THEORY OF SYSTEMS
MODELING AND ANALYSIS

Henny Sipma
Stanford University

Master class
Washington University at St Louis
November 16, 2006

COURSE OUTLINE

8:37 - 10:00	Introduction -- Computational model Fair transition systems -- Temporal logic
10:07 - 11:30	Verification methods Rules -- Diagrams -- Abstraction Traditional static analysis methods
1:07 - 2:30	Constraint-based static analysis methods Real-time and hybrid systems
2:37 - 4:00	Formalization of middleware services: Event correlation

My background

B.S. Chemistry, Groningen, The Netherlands

M.S. Chemical Engineering, idem

---

7 years as Control Engineer with Shell in
The Netherlands, Singapore and Houston, TX

M.S. Computer Science, Stanford

Ph.D. Computer Science, Stanford

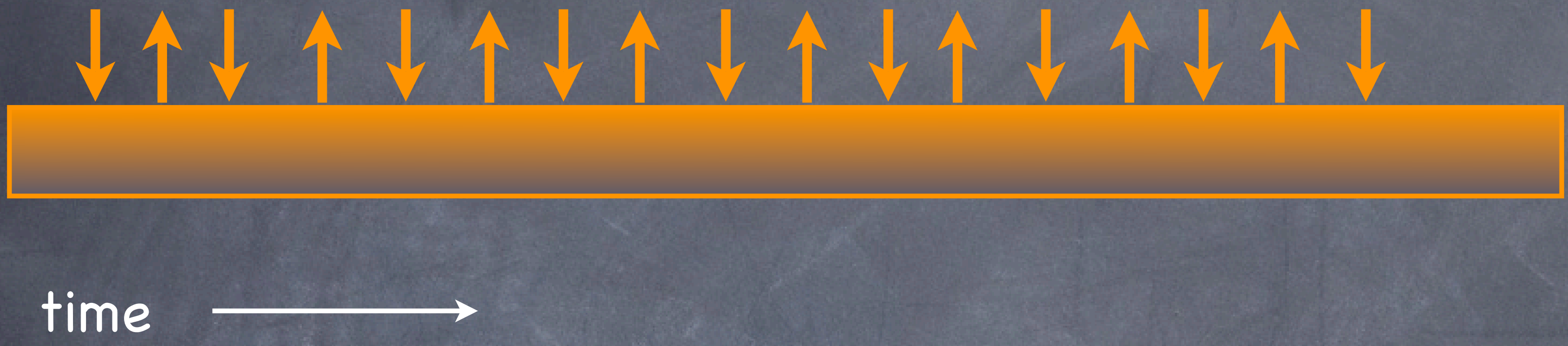
Sr. Research Associate at Stanford since 2000

Research Interests

- **Static Analysis**
  - Constraint-based reasoning
- **Runtime Analysis**
  - Monitoring of temporal properties
- **Decision Procedures**
  - Data structures
- **Formalization of Middleware services**
  - Event Correlation
  - Deadlock Avoidance

I. Introduction

Reactive Systems



Continuous interaction with the environment

Observable throughout its execution

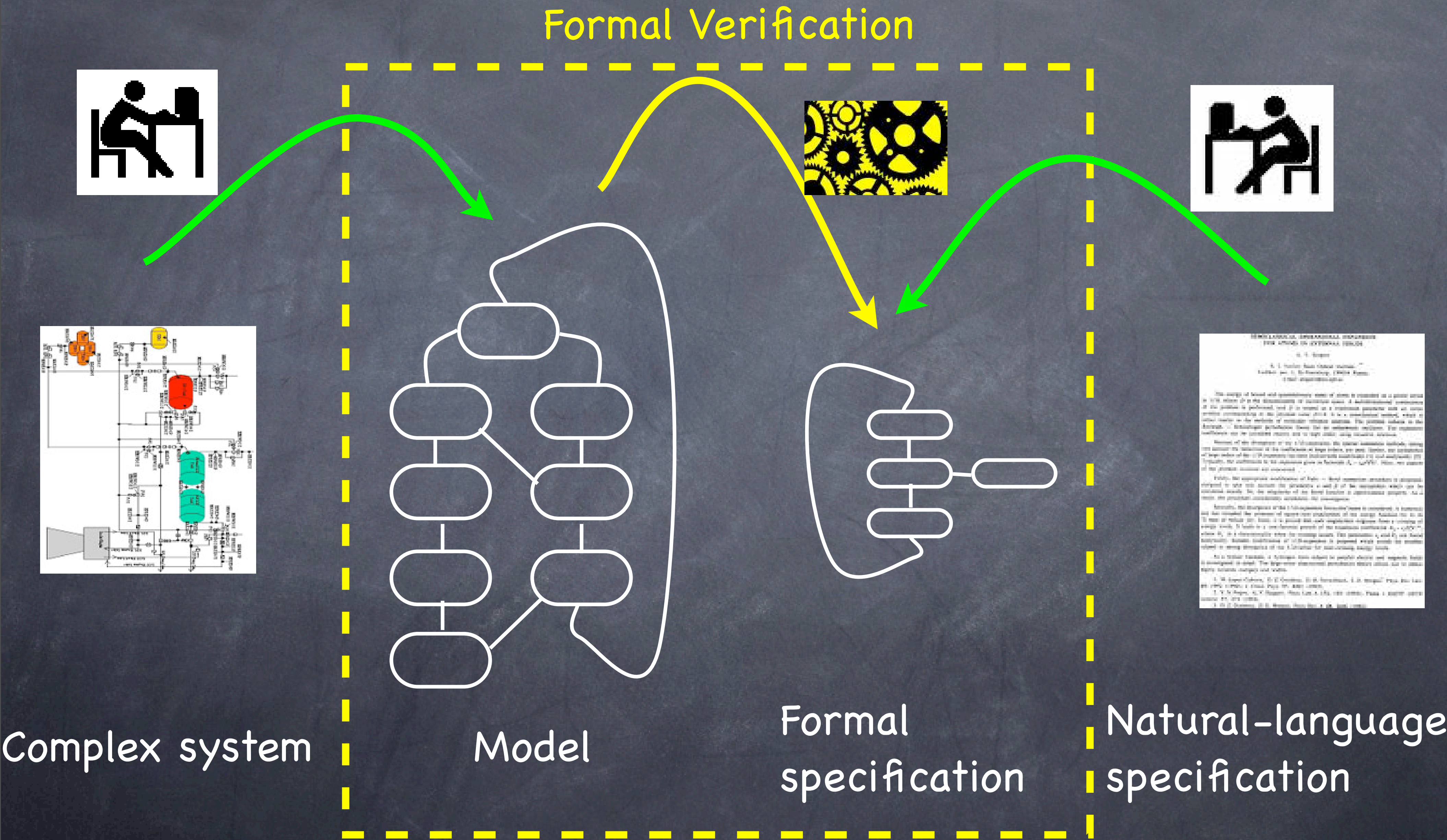
OBJECTIVE: Analysis of System Behaviors

COMPONENTS: Model (+ Specification)

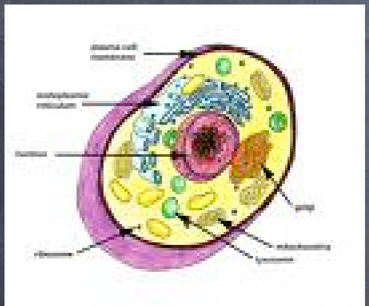
METHODS: Deductive and Algorithmic

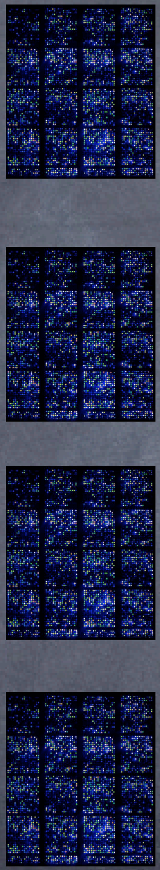
THEORY: Logic + Automata

FORMAL METHODS - Scope

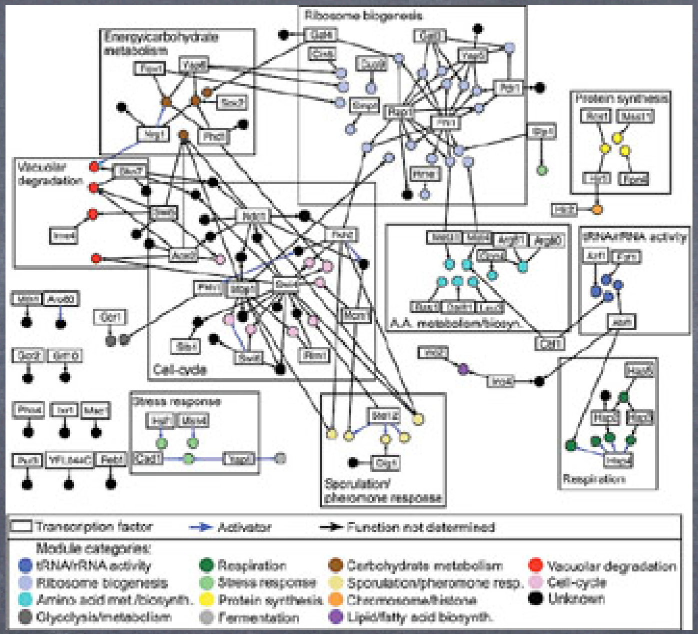


FORMAL METHODS - Scope





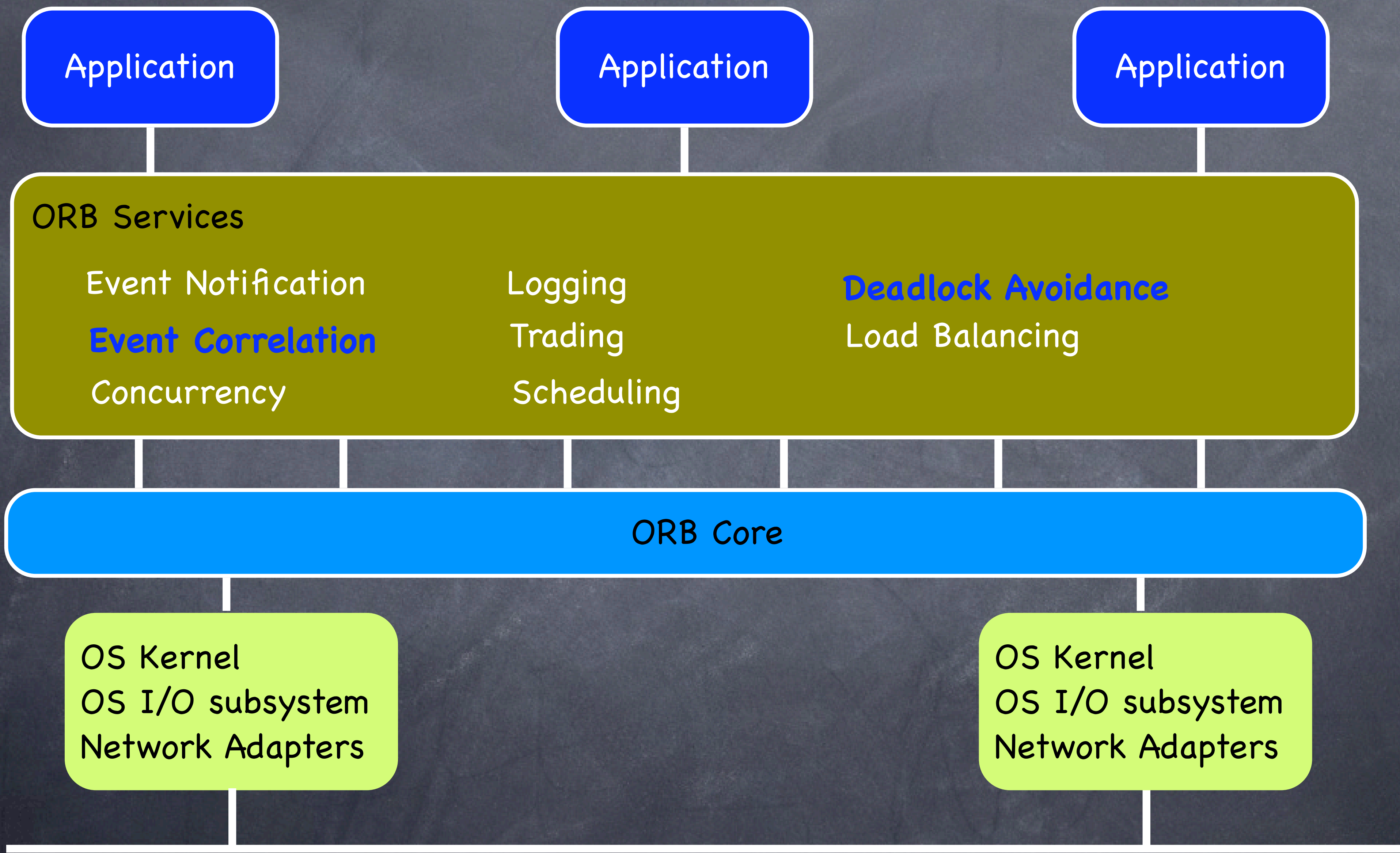
Gene regulatory network



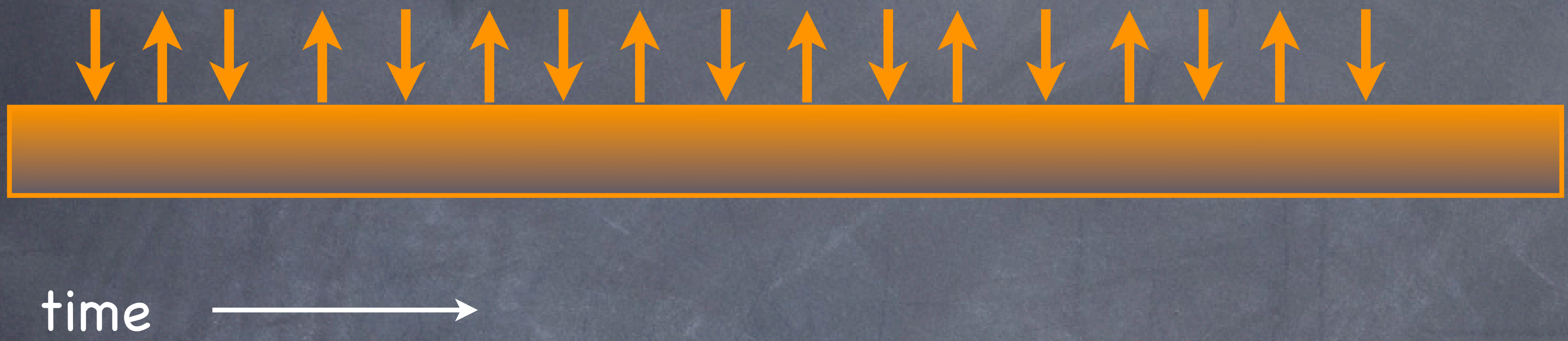
Static analysis

properties

FORMALIZATION OF MIDDLEWARE SERVICES



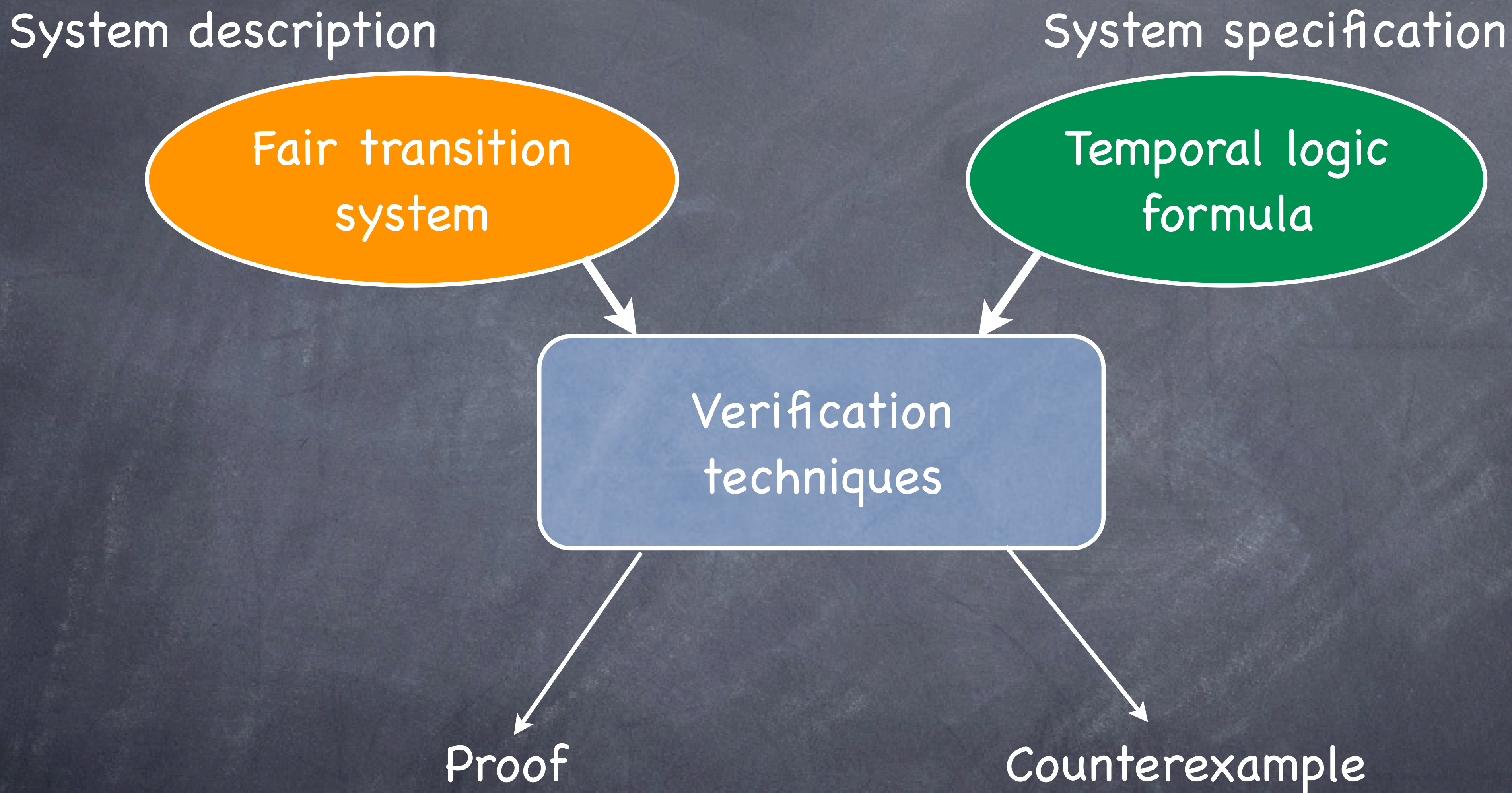
Reactive Systems



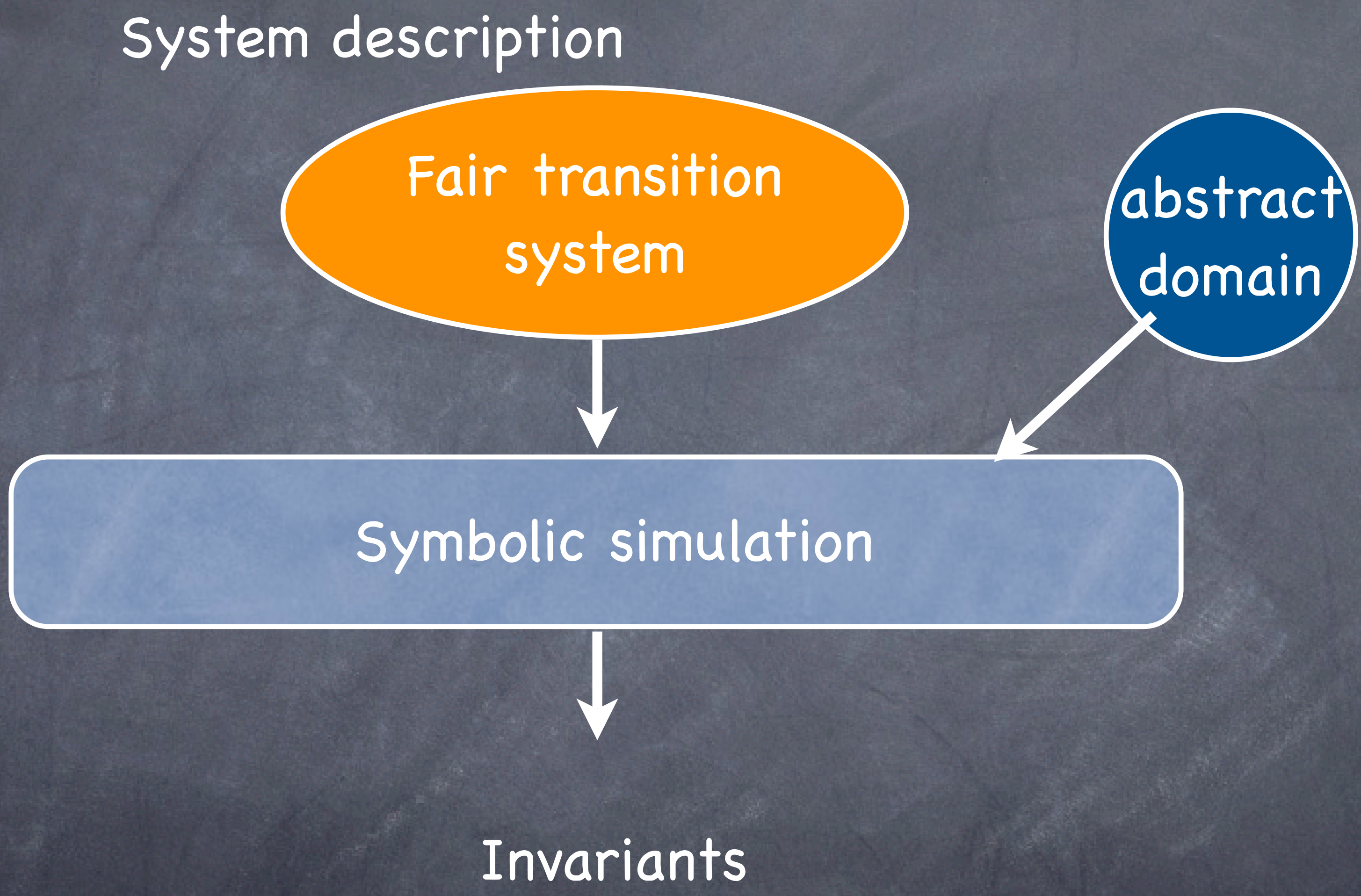
Behavior: sequences of states

Specification: temporal logic

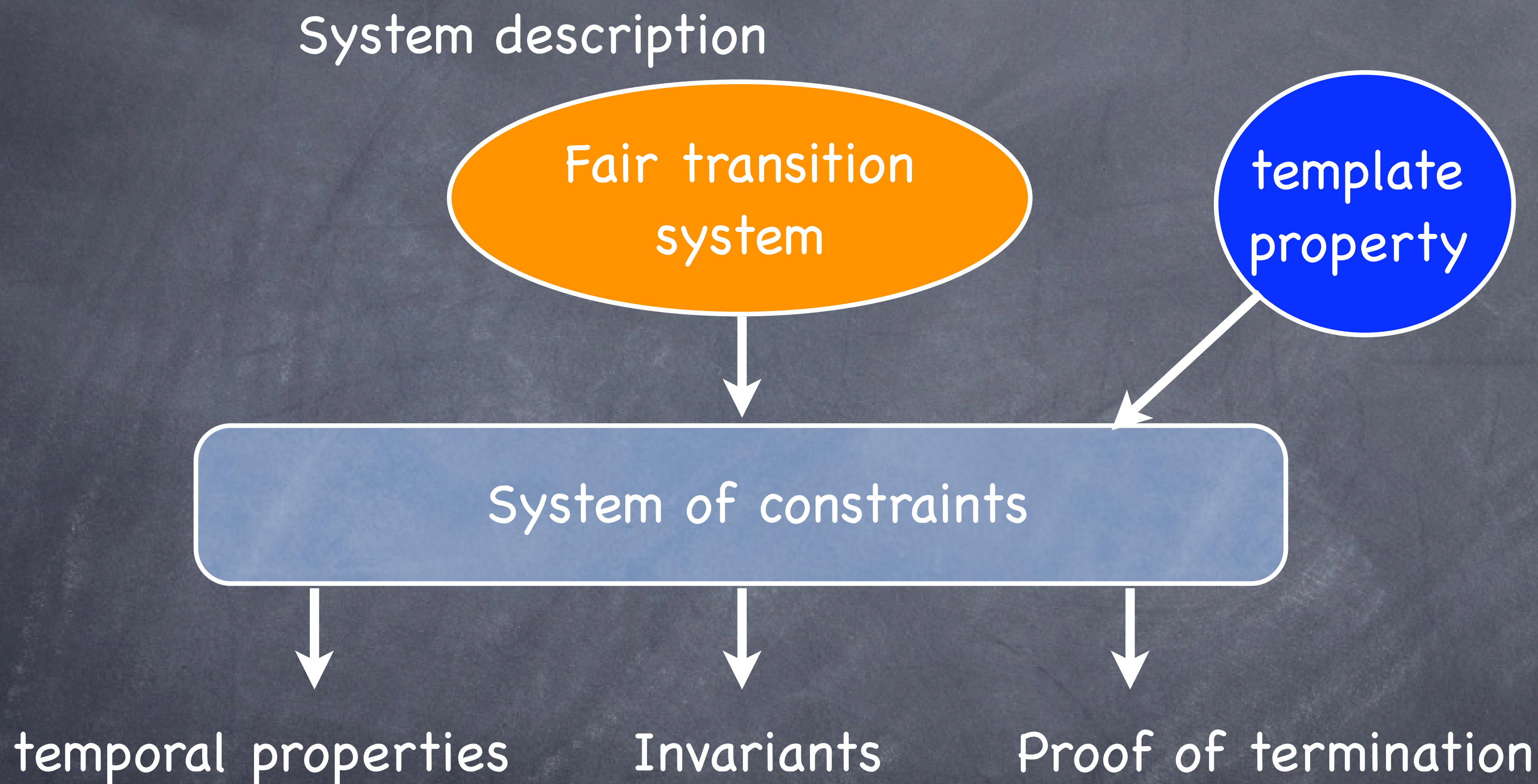
Verification process



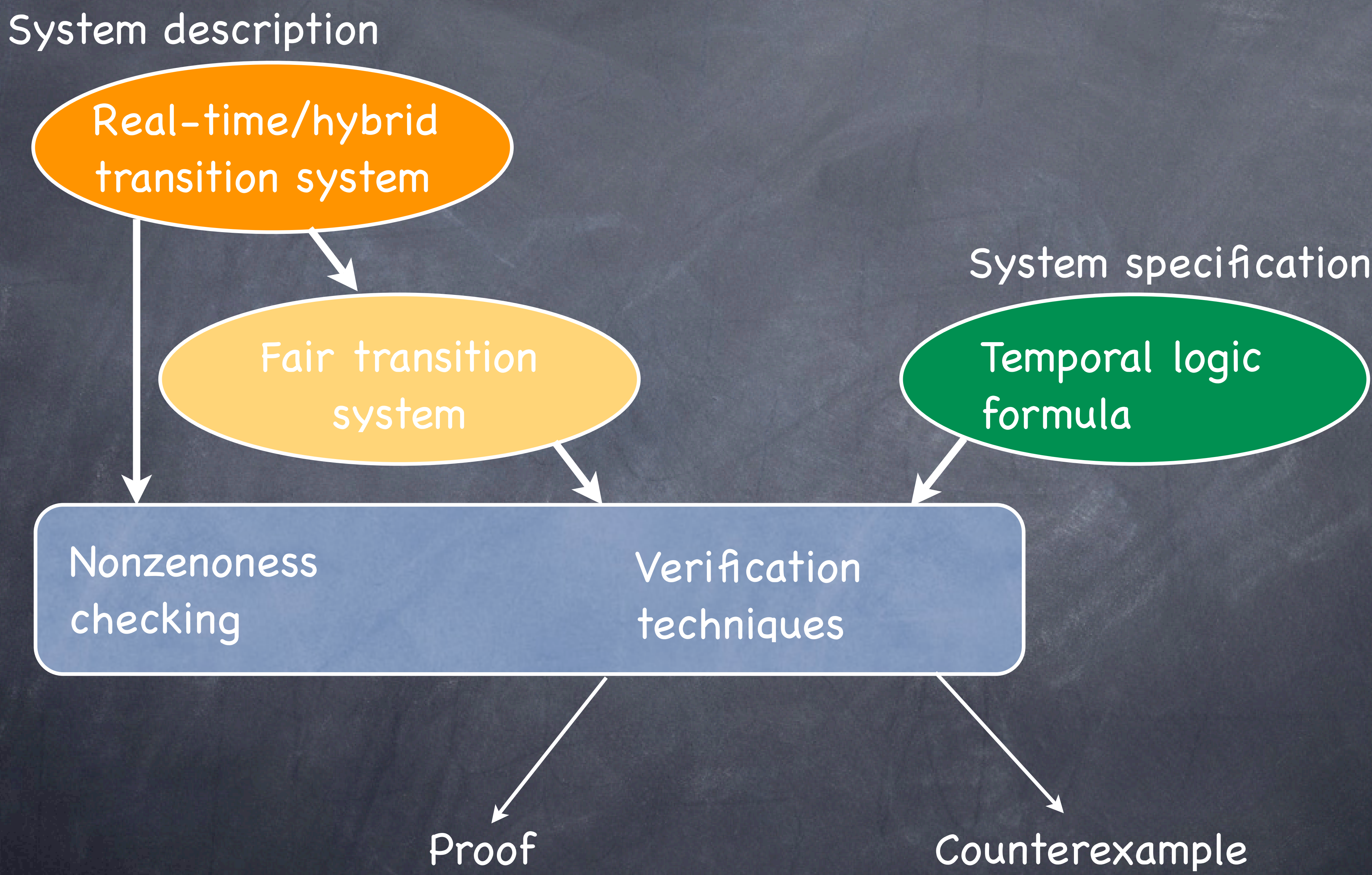
Static analysis (traditional)



Static analysis (constraint-based)



Extension to real-time and hybrid systems



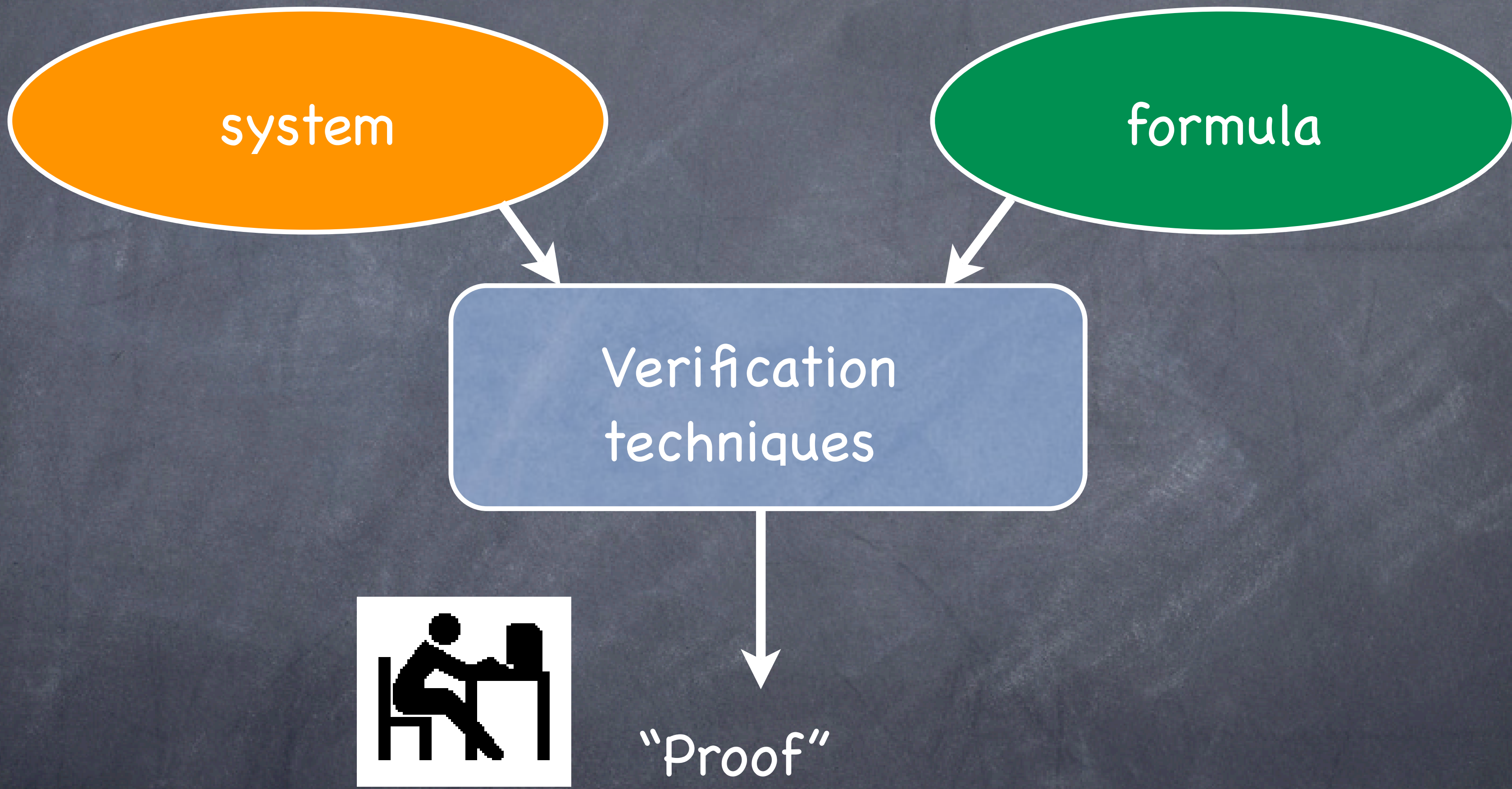
Verification techniques (1)

Algorithmic: exhaustive search for counterexamples

Issues:
- state space explosion problem
- efficient representations
- applicable to finite-state systems only

Verification techniques (2)

Deductive: “theorem proving”



Verification techniques (2)

Deductive: “theorem proving”

system

formula

Verification techniques

100s/1000s/10,000s of first-order verification conditions

Decision procedures

II. COMPUTATIONAL MODEL

- **Fair transition systems**
- Temporal logic: LTL

Reference:

Zohar Manna, Amir Pnueli, Temporal Verification of Reactive Systems: Safety, Springer-Verlag, 1995.

States

V: Vocabulary -- set of typed variables {x,y: integer, b: boolean}

expression over V x+y

assertion over V x>y

s: state -- interpretation of all variables {x:2,y:3,b:true}

s[x]=2,s[y]=3,s[b]=true

extends to expressions s[x+y]=5
and assertions s[x>y]=false

Σ: set of all states $Z \times Z \times$ {true,false}

System Description: Fair transition systems

Set of typed variables

Example: {x,y}

Fairness condition

$\mathcal{F} \subseteq \mathcal{T}$

$\Phi: \langle V, \Theta, \mathcal{T}, \mathcal{F} \rangle$

Initial condition: first-order formula

Set of transitions

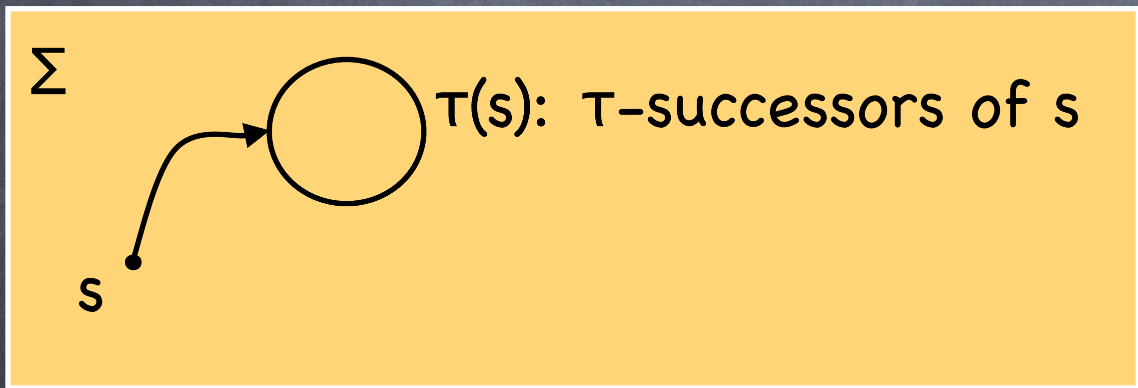
Example: $x=0 \wedge y=0$

Compact first-order representation of all sequences of states that can be generated by a system

Transitions

$\mathcal{T}$: finite set of transitions

$$\tau \in \mathcal{T}: \ \Sigma \rightarrow 2^{\Sigma}$$



Example:

ρ_τ: $x' = x+1 \vee x' = x+2$

$\tau(\langle x:2 \rangle) = \{\langle x:3 \rangle, \langle x:4 \rangle\}$

represented by a transition relation $\rho_\tau(V,V')$

V : values of variables in the current state

V′: values of variables in the next state

Runs and Computations

Infinite sequence of states

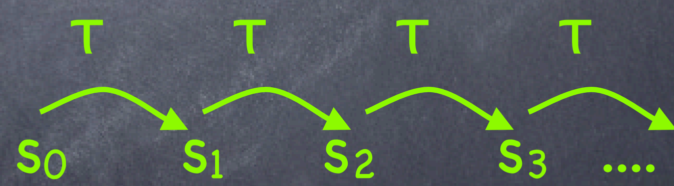
$\sigma: s_0\ s_1\ s_2\ s_3\ s_4\ \ldots\ldots\ldots\ldots$

is a **run** of Φ if

- **Initiality:** $s_0 \models \Theta$ (s₀ is an initial state)
- **Consecution:** for all $i > 0$

  s_{i+1} is a τ-successor of s_i

  for some $\tau \in \mathcal{T}$



Runs and Computations

Infinite sequence of states

σ: $s_0\ s_1\ s_2\ s_3\ s_4$ ....

is a **computation** of Φ if

- σ is a run
- **Justice:** for each $\tau \in \mathcal{J}$

  if τ is enabled infinitely often in σ
  it is taken infinitely often in σ

τ is enabled on S_i: $\tau(s_i) \neq \emptyset$

τ is taken on s_i: $s_{i+1} \in \tau(s_i)$

Runs and Computations: Example

V: {x:integer}

Θ: x=0

$\mathcal{T}$: $\{\tau_1, \tau_2, \tau_3\}$ with

$$\begin{cases} \rho_{\tau 1} : x'=x+1 \vee x'=x+3 \\ \rho_{\tau 2} : x'=x+2 \vee x'=2x \\ \rho_{\tau 3} : x'=x \end{cases}$$

$\mathcal{F}$: $\{\tau_1, \tau_2\}$

		Run?	Computation?
σ_1:	0, 1, 2, 3, 4, 5, 6, 7, ...	✓	×
σ_2:	0, 0, 0, 0, 0, 0, 0, 0	✓	×
σ_3:	0, 2, 4, 8, 16, 32, ...	✓	×
σ_4:	0, 1, 1, 3, 3, 5, 5, 7, 7, ...	✓	×
σ_5:	1, 2, 3, 5, 6, 8, 9, 11, ...	×	×

Runs and Computations: Example

V: {x:integer}

Θ: x=0

$\mathcal{T}$: {τ_1, τ_2, τ_3} with

$\rho_{\tau 1}$: $(x=0 \lor x=1) \land (x'=x+1 \lor x'=x+3)$

$\rho_{\tau 2}$: $x'=x+2 \lor x'=2x$

$\rho_{\tau 3}$: $x'=x$

$\mathcal{F}$: {τ_1, τ_2}

	Run?	Computation?
σ_1: 0, 1, 2, 3, 4, 5, 6, 7, ...	✓	×
σ_2: 0, 0, 0, 0, 0, 0, 0, 0	✓	×
σ_3: 0, 2, 4, 8, 16, 32, ...	✓	✓
σ_4: 0, 1, 1, 3, 3, 5, 5, 7, 7, ...	✓	×
σ_5: 1, 2, 3, 5, 6, 8, 9, 11, ...	×	×

System Description: Summary

Fair transition system: Φ: < V , Θ , $\mathcal{T}$, $\mathcal{F}$ >

Run: Initiality + Consecution

Computation: Run + Justice

$\mathcal{L}(\Phi)$: all computations of Φ "Behavior of the program"

(all sequences of states that satisfy Initiality, Consecution and Justice)

Reachable state space

state s is Φ-reachable if it appears in some Φ-computation

σ: s_0 s_1 s_2 s_3 s_4 ....

system Φ is finite-state if the set of Φ-reachable states is finite

Notation: Σ : state space

$\Sigma_{\Phi\triangleright}$: Φ-reachable state space

Example:

V: $\{b_1, b_2\}$

Θ: $b_1 \wedge b_2$

$\mathcal{T}$: $\{\tau\}$ with ρ_τ: $b_1' = \neg b_1 \wedge b_2' = \neg b_2$

$\Sigma = \{<t,t>,<t,f>,<f,t>,<f,f>\}$

$\Sigma_{\Phi\triangleright} = \{<t,t>,<f,f>\}$

Reachable state space

state s is **Φ-reachable** if it appears in some Φ-computation

σ: s_0 s_1 s_2 s_3 s_4 ....

system Φ is **finite-state** if the set of Φ-reachable states is finite

Notation: Σ : state space

$\Sigma_{\Phi\triangleright}$: Φ-reachable state space

Example:
V: {x}
Θ: x=0
$\mathcal{T}$: {τ} with ρ_τ: $x=0 \wedge x'=x+1$

$\Sigma = N$

$\Sigma_{\Phi\triangleright} = \{x:0, x:1\}$

Reachable state space

state s is **Φ-reachable** if it appears in some Φ-computation

σ: $s_0\ s_1\ s_2\ s_3\ s_4$ ....

system Φ is **finite-state** if the set of Φ-reachable states is finite

Notation: Σ : state space

$\Sigma_{\Phi\triangleright}$: Φ-reachable state space

Example:

V: $\{x\}$

Θ: $0 \leq x \leq M$

$\mathcal{T}$: $\{\tau_1, \tau_2\}$ with

$\rho_{\tau 1}$: $odd(x) \wedge x' = 3x+1$

$\rho_{\tau 2}$: $even(x) \wedge x' = x/2$

$\Sigma = N$

$\Sigma_{\Phi\triangleright} = ?$

Reachable state space vs Computations

System Φ may have any combination of

finite state space	$\otimes$	finite # of computations
infinite state space		infinite # of computations

II. COMPUTATIONAL MODEL

- Fair transition systems
- **Temporal logic: LTL**

Reference:
Zohar Manna, Amir Pnueli, Temporal Verification of Reactive Systems: Safety, Springer-Verlag, 1995.

Temporal Logic

Language that specifies the behavior of a reactive system

System Φ ------------------------------ System behavior: $\mathcal{L}(\Phi)$

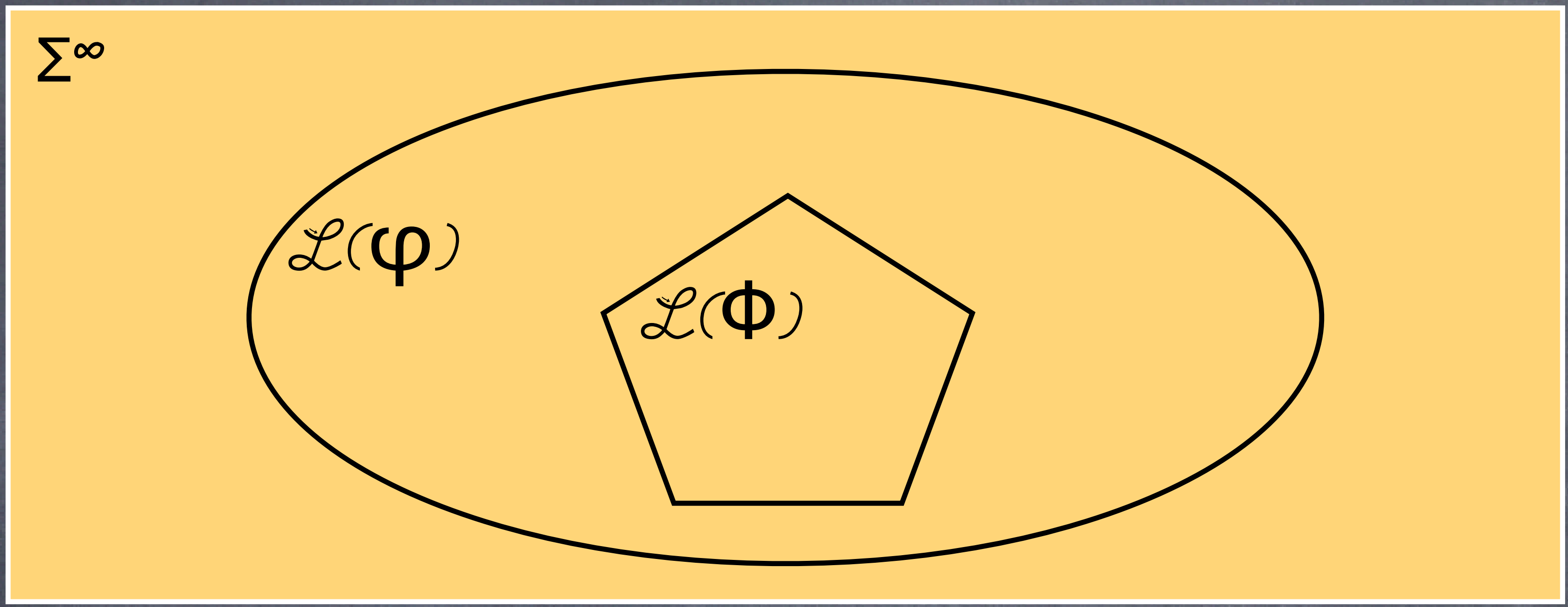
Temporal formula φ --- Sequences of states that satisfy φ: $\mathcal{L}(\varphi)$

System Φ satisfies specification φ

$$\Phi \models \varphi$$

if $\mathcal{L}(\Phi) \subseteq \mathcal{L}(\varphi)$

Temporal logic



System Φ satisfies specification φ

$$\Phi \models \varphi$$

if $\mathcal{L}(\Phi) \subseteq \mathcal{L}(\varphi)$

First-order logic --- Temporal logic

First-order logic

models are states

$\langle x{:}3, y{:}1 \rangle \Vdash x > y$

assertion p
represents
the set of states
for which p is true

Temporal logic

models are sequences of states

$\langle s_0\ s_1\ s_2\ s_3\ \ldots\rangle \vDash \varphi$

temporal formula φ
represents
the set of sequences of states
for which φ is true

Temporal logic: underlying assertion language

Assertion language $\mathcal{A}$:

first-order language over system variables
(+ theories for their domains)

Formulas in $\mathcal{A}$: state formulas (aka assertions)

evaluated over a single state

$s \Vdash p$	iff	$s[p] = \text{true}$

p holds at s
s satisfies s
s is a p-state

Example:

s: <x:4, y:1>

$s \Vdash x=0 \vee y=1$

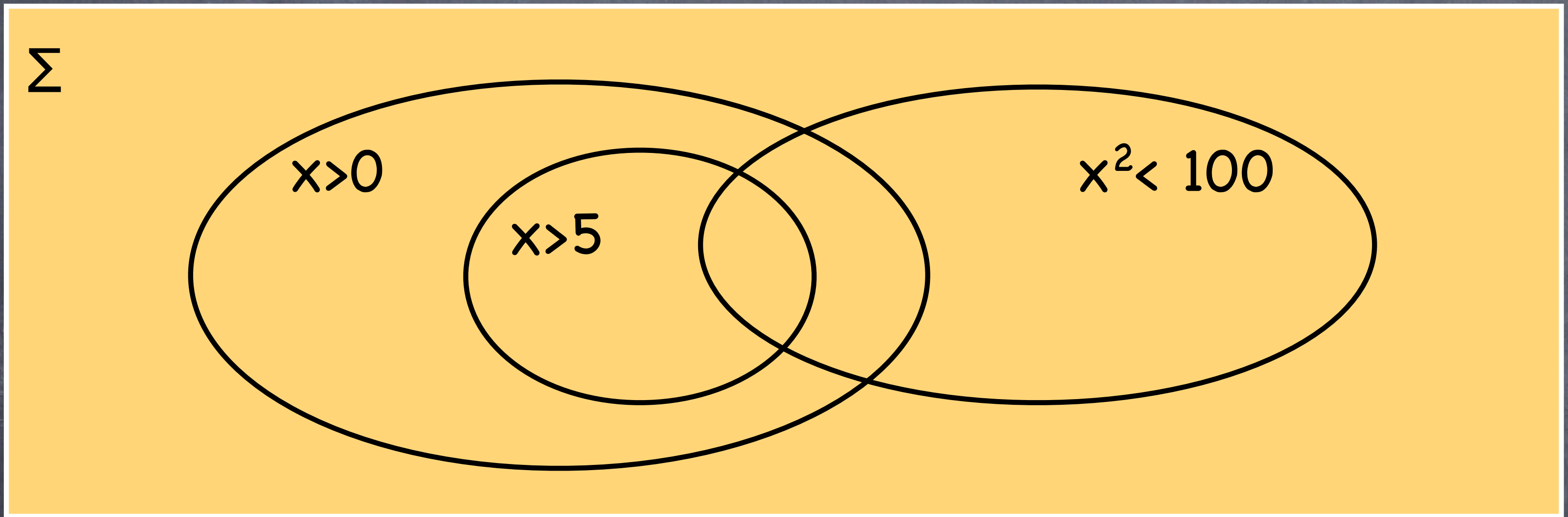
$s \Vdash x > y$

$s \Vdash x = y+3$

$s \nVdash odd(x)$

Temporal logic: underlying assertion language

Assertions represent sets of states



Temporal logic: underlying assertion language

assertion p is state-satisfiable if $s \Vdash p$ for some state $s \in \Sigma$

Example: $x>0$

assertion p is state-valid if $s \Vdash p$ for all states $s \in \Sigma$

Example: $x>y \rightarrow x+1 > y$

Temporal logic

assertions + **temporal operators**

first-order formulas describing the properties of a single state

$\square$ always
$\diamond$ eventually
$\mathcal{U}$ until
$\mathcal{W}$ wait for
$\bigcirc$ next

Temporal logic: safety versus liveness

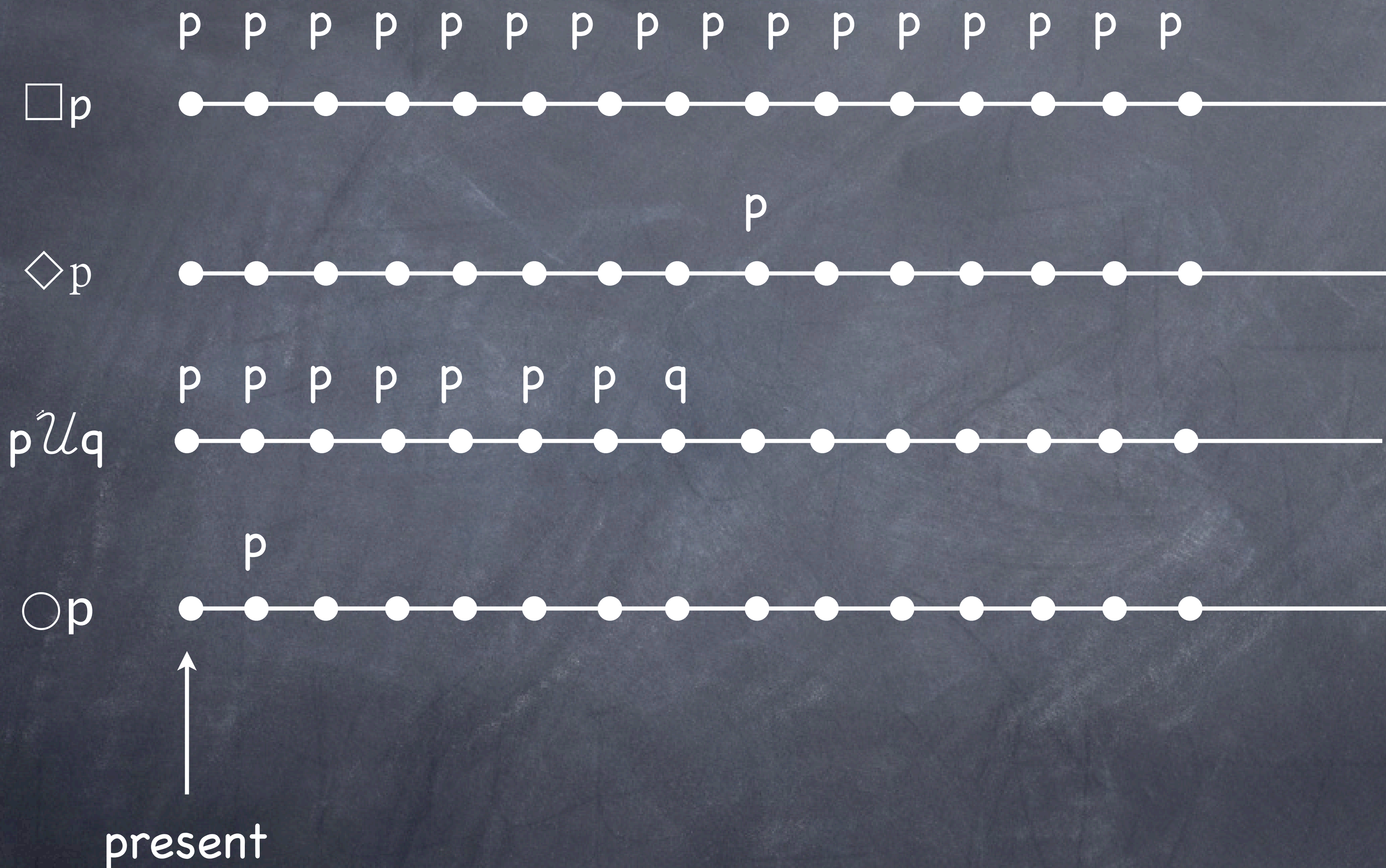
Safety property: "nothing bad will happen"

it will not happen that the train is in the crossing while the gates are open

Liveness property: "something good will happen"

the train will eventually be able to pass the crossing

Temporal logic -- informal



Temporal logic: syntax

- every assertion is a temporal formula
- if φ and ψ are temporal formulas, so are

boolean combinations: $\neg\varphi$ $\varphi\wedge\psi$ $\varphi\vee\psi$ $\varphi\rightarrow\psi$

temporal combinations: $\Diamond\varphi$ $\Box\varphi$ $\bigcirc\varphi$ $\varphi\,\mathcal{U}\,\psi$ $\varphi\,\mathcal{W}\,\psi$

Examples:

$\Box(x=0 \rightarrow \Diamond(x>0))$

$p\,\mathcal{U}\,q \rightarrow \Diamond q$

Temporal logic: semantics

Temporal formulas are evaluated over infinite sequences of states:

σ: s_0 s_1 s_2 s_3 s_4 ....

The truth value of a temporal formula φ over σ at position j in the sequence is

$(\sigma,j) \vDash \varphi$ (φ holds at position j in σ)

Temporal logic: semantics

- if φ is a state formula p

$$(\sigma,j) \vDash \varphi \quad \text{iff} \quad s_j \Vdash p$$

Example: $\sigma\langle x\rangle$: 4, 3, 1, 7, 5, 8, 0, 0, 0, 0

$(\sigma,3) \vDash x > 6$ $\quad$ $\langle x{:}7\rangle \Vdash x > 6$

$(\sigma,6) \vDash x=0$

- if φ is a temporal formula of the form (boolean operators)

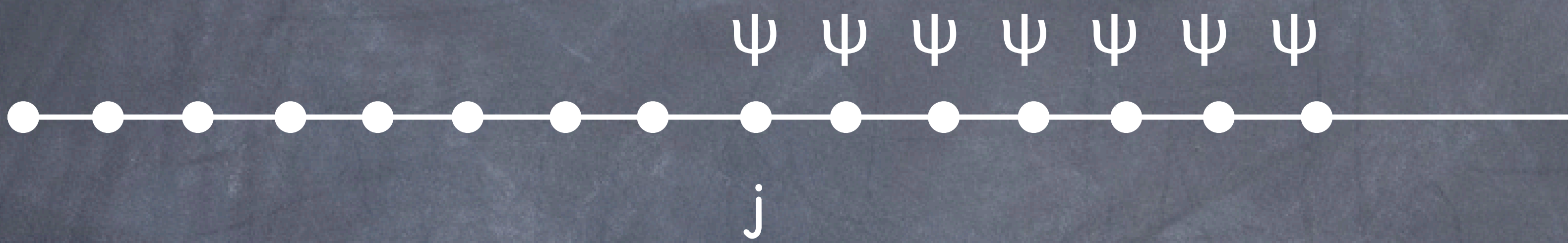
$$(\sigma,j) \vDash \neg\psi \quad \text{iff} \quad (\sigma,j) \nvDash \psi$$

$$(\sigma,j) \vDash \psi \vee \chi \quad \text{iff} \quad (\sigma,j) \vDash \psi \text{ or } (\sigma,j) \vDash \chi$$

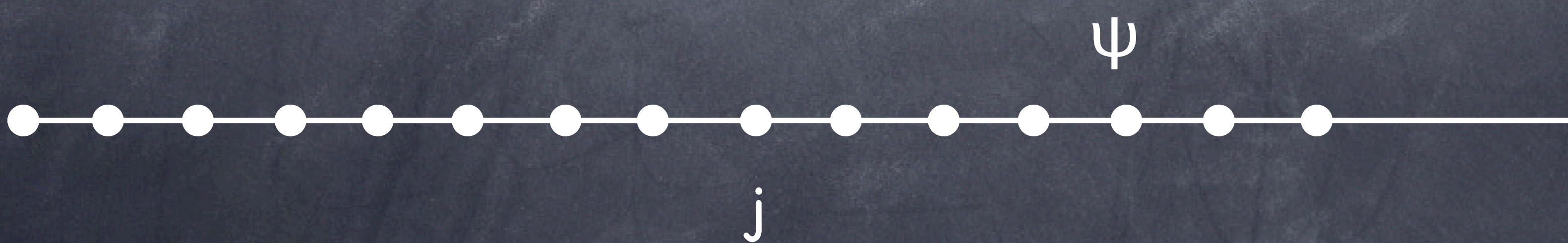
Temporal logic: semantics

- if φ is a temporal formula of the form (temporal operators)

$(\sigma,j) \models \Box\psi$ iff for all $k \geq j$, $(\sigma,j) \VDash \psi$



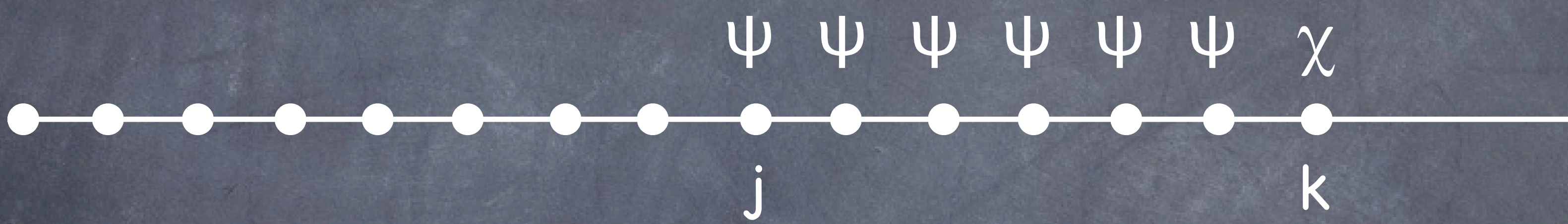
$(\sigma,j) \models \Diamond\psi$ iff for some $k \geq j$, $(\sigma,j) \VDash \psi$



Temporal logic: semantics

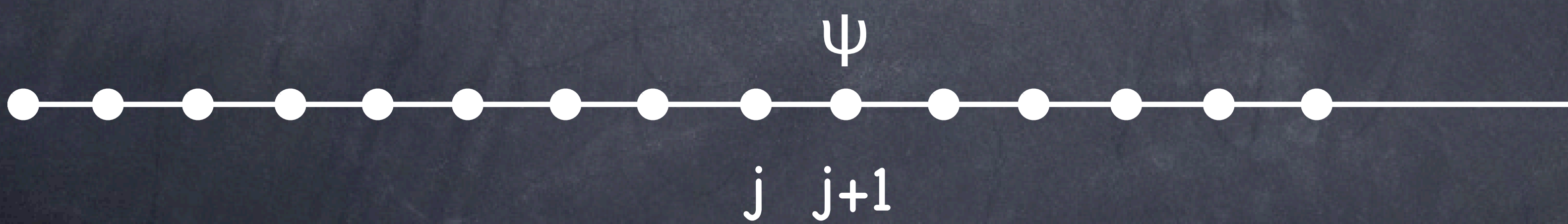
- if φ is a temporal formula of the form (temporal operators)

$(\sigma,j) \models \psi\,\mathcal{U}\,\chi$ iff for some $k \geq j$, $(\sigma,j) \Vdash \chi$
and for all i, $j \leq i < k$, $(\sigma,j) \Vdash \psi$



$(\sigma,j) \models \psi\,\mathcal{W}\,\chi$ iff $(\sigma,j) \models \psi\,\mathcal{U}\,\chi$ or $(\sigma,j) \models \Box\psi$

$(\sigma,j) \models \bigcirc\psi$ iff $(\sigma,j+1) \models \psi$



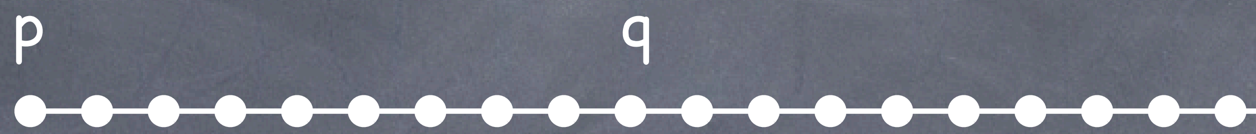
Temporal logic: semantics

A sequence of states σ satisfies a temporal formula φ

$$\sigma \vDash \varphi \qquad \text{iff} \qquad (\sigma,0) \vDash \varphi$$

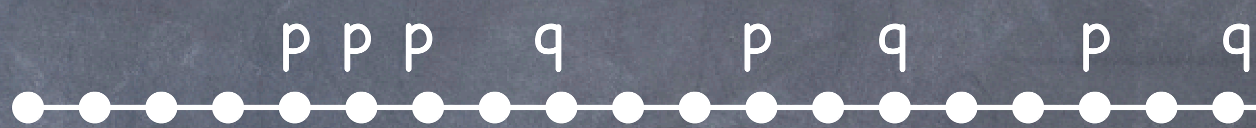
Temporal logic formulas: examples

$p \rightarrow \Diamond q$



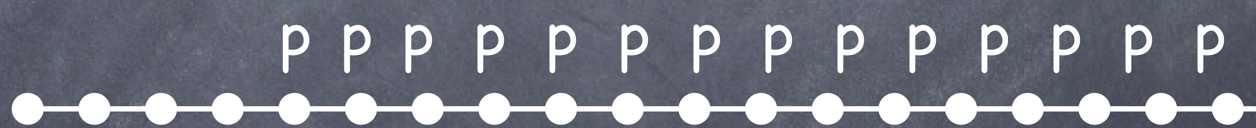
if initially p then eventually q

$\Box(p \rightarrow \Diamond q)$



every p is eventually followed by a q

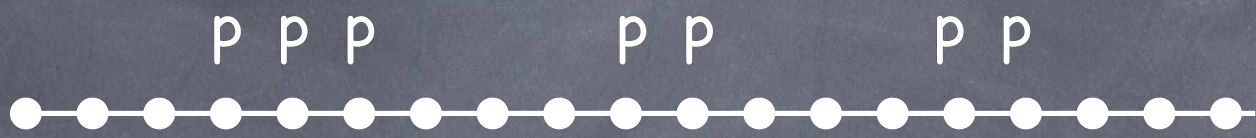
$\Box(p \rightarrow \bigcirc p)$



once p, always p

Temporal logic formulas: examples

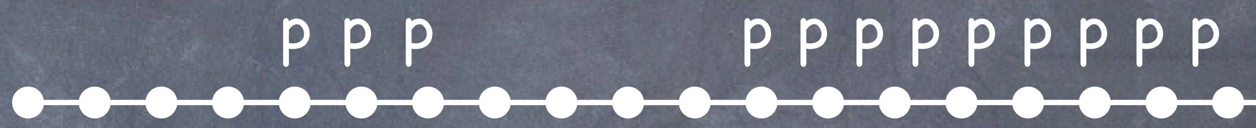
$\Box\Diamond p$



every position is eventually followed by a p

"infinitely often p"

$\Diamond\Box p$



eventually always p

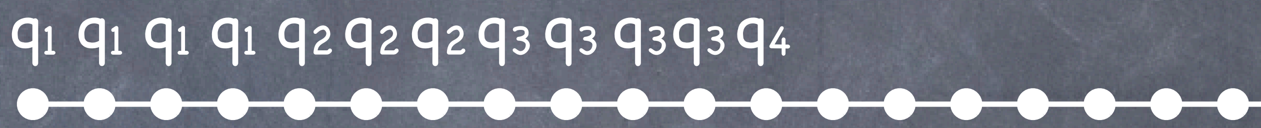
$\Box\Diamond p \rightarrow \Box\Diamond q$

if there are infinitely many p's then there are infinitely many q's

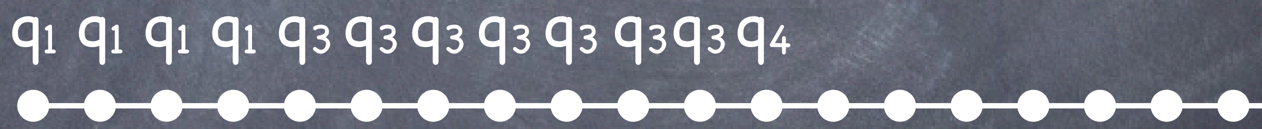
Temporal logic formulas: examples

Nested waiting-for formulas: $q_1 \mathcal{W} (q_2 \mathcal{W} (q_3 \mathcal{W} q_4))$

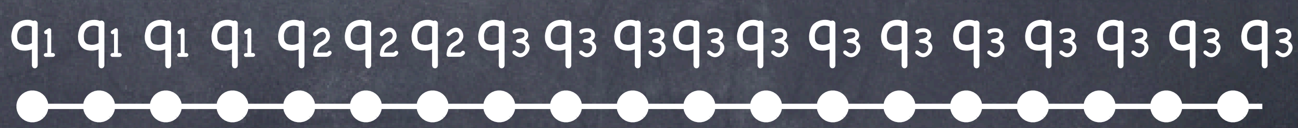
intervals of continuous q_i:



possibly empty interval:



possibly infinite interval:



Temporal logic: summary

For temporal formula φ, sequence of states σ, position $j \geq 0$:

$$(\sigma, j) \vDash \varphi$$

φ holds at position j in σ

σ satisfies φ at j

j is a φ-position in σ

For temporal formula φ and sequence of states σ

$$\sigma \vDash \varphi \quad \text{iff} \quad (\sigma, 0) \vDash \varphi$$

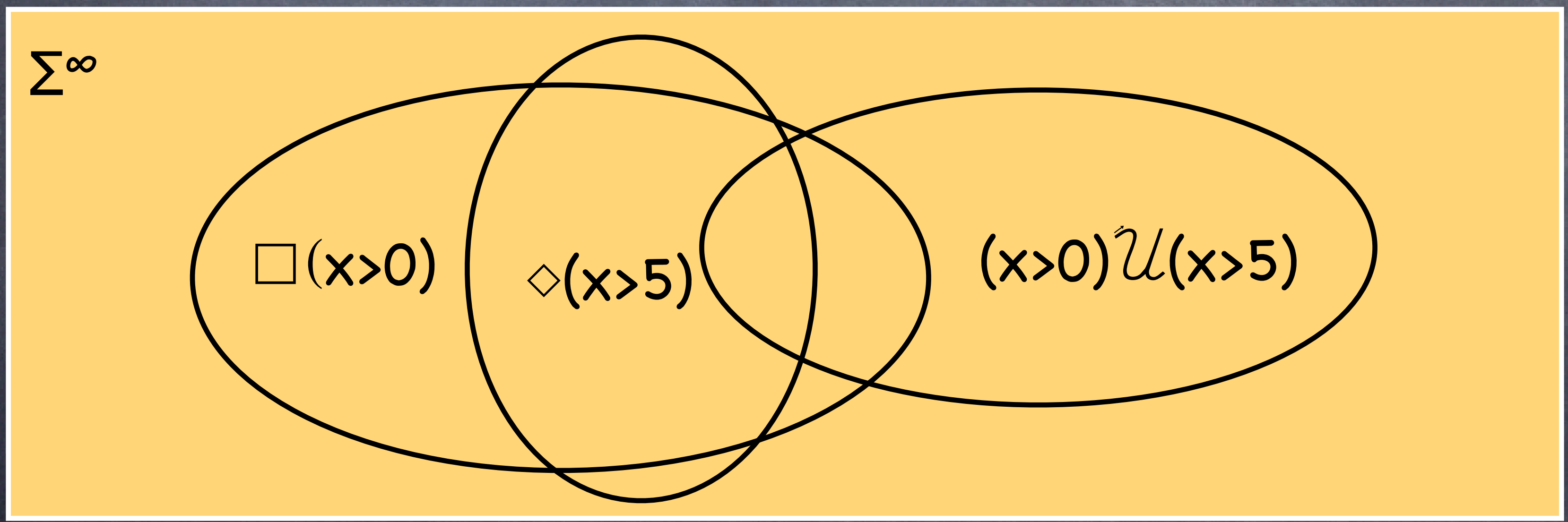
φ holds on σ

σ satisfies φ

Temporal logic: satisfiability and validity

For temporal formula φ

- φ is satisfiable if $\sigma \vDash \varphi$ for some sequence of states σ
- φ is valid if $\sigma \vDash \varphi$ for all sequences of states σ



Temporal logic: satisfiability/validity examples

	satisfiable?	valid?
$\Diamond(x=0)$	✓	×
$\Diamond(x=0) \lor \Box(x\neq 0)$	✓	✓
$\Diamond(x=0) \land \Box(x\neq 0)$	×	×
$\Diamond(x=0) \land \Diamond(x=1)$	✓	×
$\Diamond(x=0) \lor \Diamond(x=1)$	✓	×
$\Diamond\Box p \rightarrow \Box\Diamond p$	✓	✓
$\Box\Diamond p \rightarrow \Diamond\Box p$	✓	×
$p\,\mathcal{U}(q\land r) \rightarrow (\Diamond q \land \Diamond r)$	✓	✓

Temporal logic: Equivalences

Temporal formulas φ, ψ are congruent $\varphi \approx \psi$

if $\Box(\varphi \leftrightarrow \psi)$ is valid

φ and ψ have the same truth value at all positions in all models

		congruent?
$\Box(p \wedge q)$	$\Box p \wedge \Box q$	✓
$\Box(p \vee q)$	$\Box p \vee \Box q$	×
$p\,\mathcal{U}(q \vee r)$	$p\,\mathcal{U}q \vee p\,\mathcal{U}r$	✓
$p\,\mathcal{U}(q \wedge r)$	$p\,\mathcal{U}q \wedge p\,\mathcal{U}r$	×

Temporal logic: Expansions

$$\Box\varphi \approx \varphi \wedge \bigcirc\Box\varphi$$

$$\Diamond\varphi \approx \varphi \vee \bigcirc\Diamond\varphi$$

$$\varphi\,\mathcal{U}\,\psi \approx \psi \vee (\varphi \wedge \bigcirc(\varphi\,\mathcal{U}\,\psi))$$

Used in checking temporal formulas in model checking

Expressiveness

Some properties cannot be expressed in LTL:

☛ p is true, if at all, only at even positions

Not specified by

$p \wedge \Box(p \rightarrow \bigcirc\bigcirc p)$ or $p \wedge \Box(p \leftrightarrow \neg\bigcirc p)$

requires quantification

$$\exists t \; (t \wedge \Box(t \leftrightarrow \neg\bigcirc t) \wedge \Box(p \rightarrow t))$$

Temporal logic vs First-order logic

Temporal formula

$$\Box(p \rightarrow \Diamond(r \wedge \Diamond q))$$

can be expressed in first-order logic as

$$(\forall t_1 \geq 0) \left[p(t_1) \rightarrow (\exists t_2) \left[\begin{array}{c} t_1 \leq t_2 \wedge r(t_2) \wedge \\ (\exists t_3)(t_2 \leq t_3 \wedge q(t_3) \end{array} \right] \right]$$