Monitoring in practice

Ennio Visconti

Outline

Part 1 (today)

- Brief introduction
- Tools & setup
- Scripting and basic syntax (Python)
- Small exercises together

Outline

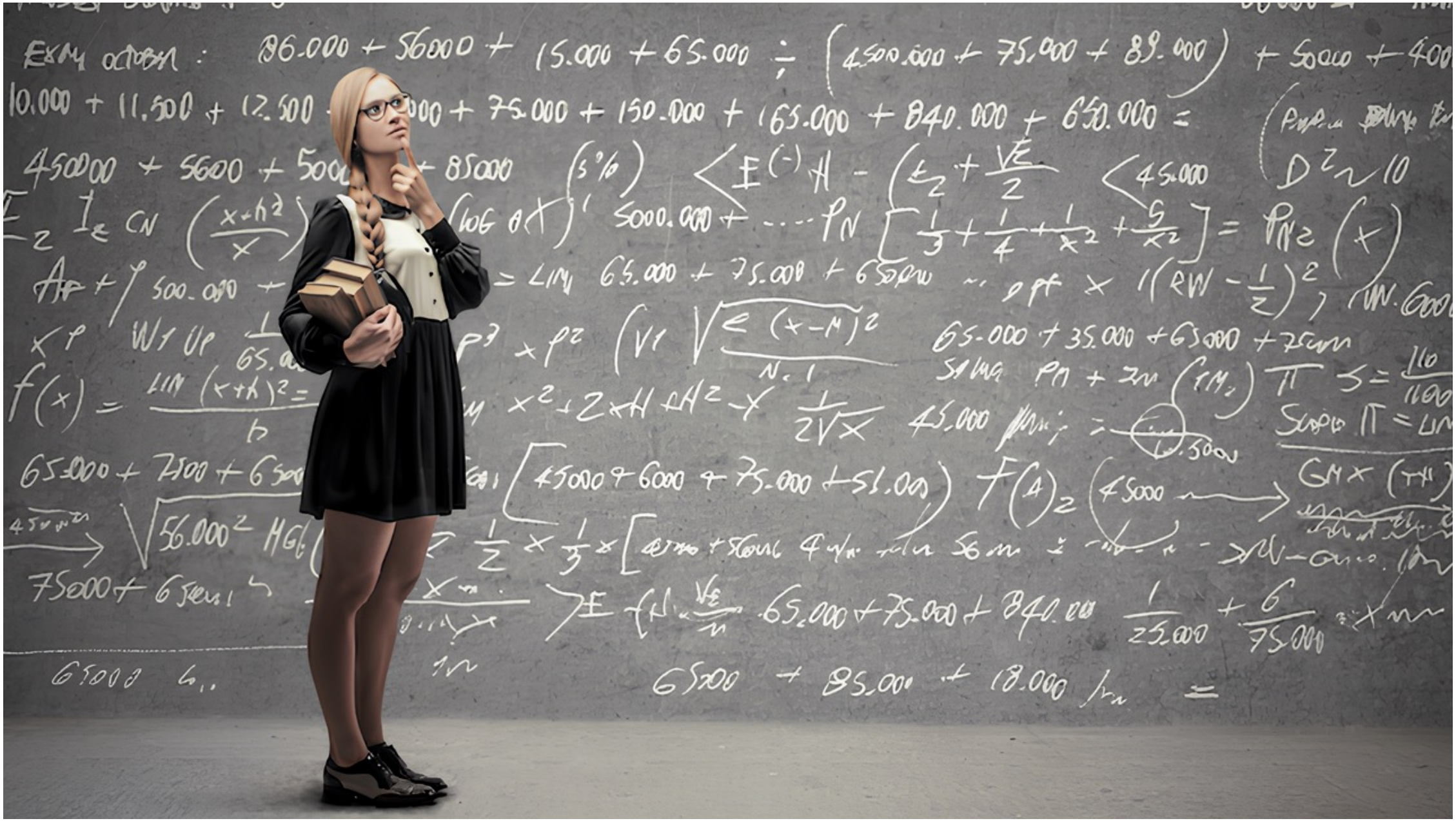
Part 2 (next Wednesday)

- A complex use case:
  - Brief introduction
  - Discussion of requirements
  - Live coding together (Java or Kotlin)

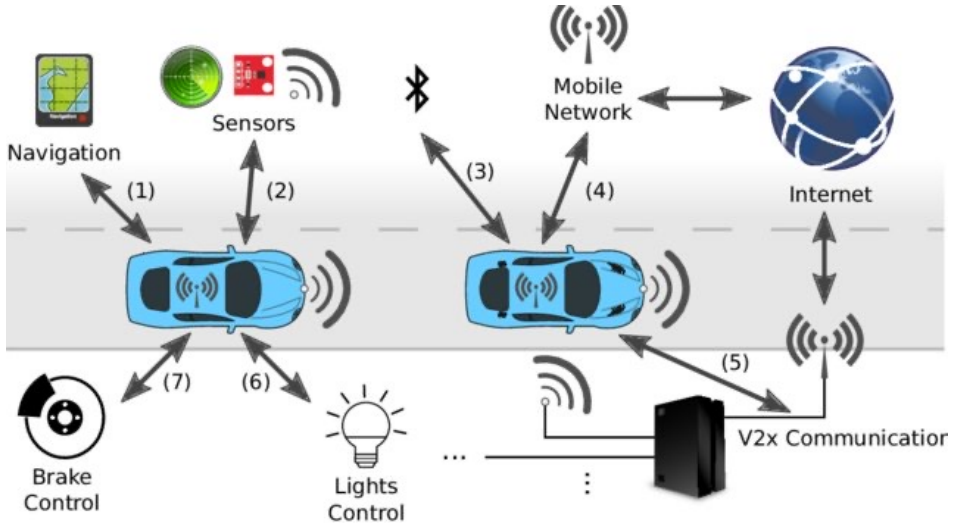
Part 2- Survey

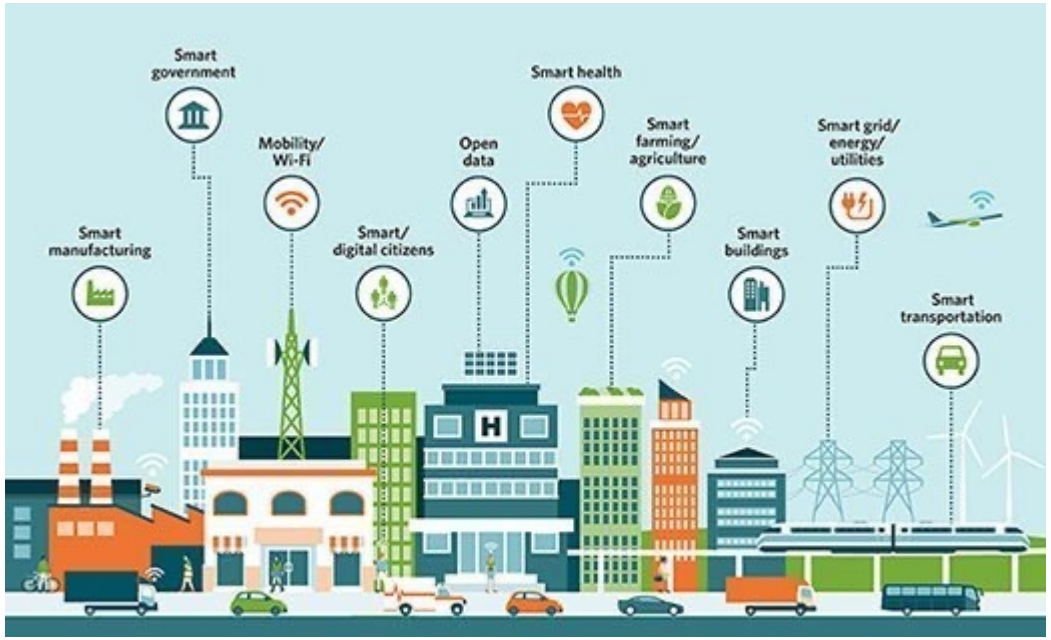
- Which languages do you know? *Java, Kotlin, Javascript, HTML, CSS…*

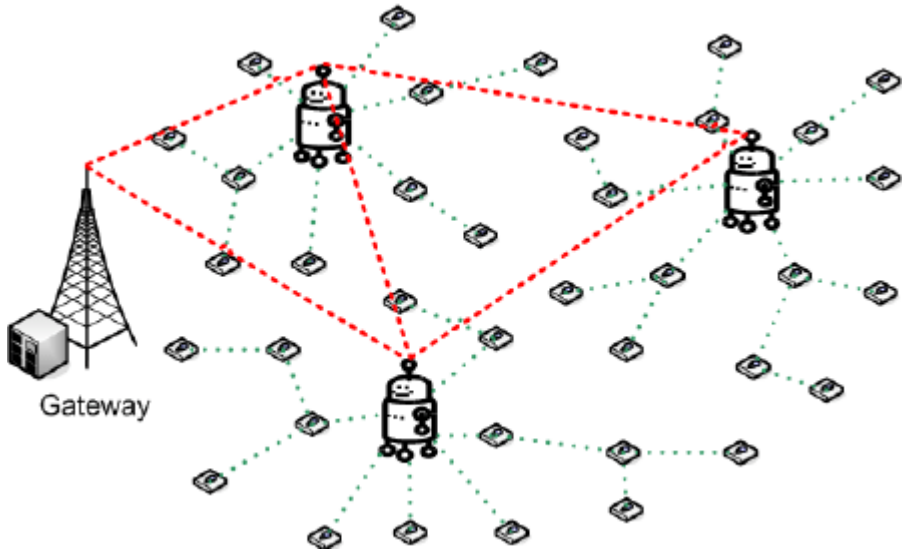
- What would you prefer? *Environmental data* vs *Web data*

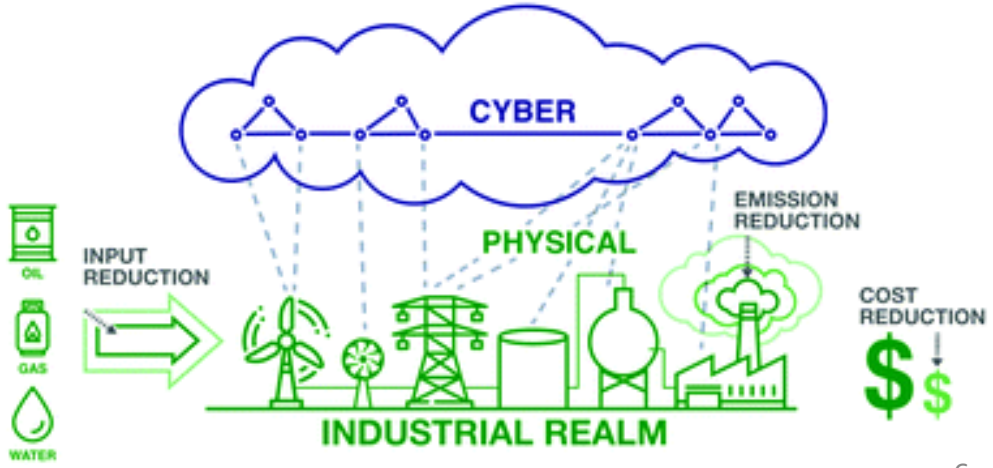


Modern systems





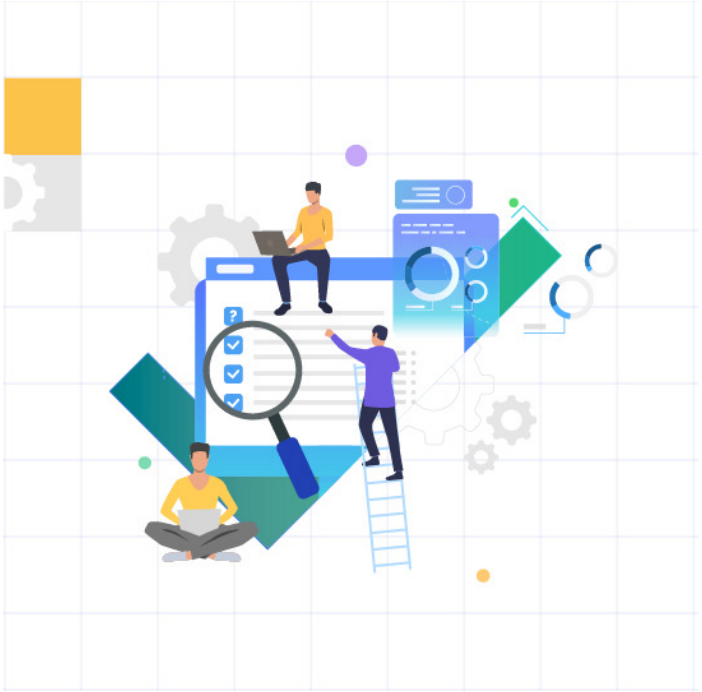




Motivation

There is a growing need to verify:

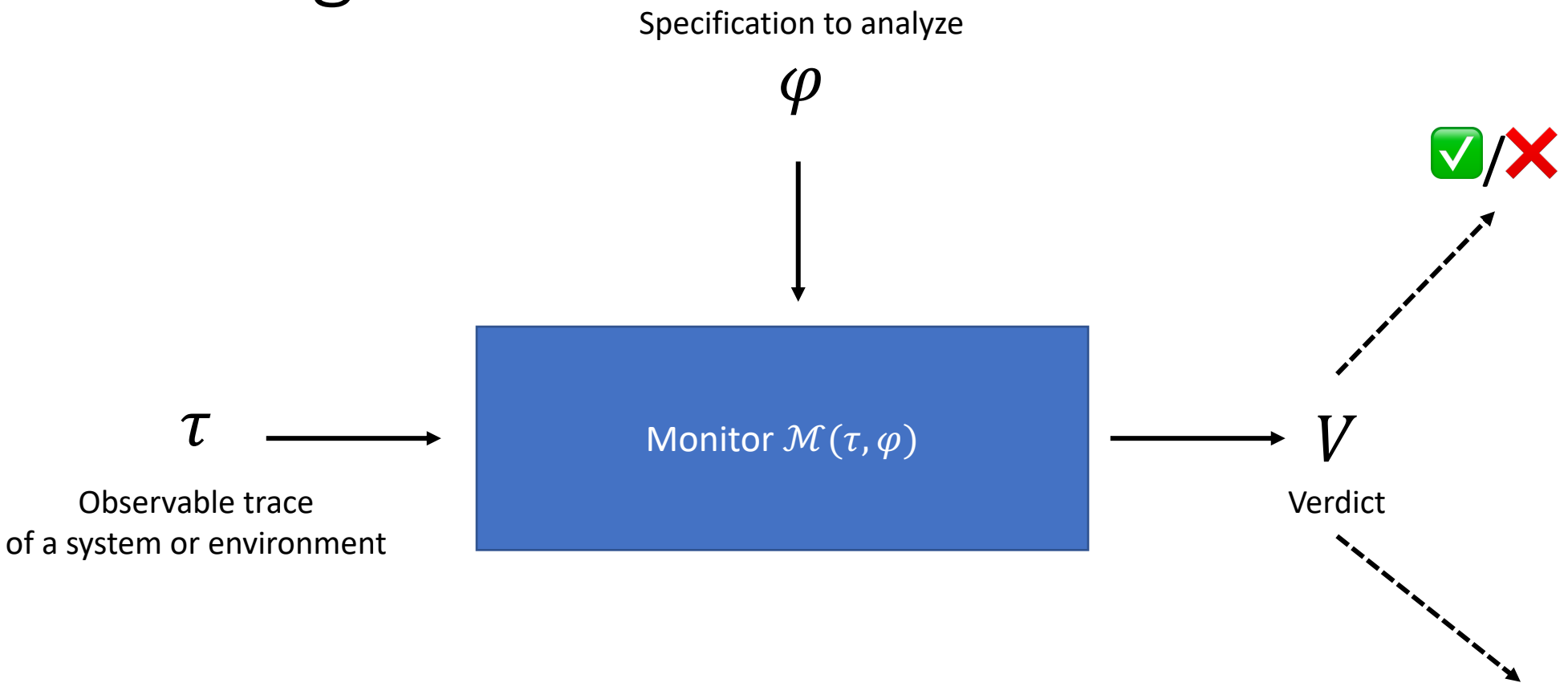
- Compliance to law
- Correct functioning of the system
- Hacking attacks
- Privacy preservation
- …



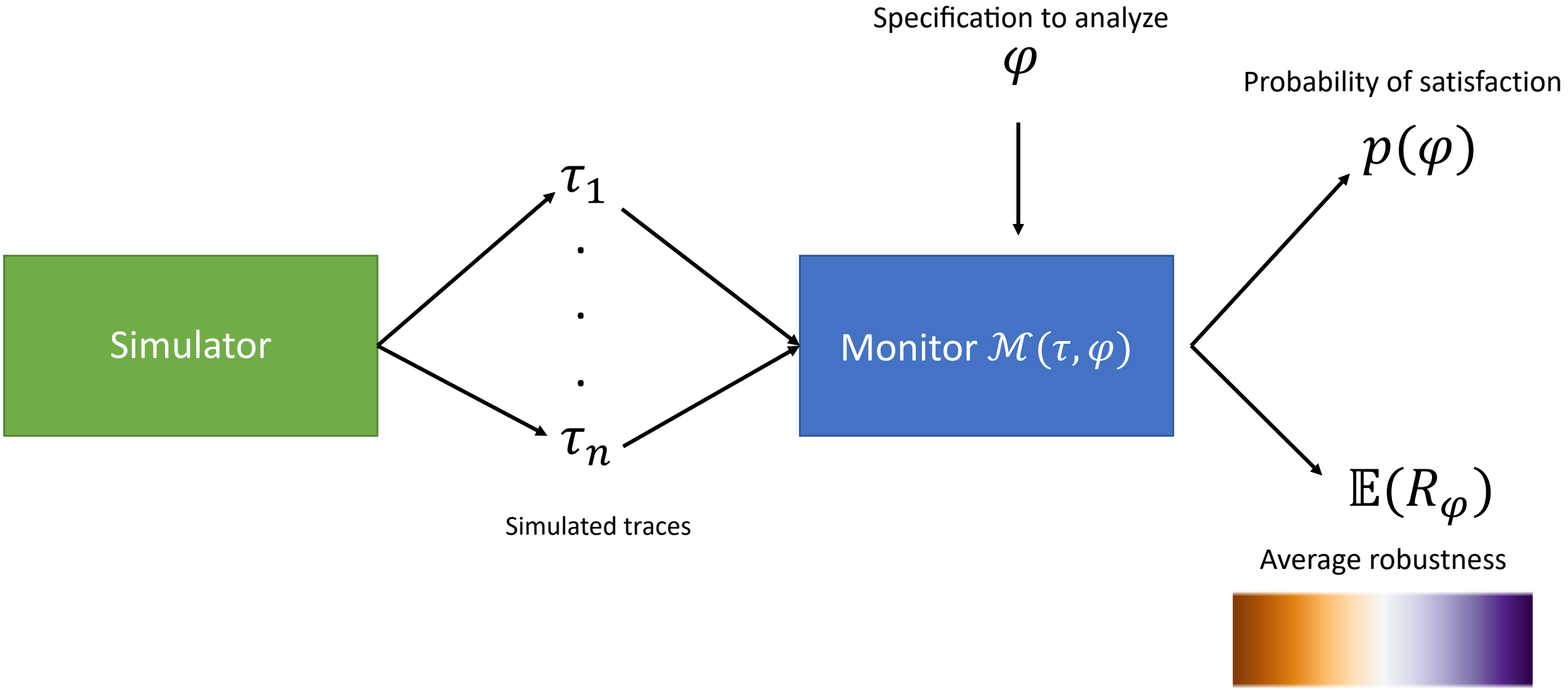
Motivation

- Classical verification techniques like
  - Model checking
  - Theorem proving
  - …
- Would easily suffer from the state-space explosion problem,
  or provide prohibitive computational times

Monitoring


Specification to analyze
φ
τ
Observable trace
of a system or environment
Monitor $\mathcal{M}(\tau, \varphi)$
V
Verdict
✅/❌

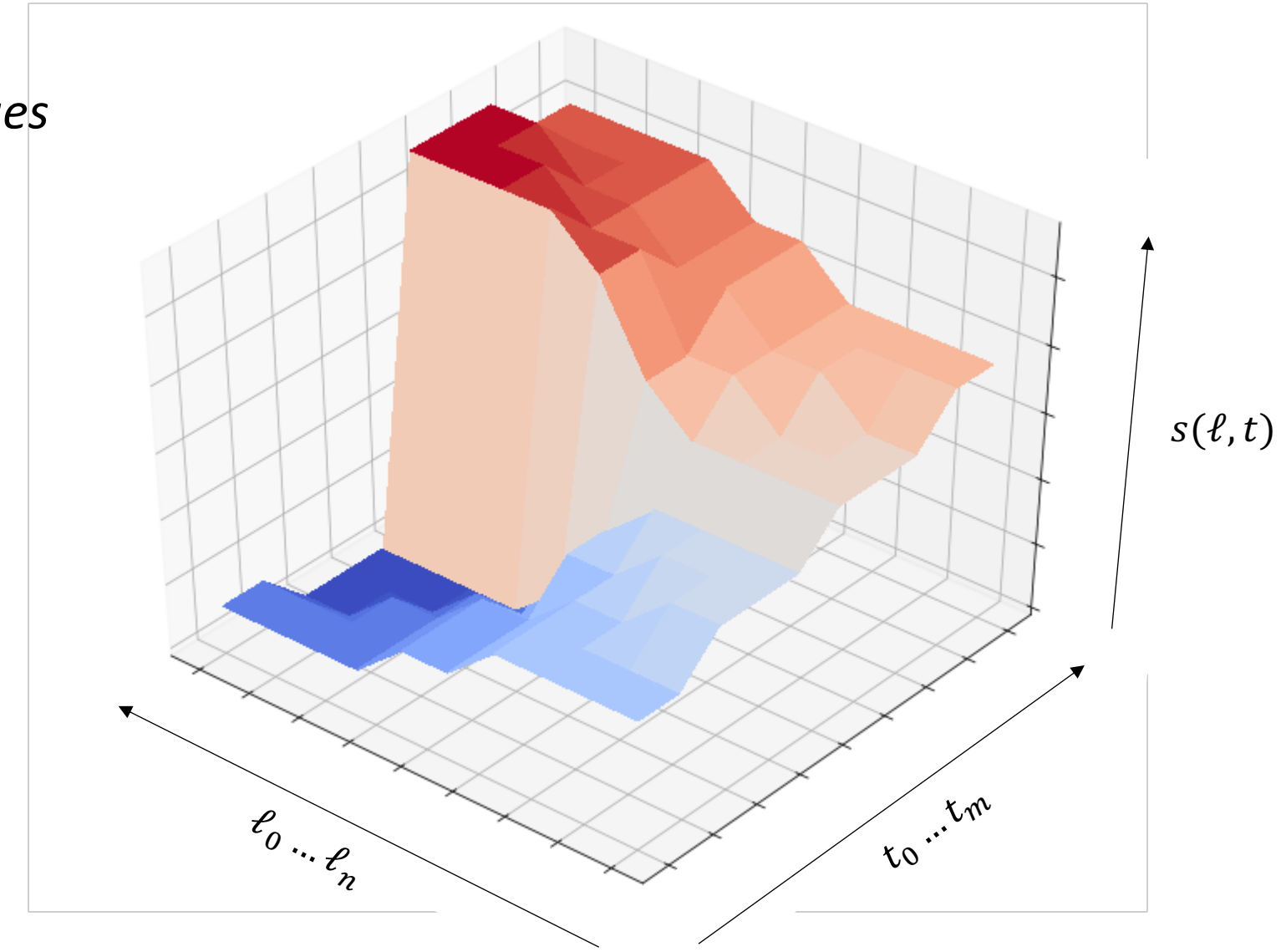
Statistical Model Checking



Signals

Signals defined over real values

$$s: L \times T \rightarrow \mathbb{R}^n$$



Specifying via STREL logic

$$\varphi := \top \mid \bot \mid p \circ c \mid \neg \varphi \mid \varphi \vee \varphi \mid$$
$$\varphi \, \mathrm{U}_I \, \varphi \mid$$
$$\varphi \, \mathcal{R}_{\leq d} \varphi \mid \mathcal{E}_{\geq d} \varphi$$

Signal Temporal Logic

Signal Temporal Reach & Escape Logic

1. Maler O., Nickovic D. - Monitoring Temporal Properties of Continuous Signals. FTRTFT 2004, FORMATS 2004.

2. Bartocci E., Bortolussi L., Loreti M., and Nenzi L. - Monitoring mobile and spatially distributed cyber-physical systems. MEMOCODE '17

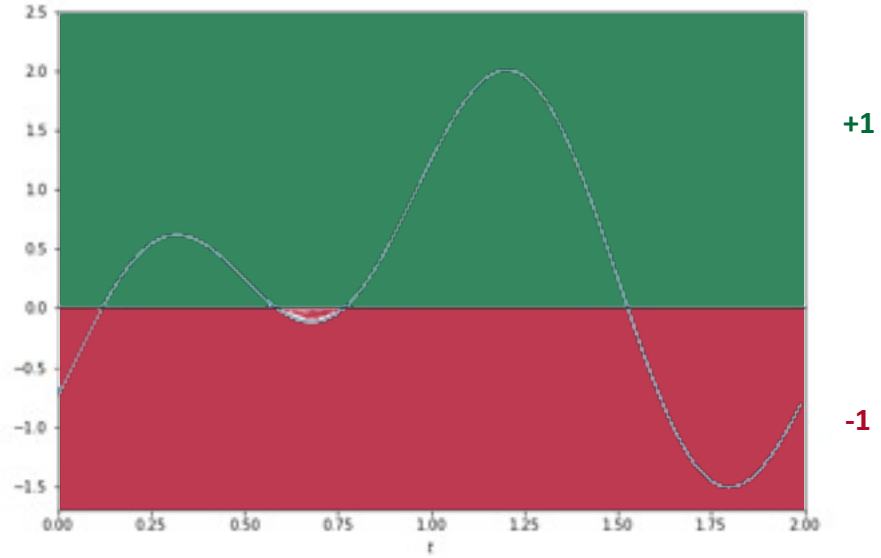
Atomic propositions

Inequalities over signals

$p > C$

χ

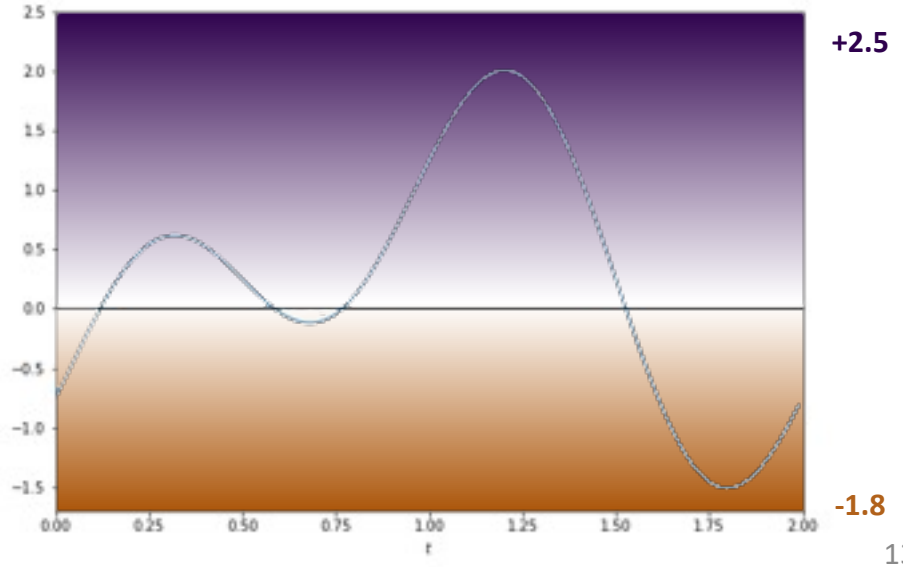
1 | -1



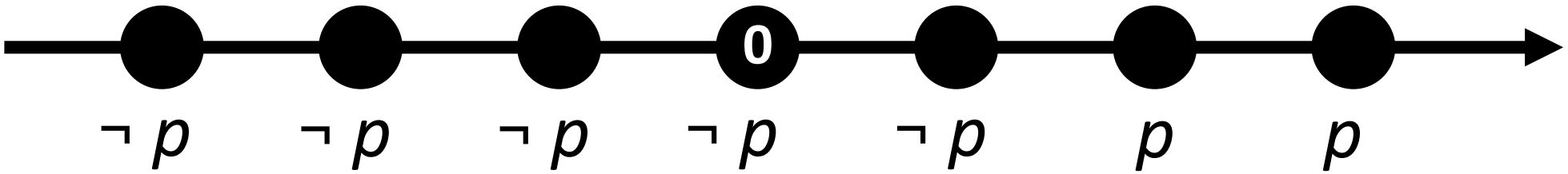
ρ

$x \in \mathbb{R}$

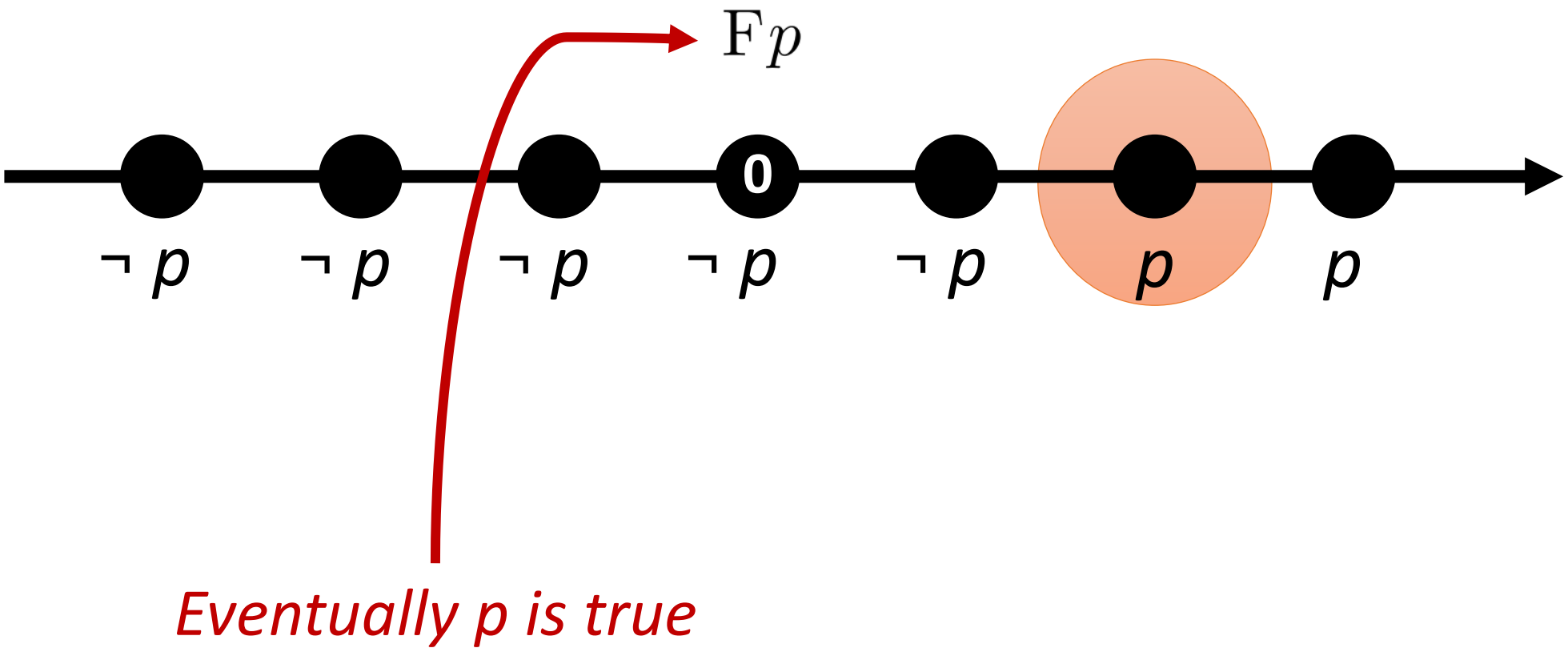
how much above/below



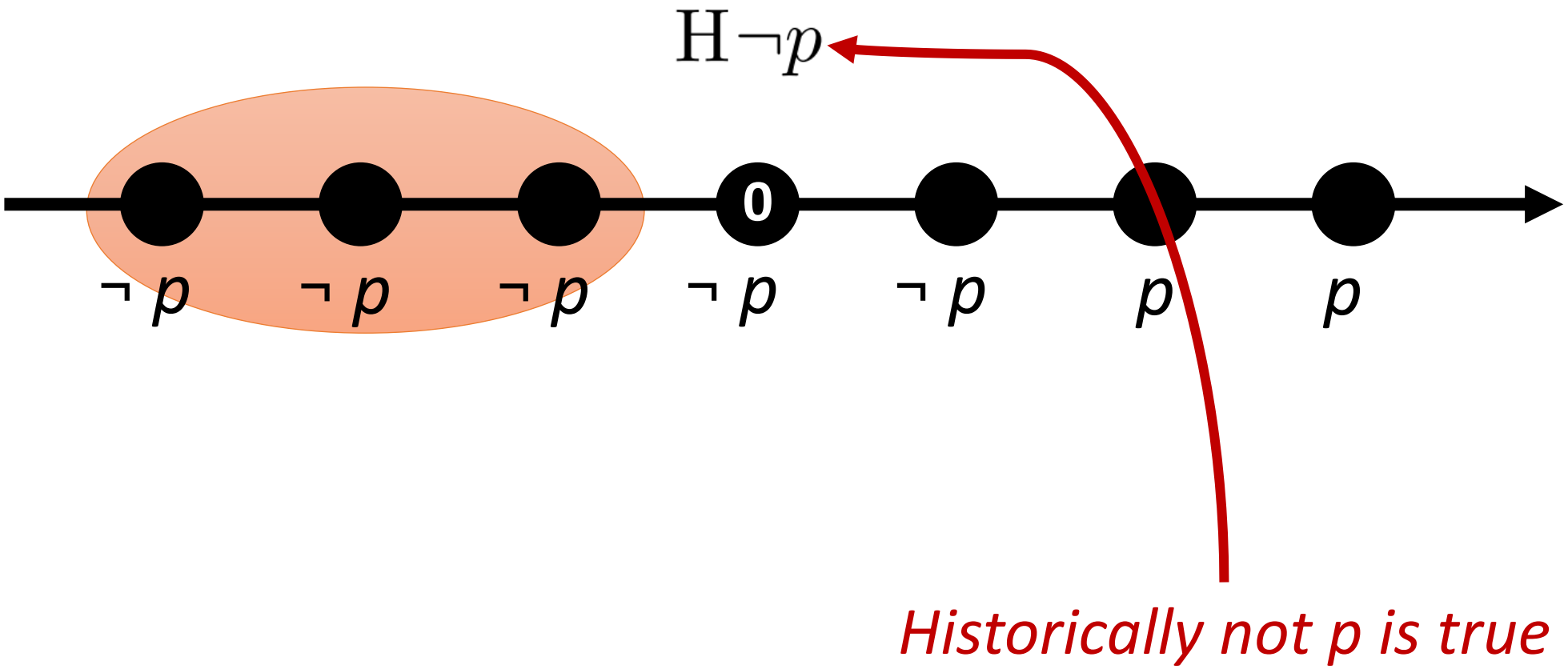
Signal Temporal



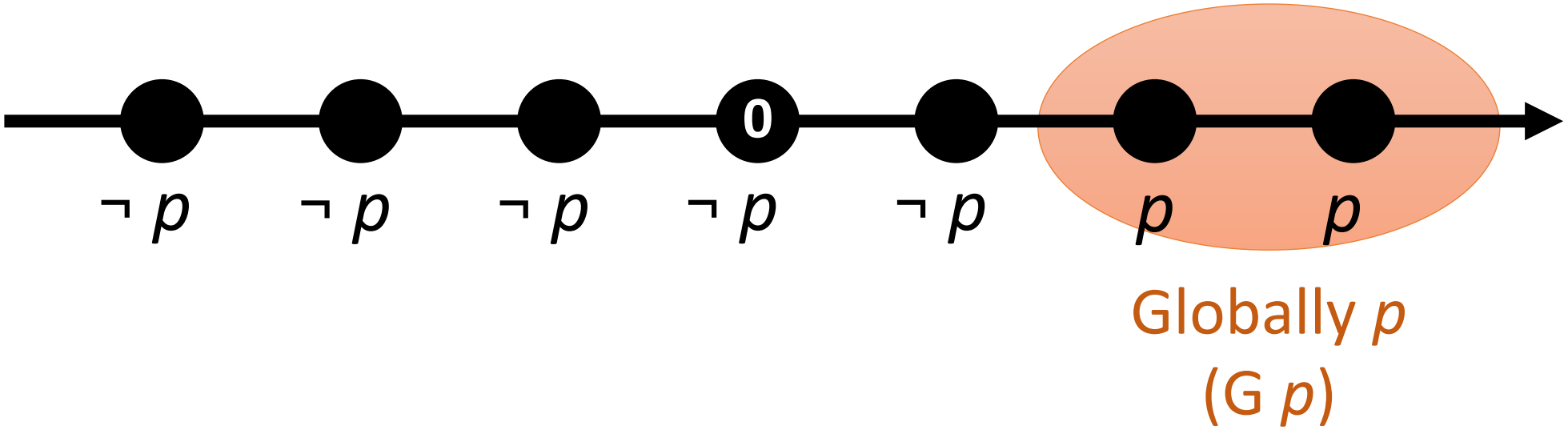
Signal Temporal Logic – Derived operators



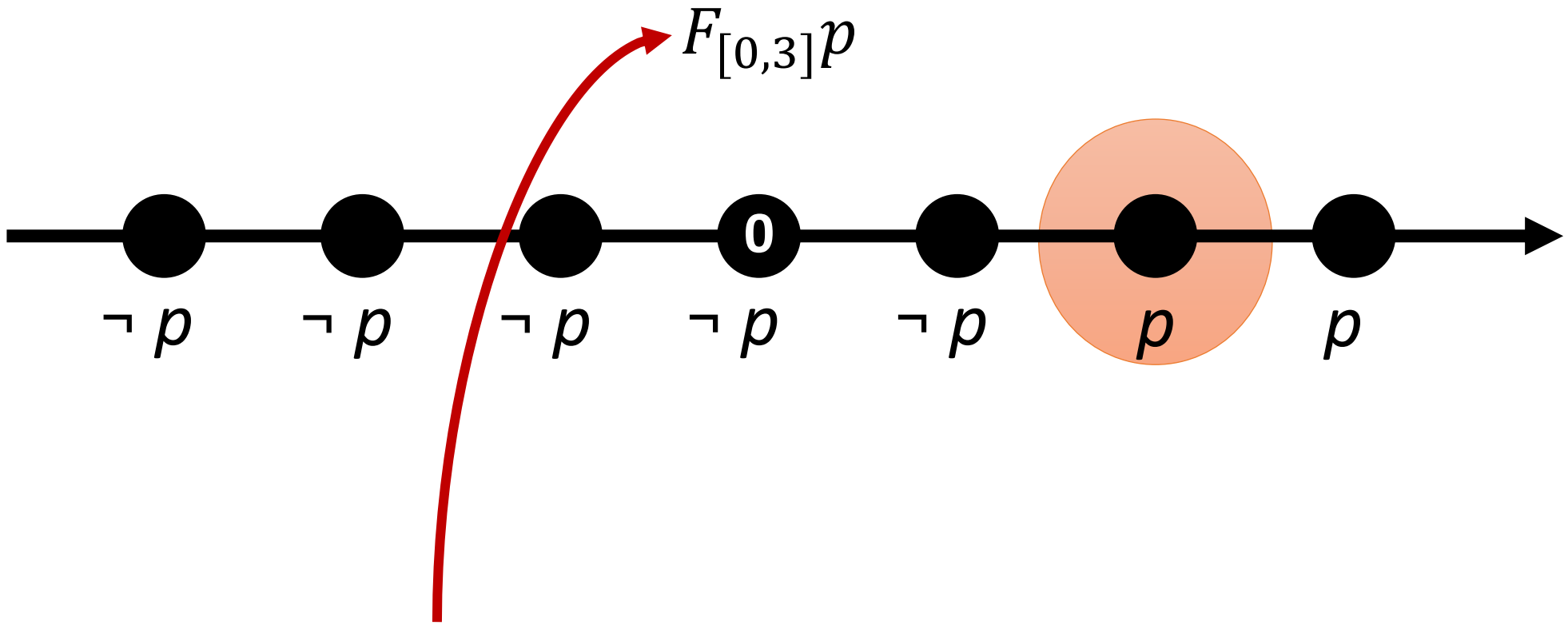
Signal Temporal Logic – Derived operators



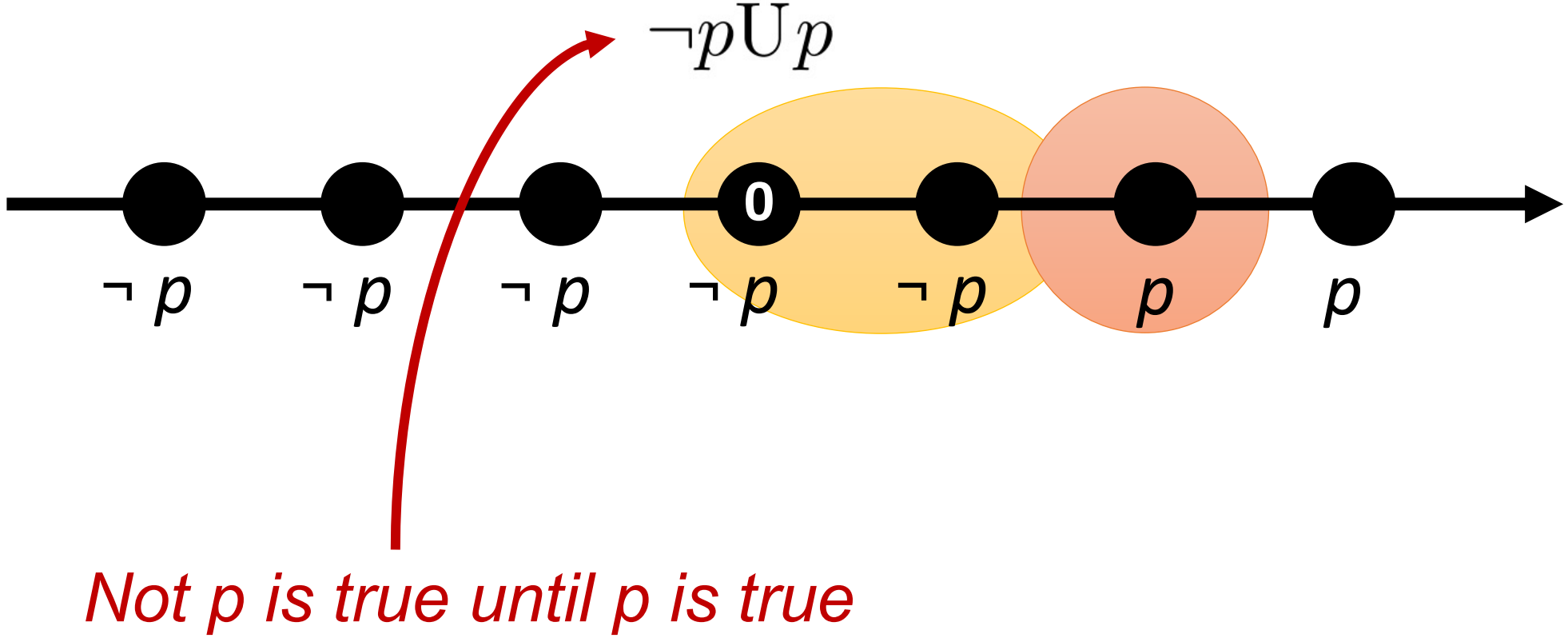
Signal Temporal Logic – Derived operators

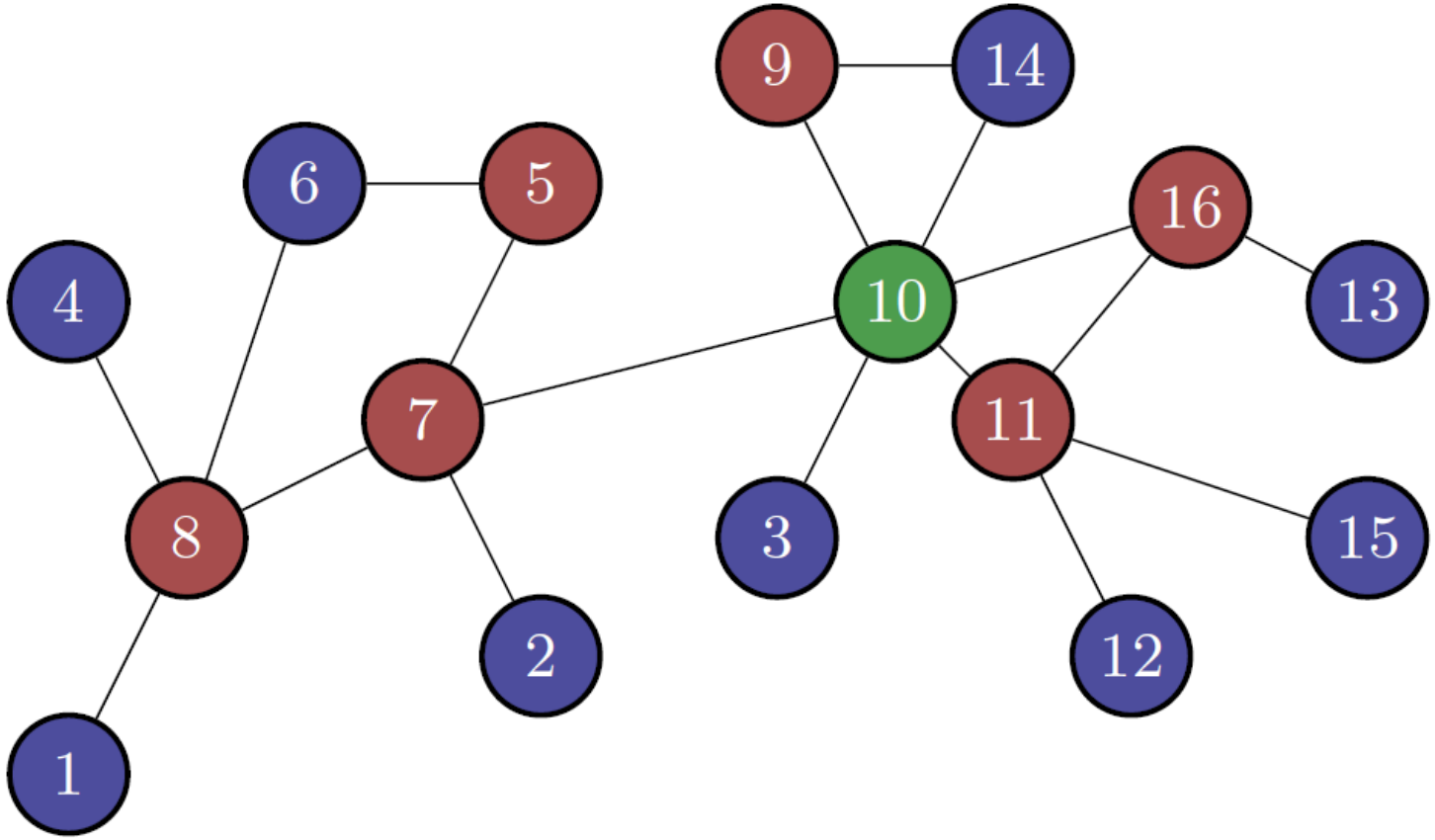


Signal Temporal Logic – Derived operators

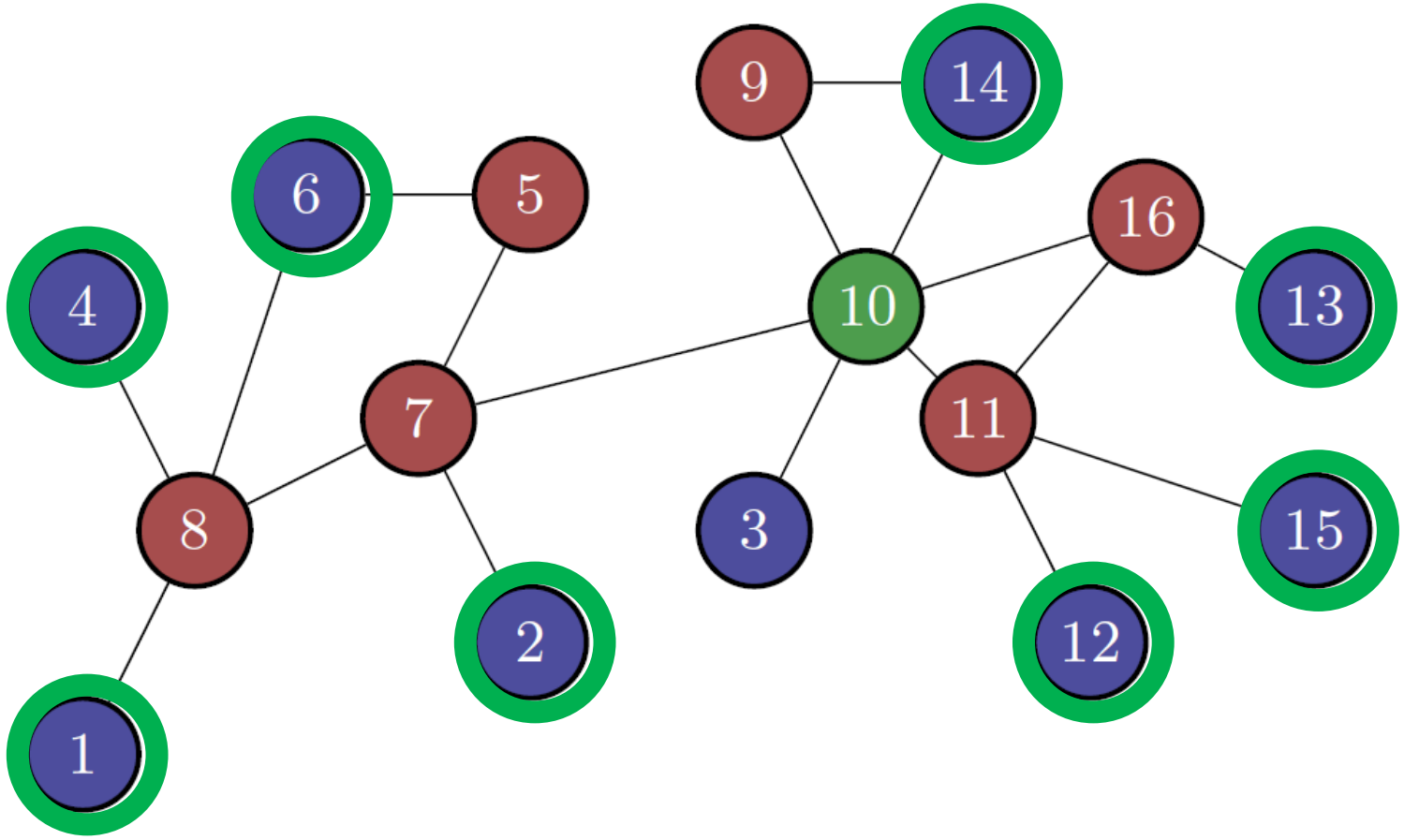


Signal Temporal Logic – Derived operators



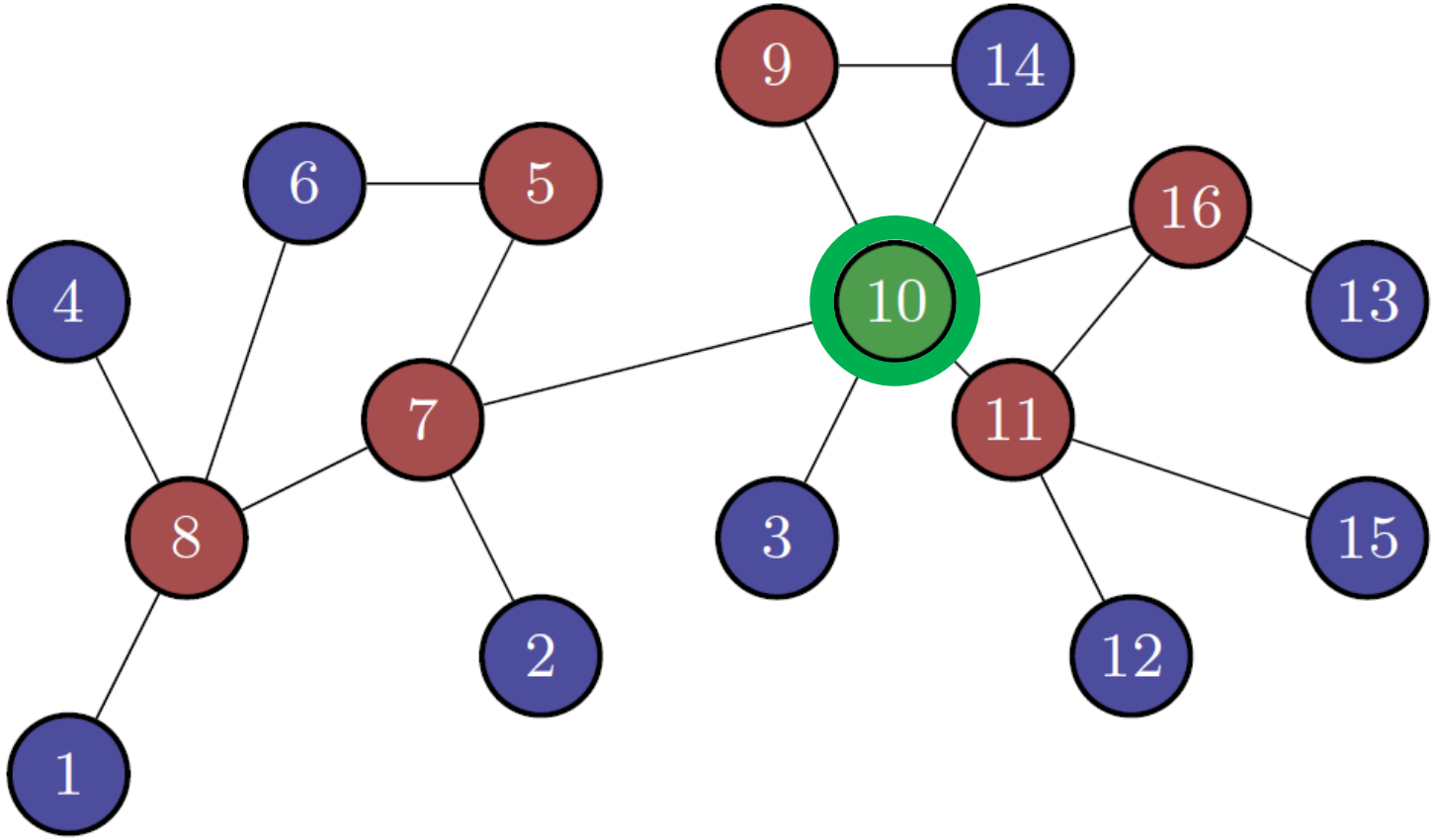


From L. Nenzi, E. Bartocci, L. Bortolussi, and M. Loreti.
A logic for monitoring dynamic networks of spatially-distributed cyber-physical systems, 2021.



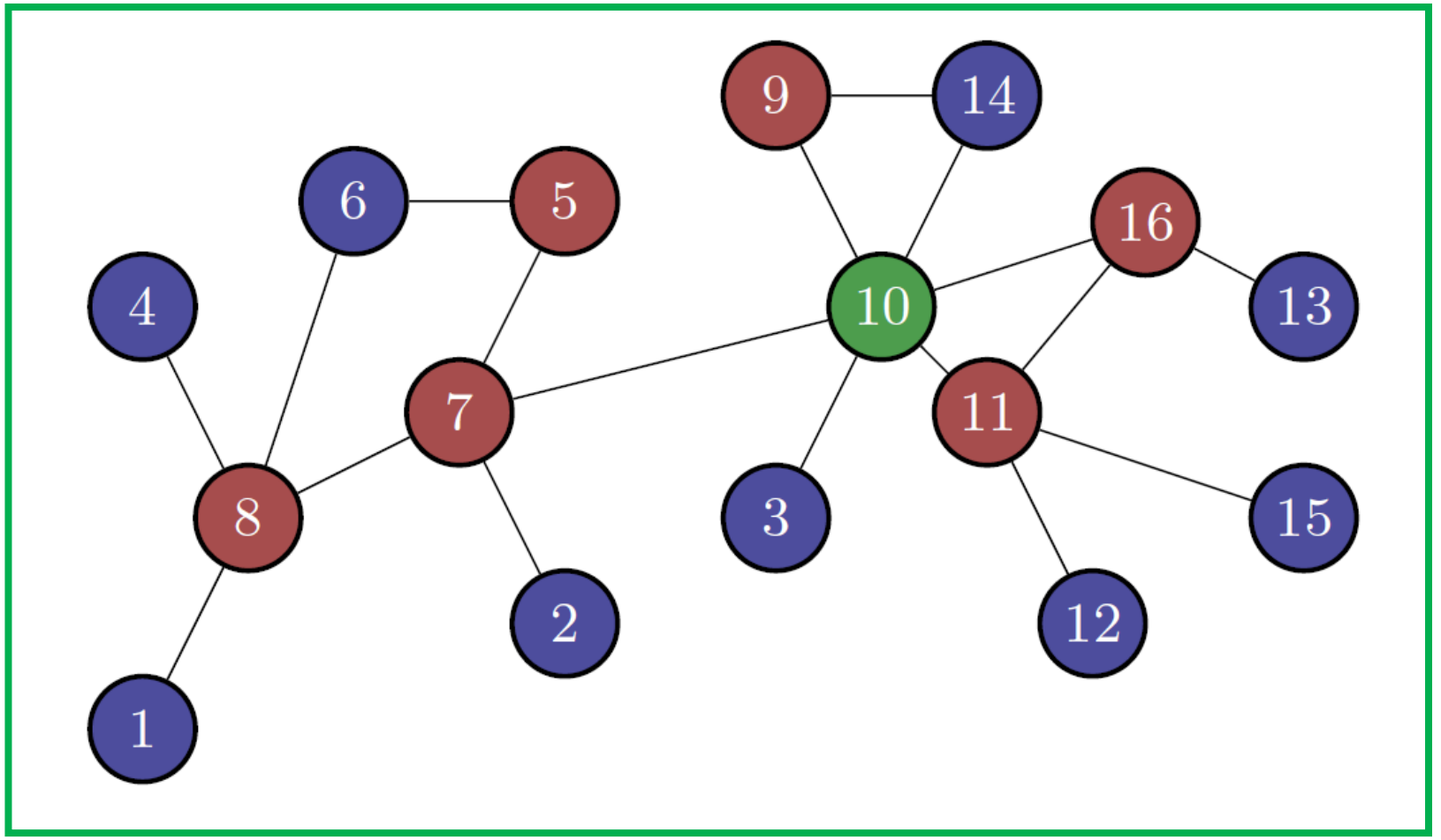
Reachability: *blue* $\mathcal{R}_{\leq 1}$ *red*.

From L. Nenzi, E. Bartocci, L. Bortolussi, and M. Loreti.
A logic for monitoring dynamic networks of spatially-distributed cyber-physical systems, 2021.



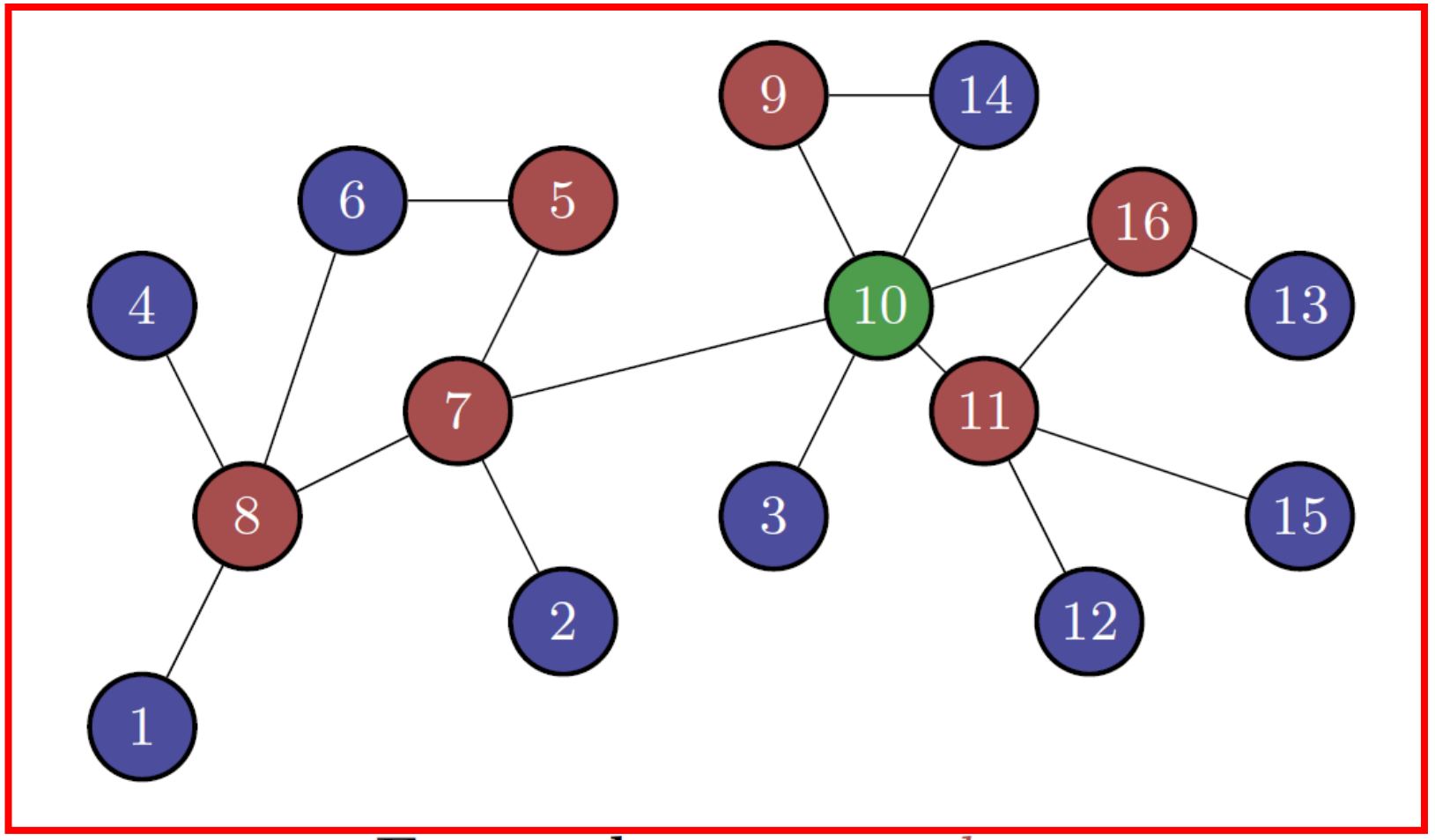
Escape: $\mathcal{E}_{\geq 2}\ \neg blue.$

From L. Nenzi, E. Bartocci, L. Bortolussi, and M. Loreti.
A logic for monitoring dynamic networks of spatially-distributed cyber-physical systems, 2021.



Somewhere: $\Diamond_{\leq 4} green.$

From L. Nenzi, E. Bartocci, L. Bortolussi, and M. Loreti.
A logic for monitoring dynamic networks of spatially-distributed cyber-physical systems, 2021.



Everywhere: $\boxdot_{\geq 2} red$.

From L. Nenzi, E. Bartocci, L. Bortolussi, and M. Loreti.
A logic for monitoring dynamic networks of spatially-distributed cyber-physical systems, 2021.

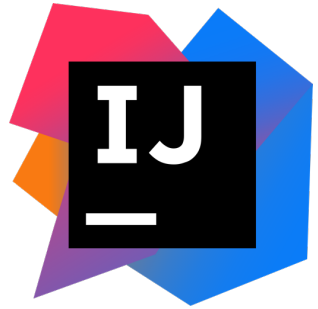
Tools

Python 3.8+

JDK 21+

Visual Studio Code

PyCharm



IntelliJ

IDEs

Extras

Windows 10+ only:

```
winget install --id Git.Git -e --source winget
```

Python poetry

```
pip install poetry
```

Steps

1. Signal definition

```
time = list(np.arange(0,10,0.05))
f1,f2 = np.sin(time),np.cos(time)
signals = list(zip(f1,f2))
```

2. Scripting

```
script = """
signal { real x; real y;}
domain boolean;
formula future = globally [0, 0.2]  (x > y);
formula past = historically [0, 0.2]  (x > y);
"""
moonlightScript = ScriptLoader.loadFromText(script)
```

Steps

3. Monitor definition

```
boolFutureMonitor = moonlightScript.getMonitor("future");
```

4. Run the monitors

```
boolFutureMonitorResult = np.array(futureMonitor.monitor(time,signals))
```

Demo time

The End