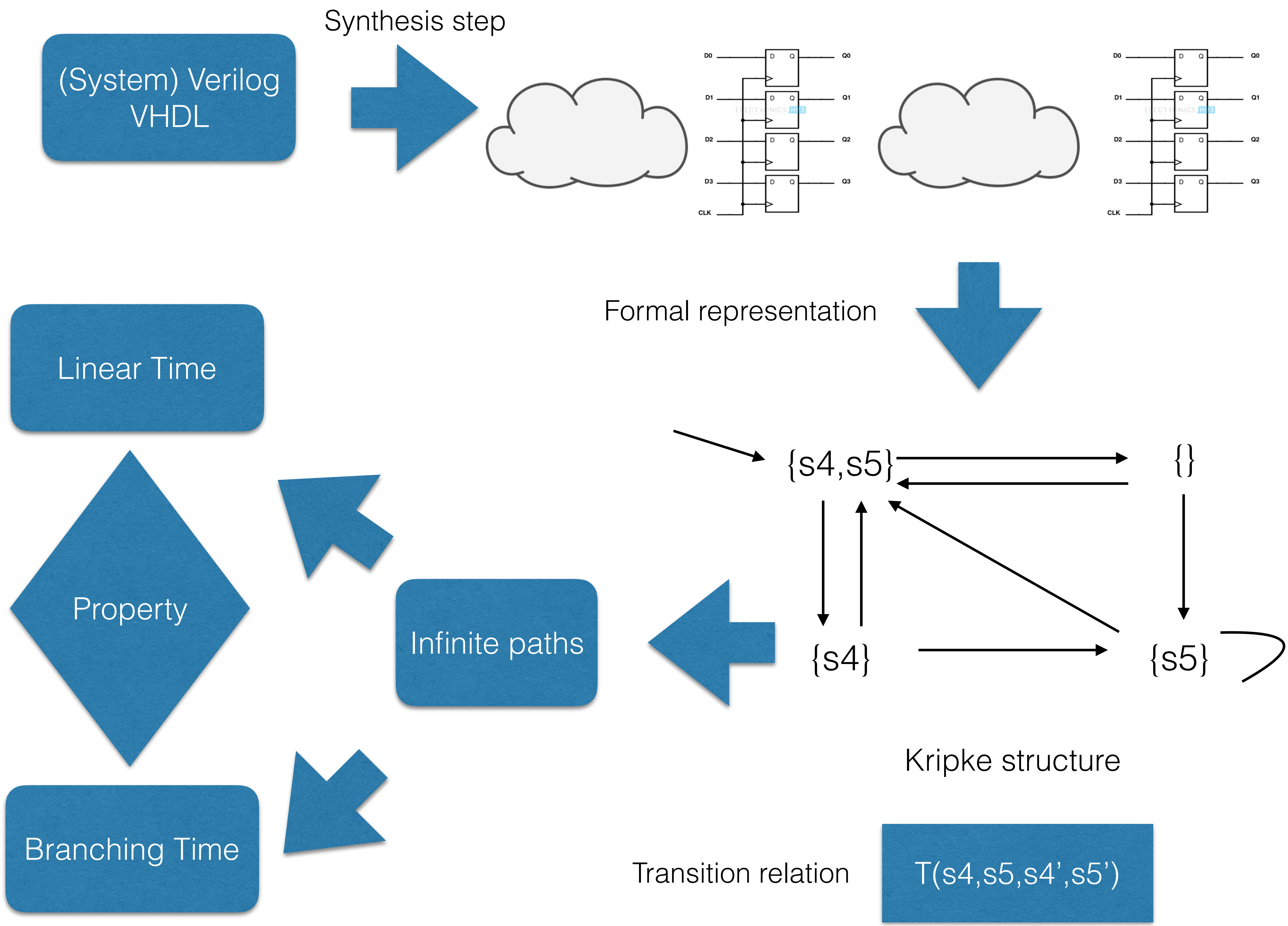
Hardware Verification
2IMF20

Julien Schmaltz

Lecture 04:
Temporal Logics
Formal semantics

Technische Universiteit **Eindhoven** University of Technology

Where innovation starts


Synthesis step
(System) Verilog
VHDL
Formal representation
Linear Time
Property
Infinite paths
Branching Time
{s4,s5}
{}
{s4}
{s5}
Kripke structure
Transition relation
T(s4,s5,s4',s5')

Formal semantics LTL, CTL, CTL*

- Last time we used different structures in our semantics
  - sometimes LTS (Labelled Transition Systems)
  - some other times Kripke Structures
- In the course, we also looked at Finite State Machines (FSMs)
- FSMs are needed to represent hardware.
- For semantics, Kripke structures are enough.
- Today, we will give formal semantics using Kripke
  - these are the only semantics you really need to know for this course

Kripke structure

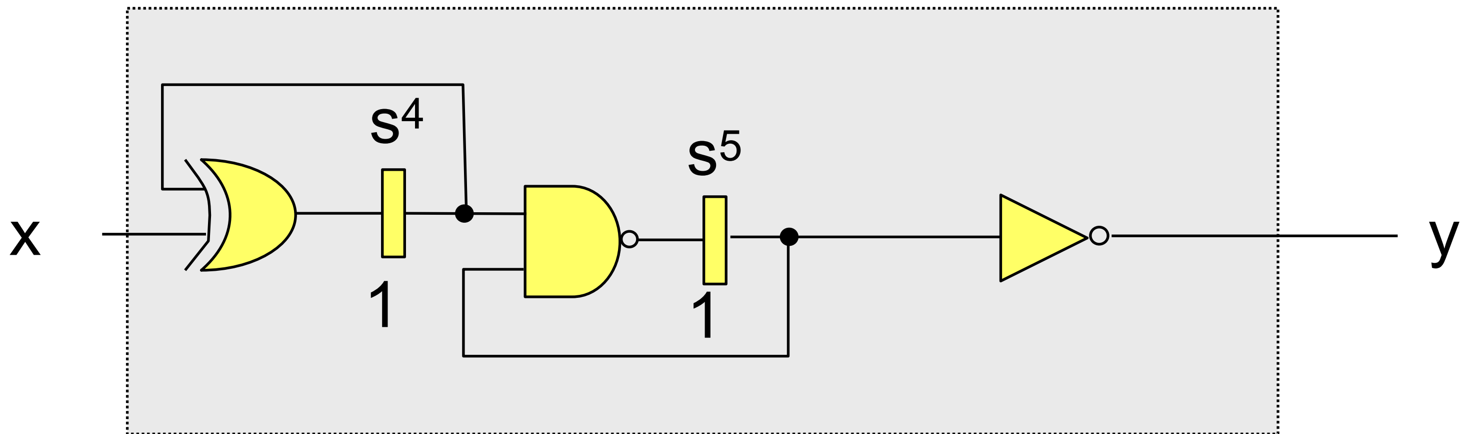
- Basic idea: information is in states

- Quadruple (S,I,R,L)
  - S: finite set of states
  - I: finite set of initial states
  - R: transition relation R is in S x S and is total
    - that is, for all states s there exists a state s' such that (s,s') is in R
  - L: labelling function. L(s) = propositions true in state s

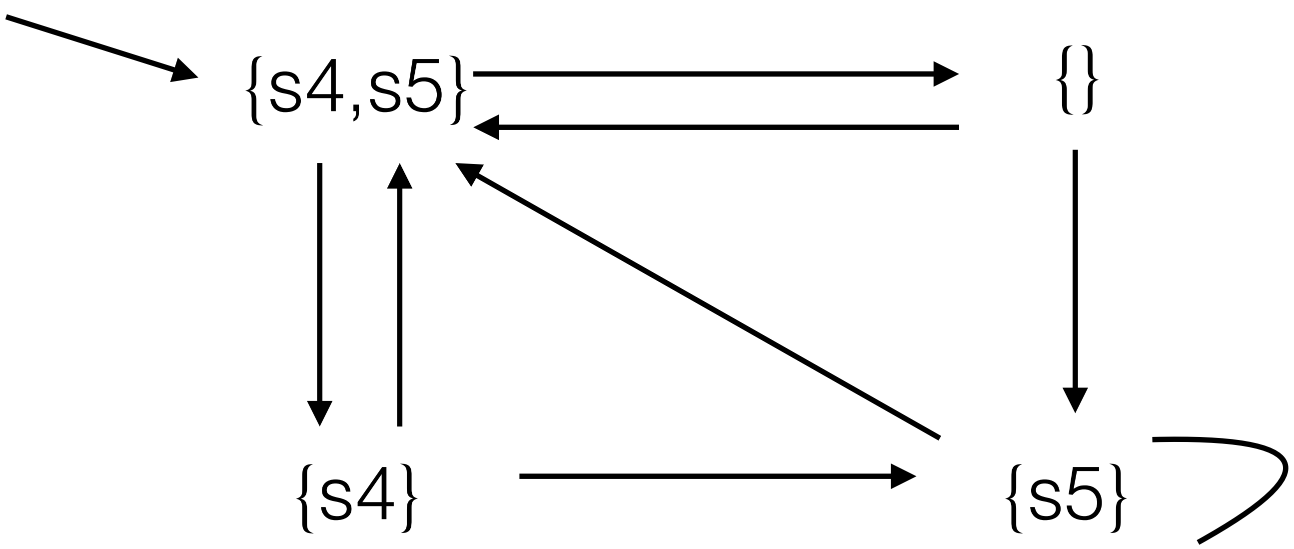
- A path is an infinite sequence of states.

- Labels are "atomic propositions", or simply "bits" or "signals".

Kripke structure - Example



sequential circuit



Kripke structure

Some syntax first

LTL - Linear Time Logic

» The only state formulas permitted are atomic propositions.

CTL - Computation Tree Logic

- Main restriction
  - Temporal operators (e.g. X,U) **must be immediately** preceded by a path quantifier

LTL - Syntax

» If p is an atomic proposition, then it is a path formula

» If f and g are path formulas, then the following are also path formulas:

$$\neg f \qquad f \wedge g \qquad \mathbf{X}\ f \qquad f\ \mathbf{U}\ g$$

CTL - Syntax

» State formulas are either atomic propositions or if f and g are state formulas, the following are also state formulas:

$$\neg f \qquad f \wedge g \qquad \mathbf{E} f \qquad \mathbf{A} f$$

» If f and g are **state** formulas, then they are path formulas.

» Additional path formulas can be created using:

$$\mathbf{X}\ f \qquad f\ \mathbf{U}\ g$$

Syntax of CTL*

- CTL* state-formulae are formed according to:

$$\phi ::= \top \mid a \mid \phi_1 \wedge \phi_2 \mid \neg\phi \mid \mathbf{E}\psi$$

  where $a \in AP$, ϕ, ϕ_1, ϕ_2 are state-formulae, and ψ is a path-formula
- CTL* path-formulae are formed according to:

$$\psi ::= \phi \mid \psi_1 \wedge \psi_2 \mid \neg\psi \mid \mathbf{X}\psi \mid \psi_1 \, \mathbf{U} \psi_2$$

  where ϕ is a state-formula, and ψ, ψ_1, ψ_2 are path-formulae
- Path-quantifiers and temporal operators do not have to alternate anymore
- In CTL* we can define $\mathbf{A}\psi = \neg\mathbf{E}\neg\psi$ which is not possible in CTL!

Some notations

» A path is noted π

» Suffix of a path starting at index i is noted π^i

» State at index i in a path is noted $\pi(i)$

» Kripke structure M satisfies a property for a path is noted $M, \pi \models f$

LTL semantics (1)

$M, \pi \models p$	iff	$p \in L(\pi(0))$
$M, \pi \models \neg f$	iff	$M, \pi \not\models f$
$M, \pi \models f \wedge g$	iff	$M, \pi \models f \text{ and } M, \pi \models g$
$M, \pi \models \mathbf{X}\ f$	iff	$M, \pi^1 \models f$
$M, \pi \models f\ \mathbf{U}\ g$	iff	$\exists i. M, \pi^i \models g \text{ and} \forall j < i. M, \pi^j \models f$

All other formulas can be derived from the ones on this slide.

Derived operators

- $\varphi \vee \psi \equiv \neg(\neg\varphi \wedge \neg\psi)$
- $\varphi \Rightarrow \psi \equiv \neg\varphi \vee \psi$
- $\varphi \Leftrightarrow \psi \equiv (\varphi \Rightarrow \psi) \wedge (\psi \Rightarrow \varphi)$
- True (or $\top$) $\equiv \varphi \vee \neg\varphi$
- False (or $\bot$) $\equiv \neg\top$
- $\mathbf{F}\varphi$ (also noted $\Diamond\varphi$) $\equiv \top\ \mathbf{U}\ \varphi$ "eventually φ"
- $\mathbf{G}\varphi$ (also noted $\Box\varphi$) $\equiv \neg\mathbf{F}\neg\varphi$ "globally φ"

CTL Semantics - state formulas

$M, s \models p$	iff	$p \in L(s)$
$M, s \models \neg f$	iff	$M, s \not\models f$
$M, s \models f \wedge g$	iff	$M, s \models f \text{ and } M, s \models g$
$M, s \models \mathbf{E}\ f$	iff	$\exists \pi . \pi(0) = s \wedge M, \pi \models f$
$M, s \models \mathbf{A}\ f$	iff	$\forall \pi . \pi(0) = s \wedge M, \pi \models f$

CTL Semantics - path formulas

$M, \pi \models p$	iff	$M, \pi(0) \models p$
$M, \pi \models \neg f$	iff	$M, \pi \not\models f$
$M, \pi \models f \wedge g$	iff	$M, \pi \models f \text{ and } M, \pi \models g$
$M, \pi \models \mathbf{X}\ f$	iff	$M, \pi^1 \models f$
$M, \pi \models f\ \mathbf{U}\ g$	iff	$\exists i. M, \pi^i \models g \text{ and} \forall j < i. M, \pi^j \models f$

CTL* Semantics - state formulas

» Same as for CTL

CTL* Semantics - path formulas

$M, \pi \models f$ iff $M, \pi(0) \models f$

$M, \pi \models \neg f$ iff $M, \pi \not\models f$

$M, \pi \models f \wedge g$ iff $M, \pi \models f \text{ and } M, \pi \models g$

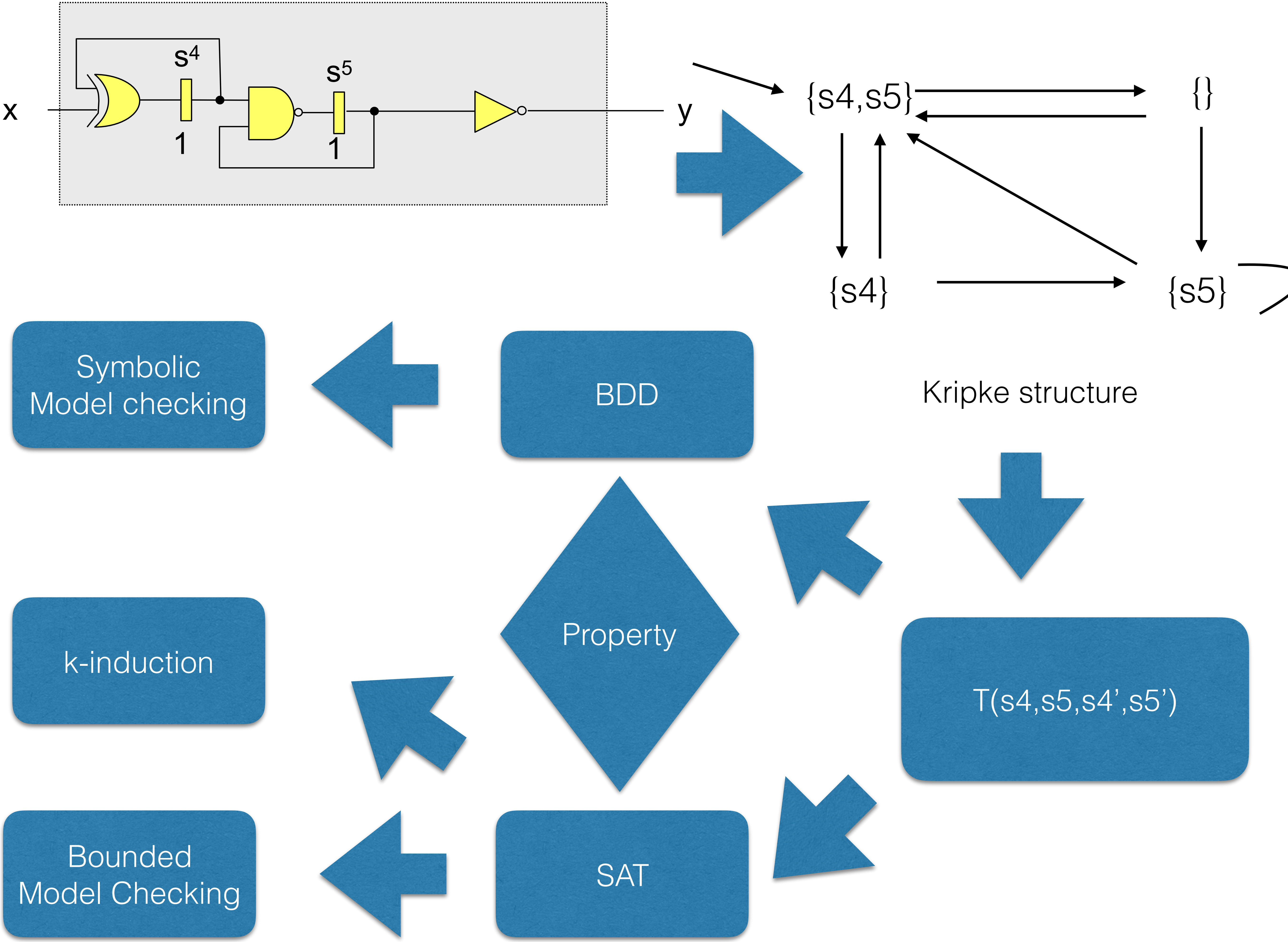
$M, \pi \models p$ iff $M, \pi(0) \models p$

$M, \pi \models \neg f$ iff $M, \pi \not\models f$

$M, \pi \models f \wedge g$ iff $M, \pi \models f \text{ and } M, \pi \models g$

$M, \pi \models \mathbf{X}\ f$ iff $M, \pi^1 \models f$

$M, \pi \models f\ \mathbf{U}\ g$ iff $\exists i. M, \pi^i \models g \text{ and} \forall j < i. M, \pi^j \models f$



x
s4
1
s5
1
y
{s4,s5}
{}
{s4}
{s5}
Kripke structure
Symbolic
Model checking
BDD
Property
T(s4,s5,s4',s5')
k-induction
Bounded
Model Checking
SAT