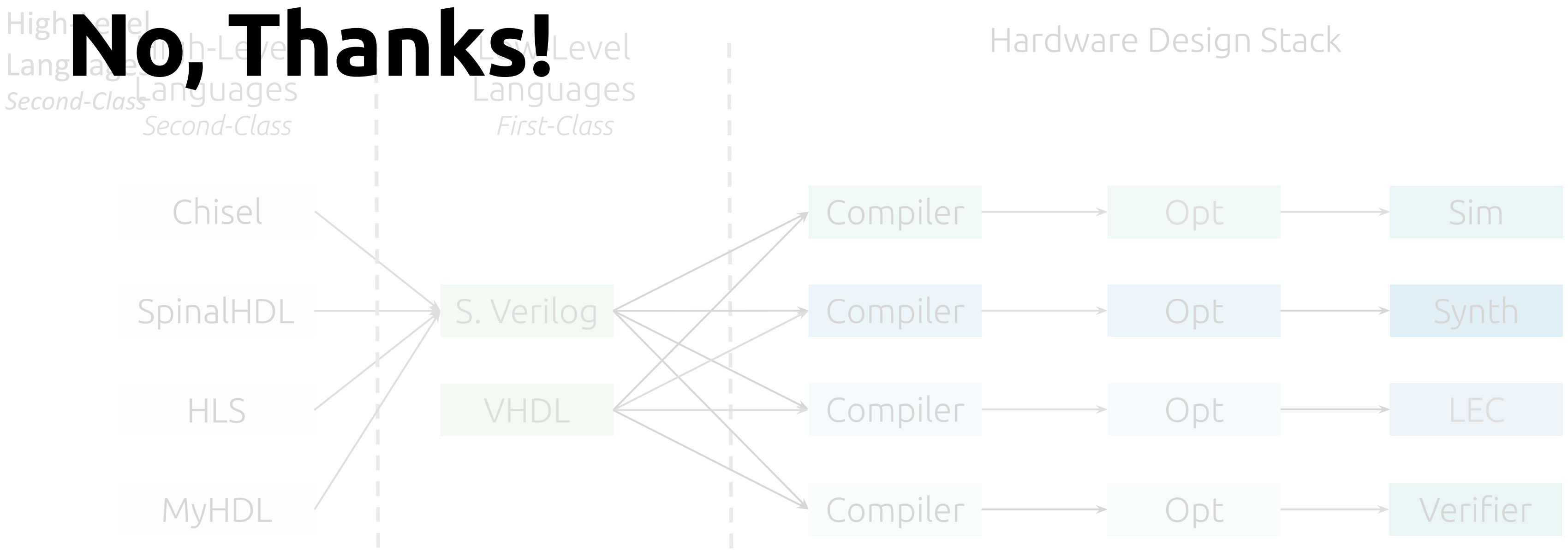
what’s that?

Traditional Hardware Design



Traditional Hardware Design

No, Thanks!



Traditional Hardware Design

No, Thanks!

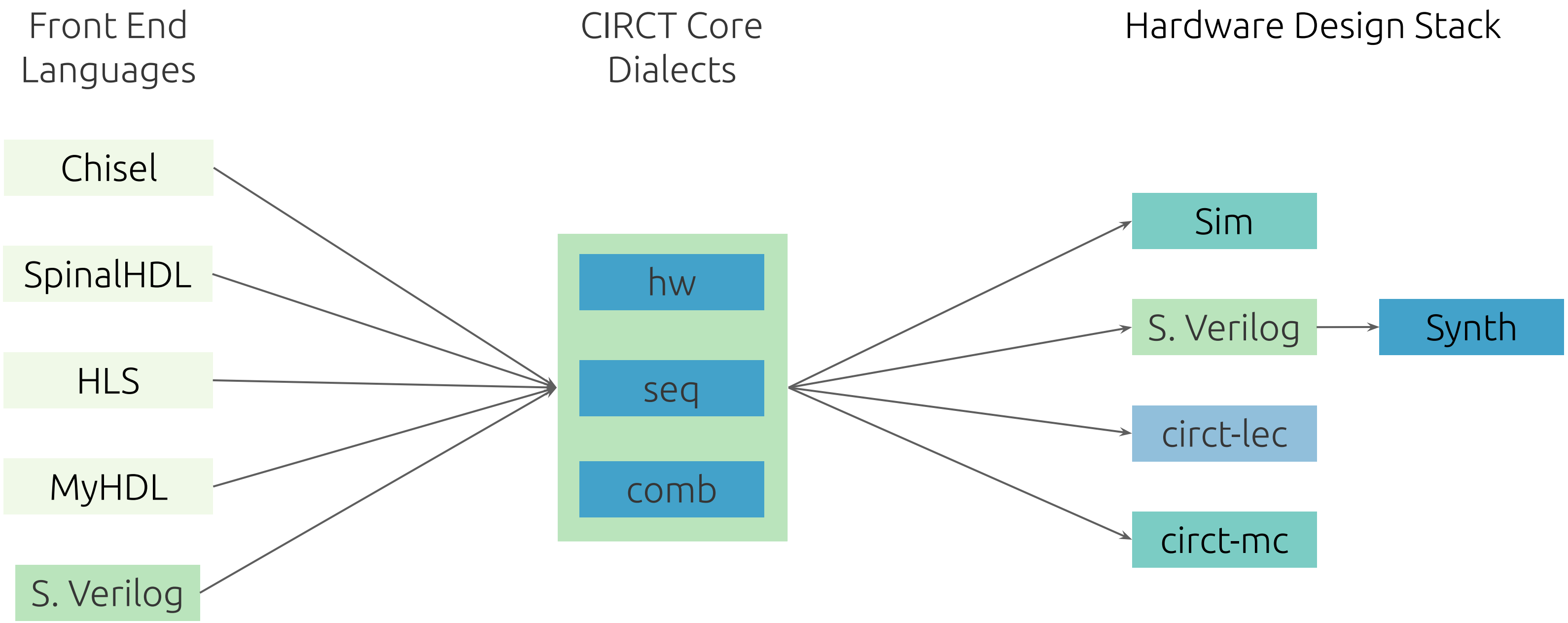
Many tools to learn
Little Reuse
Implementations Repeated
Redundancy

Traditional Hardware Design

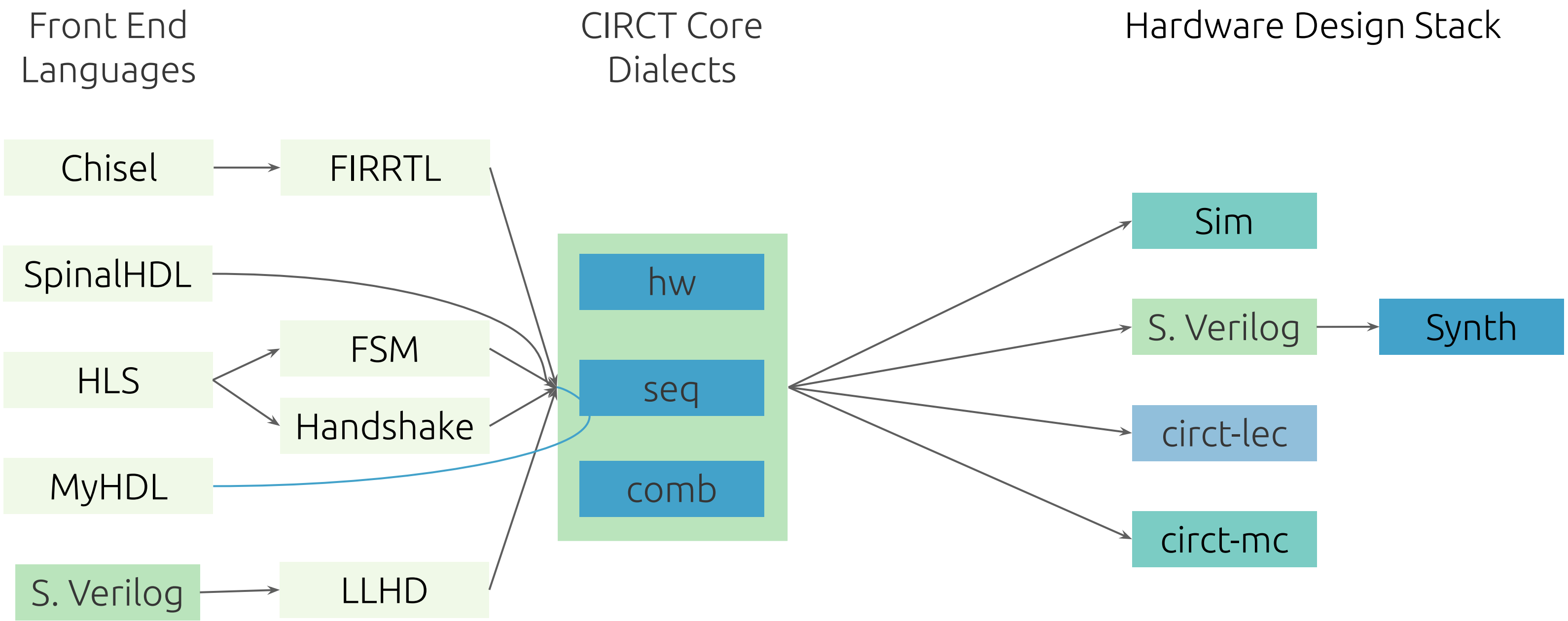
No, Thanks!

Many tools to learn
Little Reuse
Implementations Repeated
Redundancy

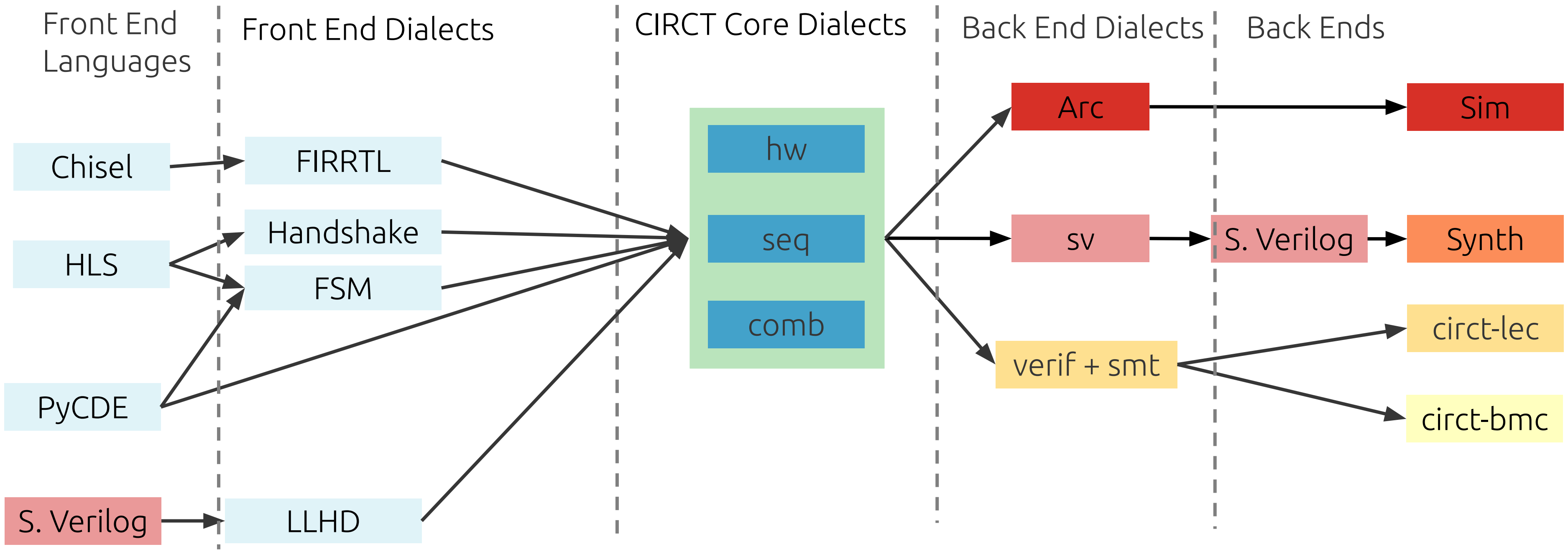
CIRCT

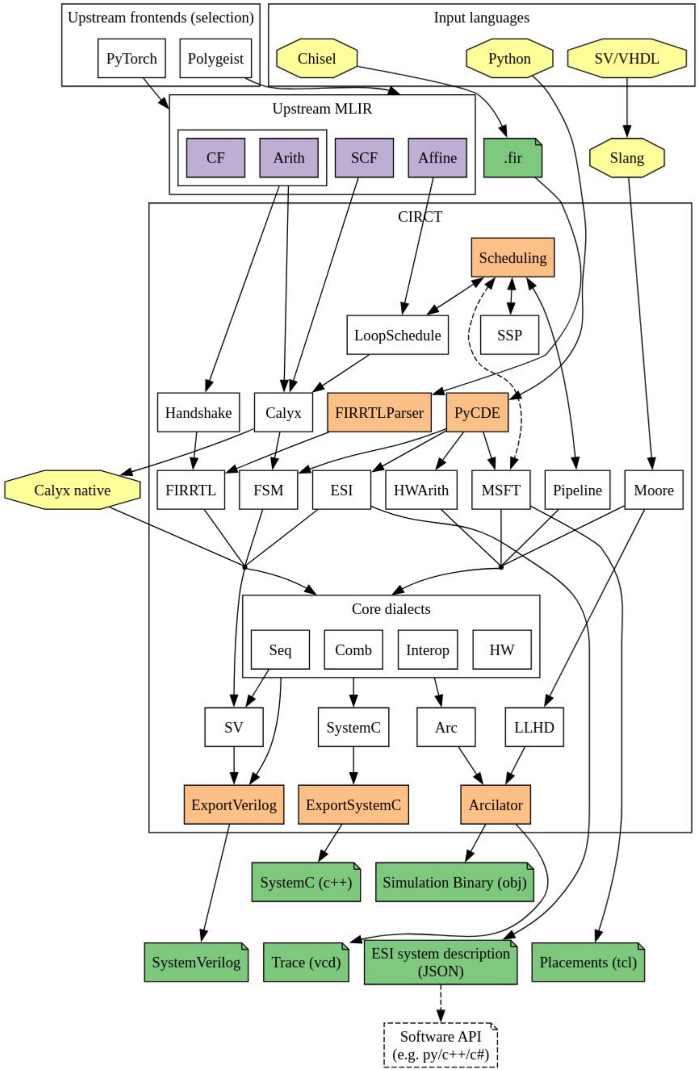

Front End Languages
CIRCT Core Dialects
Hardware Design Stack
Chisel
SpinalHDL
HLS
MyHDL
S. Verilog
hw
seq
comb
Sim
S. Verilog
Synth
circt-lec
circt-mc

Traditional Hardware Design



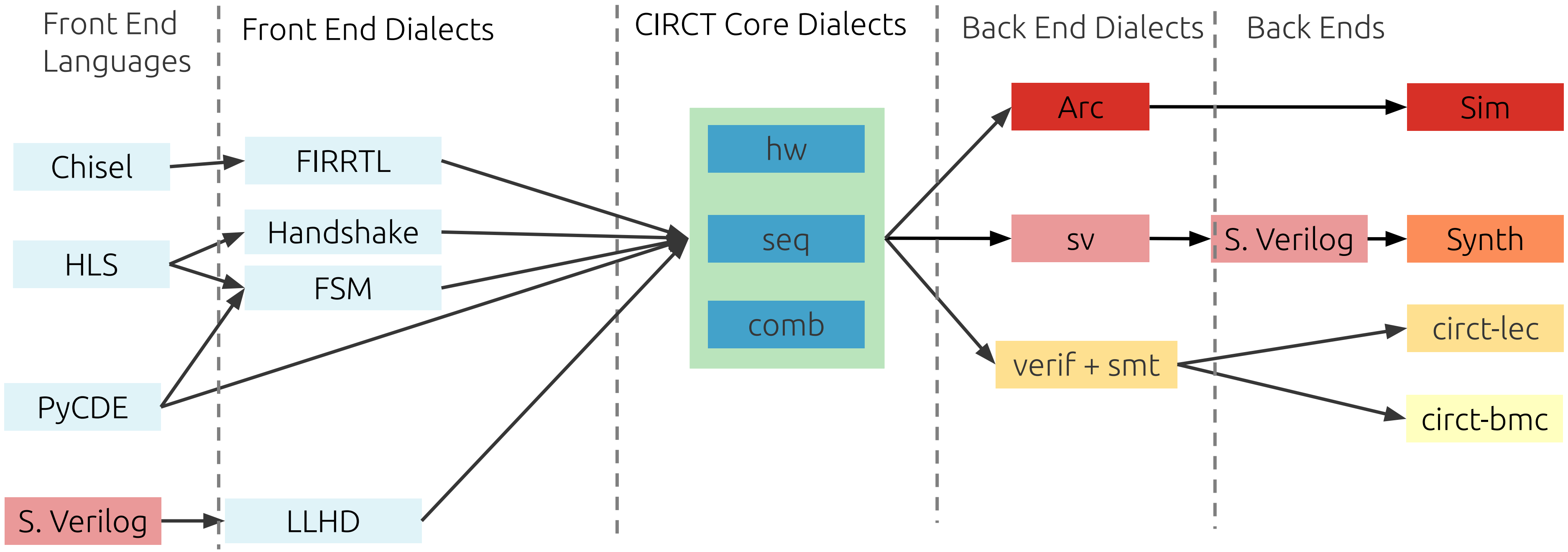
CIRCT


Front End Languages
Front End Dialects
CIRCT Core Dialects
Back End Dialects
Back Ends
Chisel
HLS
PyCDE
S. Verilog
FIRRTL
Handshake
FSM
LLHD
hw
seq
comb
Arc
sv
verif + smt
Sim
S. Verilog
Synth
circt-lec
circt-bmc

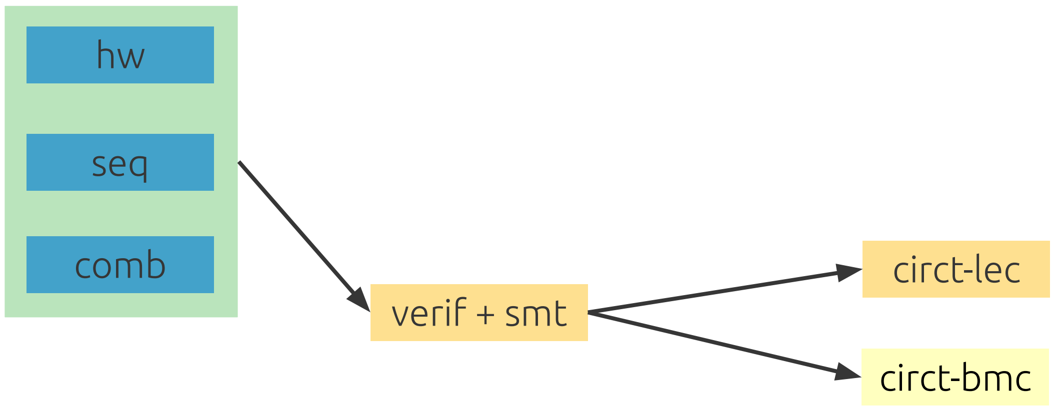


Upstream frontends (selection)
PyTorch
Polygeist
Input languages
Chisel
Python
SV/VHDL
Upstream MLIR
CF
Arith
SCF
Affine
.fir
Slang
CIRCT
Scheduling
LoopSchedule
SSP
Handshake
Calyx
FIRRTLParser
PyCDE
Calyx native
FIRRTL
FSM
ESI
HWArith
MSFT
Pipeline
Moore
Core dialects
Seq
Comb
Interop
HW
SV
SystemC
Arc
LLHD
ExportVerilog
ExportSystemC
Arcilator
SystemC (c++)
Simulation Binary (obj)
SystemVerilog
Trace (vcd)
ESI system description (JSON)
Placements (tcl)
Software API (e.g. py/c++/c#)

CIRCT


Front End Languages
Front End Dialects
CIRCT Core Dialects
Back End Dialects
Back Ends
Chisel
HLS
PyCDE
S. Verilog
FIRRTL
Handshake
FSM
LLHD
hw
seq
comb
Arc
sv
verif + smt
Sim
S. Verilog
Synth
circt-lec
circt-bmc

CIRCT - Verification



What about Verification?

Safety Properties

"Something never happens"

What about Verification?

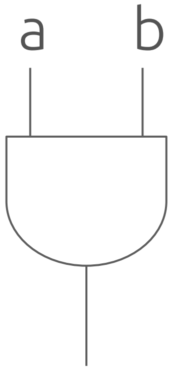
Safety Properties

"Something never happens"

Reachability Properties

"A certain state is eventually reached in any execution"

CIRCT-MC: RTL-level Model Checking

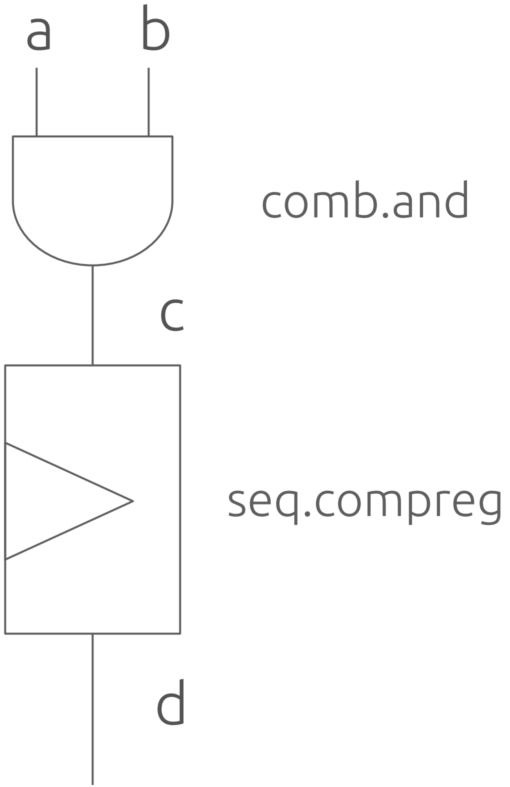


comb.and

CIRCT:

`%c = comb.and %a, %b : i1`

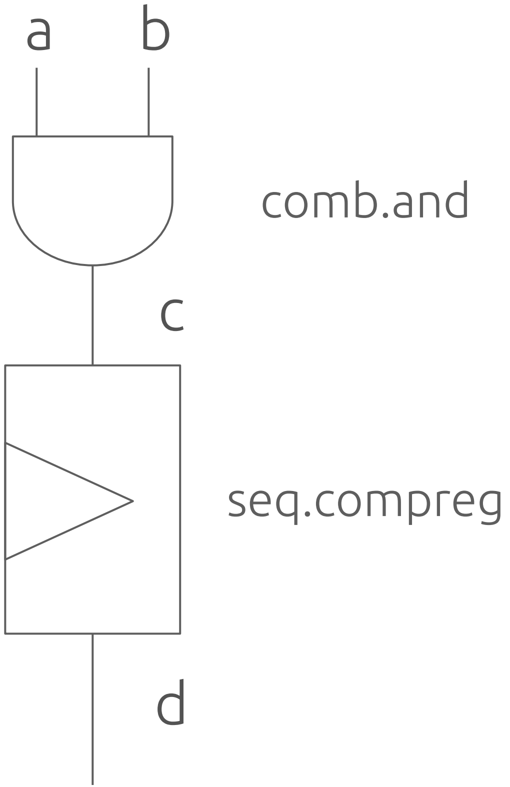
CIRCT-MC: RTL-level Model Checking



CIRCT:

```
%c = comb.and %a, %b : i1
%d = seq.compreg %c, clk %clk : i1
```

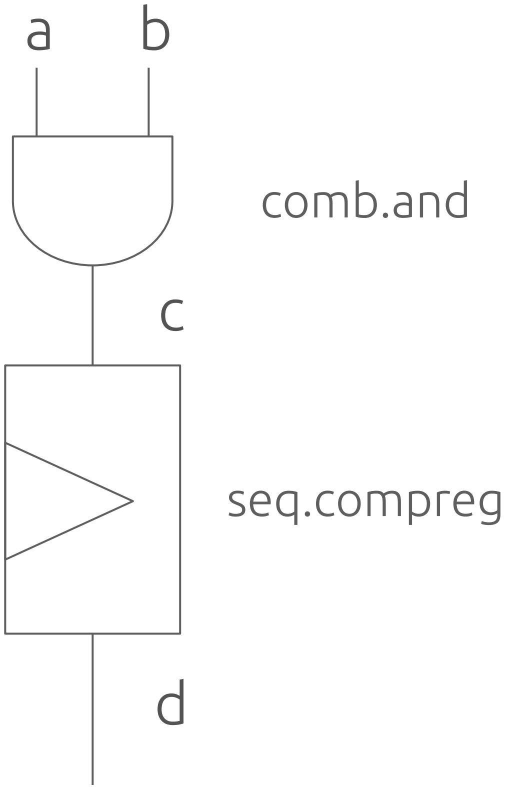
CIRCT-MC: RTL-level Model Checking



CIRCT:

$$\underbrace{\begin{array}{l}\%c = \text{comb.and } \%a, \%b : i1 \\ \%d = \text{seq.compreg } \%c, \text{clk } \%clk : i1\end{array}}_{\text{SMT Variables}}$$

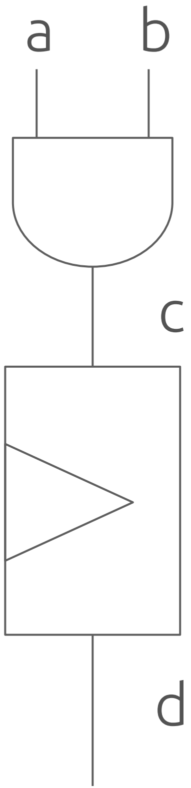
CIRCT-MC: RTL-level Model Checking



CIRCT:

$$\underbrace{\begin{array}{l}\text{\%c = comb.and \%a, \%b : i1} \\ \text{\%d = seq.compreg \%c, clk \%clk : i1}\end{array}}_{\text{SMT Formula}}$$

CIRCT-MC: RTL-level Model Checking



comb.and

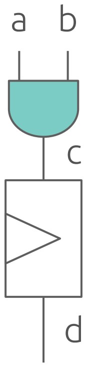
seq.compreg

CIRCT:

```
%c = comb.and %a, %b : i1
%d = seq.compreg %c, clk %clk : i1
```

Register List

CIRCT-MC: RTL-level Model Checking



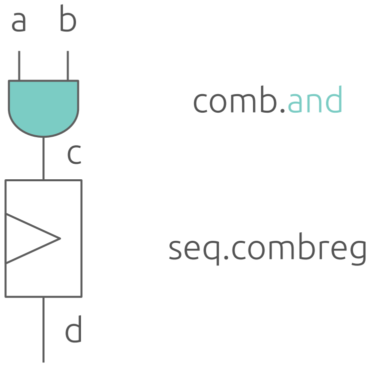
comb.and

seq.combreg

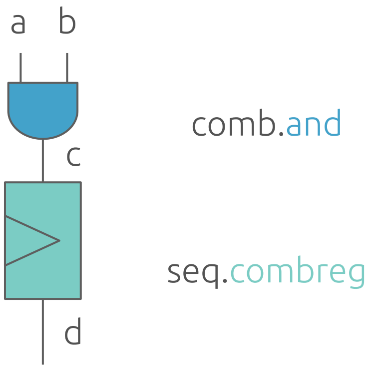
clock cycle = 0

```
Vars: x, y, z
%a = x
%b = y
%c = x && y
%d = z
```

CIRCT-MC: RTL-level Model Checking



clock cycle = 0

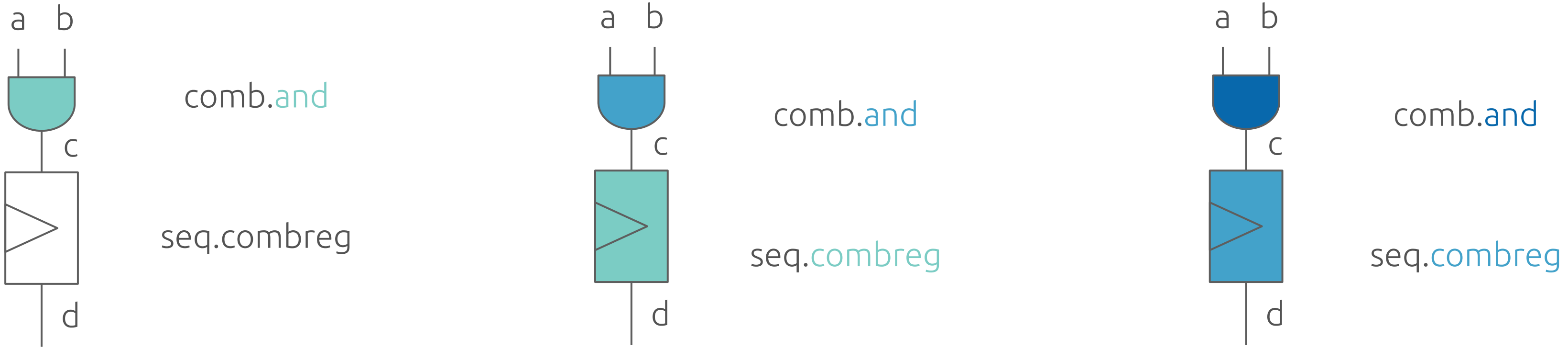


clock cycle = 1

Vars: x, y, z
%a = x
%b = y
%c = x && y
%d = z

Vars: x, y, z, x_1, y_1
%a = x_1
%b = y_1
%c = x_1 && y_1
%d = x && y

CIRCT-MC: RTL-level Model Checking



clock cycle = 0

Vars: x, y, z
%a = x
%b = y
%c = x && y
%d = z

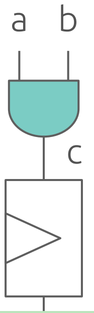
clock cycle = 1

Vars: x, y, z, x_1, y_1
%a = x_1
%b = y_1
%c = x_1 && y_1
%d = x && y

clock cycle = 2

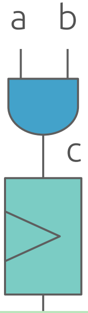
Vars: x, y, z, x_1, y_1, x_2, y_2
%a = x_2
%b = y_2
%c = x_2 && $_2$
%d = x_1 && y_1

CIRCT-MC: RTL-level Model Checking



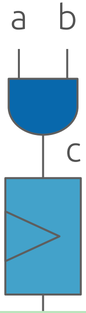
comb.and

seq.combreg



comb.and

seq.combreg



comb.and

seq.combreg

Unrolled Bounded Model Checking

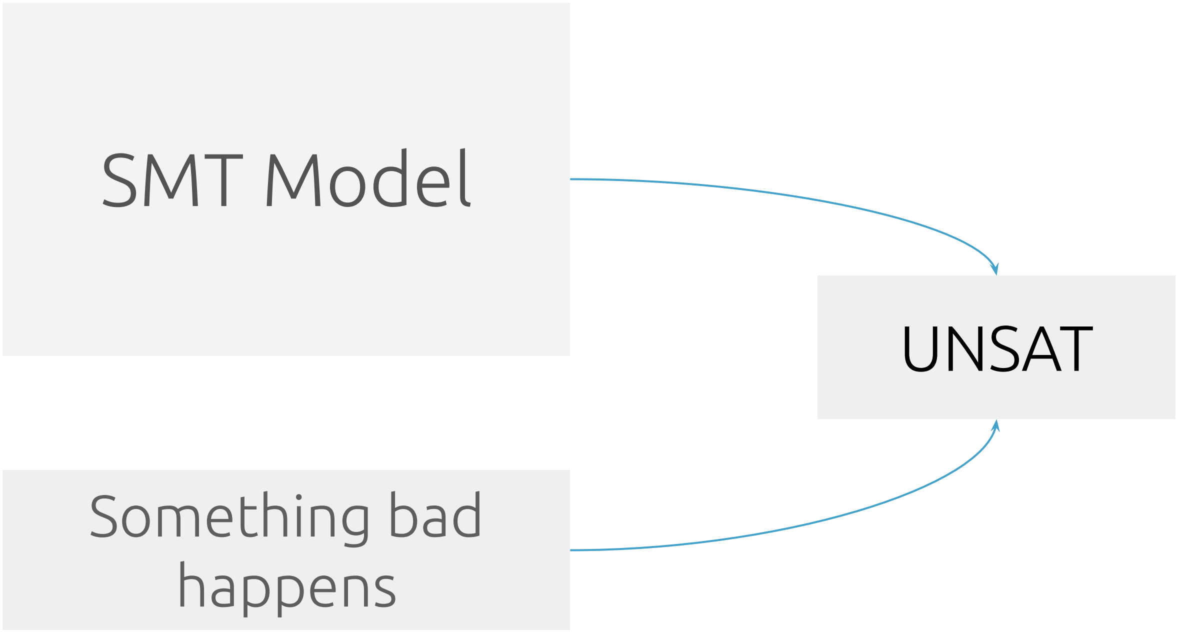
$\%b = y$
$\%c = x \ \&\& \ y$
$\%d = z$

$\%b = y_1$
$\%c = x_1 \ \&\& \ y_1$
$\%d = x \ \&\& \ y$

$\%a = x_2$
$\%b = y_2$
$\%c = x_2 \ \&\& \ _2$
$\%d = x_1 \ \&\& \ y_1$

CIRCT-MC: RTL-level Model Checking

∀ clock cycle:



CIRCT-MC: RTL-level Model Checking

$\forall$ clock cycle:

SMT Model

$\neg$ Safety Property

UNSAT

CIRCT-MC: RTL-level Model Checking

∀ clock cycle:

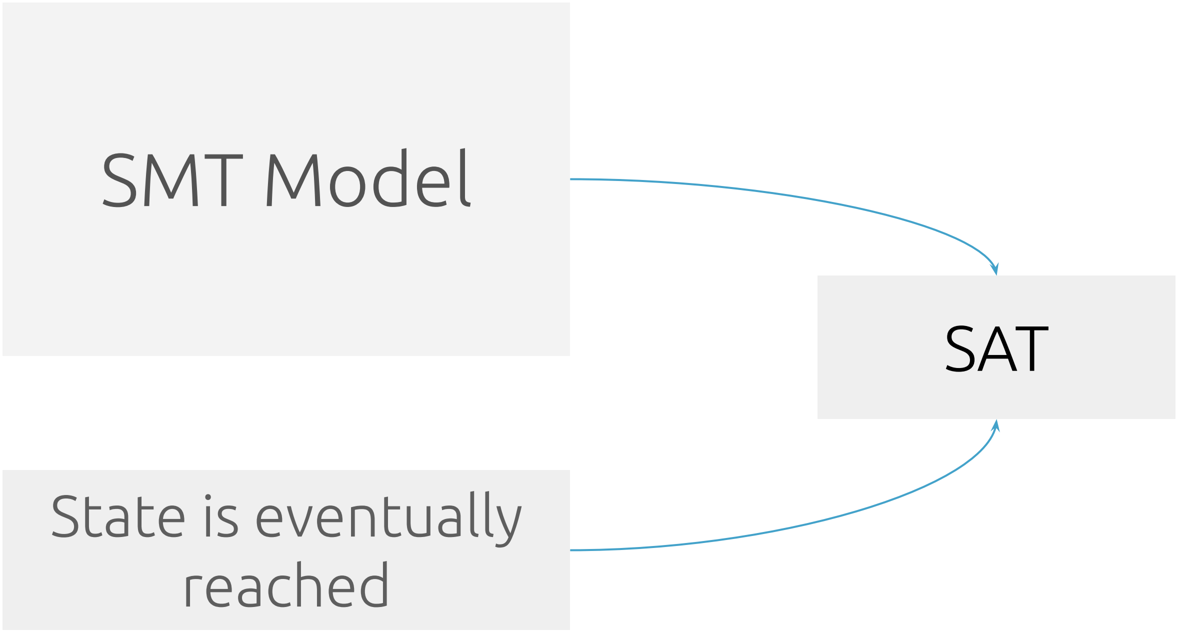
SMT Model

UNSAT if the safety property holds

¬ Safety Property

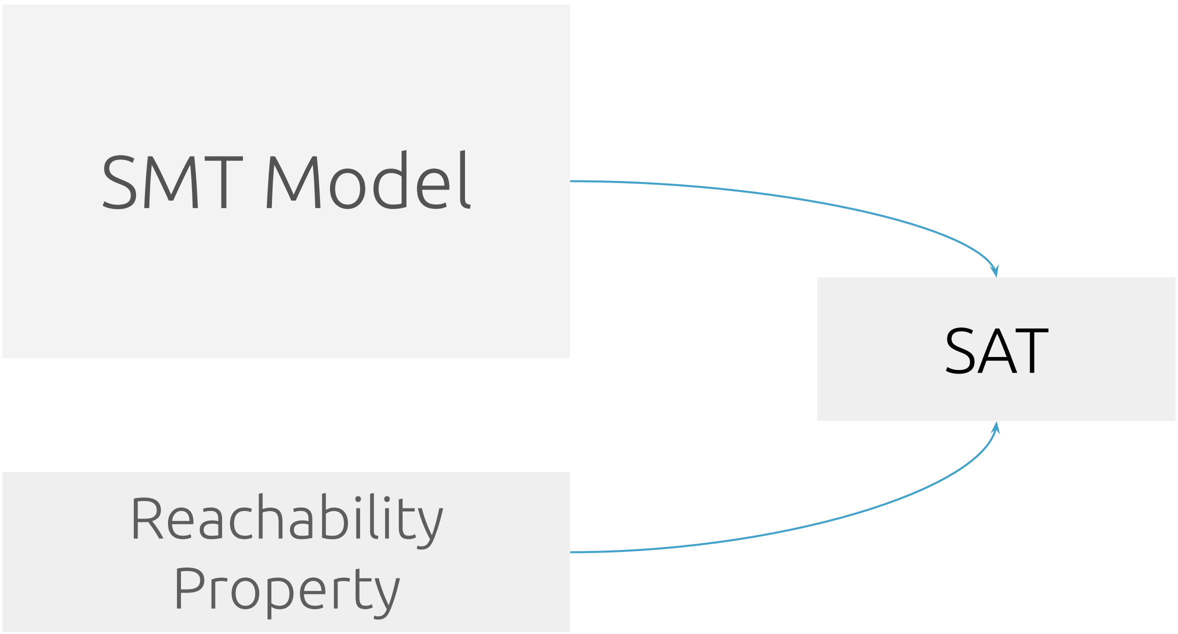
CIRCT-MC: RTL-level Model Checking

∀ clock cycle:



CIRCT-MC: RTL-level Model Checking

∀ clock cycle:



CIRCT-MC: RTL-level Model Checking

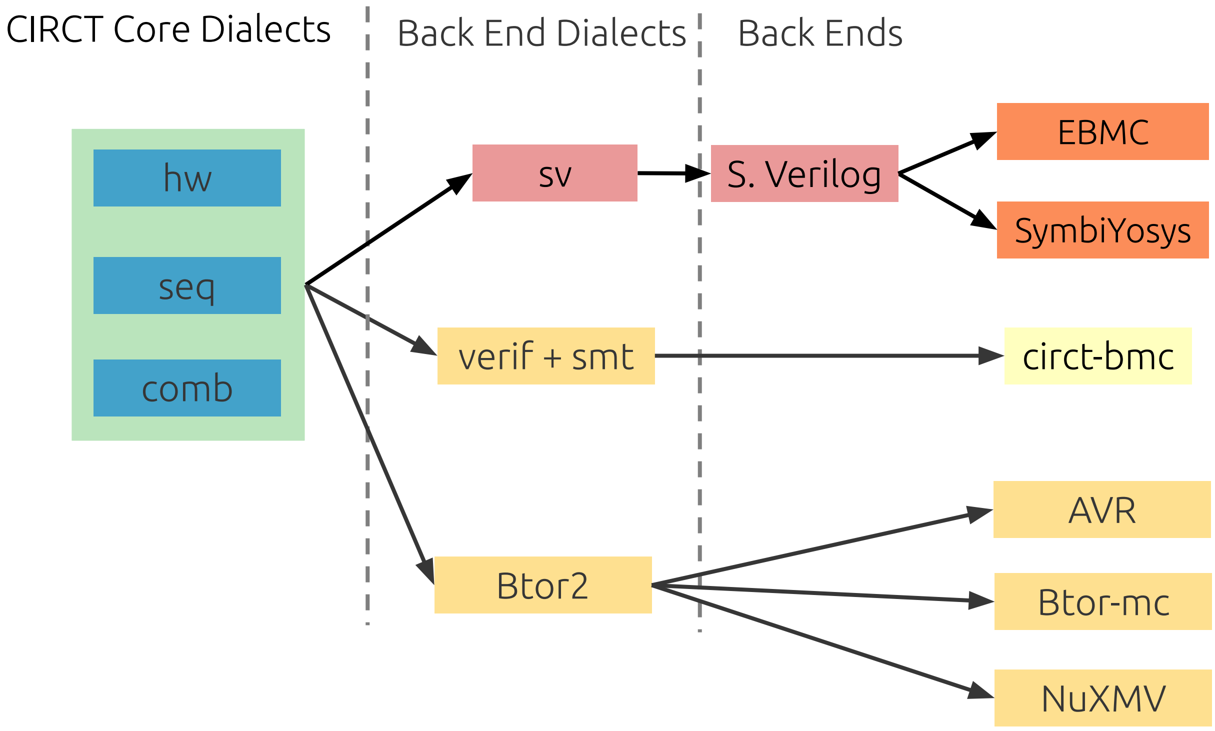
∀ clock cycle:

SMT Model

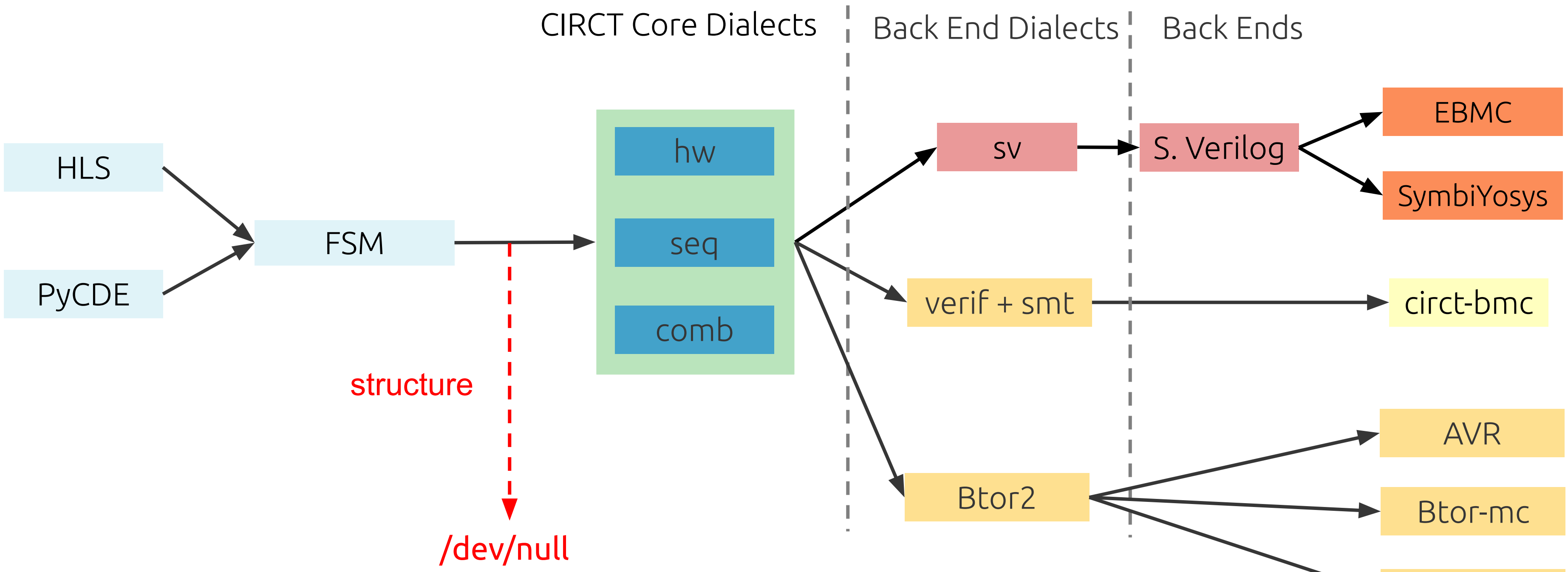
SAT if the liveness property is covered

Reachability
Property

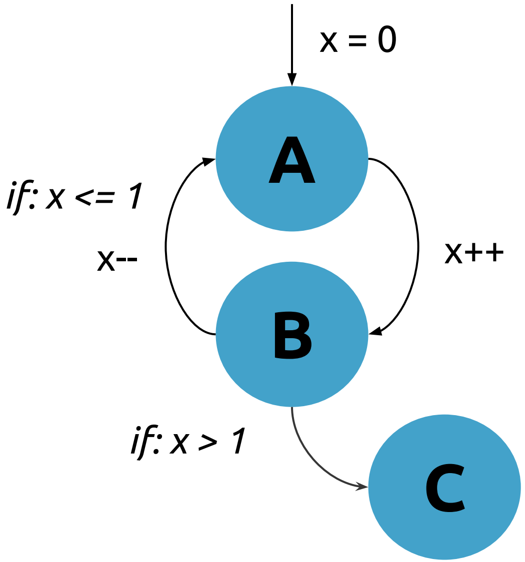
CIRCT - Verification



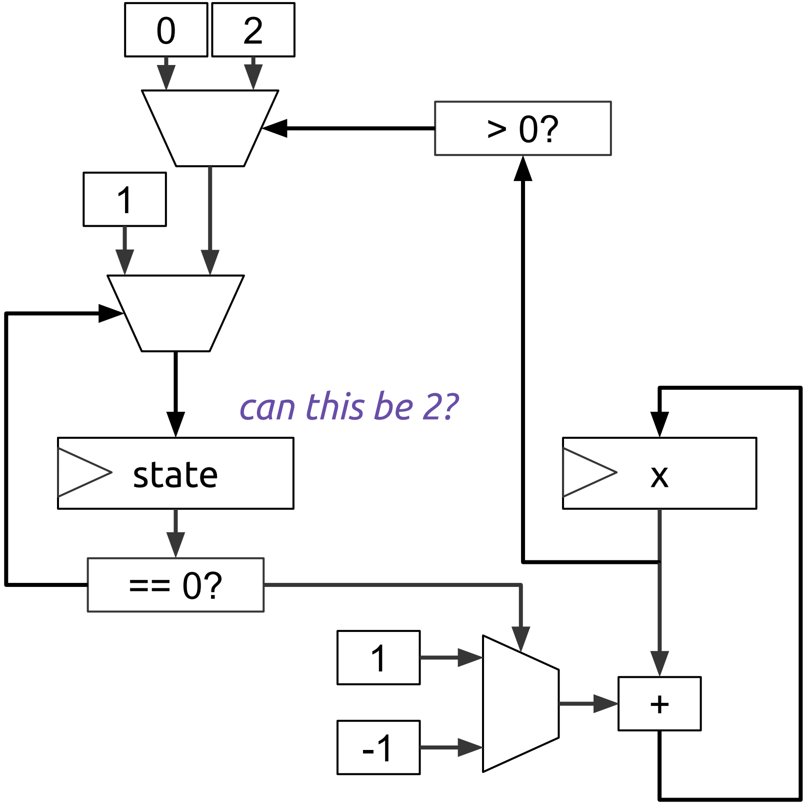
CIRCT - Verification



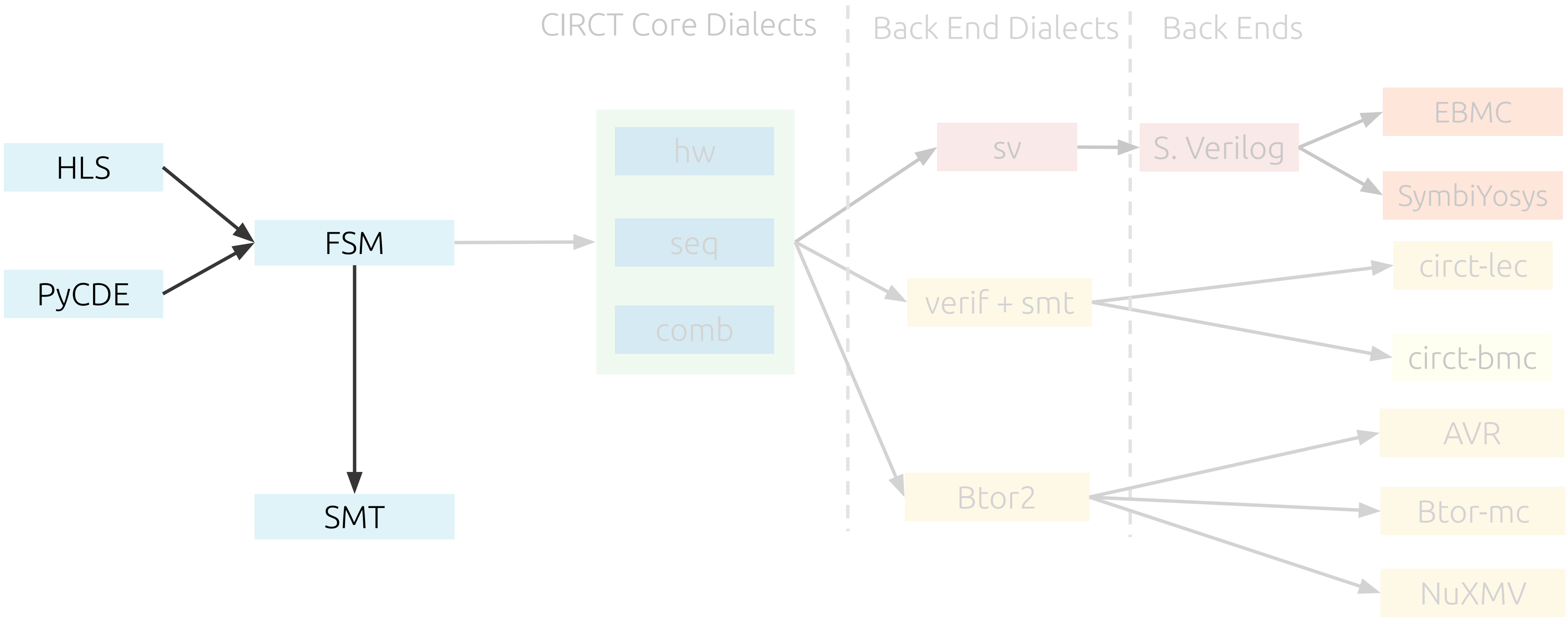
CIRCT - Verification



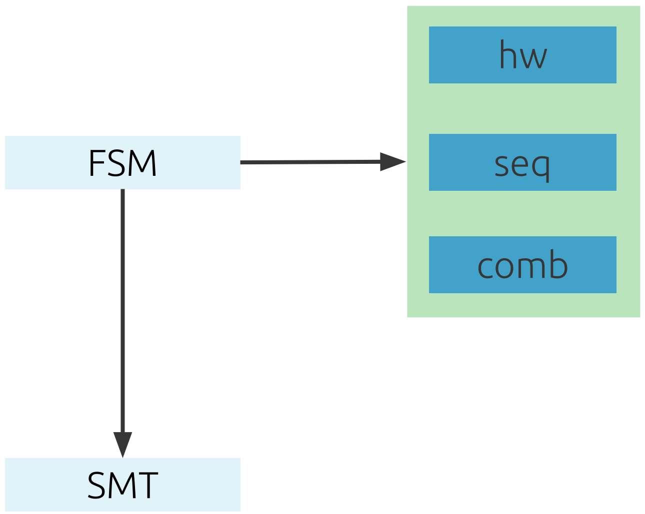
can state C be reached?



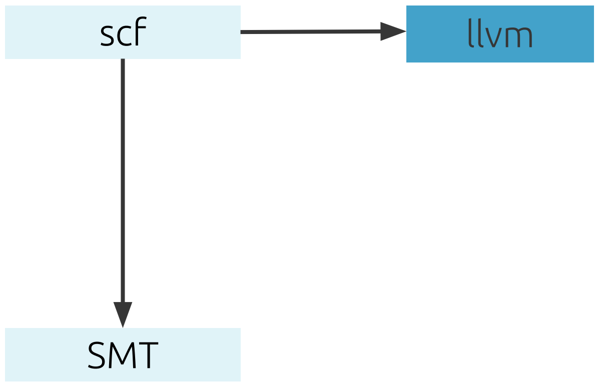
CIRCT - Verification



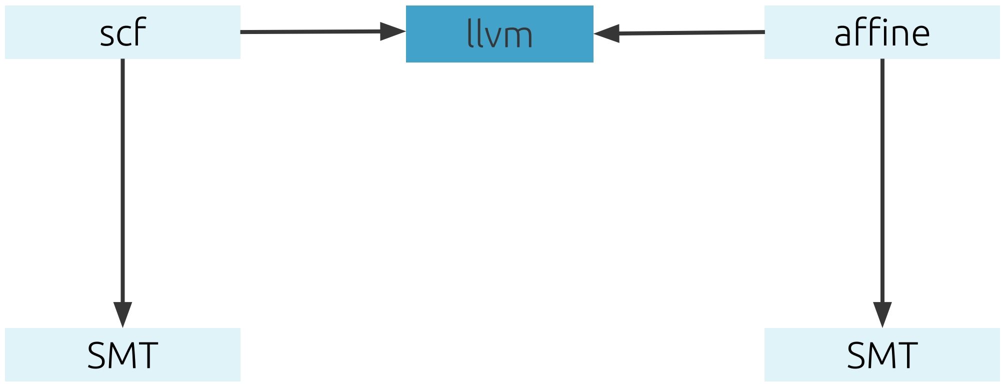
CIRCT - Verification



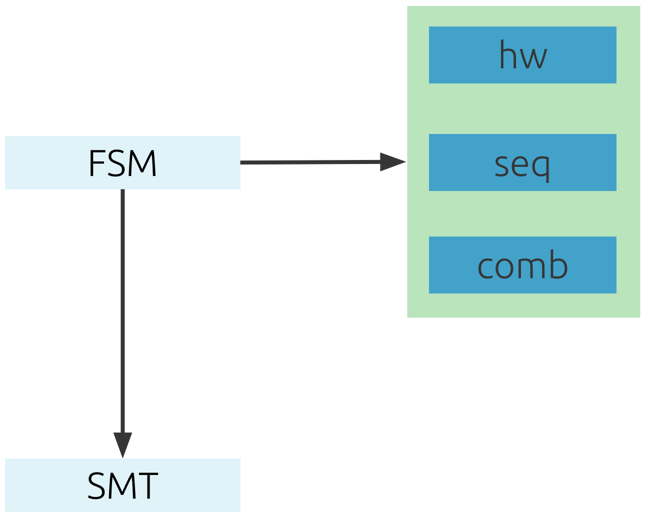
Not just for CIRCT!



Not just for CIRCT!



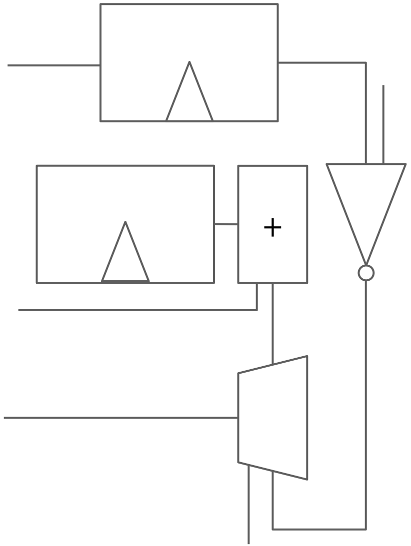
CIRCT - Verification



Traditional Hardware Design

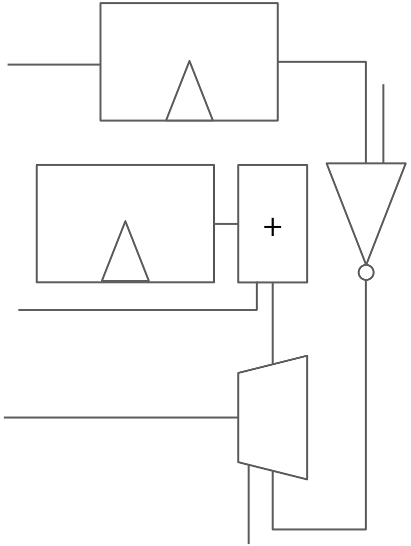
Frontends

Traditional Hardware Design



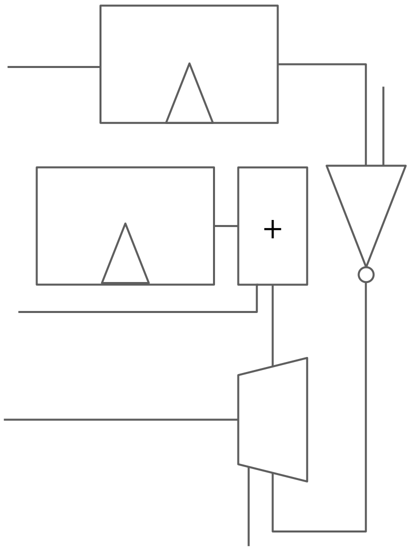
Frontends → RTL

Traditional Hardware Design



Frontends → RTL → Verilog

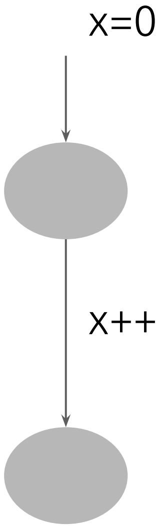
Multi-Level Hardware Design in CIRCT[1]

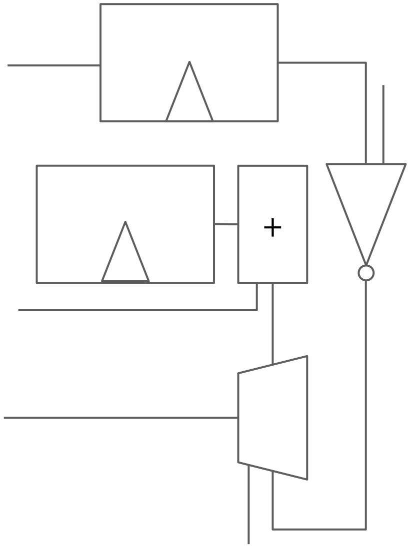


Frontends → High Level IR → RTL → Verilog

[1] https://circt.llvm.org/

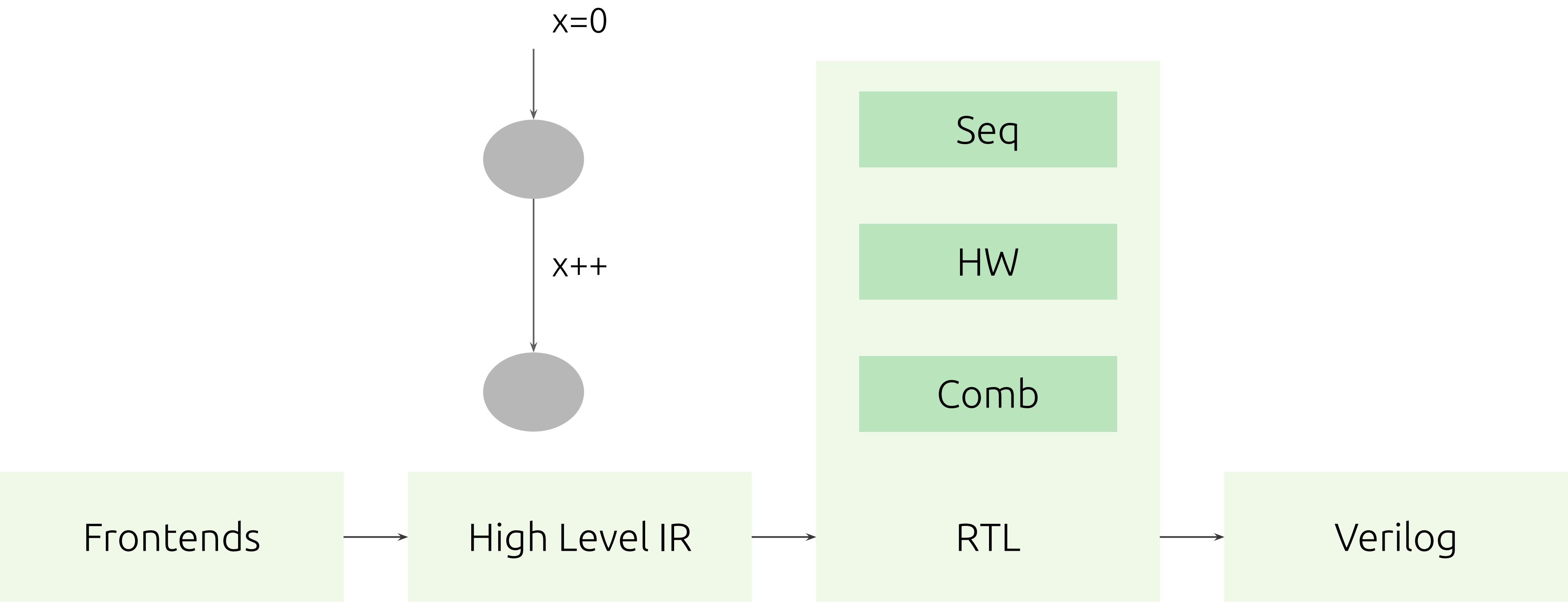
Multi-Level Hardware Design in CIRCT



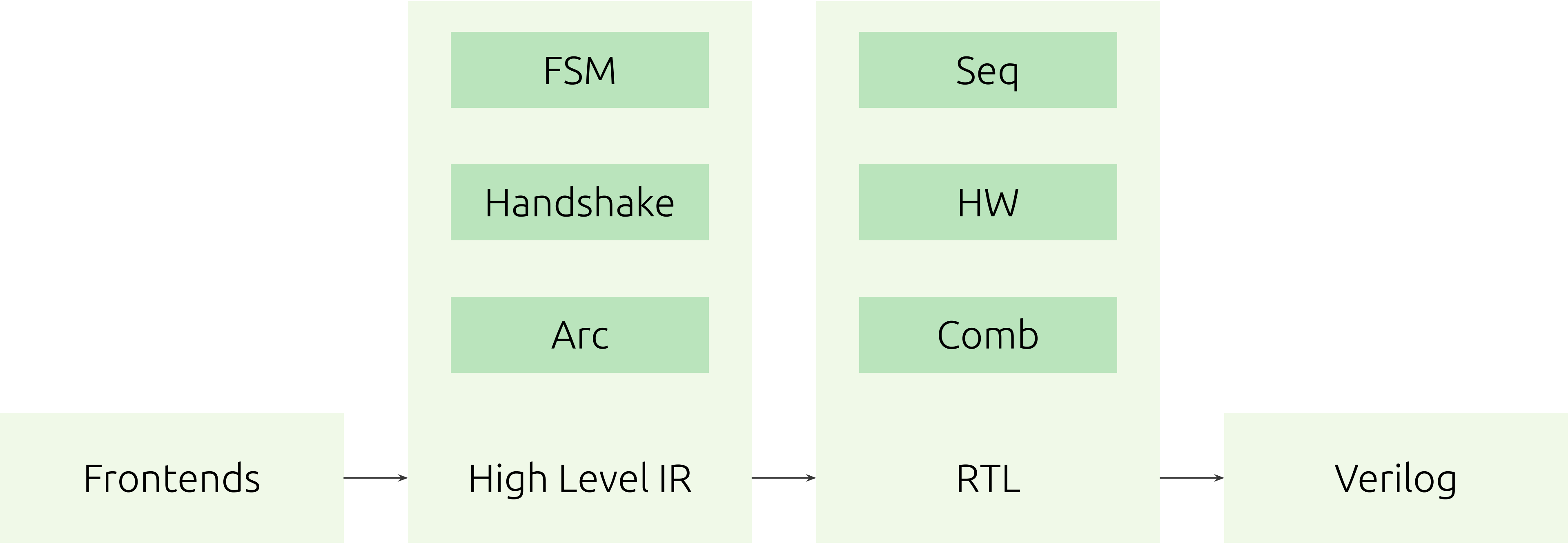


Frontends → High Level IR → RTL → Verilog

Multi-Level Hardware Design in CIRCT



Multi-Level Hardware Design in CIRCT



What about Verification?

What about Verification?

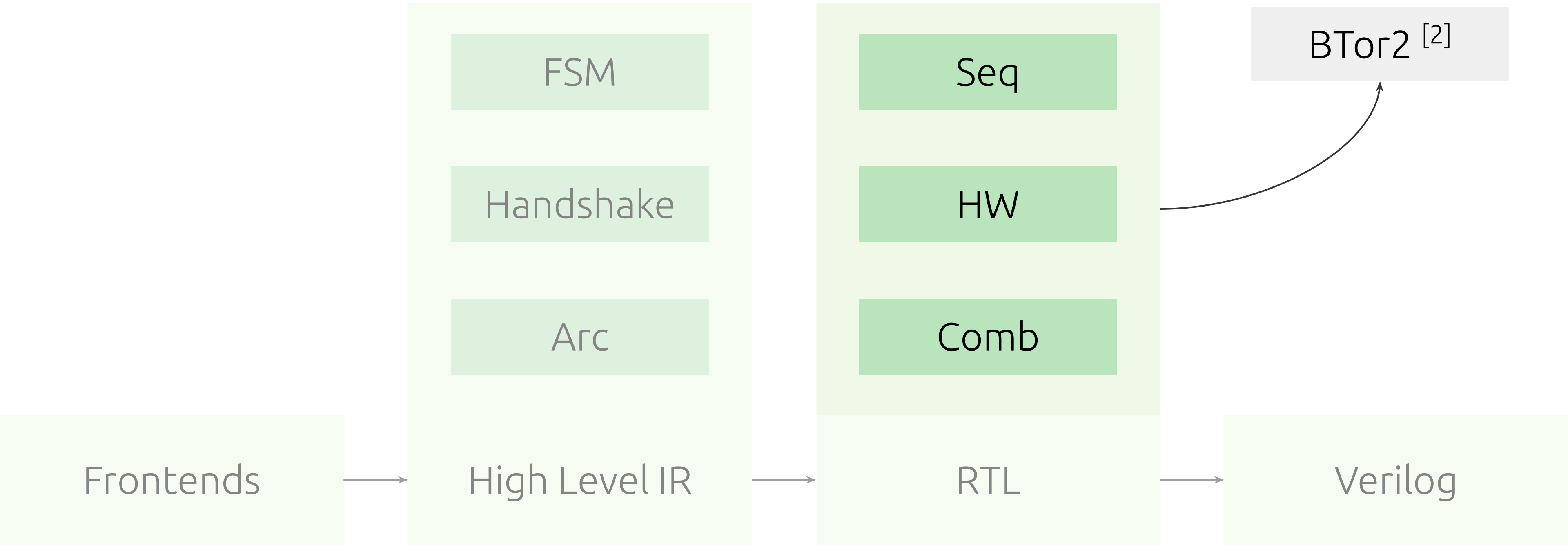
Safety Properties

“Something never happens”

What about Verification?

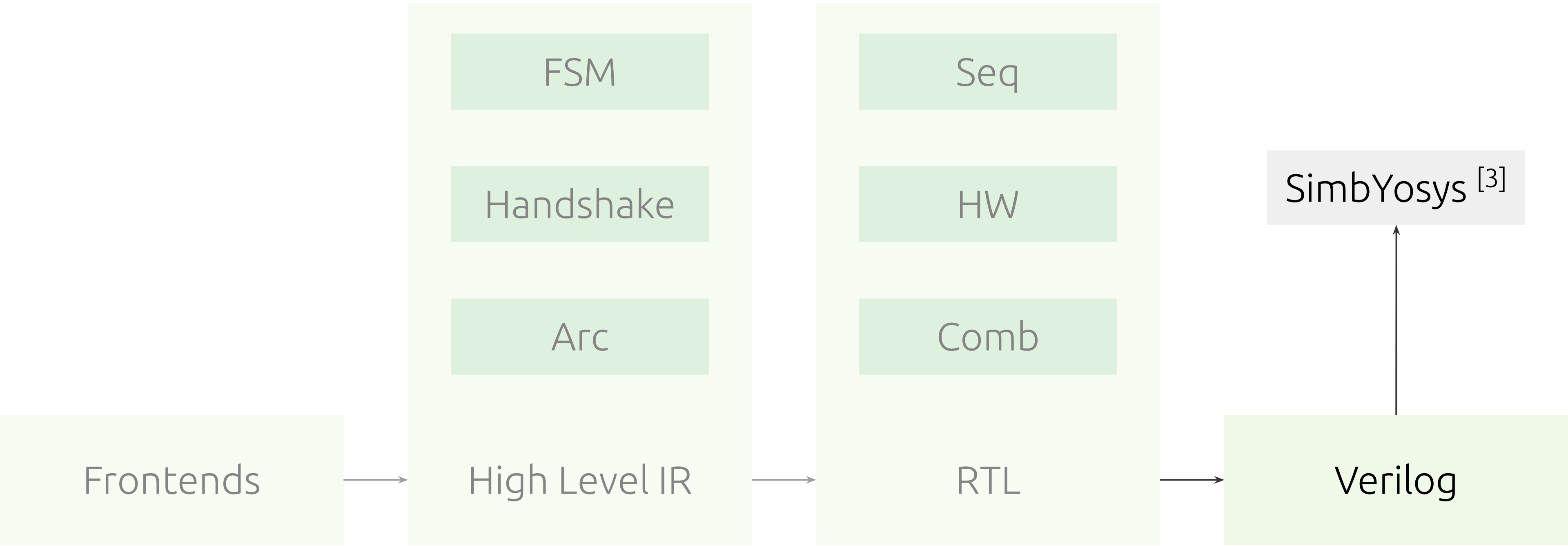
Safety Properties	“Something never happens”
Reachability Properties	“A certain state is eventually reached”

Verification in CIRCT



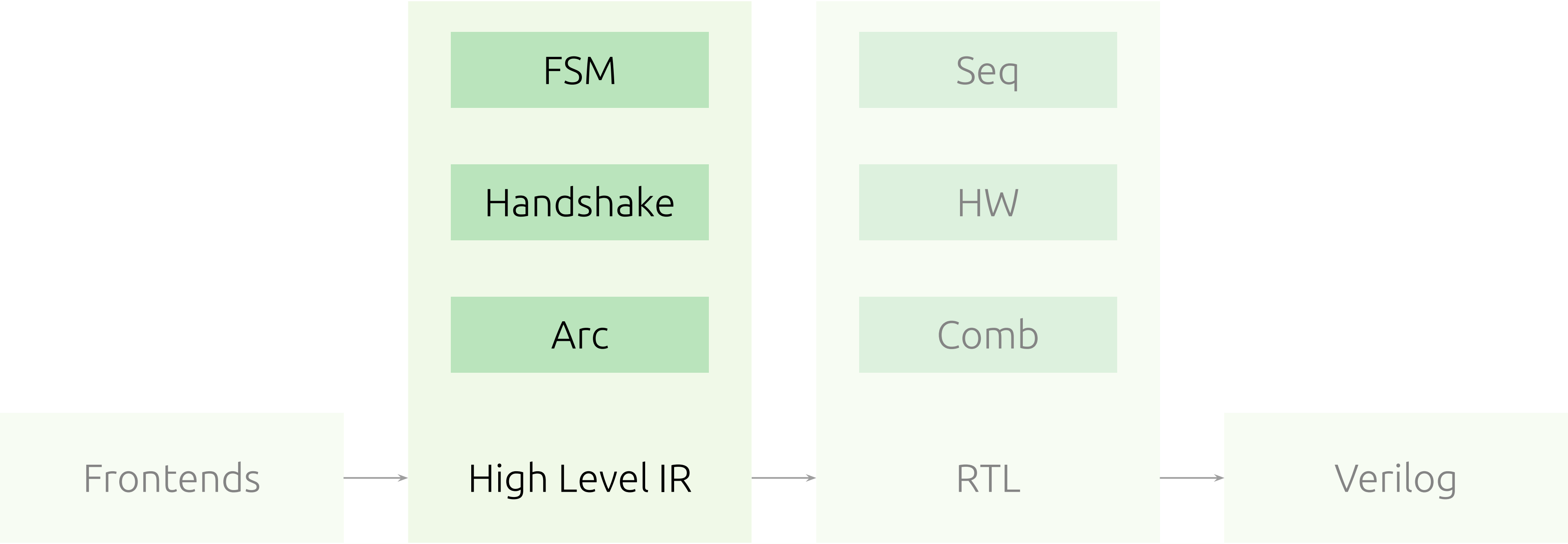
[2] Dobis et al, 2023, Verification of Chisel Hardware Designs with ChiselVerify. Microprocessors and Microsystems 96 (2023), 104737

Verification in CIRCT



[3] SymbiYosys (sby) – Front-end for Yosys-based formal verification flows. ([n. d.]). https://github.com/YosysHQ/SymbiYosys, Accessed 24/11/21

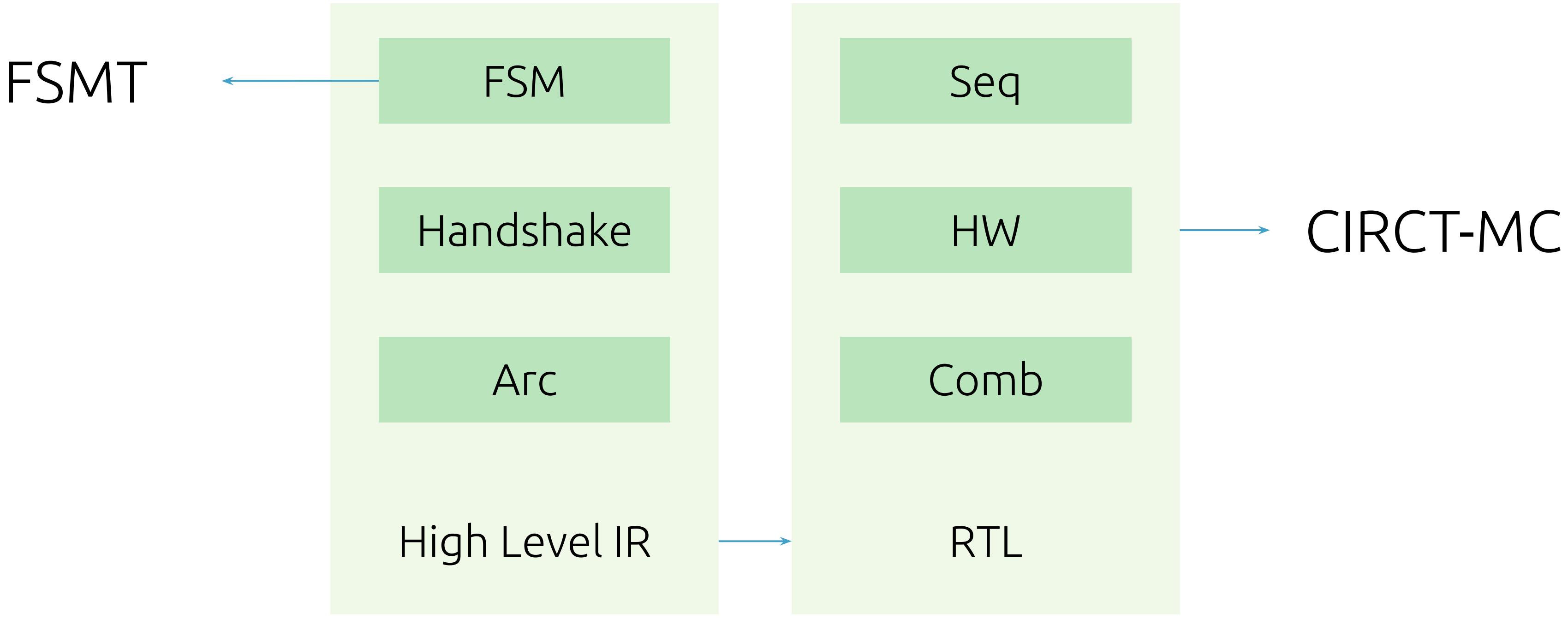
Verification in CIRCT



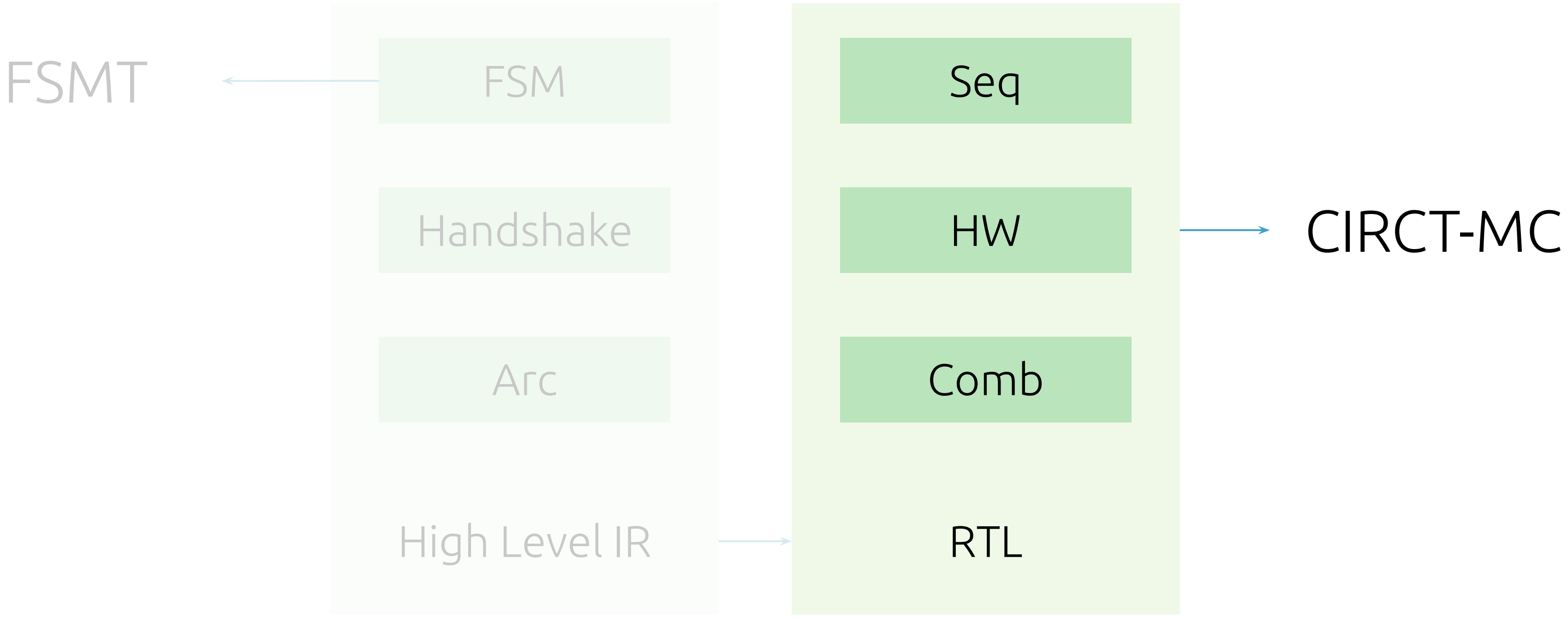
Verification in CIRCT

Can we exploit higher level abstractions for model checking?

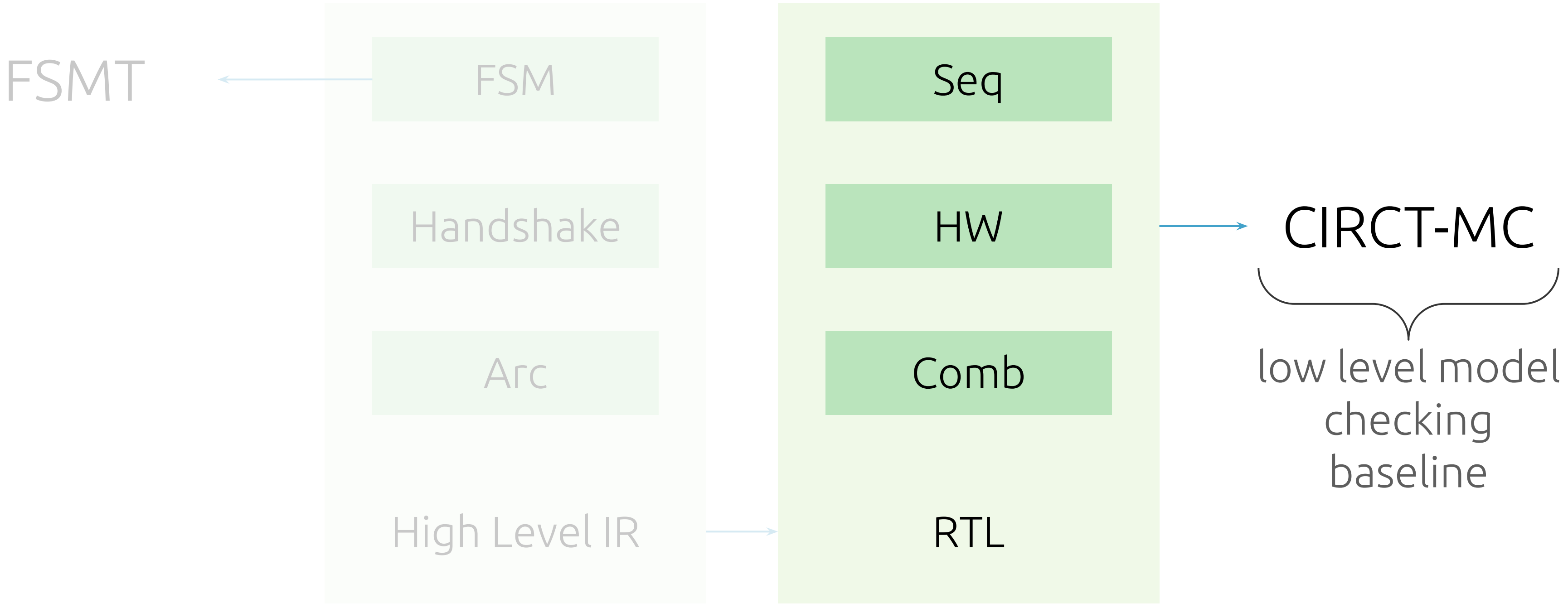
Our effort in CIRCT for Verification



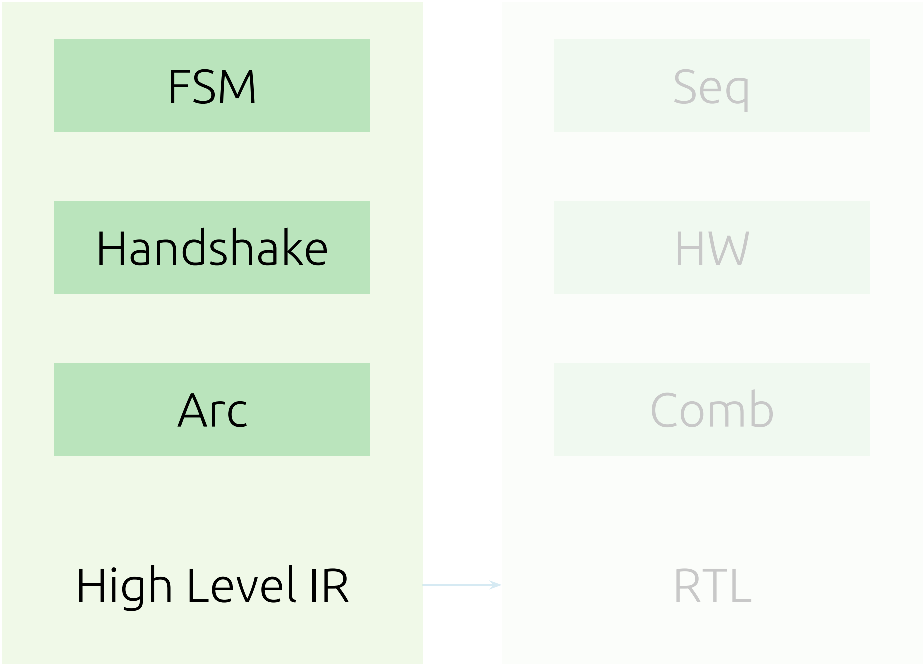
Our effort in CIRCT for Verification



Our effort in CIRCT for Verification



Our effort in CIRCT for Verification



Our effort in CIRCT for Verification

FSM

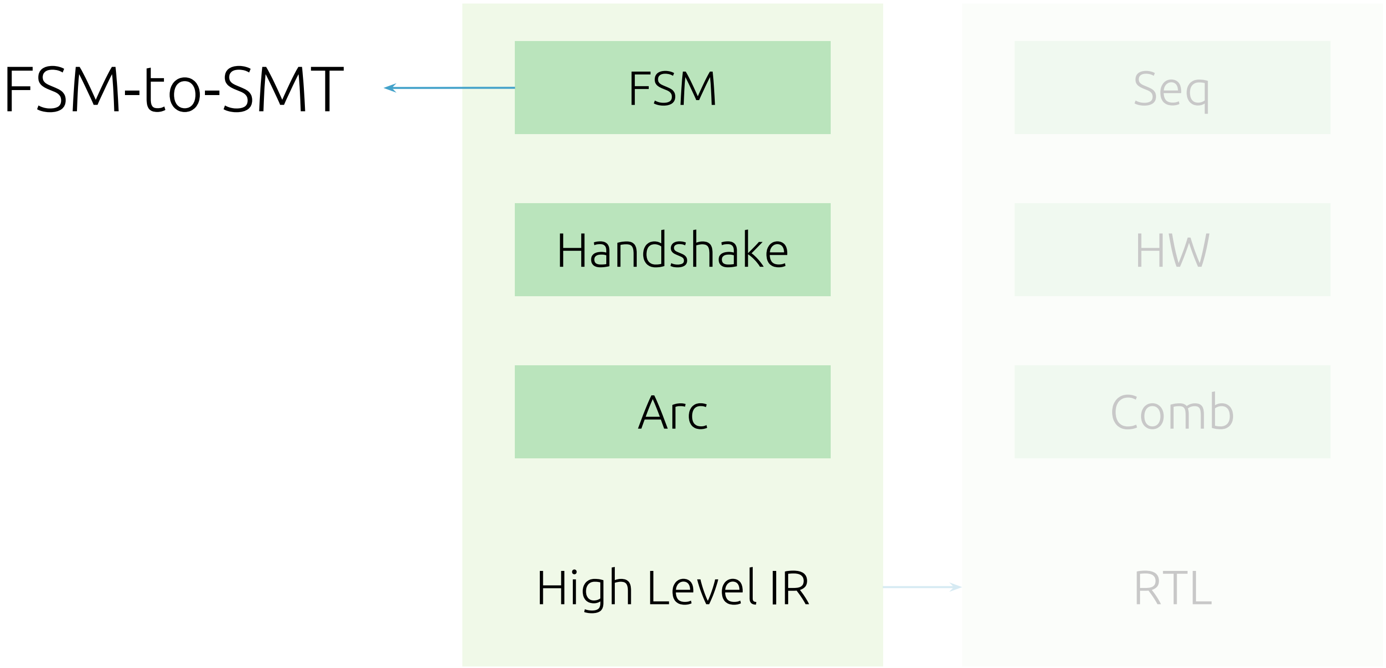
Seq

Can we check something useful at FSM level?

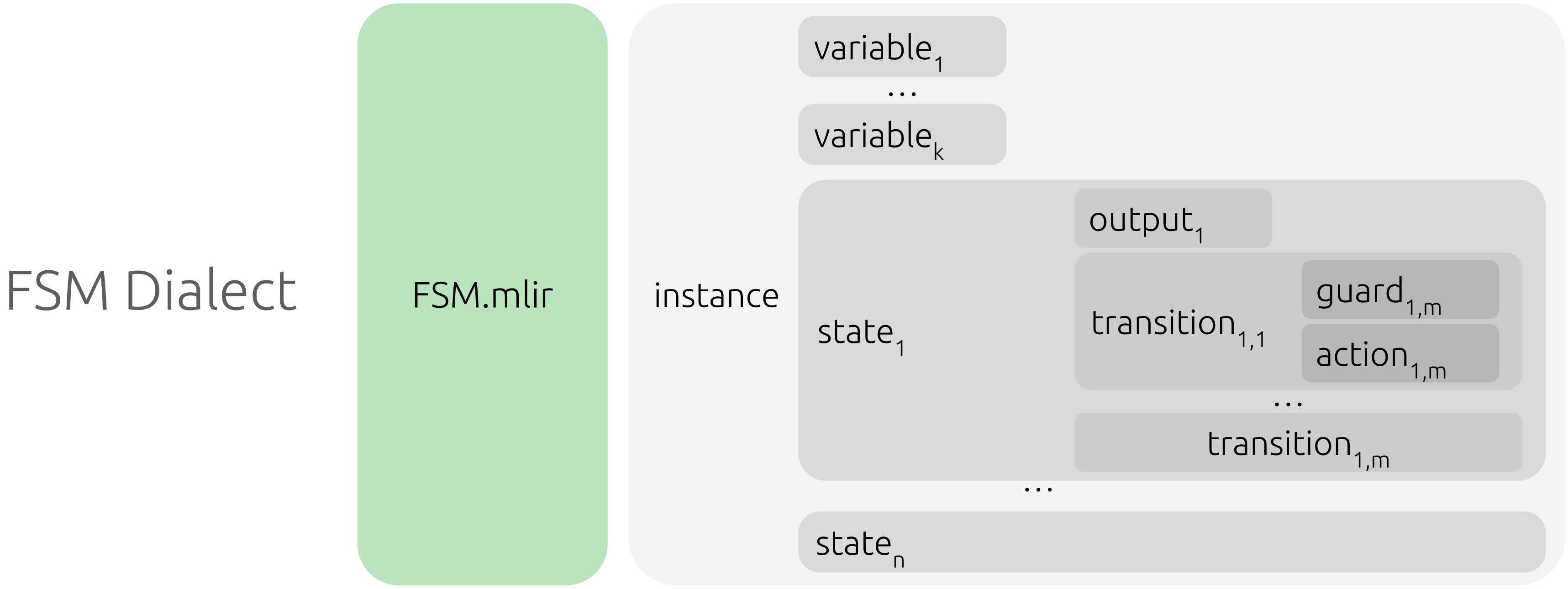
High Level IR

RTL

Our effort in CIRCT for Verification



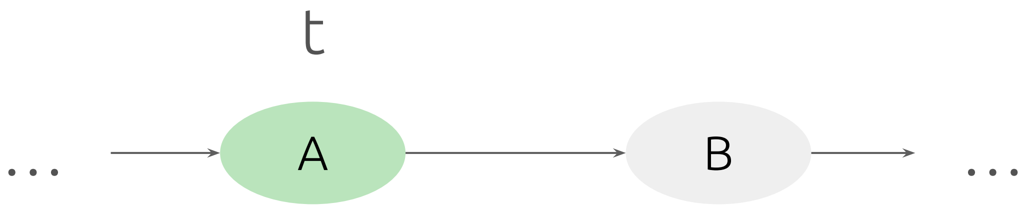
FSMT: FSM-level Model Checking



FSMT: FSM-level Model Checking

$g(\underline{x}, \underline{in}(t)), a(\underline{x}, \underline{in}(t))$

FSMT: FSM-level Model Checking



$$g(\underline{x}, \underline{in}(t)), a(\underline{x}, \underline{in}(t))$$

FSMT: FSM-level Model Checking

$\underline{in}(t)$

t

$g(\underline{x}, \underline{in}(t)), a(\underline{x}, \underline{in}(t))$

FSMT: FSM-level Model Checking

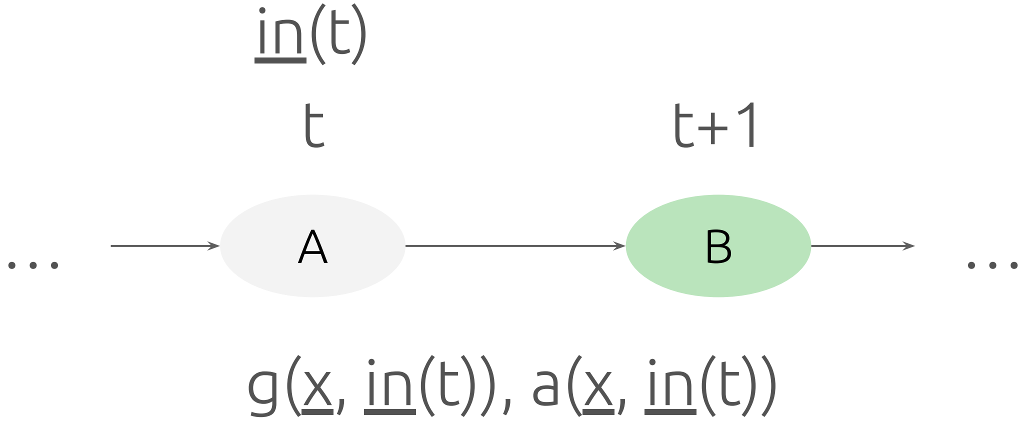
$\underline{in}(t)$

t

$g(\underline{x}, \underline{in}(t)), a(\underline{x}, \underline{in}(t))$

$F_A(t, \underline{x})$

FSMT: FSM-level Model Checking



$F_A(t, \underline{x})$

FSMT: FSM-level Model Checking

$\underline{in}(t)$ $\underline{in}(t+1)$

t $t+1$

$g(\underline{x}, \underline{in}(t)), a(\underline{x}, \underline{in}(t))$

$F_A(t, \underline{x})$

FSMT: FSM-level Model Checking

$\underline{in}(t)$ $\quad\quad$ $\underline{in}(t+1)$

t $\quad\quad$ $t+1$

$g(\underline{x}, \underline{in}(t)), a(\underline{x}, \underline{in}(t))$

$F_A(t, \underline{x})$ $\quad\quad$ $F_B(\underline{x}, t+1)$

FSMT: FSM-level Model Checking

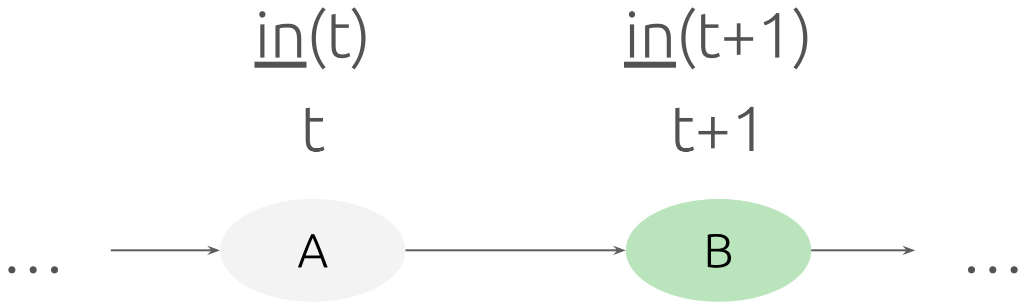
$\underline{in}(t)$ $\underline{in}(t+1)$

t $t+1$

$g(\underline{x}, \underline{in}(t)), a(\underline{x}, \underline{in}(t))$

$$F_A(t, \underline{x}) \ \&\& \ g(\underline{x}, \underline{in}(t)) \Rightarrow F_B(a(\underline{x}, \underline{in}(t+1)), t+1)$$

FSMT: FSM-level Model Checking

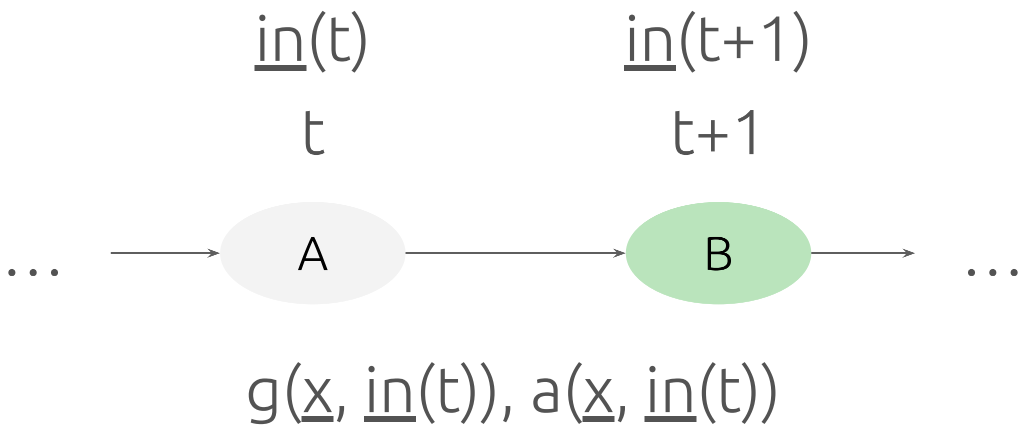


$g(\underline{x}, \underline{in}(t)), a(\underline{x}, \underline{in}(t))$

$F_A(t, \underline{x})$ && $g(\underline{x}, \underline{in}(t)) \Rightarrow F_B(a(\underline{x}, \underline{in}(t+1)), t+1)$

Uninterpreted Bool functions

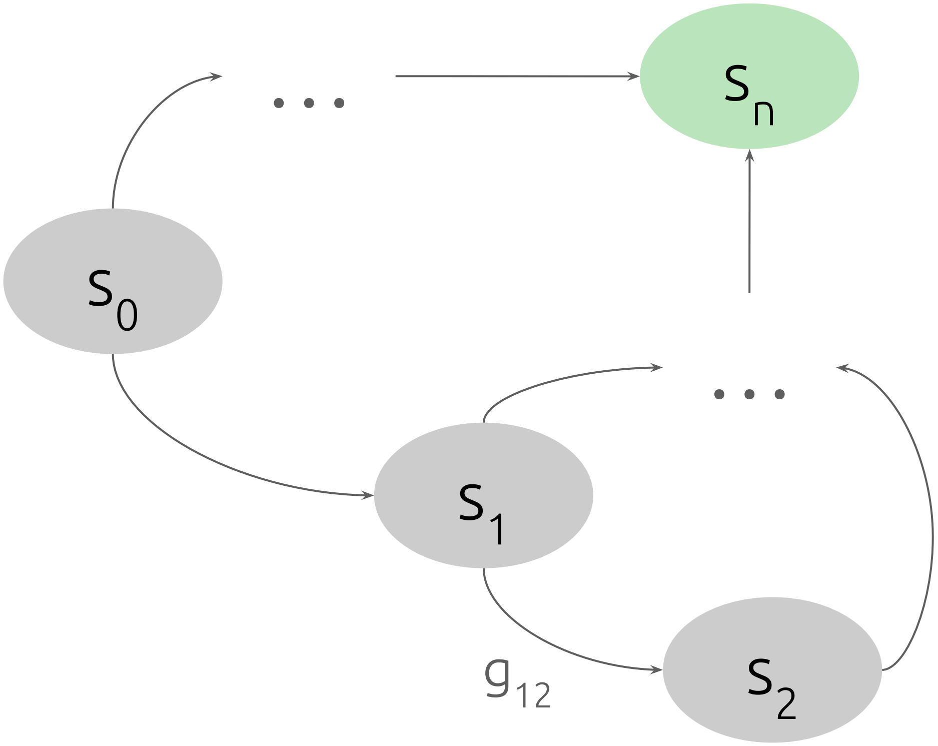
FSMT: FSM-level Model Checking



$$F_A(t,\ \underline{x})\ \&\&\ g(\underline{x},\ \underline{in}(t)) \boxed{\Longrightarrow} F_B(a(\underline{x},\ \underline{in}(t+1)),\ t+1)$$

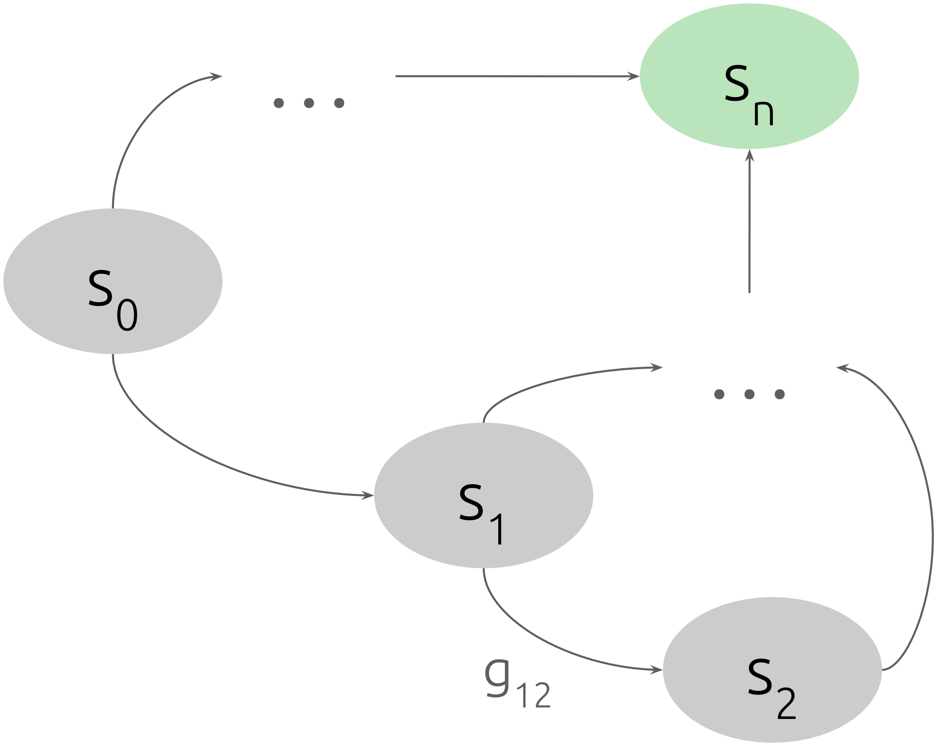
Necessary condition only

Problem: SMT Solvers are Dumb



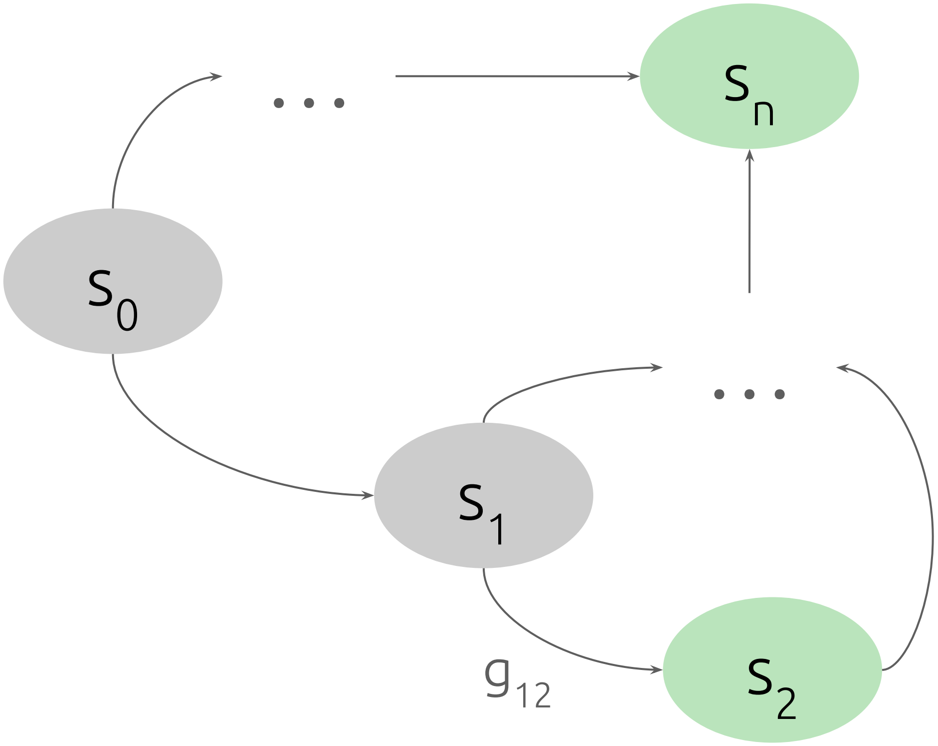
$s_0(t)$	
$s_1(t)$	
$s_2(t)$	
…	
$s_n(t)$	

Problem: SMT Solvers are Dumb



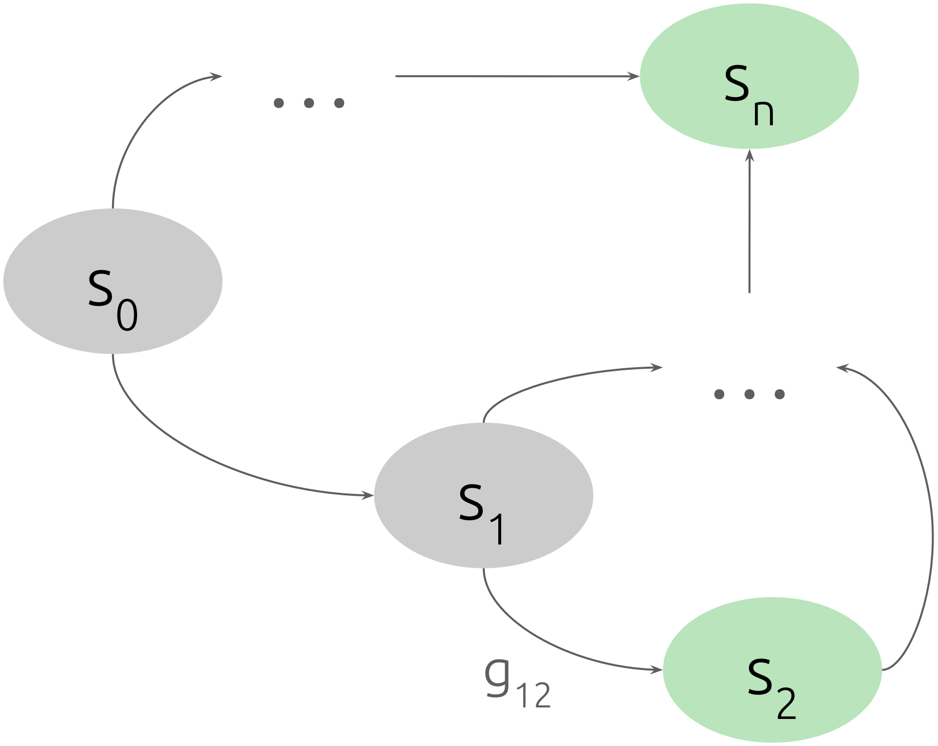
$s_0(t)$	$t = 0$
$s_1(t)$	$t = 1$
$s_2(t)$	$t = 2$ && g_{12}
...	...
$s_n(t)$	$t = m$

Problem: SMT Solvers are Dumb



$\exists t, \exists\ x \in \underline{x}$:
$F_{sn}(t)\ \&\&\ F_{s1}(t)$

Problem: SMT Solvers are Dumb



$s_0(t)$	$t = 0$
$s_1(t)$	$t = 1$
$s_2(t)$	$t = 2$ && g_{12}
...	...
$s_n(t)$	$t = m \mid\mid t = 1$

Solution: Mutual Exclusion

$$F_A(t, \underline{x})\ \&\&\ g_{AB}(\underline{x}, \underline{in}(t)) \Rightarrow F_B(a_{AB}(\underline{x}, \underline{in}(t+1), t+1)$$
$$F_B(t, \underline{x})\ \&\&\ g_{BC}(\underline{x}, \underline{in}(t)) \Rightarrow F_C(a_{BC}(\underline{x}, \underline{in}(t+1), t+1)$$

…

; property

$$F_A(t, \underline{x}) \Rightarrow !F_B(\ldots, t)$$
$$F_A(t, \underline{x}) \Rightarrow !F_C(\ldots, t)$$

…

$$F_A(t, \underline{x}) \Rightarrow !F_N(\ldots, t)$$

Solution: Mutual Exclusion

$$F_A(t,\ \underline{x})\ \&\&\ g_{AB}(\underline{x},\ \underline{in}(t)) \Rightarrow F_B(a_{AB}(\underline{x},\ \underline{in}(t+1),\ t+1)$$
$$F_B(t,\ \underline{x})\ \&\&\ g_{BC}(\underline{x},\ \underline{in}(t)) \Rightarrow F_C(a_{BC}(\underline{x},\ \underline{in}(t+1),\ t+1)$$

…

; property

$$F_A(t,\ \underline{x}) \Rightarrow\ !F_B(\ldots,\ t)$$
$$F_A(t,\ \underline{x}) \Rightarrow\ !F_C(\ldots,\ t)$$

…

$$F_A(t,\ \underline{x}) \Rightarrow\ !F_N(\ldots,\ t)$$

Guarantee one active state at any time-step t

FSMT: Reachability Verification

FSMT: Reachability Verification

$$F_A(t,\ \underline{x})\ \&\&\ g_{AB}(\underline{x},\ \underline{in}(t)) \Rightarrow F_B(a_{AB}(\underline{x},\ \underline{in}(t+1),\ t+1)$$
$$F_B(t,\ \underline{x})\ \&\&\ g_{BC}(\underline{x},\ \underline{in}(t)) \Rightarrow F_C(a_{BC}(\underline{x},\ \underline{in}(t+1),\ t+1)$$

…

"eventually state B will be reached"

$$\forall t, \forall\ x \in \underline{x} : F_B(t,\ \underline{x}) \Rightarrow \text{false}$$

FSMT: Reachability Verification

$F_A(t, \underline{x})$ && $g_{AB}(\underline{x}, \underline{in}(t)) \Rightarrow F_B(a_{AB}(\underline{x}, \underline{in}(t+1), t+1)$
$F_B(t, \underline{x})$ && $g_{BC}(\underline{x}, \underline{in}(t)) \Rightarrow F_C(a_{BC}(\underline{x}, \underline{in}(t+1), t+1)$

…

"eventually state B will be reached"

$$\forall t, \forall\ x \in \underline{x} : F_B(t, \underline{x}) \Rightarrow false$$
$$= !F_B(t, \underline{x}) \mid\mid false$$

FSMT: Reachability Verification

$F_A(t, \underline{x})$ && $g_{AB}(\underline{x}, \underline{in}(t)) \Rightarrow F_B(a_{AB}(\underline{x}, \underline{in}(t+1), t+1)$
$F_B(t, \underline{x})$ && $g_{BC}(\underline{x}, \underline{in}(t)) \Rightarrow F_C(a_{BC}(\underline{x}, \underline{in}(t+1), t+1)$

…

"eventually state B will be reached"

$$\begin{aligned} \forall t, \forall\ x \in \underline{x} : F_B(t, \underline{x}) &\Rightarrow \text{false} \\ &= !F_B(t, \underline{x}) \mid\mid \text{false} \\ &= !F_B(t, \underline{x}) \end{aligned}$$

FSMT: Reachability Verification

$$F_A(t,\ \underline{x})\ \&\&\ g_{AB}(\underline{x},\ \underline{in}(t)) \Rightarrow F_B(a_{AB}(\underline{x},\ \underline{in}(t+1),\ t+1)$$
$$F_B(t,\ \underline{x})\ \&\&\ g_{BC}(\underline{x},\ \underline{in}(t)) \Rightarrow F_C(a_{BC}(\underline{x},\ \underline{in}(t+1),\ t+1)$$

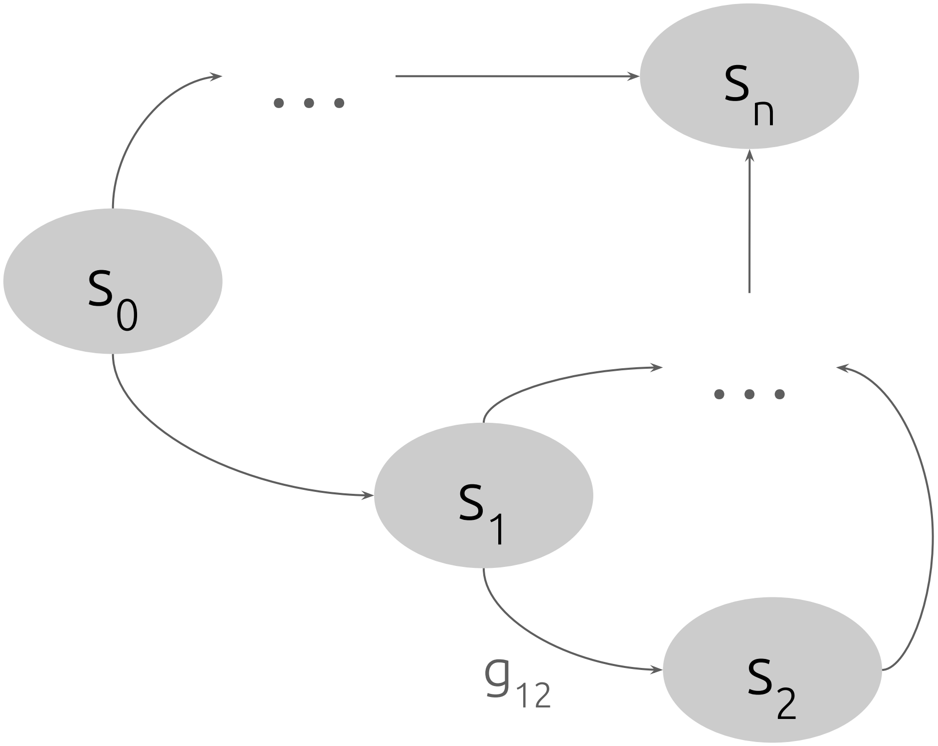
…

"eventually state B will be reached"

$$\begin{aligned}\forall t, \forall\ x \in \underline{x} :\ & F_B(t,\ \underline{x}) \Rightarrow \text{false} \\ &= !F_B(t,\ \underline{x})\ ||\ \text{false} \\ &= !F_B(t,\ \underline{x})\end{aligned}$$

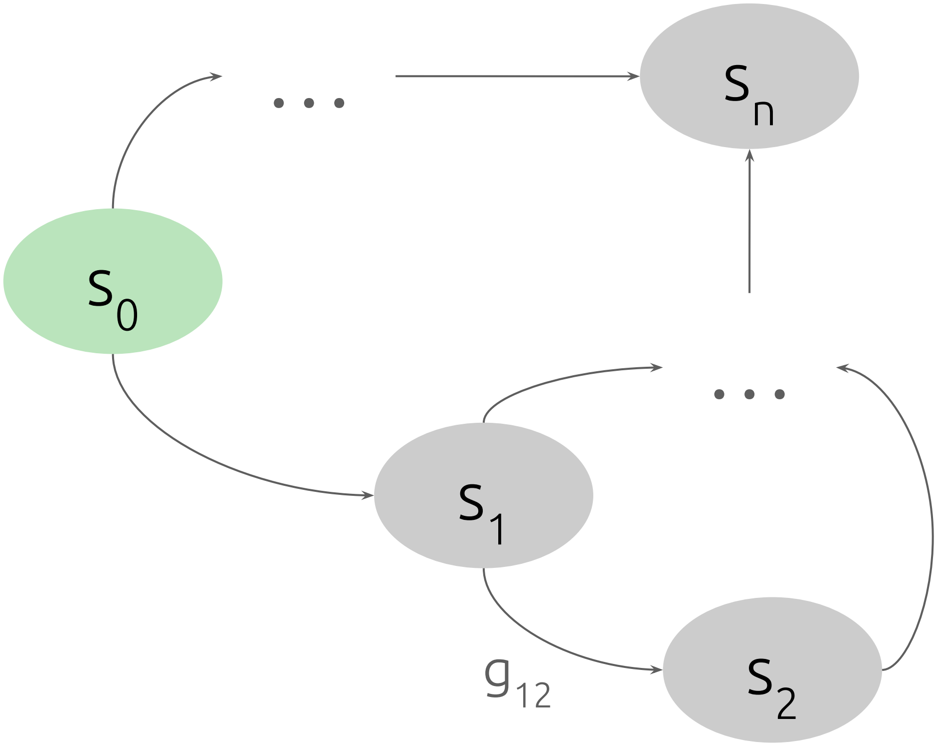
UNSAT

FSMT: Reachability Verification



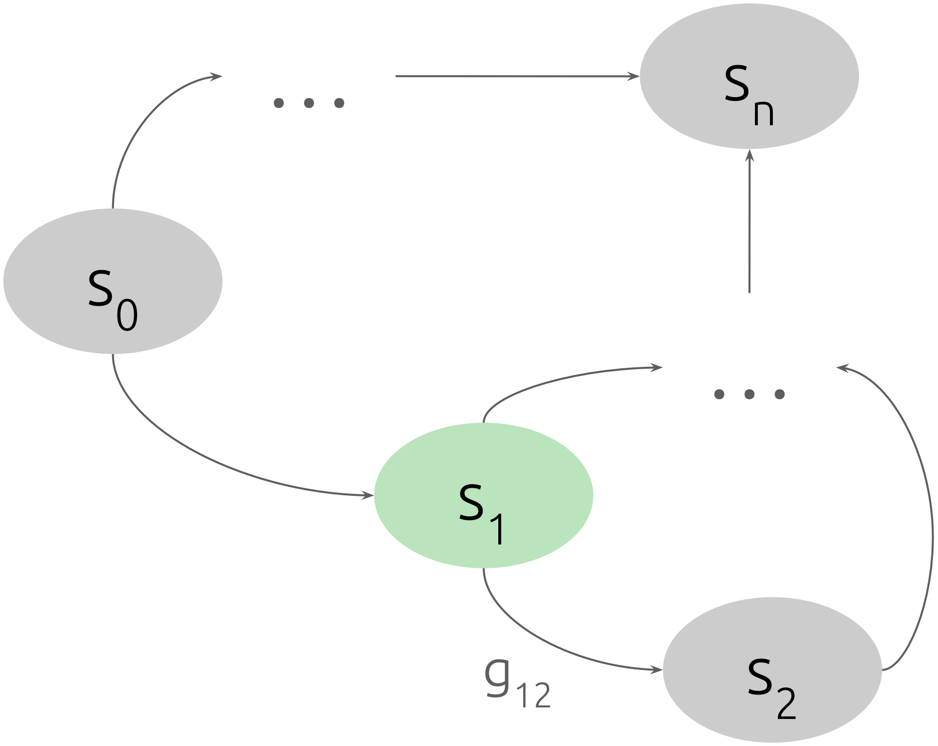
$$\forall t, \forall\ x \in \underline{x} : !F_{sn}(t)$$

FSMT: Reachability Verification



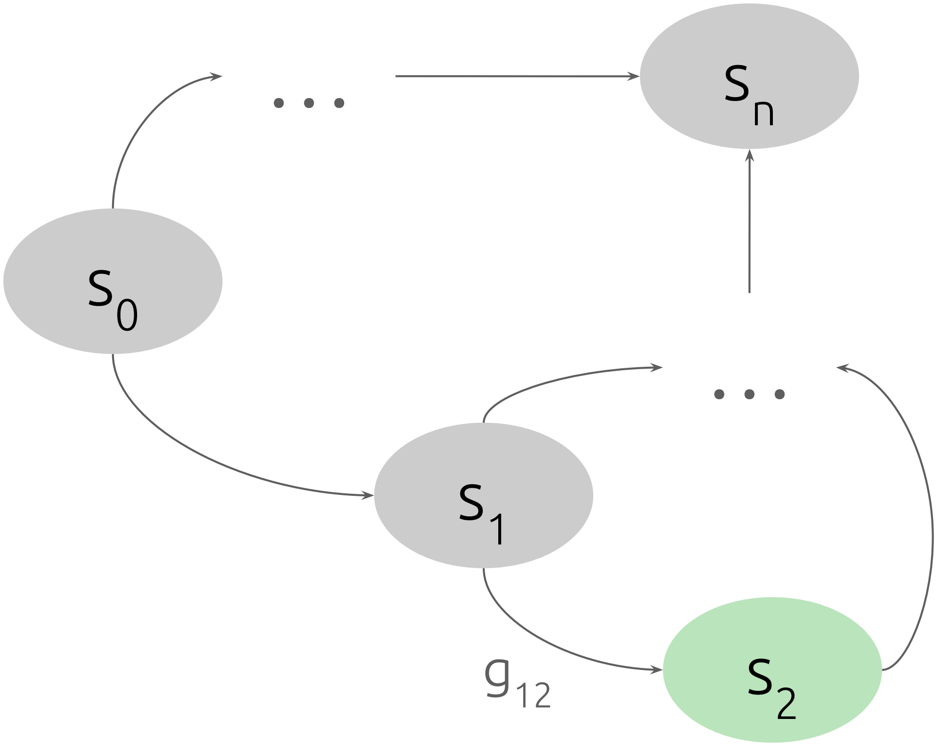
$\forall t, \forall\ x \in \underline{x}\ :\ !F_{sn}(t)$

FSMT: Reachability Verification



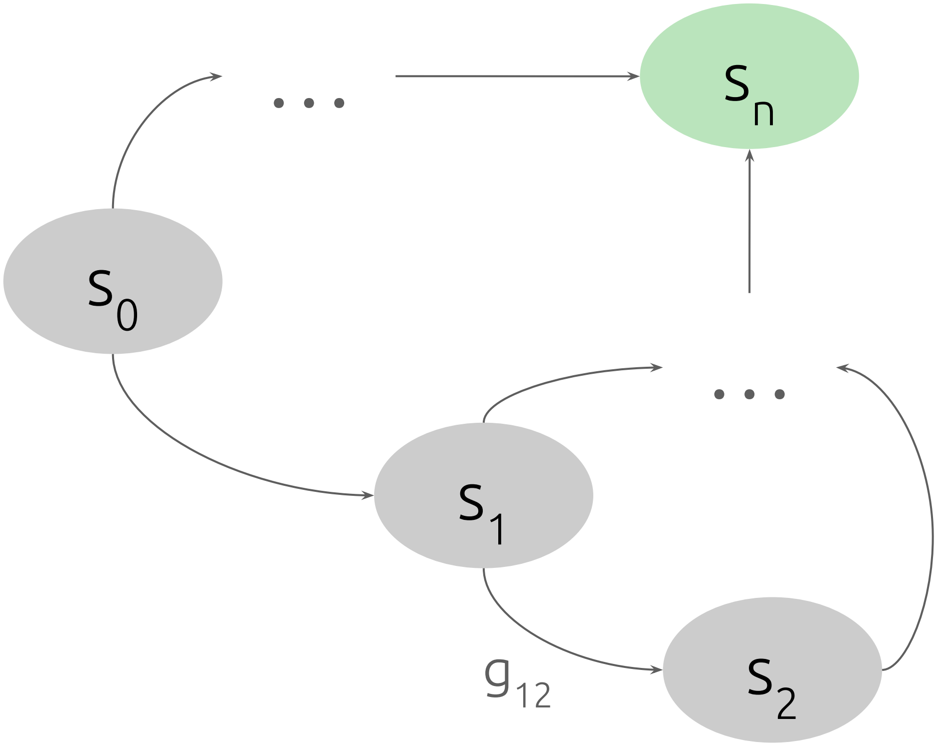
$\forall t, \forall\ x \in \underline{x} : !F_{sn}(t)$

FSMT: Reachability Verification



$\forall t, \forall\ x \in \underline{x} : \ !F_{sn}(t)$

FSMT: Reachability Verification



$\forall t, \forall\ x \in \underline{x}\ :\ !F_{sn}(t)$

UNSAT

FSMT: Safety Properties Verification

$$F_A(t, \underline{x}) \mathbin{\&\&} g_{AB}(\underline{x}, \underline{in}(t)) \Rightarrow F_B(a_{AB}(\underline{x}, \underline{in}(t+1), t+1)$$
$$F_B(t, \underline{x}) \mathbin{\&\&} g_{BC}(\underline{x}, \underline{in}(t)) \Rightarrow F_C(a_{BC}(\underline{x}, \underline{in}(t+1), t+1)$$

…

FSMT: Safety Properties Verification

$$F_A(t,\ \underline{x})\ \&\&\ g_{AB}(\underline{x},\ \underline{in}(t))\ \Rightarrow\ F_B(a_{AB}(\underline{x},\ \underline{in}(t+1),\ t+1)$$
$$F_B(t,\ \underline{x})\ \&\&\ g_{BC}(\underline{x},\ \underline{in}(t))\ \Rightarrow\ F_C(a_{BC}(\underline{x},\ \underline{in}(t+1),\ t+1)$$

…

“x_i is always 1 in state B”

FSMT: Safety Properties Verification

$$F_A(t, \underline{x}) \;\&\&\; g_{AB}(\underline{x}, \underline{in}(t)) \Rightarrow F_B(a_{AB}(\underline{x}, \underline{in}(t+1), t+1)$$
$$F_B(t, \underline{x}) \;\&\&\; g_{BC}(\underline{x}, \underline{in}(t)) \Rightarrow F_C(a_{BC}(\underline{x}, \underline{in}(t+1), t+1)$$

…

"x_i is always 1 in state B"

$$\forall t, \forall\ x \in \underline{x} : F_B(t, \underline{x}) \Rightarrow x_i = 1$$

FSMT: Safety Properties Verification

$F_A(t, \underline{x})$ && $g_{AB}(\underline{x}, \underline{in}(t)) \Rightarrow F_B(a_{AB}(\underline{x}, \underline{in}(t+1), t+1)$
$F_B(t, \underline{x})$ && $g_{BC}(\underline{x}, \underline{in}(t)) \Rightarrow F_C(a_{BC}(\underline{x}, \underline{in}(t+1), t+1)$

…

"x_i is always 1 in state B"

$$\forall t, \forall\ x \in \underline{x} : F_B(t, \underline{x}) \Rightarrow x_i = 1$$

FSMT: Safety Properties Verification

$F_A(t, \underline{x})$ && $g_{AB}(\underline{x}, \underline{in}(t)) \Rightarrow F_B(a_{AB}(\underline{x}, \underline{in}(t+1), t+1)$
$F_B(t, \underline{x})$ && $g_{BC}(\underline{x}, \underline{in}(t)) \Rightarrow F_C(a_{BC}(\underline{x}, \underline{in}(t+1), t+1)$

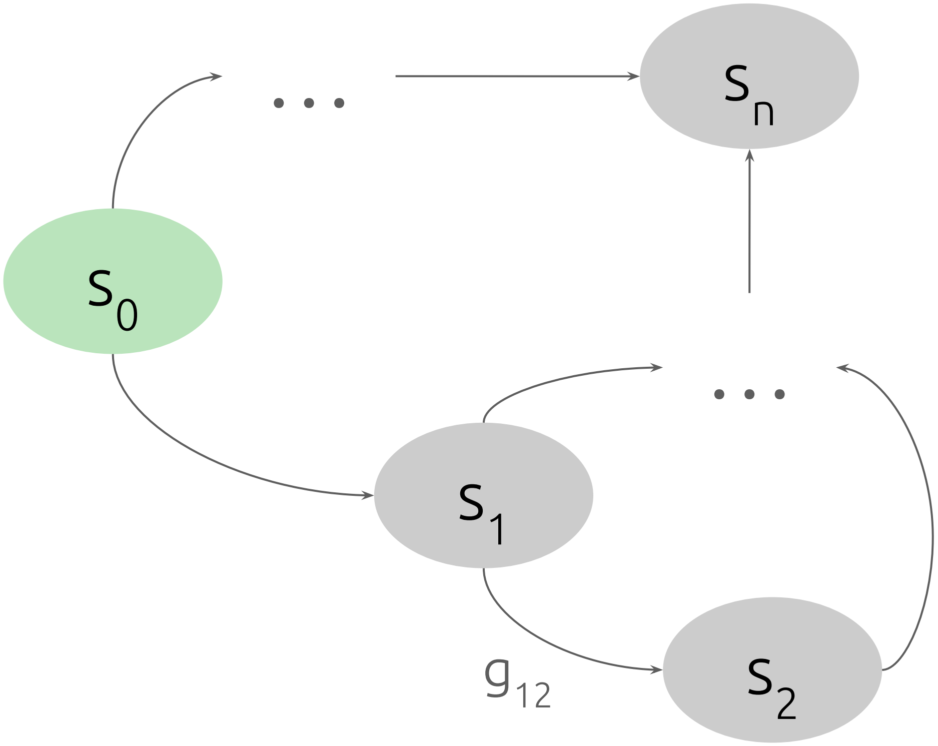
…

"x_i is always 1 in state B"

$\forall t, \forall\ x \in \underline{x}\ :\ F_B(t, \underline{x}) \Rightarrow x_i = 1$

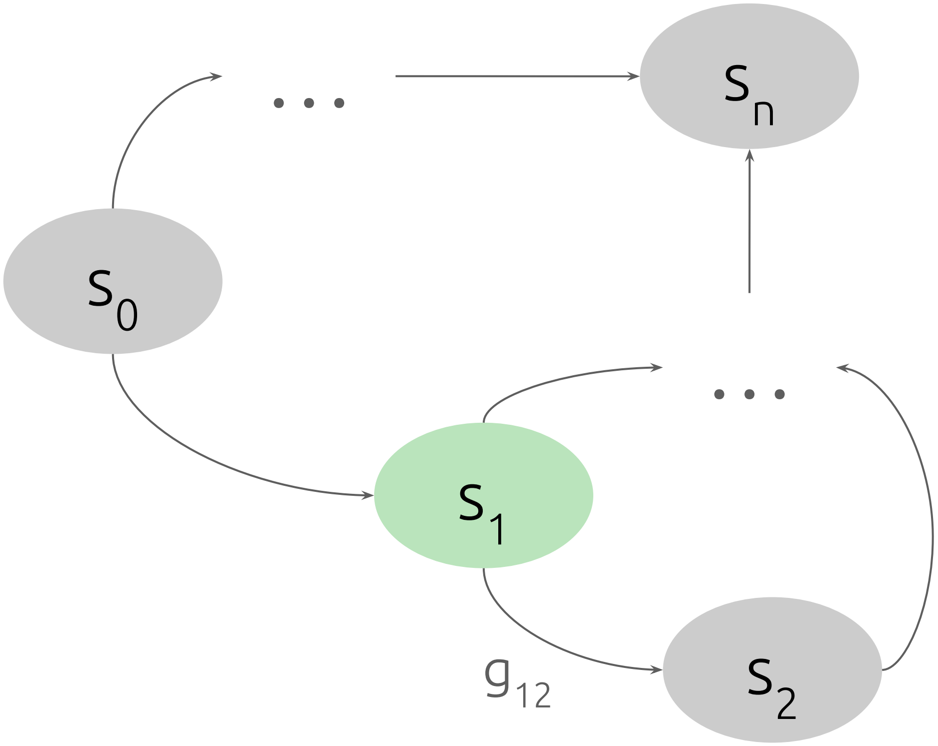
SAT

FSMT: Safety Properties Verification



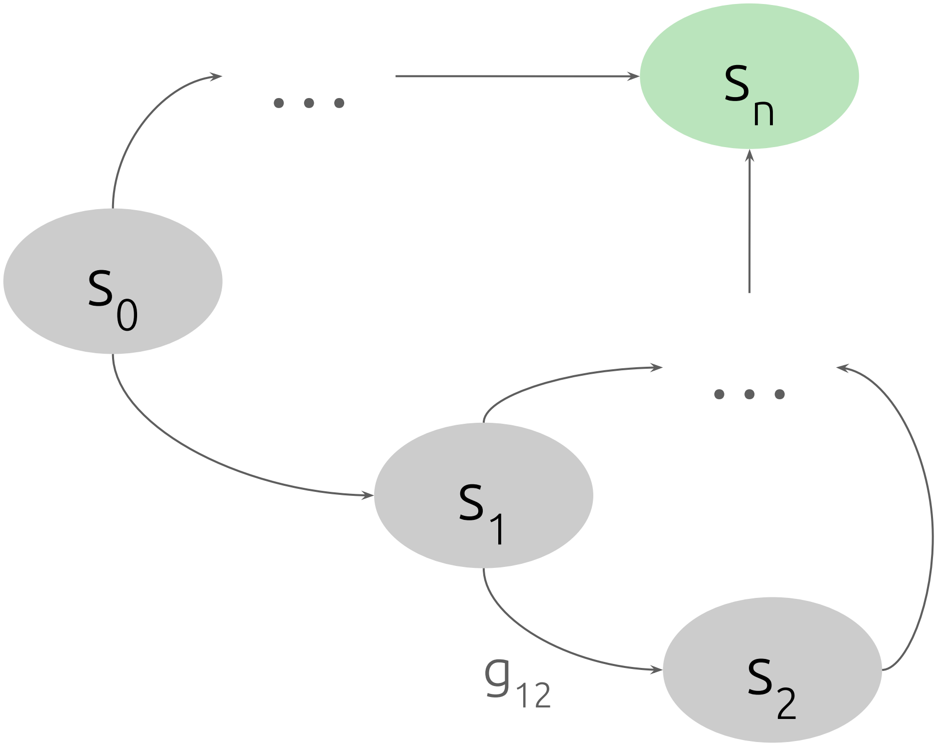
$$\forall t, \forall\ x \in \underline{x} : F_{x2}(t, \underline{x}) \Rightarrow x = 2$$

FSMT: Safety Properties Verification



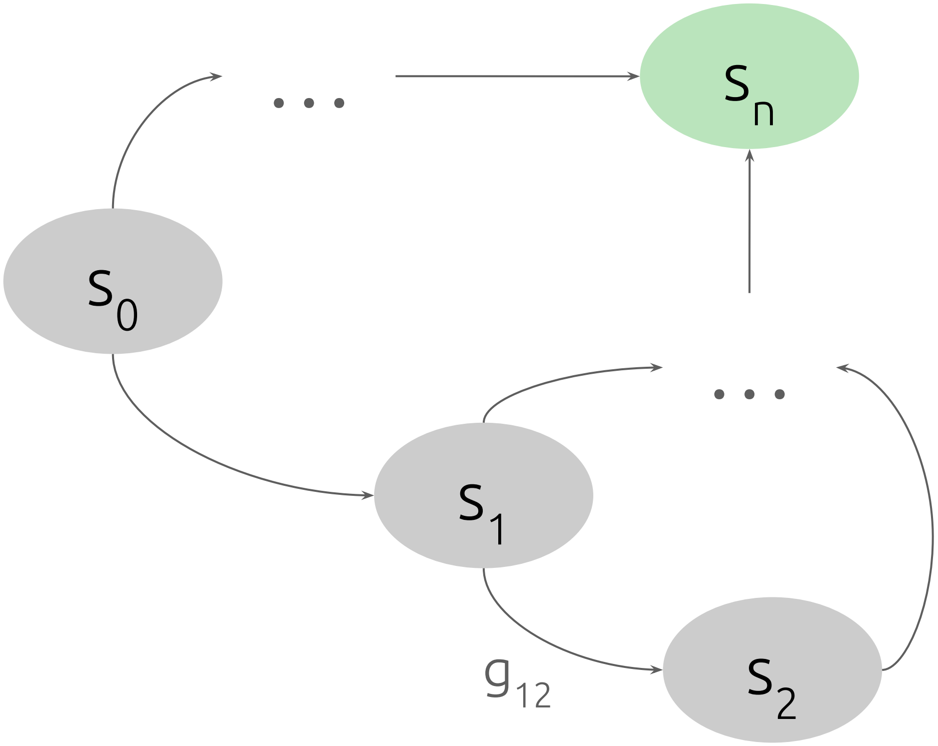
$$\forall t, \forall\ x \in \underline{x} : F_{x2}(t, \underline{x}) \Rightarrow x = 2$$

FSMT: Safety Properties Verification



$$\forall t, \forall\ x \in \underline{x} : F_{x2}(t, \underline{x}) \Rightarrow x = 2$$

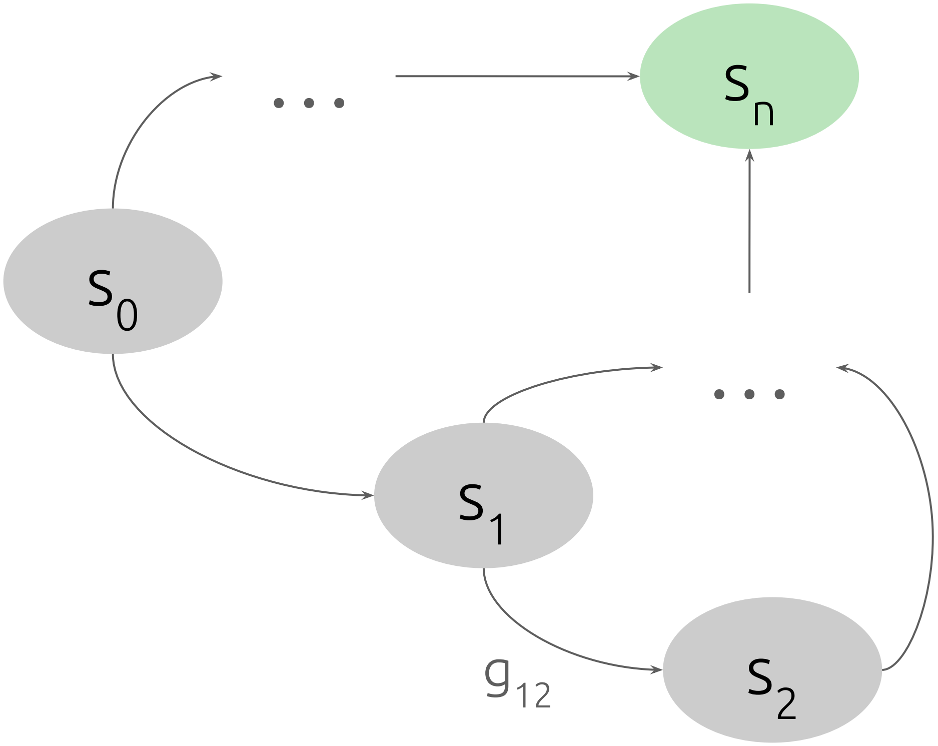
FSMT: Safety Properties Verification



$$\forall t, \forall\ x \in \underline{x}\ :\ F_{x2}(t,\ \underline{x}) \Rightarrow x = 2$$

$$= \ !F_{x2}(t,\ \underline{x})\ ||\ x = 2$$

FSMT: Safety Properties Verification

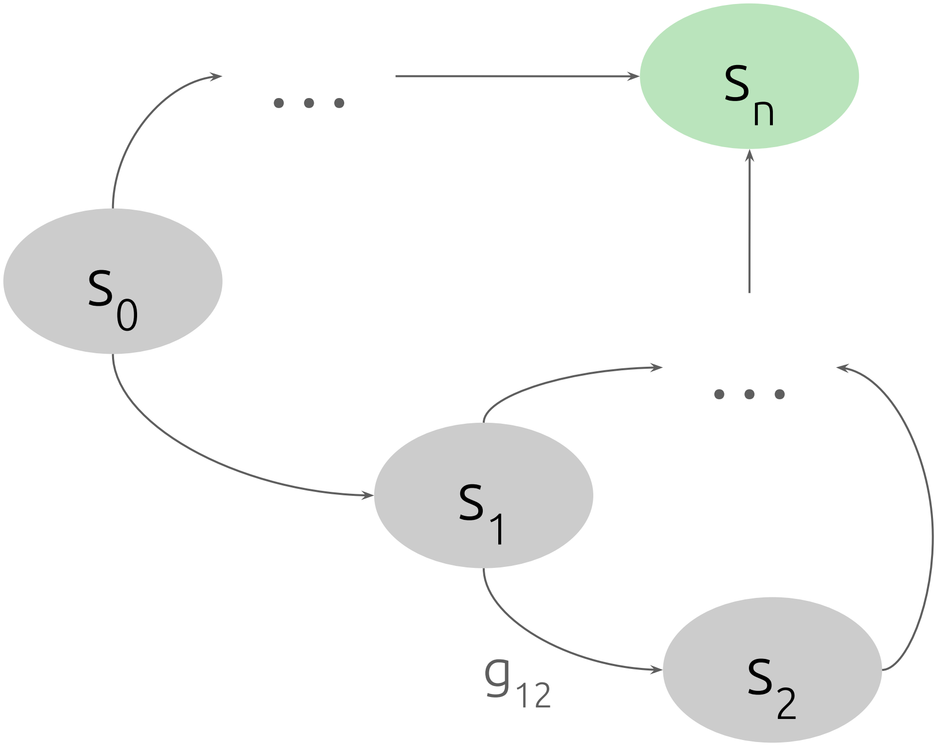


$$\forall t, \forall\ x \in \underline{x} : F_{x2}(t, \underline{x}) \Rightarrow x = 2$$

$$= \boxed{!F_{x2}(t, \underline{x})} \ || \ x = 2$$

SAT

FSMT: Safety Properties Verification

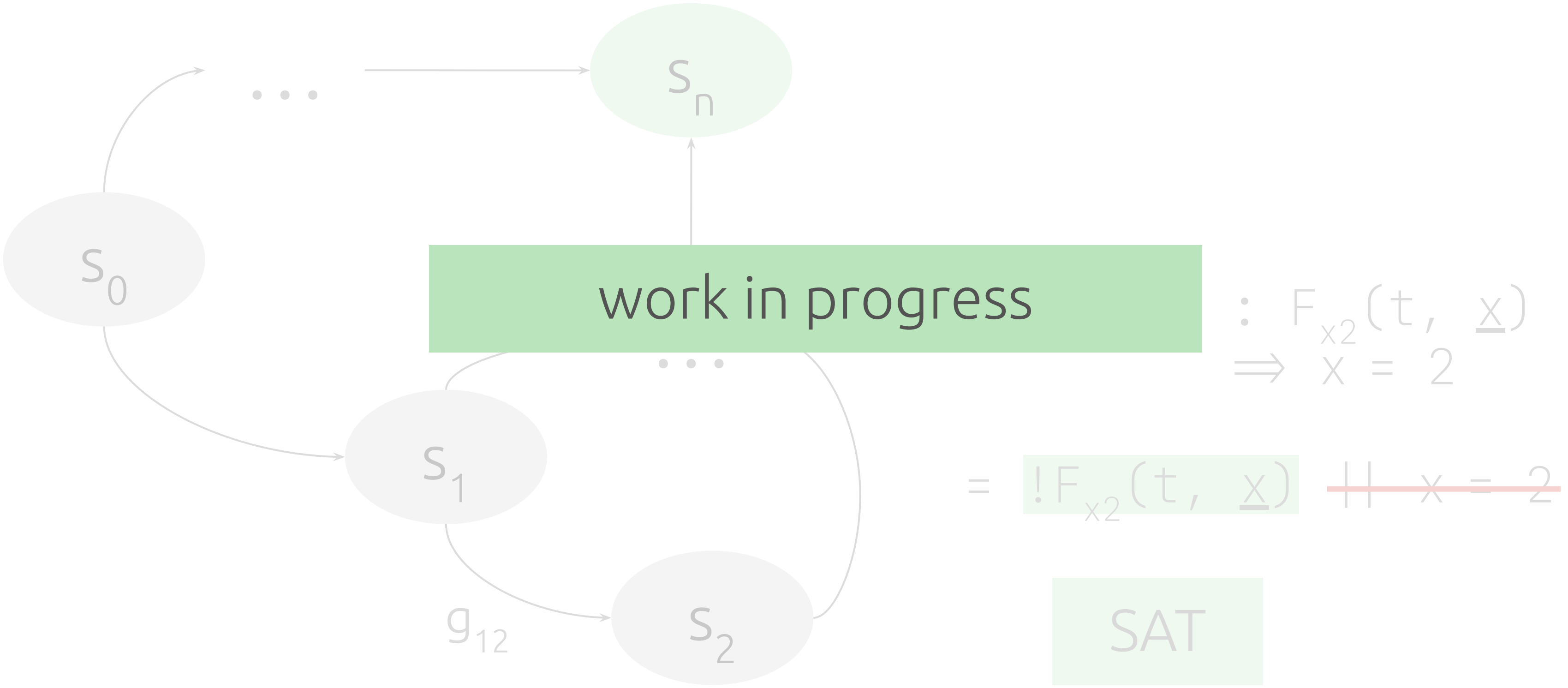


$$\forall t, \forall\ x \in \underline{x} : F_{x2}(t, \underline{x}) \Rightarrow x = 2$$

$$= \boxed{!F_{x2}(t, \underline{x})}\ \cancel{||\ x = 2}$$

SAT

FSMT: Safety Properties Verification



FSMT: Testing

HLS Real FSMs [HLS]
Synthetic FSMs [SYN]

FSM

FSMT: Testing

HLS Real FSMs [HLS]
Synthetic FSMs [SYN]

FSM → SMT

FSMT: Testing

HLS Real FSMs [HLS]
Synthetic FSMs [SYN]

FSM → SMT → SMT-LIB

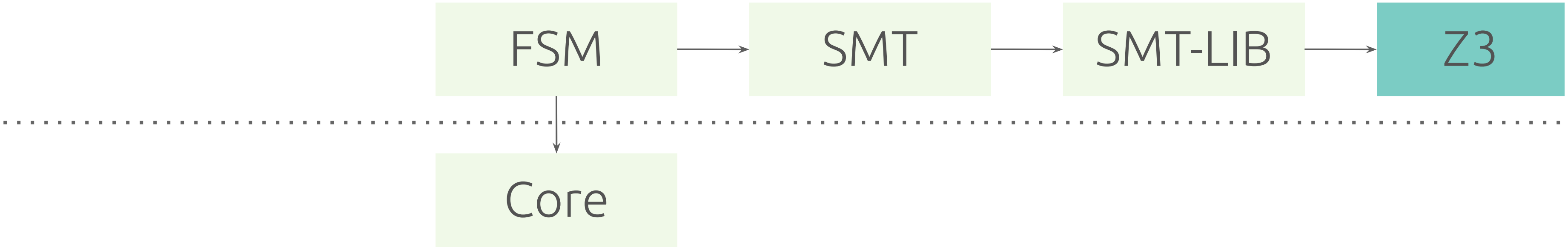
FSMT: Testing

HLS Real FSMs [HLS]
Synthetic FSMs [SYN]

FSM → SMT → SMT-LIB → Z3

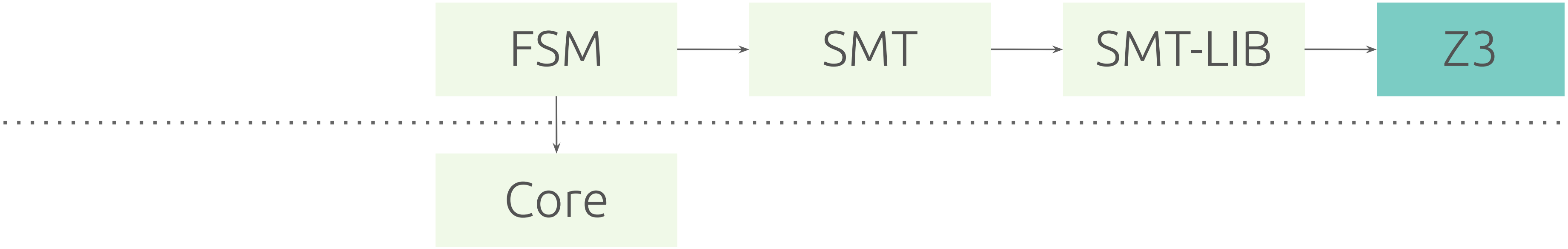
FSMT: Testing

HLS Real FSMs [HLS]
Synthetic FSMs [SYN]



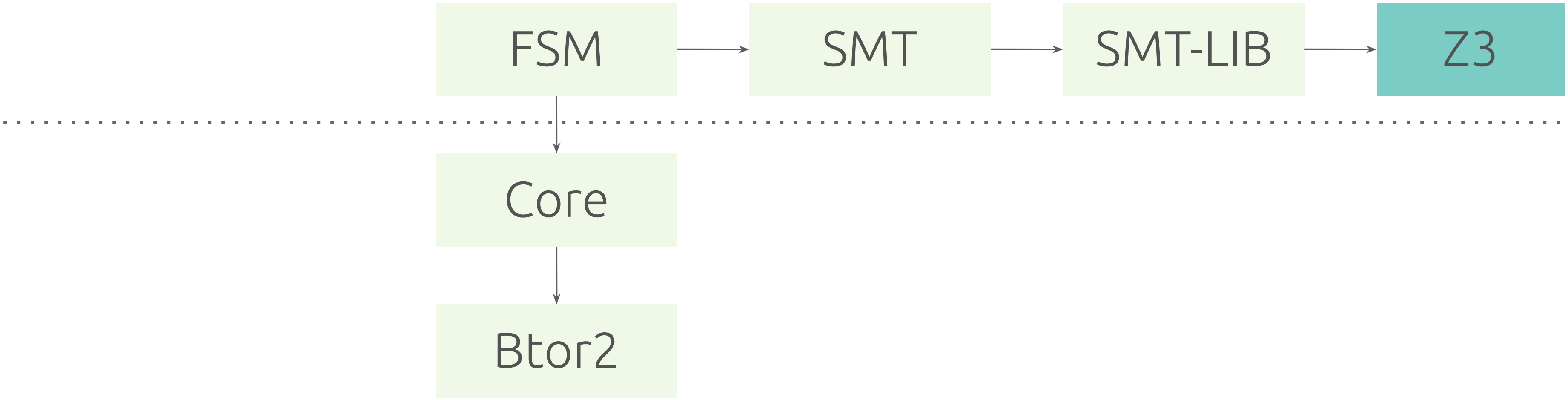
FSMT: Testing

HLS Real FSMs [HLS]
Synthetic FSMs [SYN]



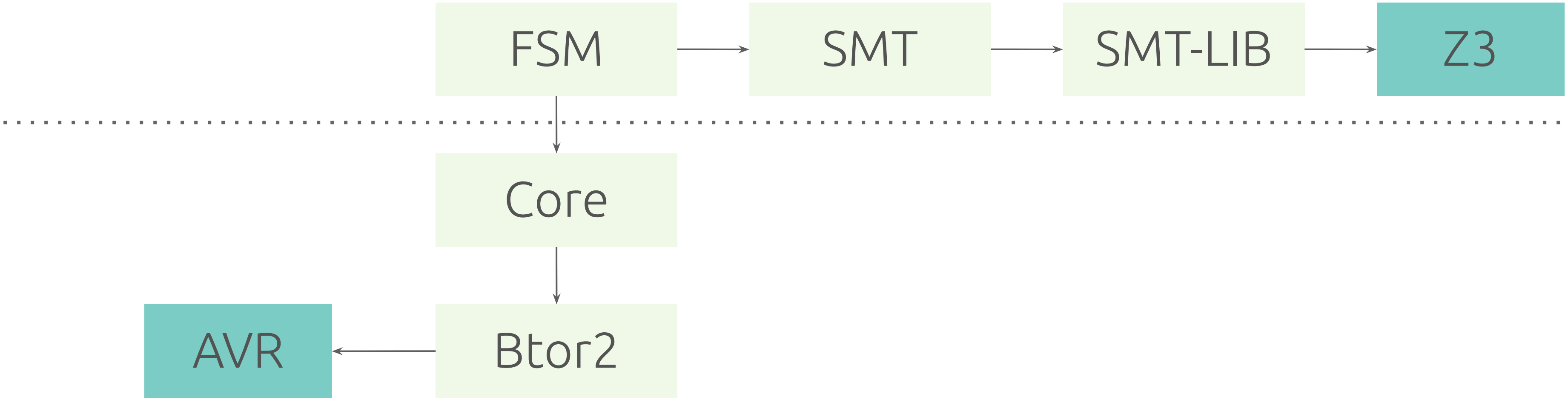
FSMT: Testing

HLS Real FSMs [HLS]
Synthetic FSMs [SYN]



FSMT: Testing

HLS Real FSMs [HLS]
Synthetic FSMs [SYN]

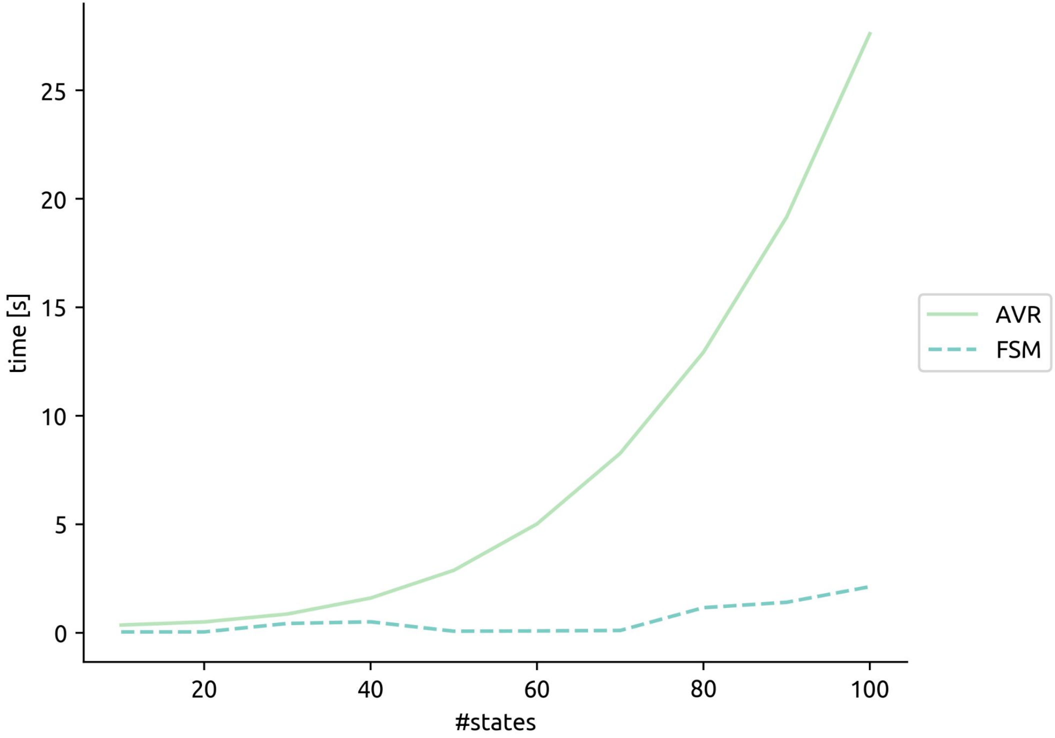


FSMT: Testing

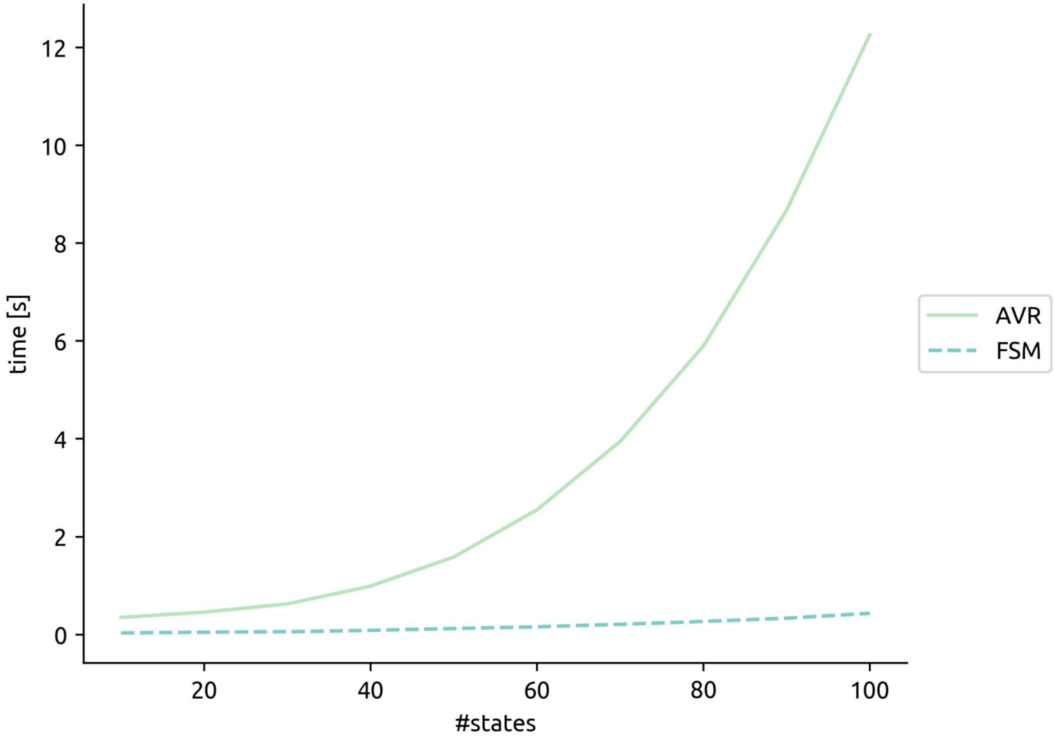
HORN
AUFLIA

Z3

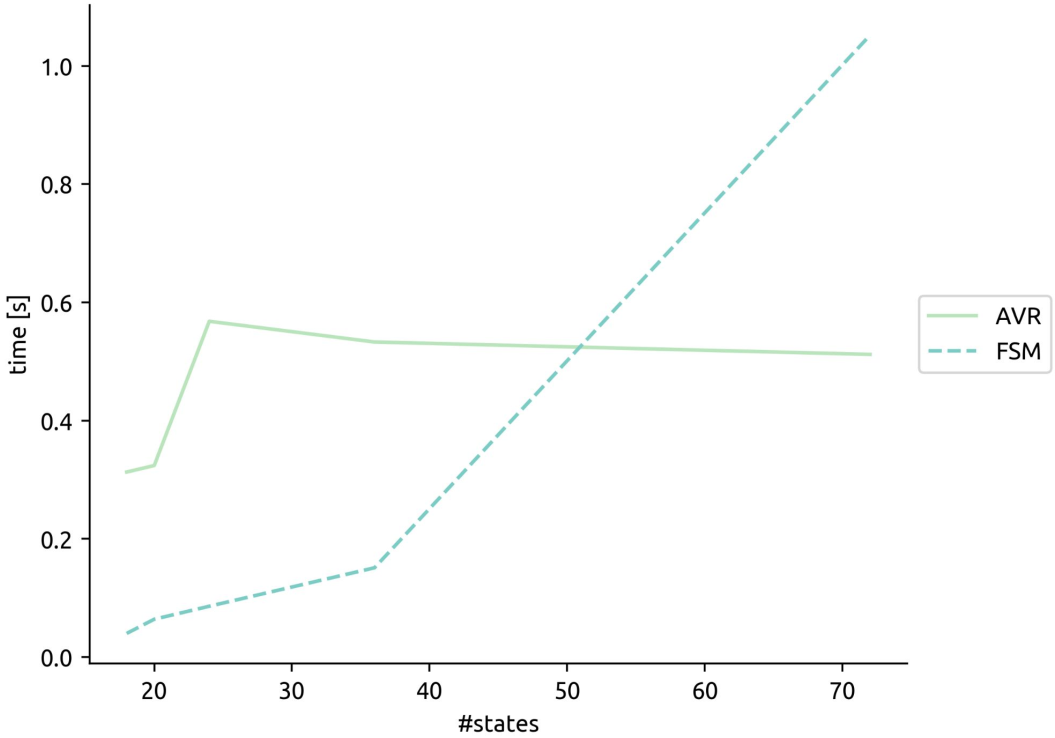
FSMT vs. AVR: reachability property - SYN



FSMT vs. AVR: reachability property - SYN



FSMT vs. AVR: reachability property - HLS



FSMT vs. AVR: reachability property