COURSE 2.8.1

Non-Sequential Theory of Distributed Systems

Lecturers

* Benedikt Bollig (responsible)
* Paul Gastin

Dates

The lectures will take place on Wednesday, 13:15-15:45, in Bat. Sophie Germain, **room 1004**.

The first lecture is on Wednesday, 12th September 2018.

Outline of the course

1st Lecture	12.09.2018	motivating examples, automata models of concurrent processes with data structures
2nd Lecture		operational semantics; graph semantics; MSO logic
3rd Lecture		expressive power of MSO logic; underapproximate verification
4th Lecture		special tree-width, decomposition game
5th Lecture		decision procedures via tree automata
6th Lecture		propositional dynamic logic (PDL)
7th Lecture		translation of PDL formulas into CPDS
8th Lecture		asynchronous automata
9th Lecture		revision and exercises
Exam		

Lecture Notes

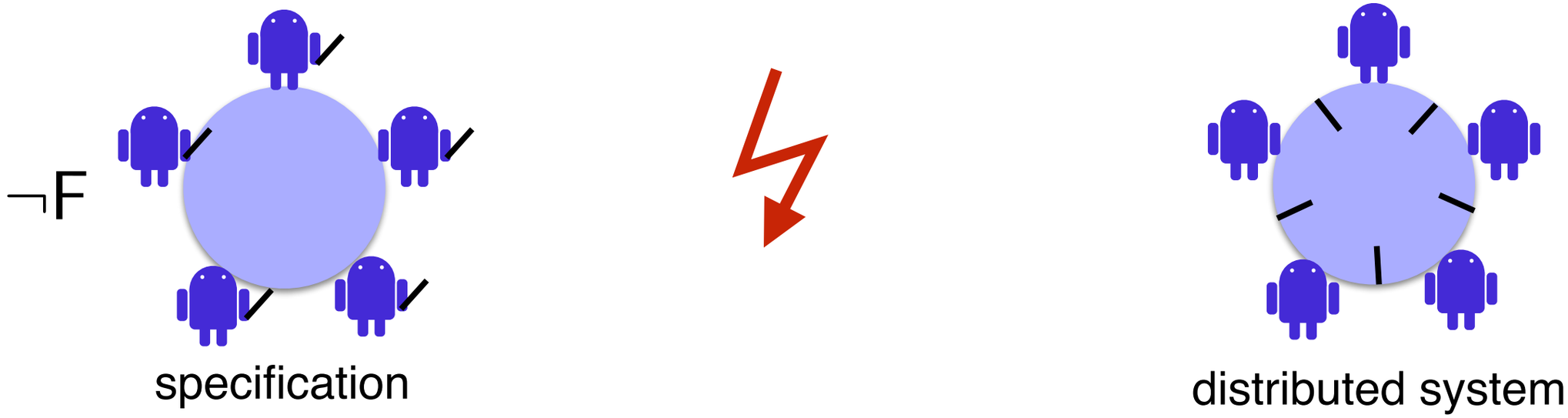
non-sequential.pdf (version as of 31st October 2017, covering lectures 1-7)

Prerequisites

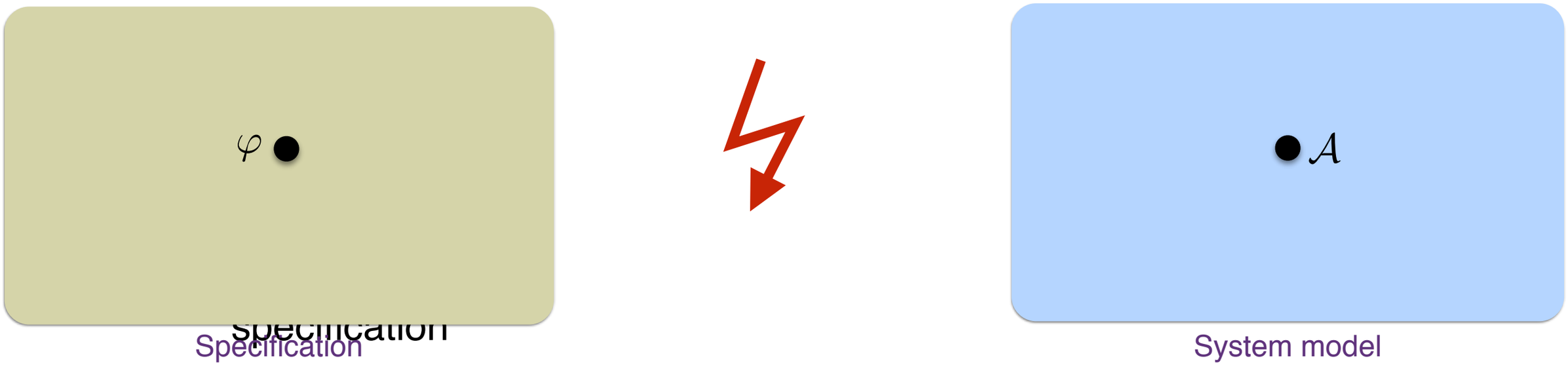
Basic knowledge in Automata, Logics, Complexity.

While not mandatory, it is useful to know the basics of verification. See for instance the level 1 course 1-22 Basics of verification which can also be taken during M2.

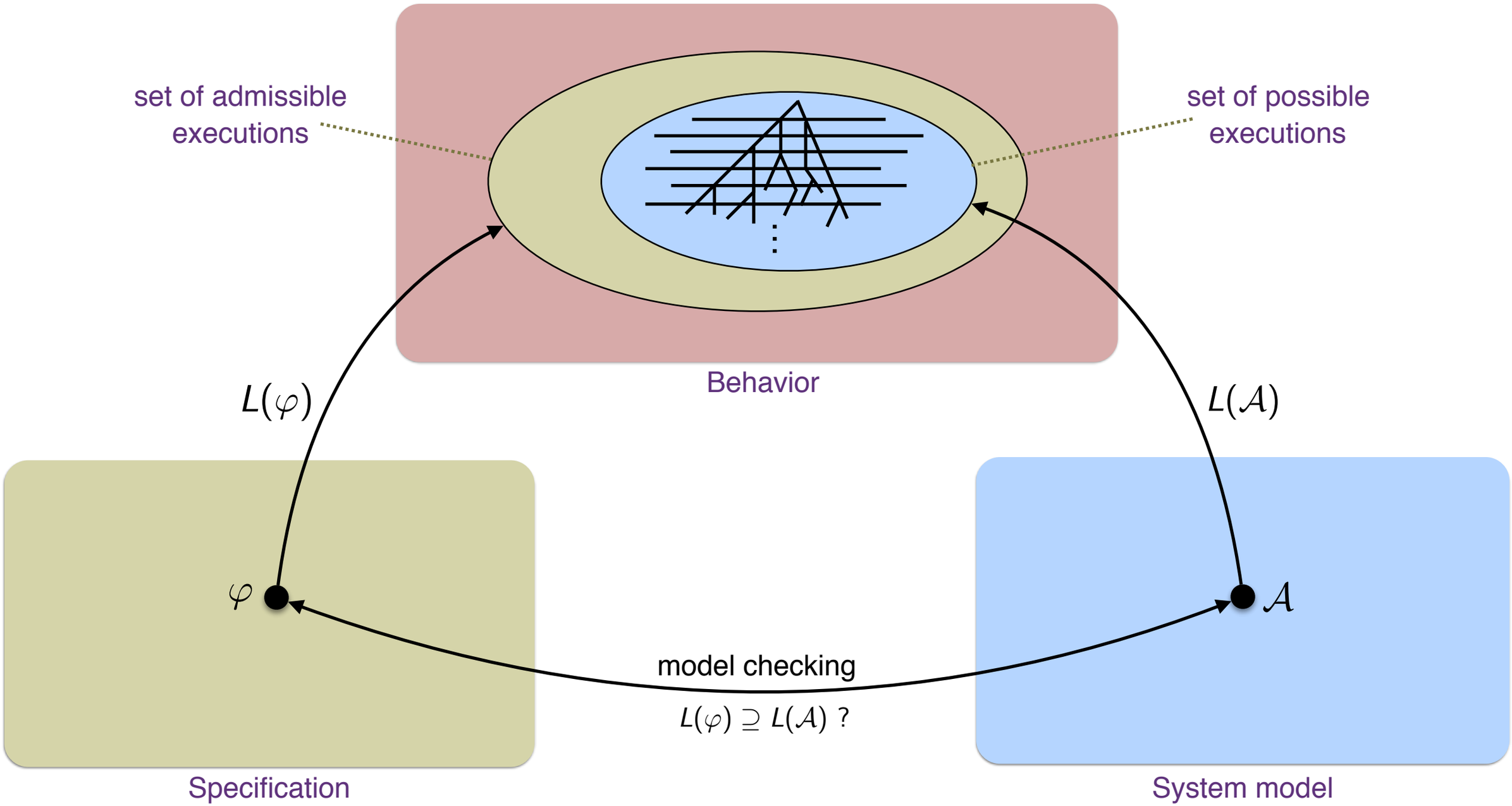
Landscape & Objectives



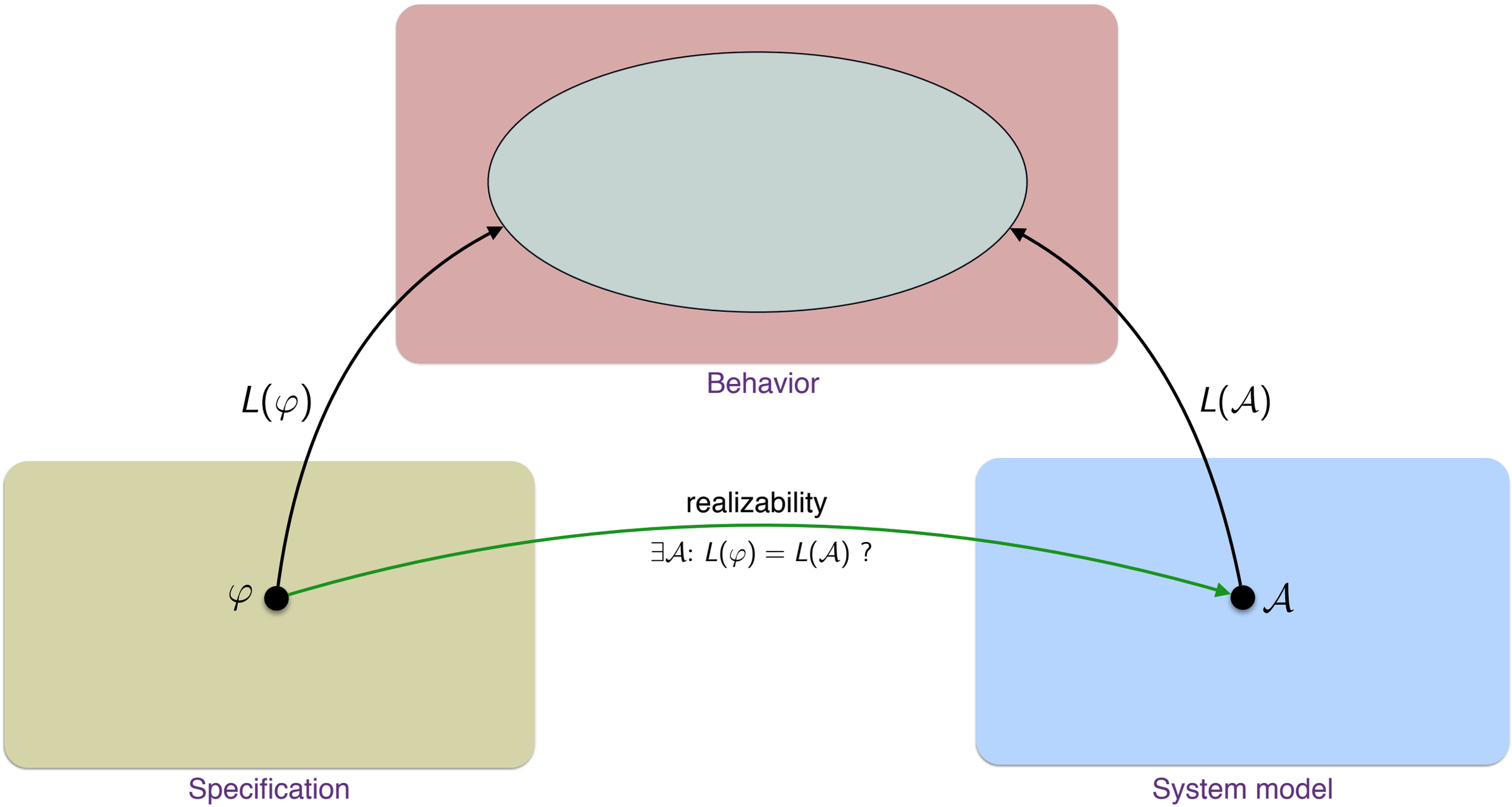
Landscape & Objectives



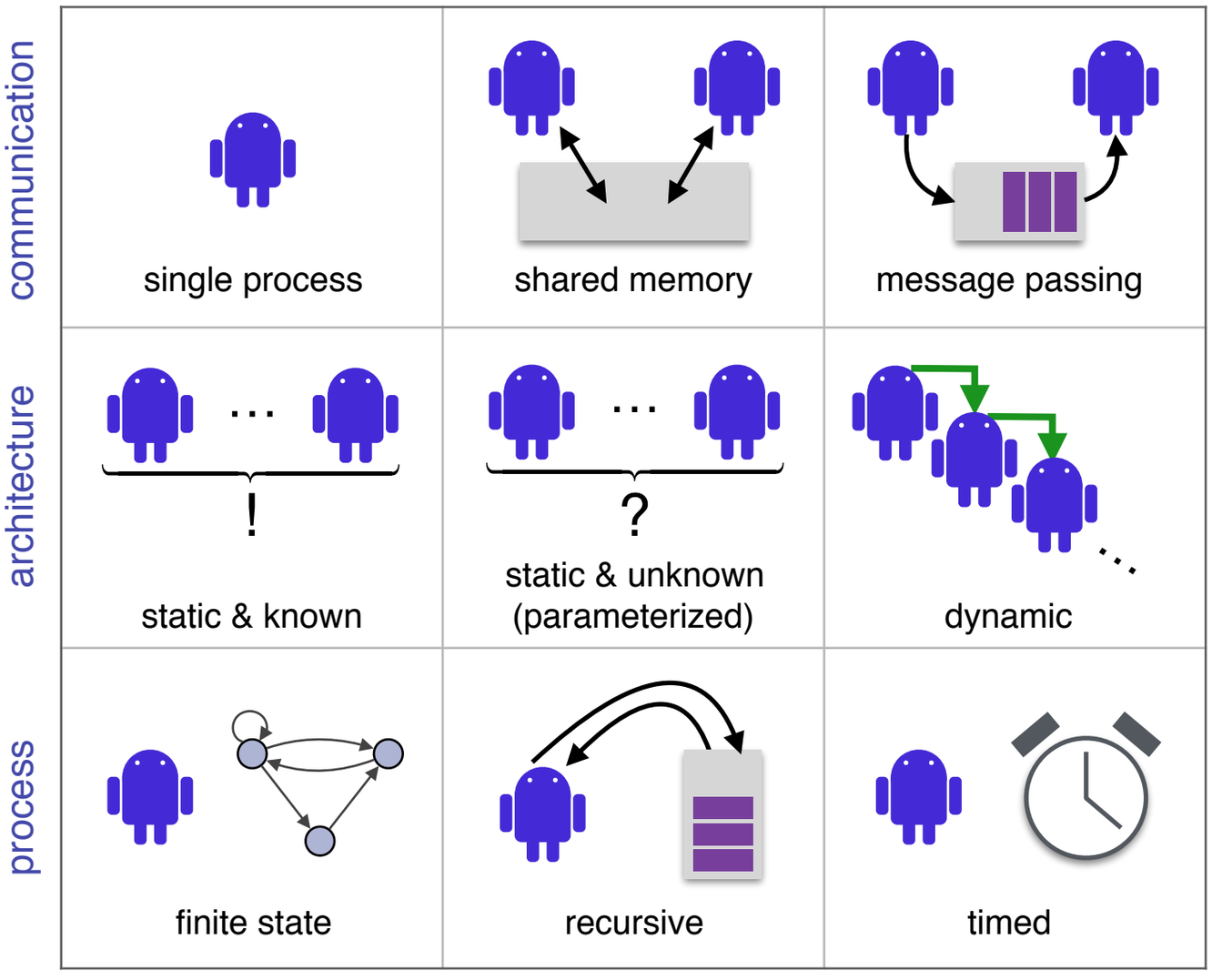
Landscape & Objectives



Landscape & Objectives



Various settings



Finite automata

communication	single process	shared memory	message passing
architecture	static & known	static & unknown (parameterized)	dynamic
process	finite state	recursive	timed

Behavior

- Words

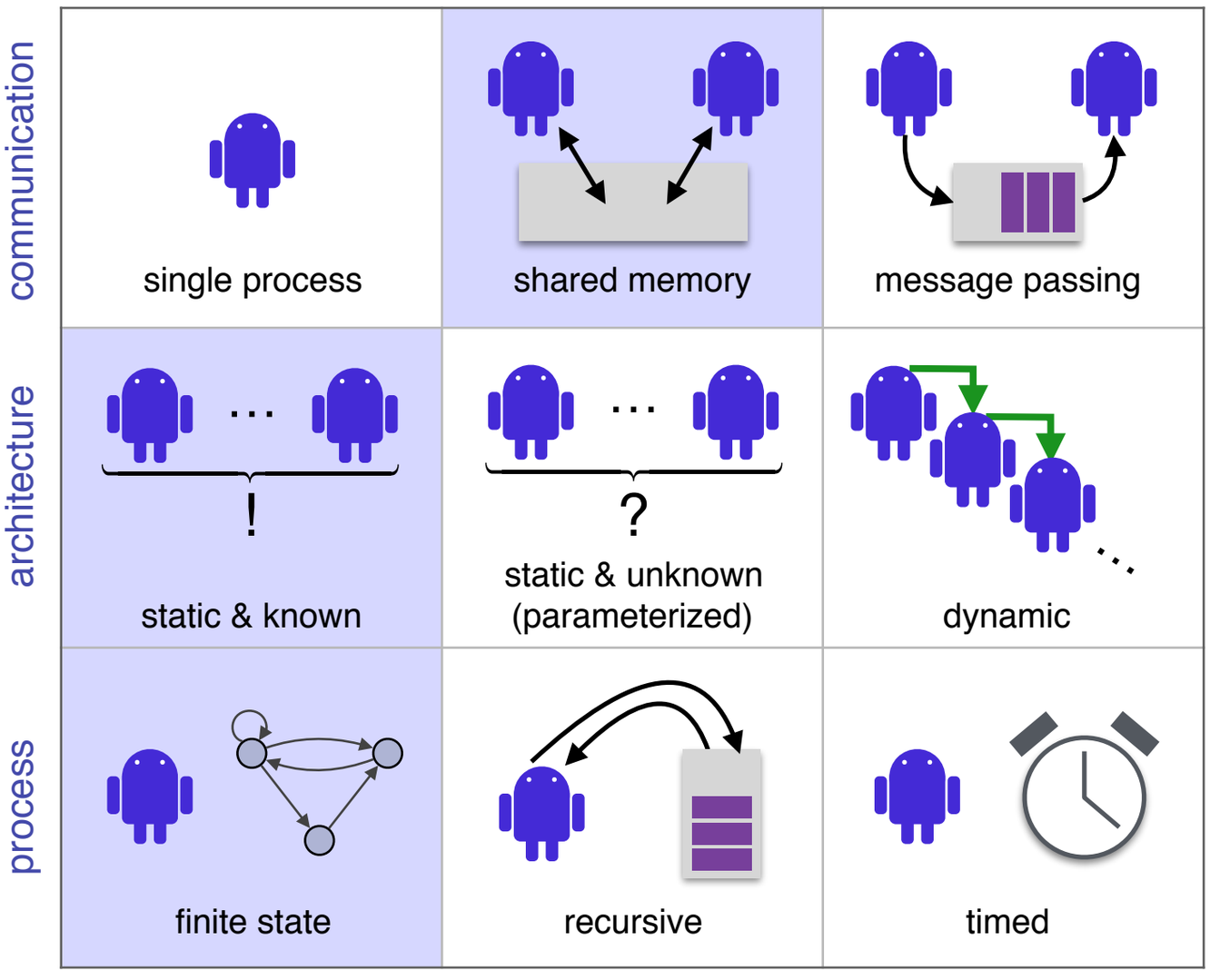
System model

- Finite automata

Specification

- Linear-time temporal logic (LTL)
- Monadic second-order logic (MSO)

Asynchronous automata



Behavior

- Mazurkiewicz traces
  [Mazurkiewicz '70s]

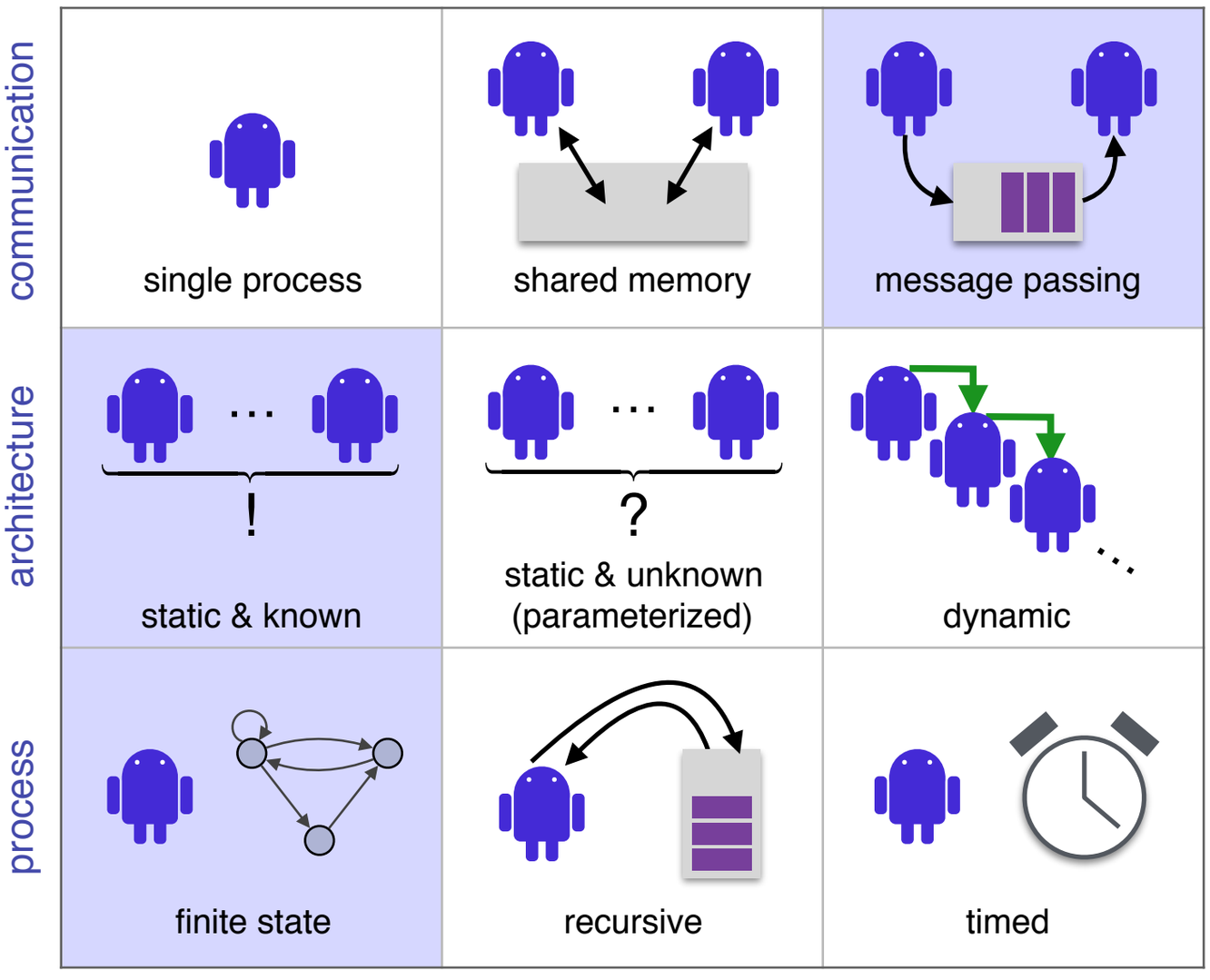
System model

- Asynchronous automata
  [Zielonka '87]

Specification

- Linear-time temporal logic (LTL)
- Monadic second-order logic (MSO)

Communicating automata



Behavior

- Message sequence charts

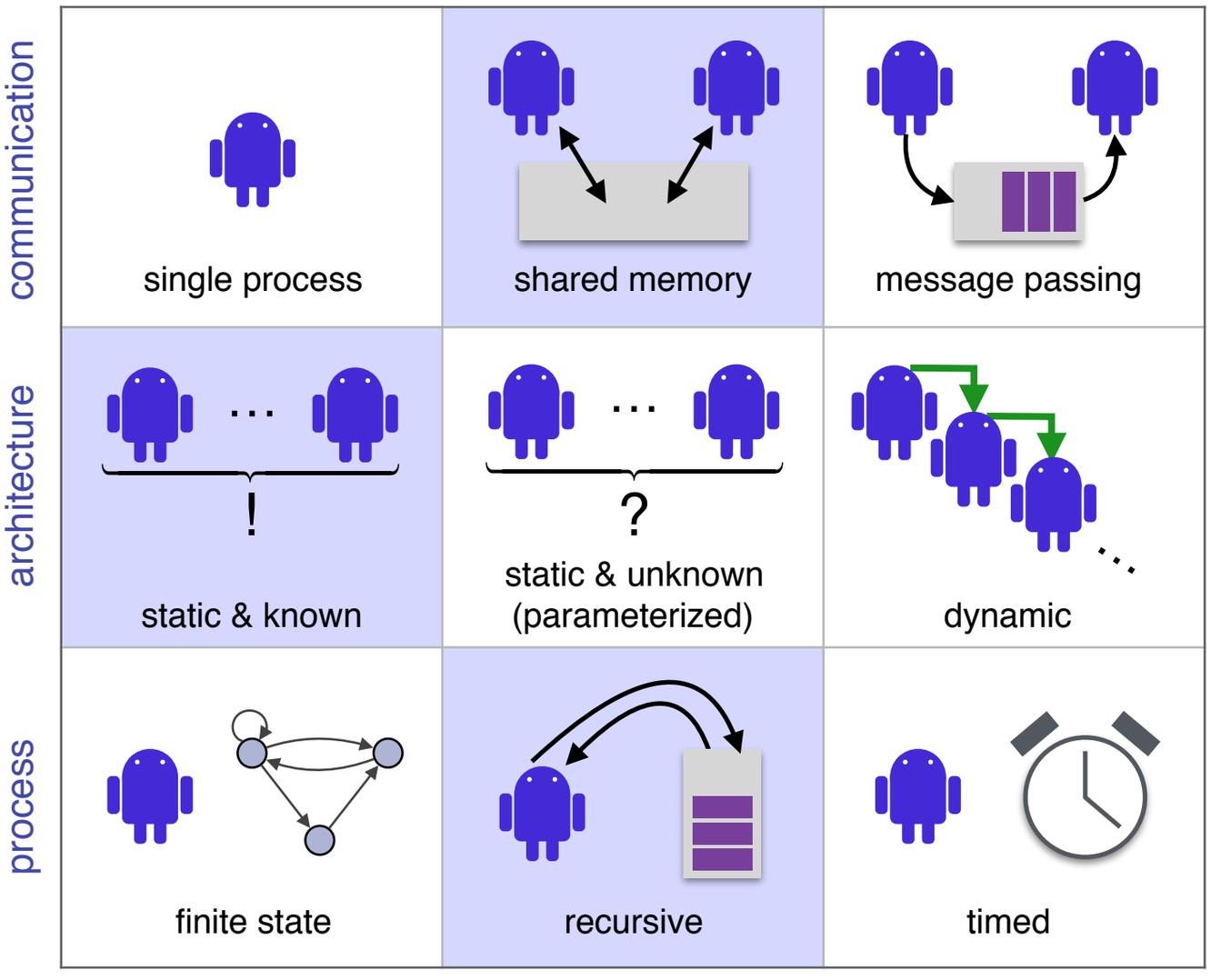
System_model

- Communicating automata
  [Brand-Zafiropulo '83]

Specification

- Linear-time temporal logic (LTL)
- Monadic second-order logic (MSO)

Multi-pushdown automata



Behavior

- (Muliply) Nested words

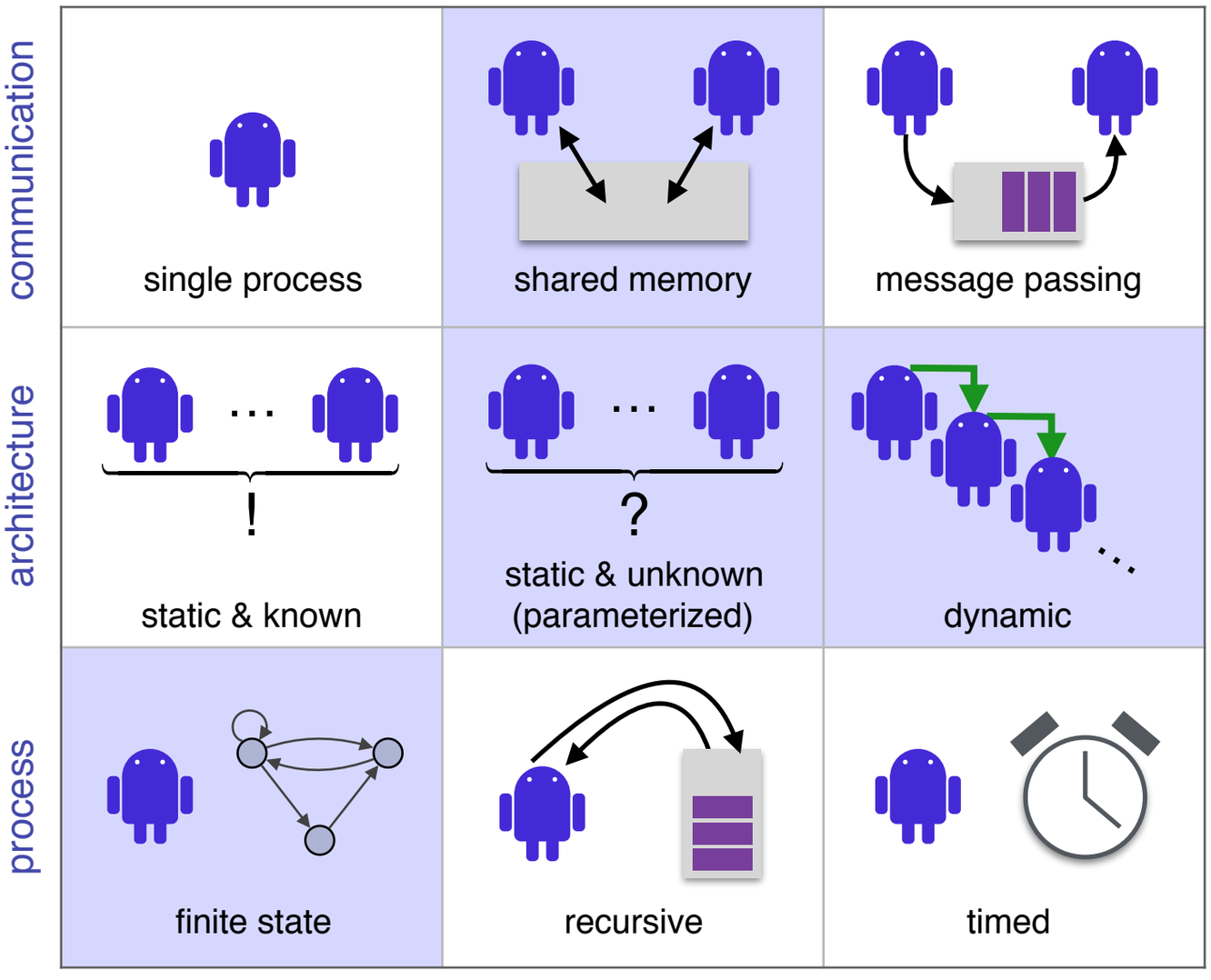
System_model

- Visibly multi-pushdown automata
  [Alur-Madhusudan '04]
  [La Torre-Madhusudan-Parlato '07]

Specification

- Linear-time temporal logic (LTL)
- Monadic second-order logic (MSO)

Data automata



Behavior

- Data words (over infinite alphabet)

System_model

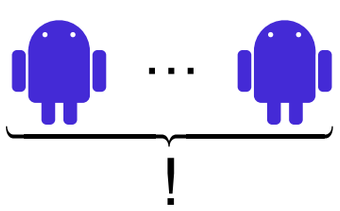
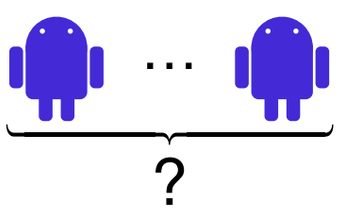
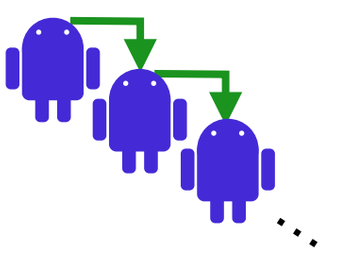
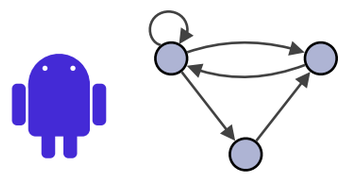
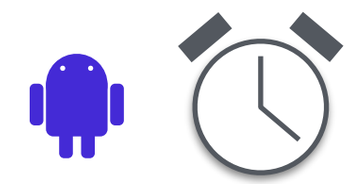
- Data/Class-memory automata
  [Bojanczyk et al. ’06]
  [Björklund-Schwentick’07]

Specification

- Linear-time temporal logic (LTL)
- Monadic second-order logic (MSO)

Many different models …

Graph automata

communication / architecture	single process	shared memory	message passing
	 static & known	 static & unknown (parameterized)	 dynamic
process	 finite state	 recursive	 timed

Behavior

- Graphs/Partial orders

System model

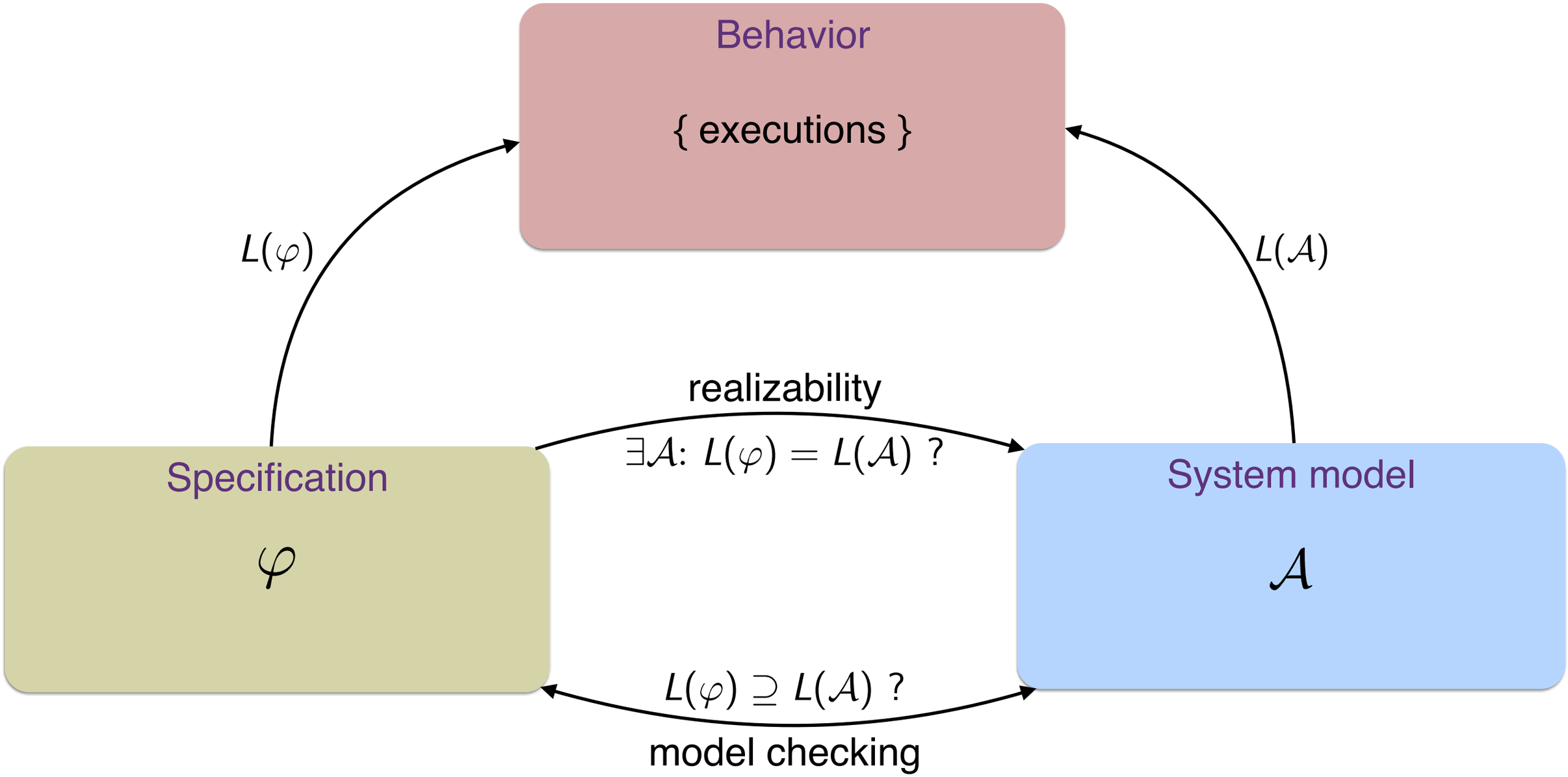
- Graph automata

Specification

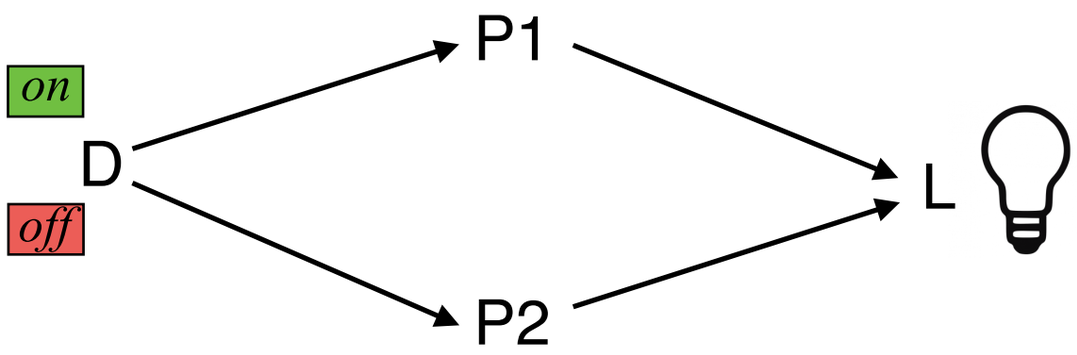
- Linear-time temporal logic (LTL)
- Monadic second-order logic (MSO)

… in a uniform framework.

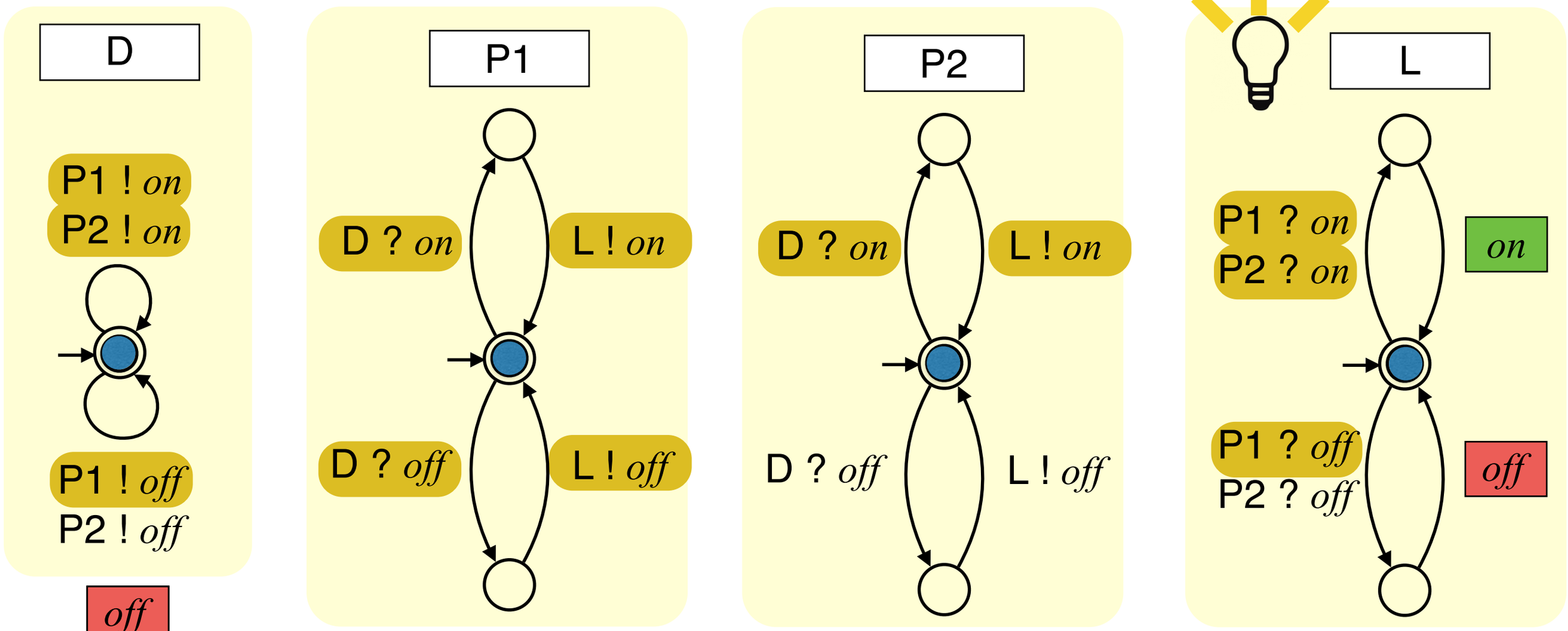
Landscape & Objectives

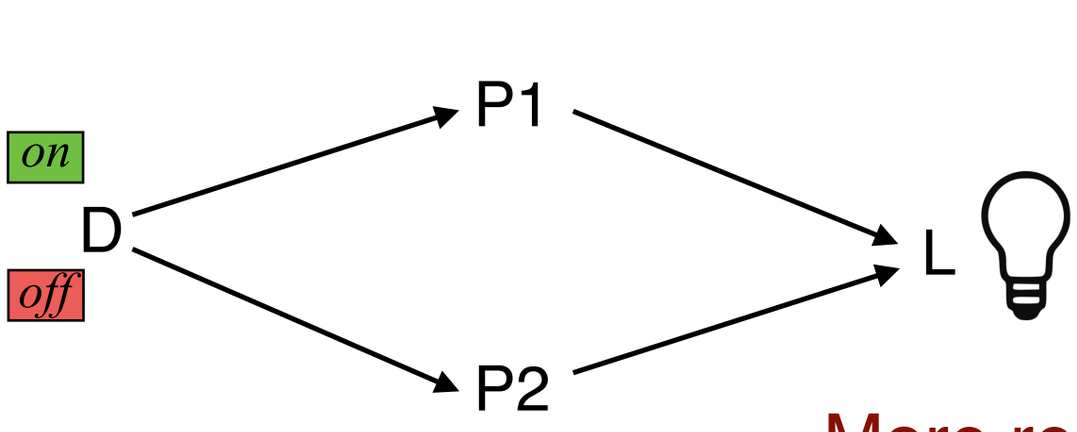


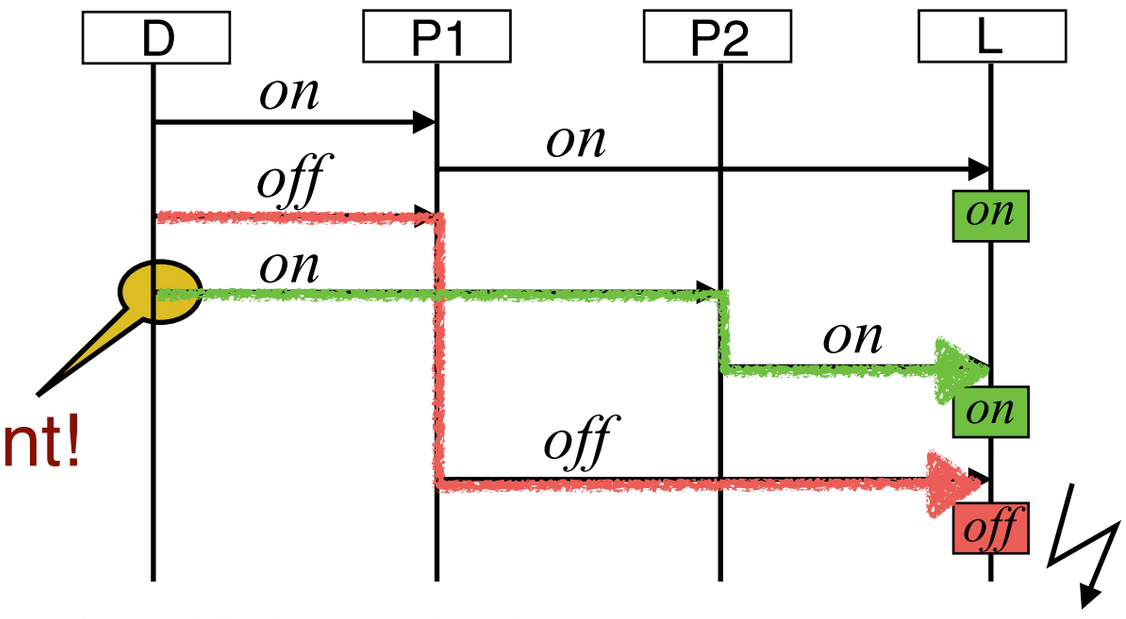
Motivating examples



synchronous communication
(rendez-vous, handshake)

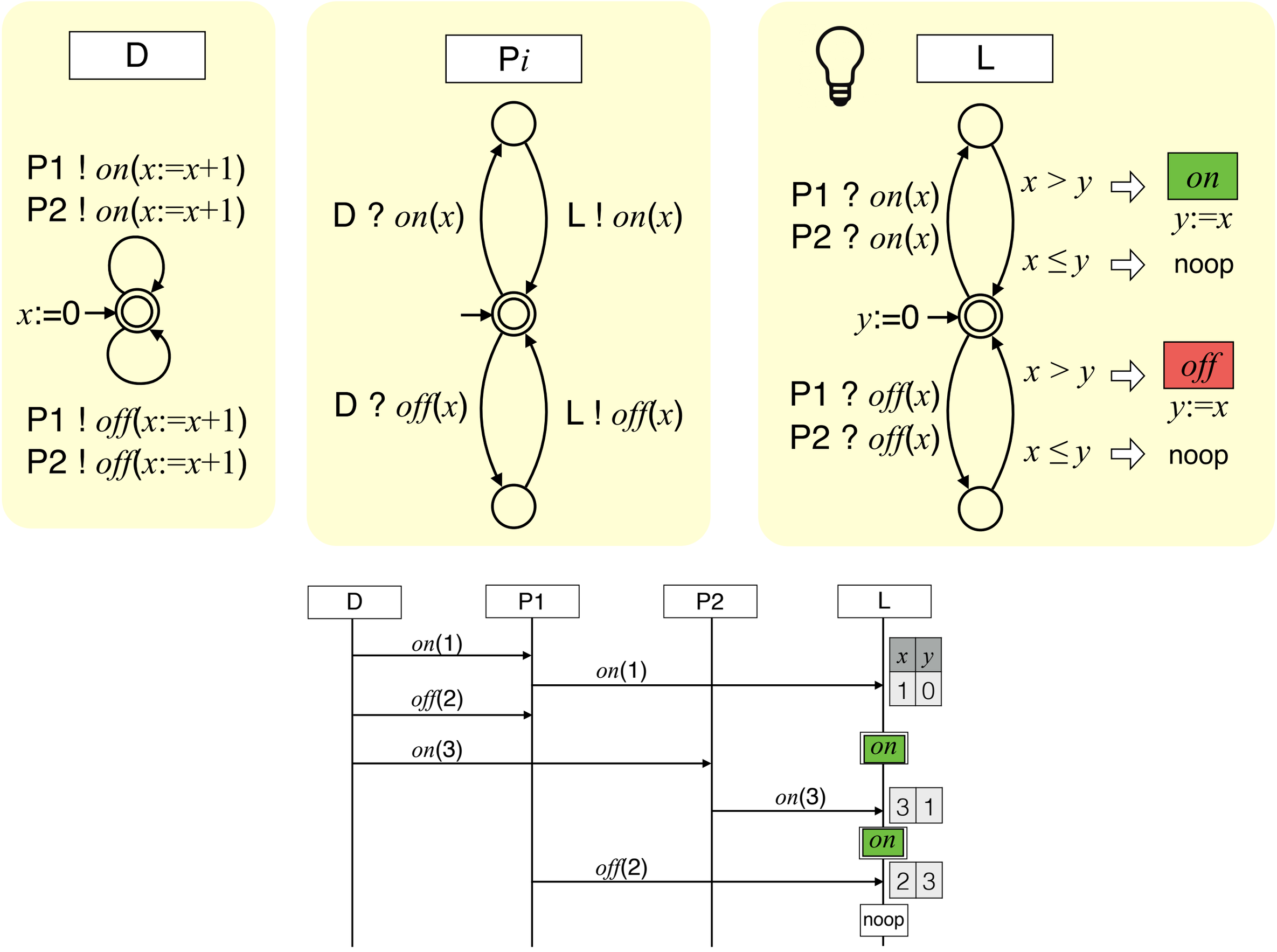


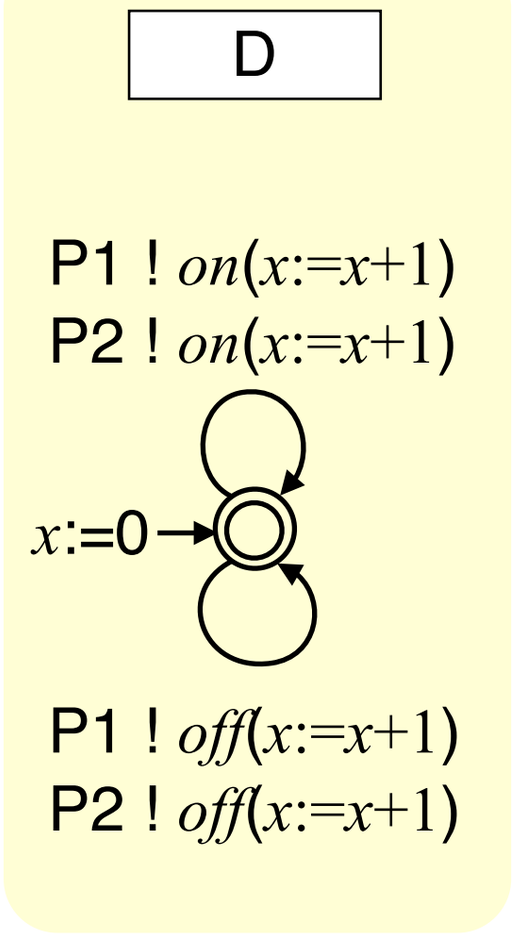


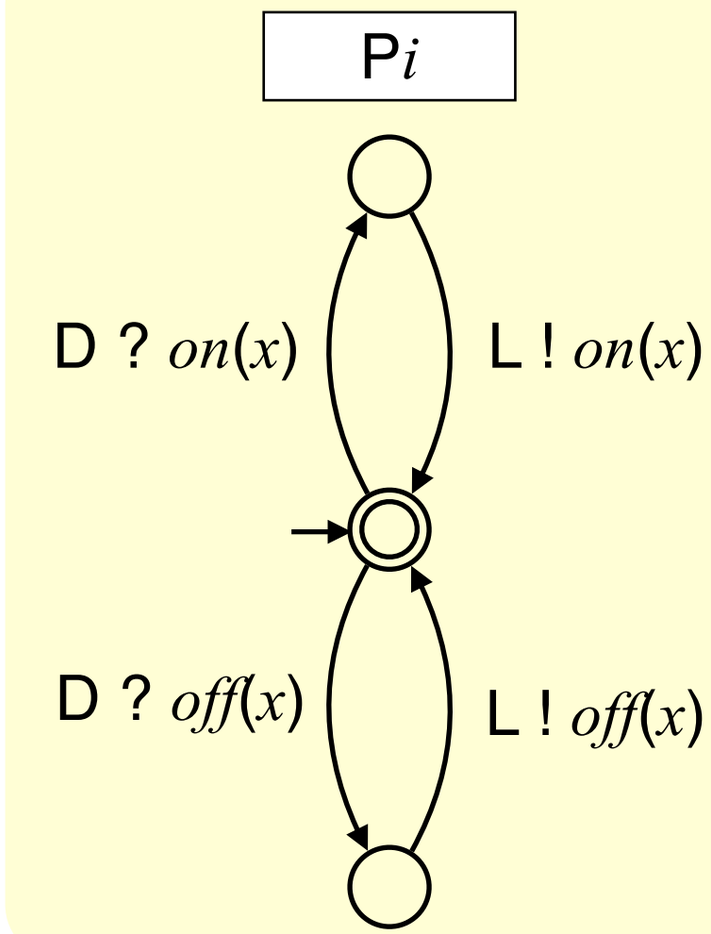


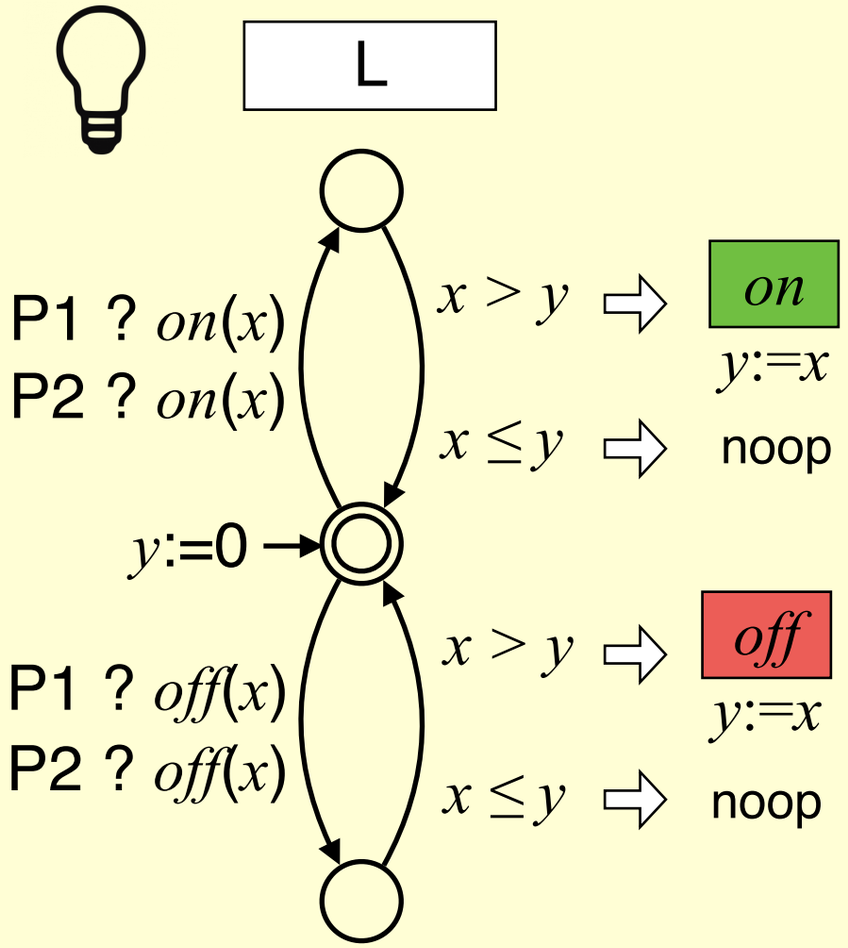
More recent!

Display is updated based on outdated information!


D
P1 ! on(x:=x+1)
P2 ! on(x:=x+1)
x:=0
P1 ! off(x:=x+1)
P2 ! off(x:=x+1)
Pi
D ? on(x)
L ! on(x)
D ? off(x)
L ! off(x)
L
P1 ? on(x)
P2 ? on(x)
x > y
y:=x
on
x ≤ y
noop
y:=0
P1 ? off(x)
P2 ? off(x)
x > y
y:=x
off
x ≤ y
noop
D
P1
P2
L
on(1)
on(1)
off(2)
on(3)
x
y
1
0
on
on(3)
3
1
on
off(2)
2
3
noop







Protocol is correct.

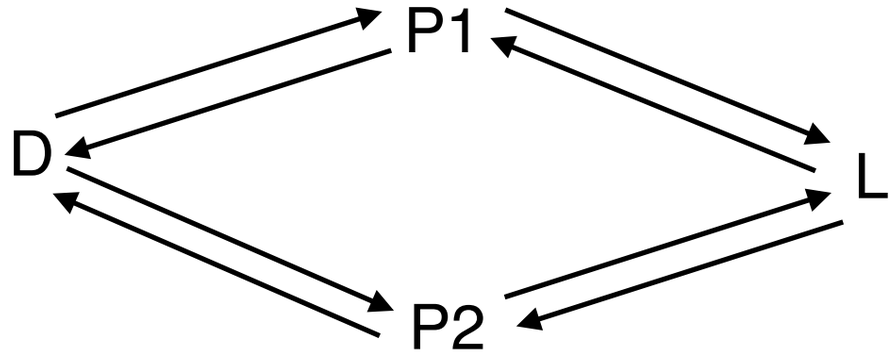
But unbounded counters are needed.

How about finite-state solution (i.e., with a bounded number of messsages)?

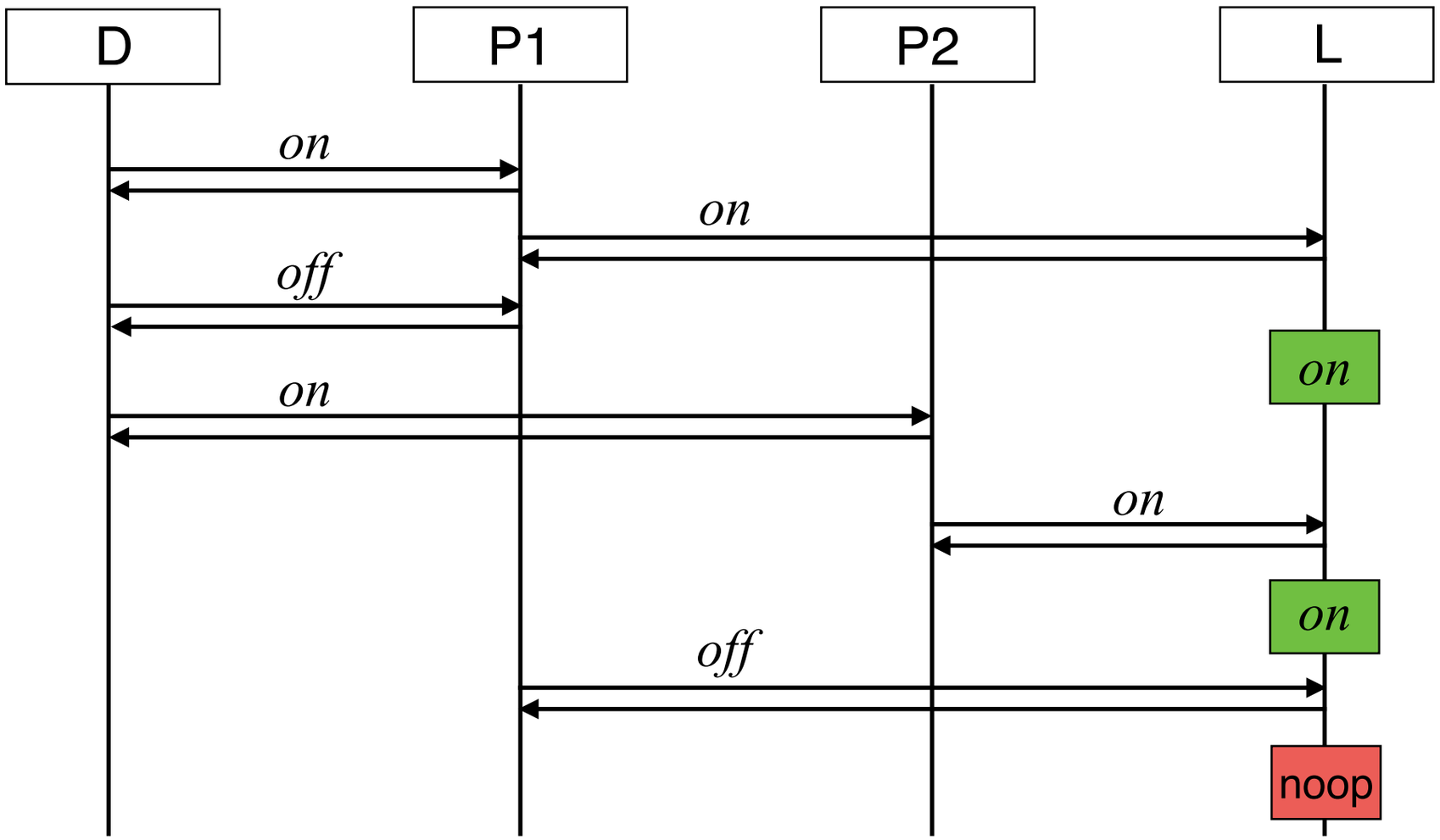
Difficult to design.

Impossible in our example (no « feedback »).

A solution exists for:



… but it is still very difficult to obtain. (When can a time stamp be reused?)

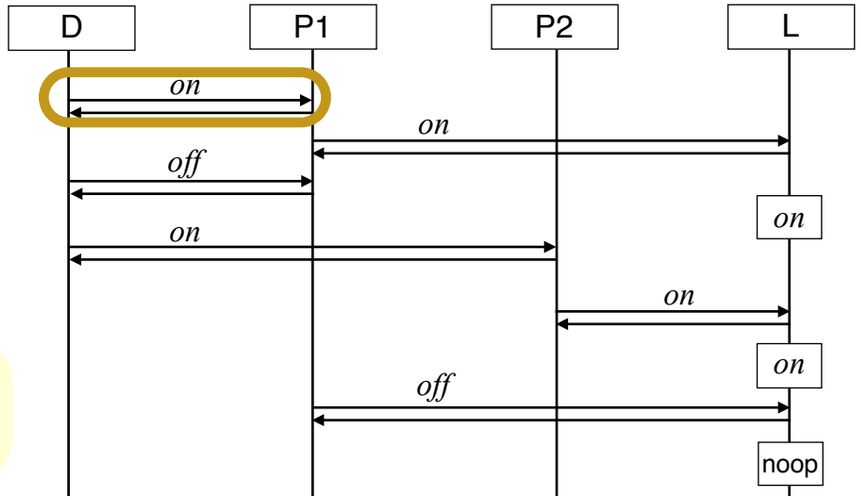


Is there an automatic way of synthesizing a distributed program from a specification?

$$L_{spec} \subseteq \Sigma^*$$

$$\begin{aligned}\Sigma = \quad & \{\langle \mathsf{D} \overset{on}{\rightleftarrows} \mathsf{P1}\rangle, \langle \mathsf{D} \overset{on}{\rightleftarrows} \mathsf{P2}\rangle, \langle \mathsf{D} \overset{off}{\rightleftarrows} \mathsf{P1}\rangle, \langle \mathsf{D} \overset{off}{\rightleftarrows} \mathsf{P2}\rangle\} \\ \cup \quad & \{\langle \mathsf{P1} \overset{on}{\rightleftarrows} \mathsf{L}\rangle, \langle \mathsf{P2} \overset{on}{\rightleftarrows} \mathsf{L}\rangle, \langle \mathsf{P1} \overset{off}{\rightleftarrows} \mathsf{L}\rangle, \langle \mathsf{P2} \overset{off}{\rightleftarrows} \mathsf{L}\rangle\} \\ \cup \quad & \{on, off, \mathsf{noop}\}\end{aligned}$$

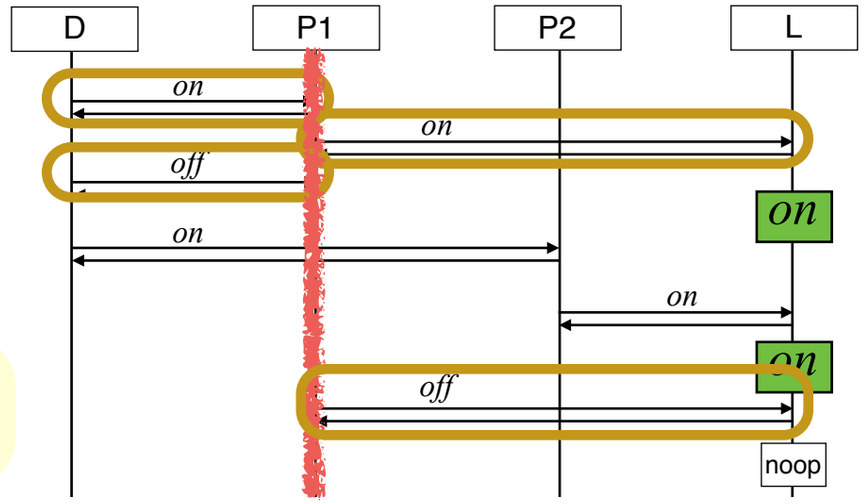
$$\langle \mathsf{D} \overset{on}{\rightleftarrows} \mathsf{P1}\rangle \langle \mathsf{P1} \overset{on}{\rightleftarrows} \mathsf{L}\rangle \langle \mathsf{D} \overset{off}{\rightleftarrows} \mathsf{P1}\rangle\ on\ \langle \mathsf{D} \overset{on}{\rightleftarrows} \mathsf{P2}\rangle \langle \mathsf{P2} \overset{on}{\rightleftarrows} \mathsf{L}\rangle\ on\ \langle \mathsf{P1} \overset{off}{\rightleftarrows} \mathsf{L}\rangle\ \mathsf{noop}$$



$$L_{spec} \subseteq \Sigma^*$$

$$\begin{aligned}\Sigma = \quad & \{\langle \mathsf{D} \overset{on}{\rightleftarrows} \mathsf{P1}\rangle, \langle \mathsf{D} \overset{on}{\rightleftarrows} \mathsf{P2}\rangle, \langle \mathsf{D} \overset{off}{\rightleftarrows} \mathsf{P1}\rangle, \langle \mathsf{D} \overset{off}{\rightleftarrows} \mathsf{P2}\rangle\} \\ \cup \quad & \{\langle \mathsf{P1} \overset{on}{\rightleftarrows} \mathsf{L}\rangle, \langle \mathsf{P2} \overset{on}{\rightleftarrows} \mathsf{L}\rangle, \langle \mathsf{P1} \overset{off}{\rightleftarrows} \mathsf{L}\rangle, \langle \mathsf{P2} \overset{off}{\rightleftarrows} \mathsf{L}\rangle\} \\ \cup \quad & \{on, off, \mathsf{noop}\}\end{aligned}$$

$\langle \mathsf{D} \overset{on}{\rightleftarrows} \mathsf{P1}\rangle \langle \mathsf{P1} \overset{on}{\rightleftarrows} \mathsf{L}\rangle \langle \mathsf{D} \overset{off}{\rightleftarrows} \mathsf{P1}\rangle \; on \; \langle \mathsf{D} \overset{on}{\rightleftarrows} \mathsf{P2}\rangle \langle \mathsf{P2} \overset{on}{\rightleftarrows} \mathsf{L}\rangle \; on \; \langle \mathsf{P1} \overset{off}{\rightleftarrows} \mathsf{L}\rangle \; \mathsf{noop}$



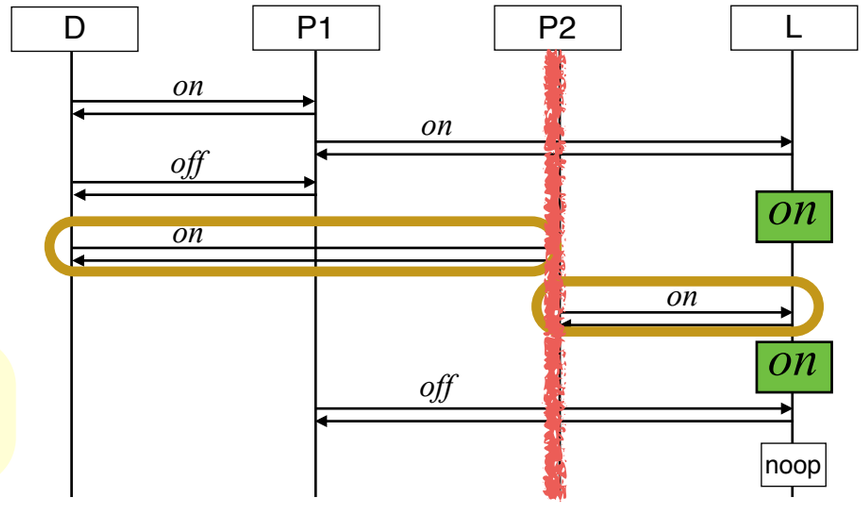
1. The projection to P1 is in:

$$(\langle \mathsf{D} \rightleftarrows \mathsf{P1}\rangle \langle \mathsf{P1} \rightleftarrows \mathsf{L}\rangle + \langle \mathsf{D} \overset{on}{\rightleftarrows} \mathsf{P1}\rangle \langle \mathsf{P1} \overset{on}{\rightleftarrows} \mathsf{L}\rangle + \langle \mathsf{D} \overset{off}{\rightleftarrows} \mathsf{P1}\rangle \langle \mathsf{P1} \overset{off}{\rightleftarrows} \mathsf{L}\rangle)^* .$$

$$L_{spec} \subseteq \Sigma^*$$

$$\begin{aligned}\Sigma = \quad & \{\langle \mathsf{D} \overset{on}{\rightleftarrows} \mathsf{P1}\rangle, \langle \mathsf{D} \overset{on}{\rightleftarrows} \mathsf{P2}\rangle, \langle \mathsf{D} \overset{off}{\rightleftarrows} \mathsf{P1}\rangle, \langle \mathsf{D} \overset{off}{\rightleftarrows} \mathsf{P2}\rangle\} \\ \cup \quad & \{\langle \mathsf{P1} \overset{on}{\rightleftarrows} \mathsf{L}\rangle, \langle \mathsf{P2} \overset{on}{\rightleftarrows} \mathsf{L}\rangle, \langle \mathsf{P1} \overset{off}{\rightleftarrows} \mathsf{L}\rangle, \langle \mathsf{P2} \overset{off}{\rightleftarrows} \mathsf{L}\rangle\} \\ \cup \quad & \{on, off, \mathsf{noop}\}\end{aligned}$$

$$\langle \mathsf{D} \overset{on}{\rightleftarrows} \mathsf{P1}\rangle \langle \mathsf{P1} \overset{on}{\rightleftarrows} \mathsf{L}\rangle \langle \mathsf{D} \overset{off}{\rightleftarrows} \mathsf{P1}\rangle \, on \, \langle \mathsf{D} \overset{on}{\rightleftarrows} \mathsf{P2}\rangle \langle \mathsf{P2} \overset{on}{\rightleftarrows} \mathsf{L}\rangle \, on \, \langle \mathsf{P1} \overset{off}{\rightleftarrows} \mathsf{L}\rangle \, \mathsf{noop}$$



1. The projection to P1 is in:

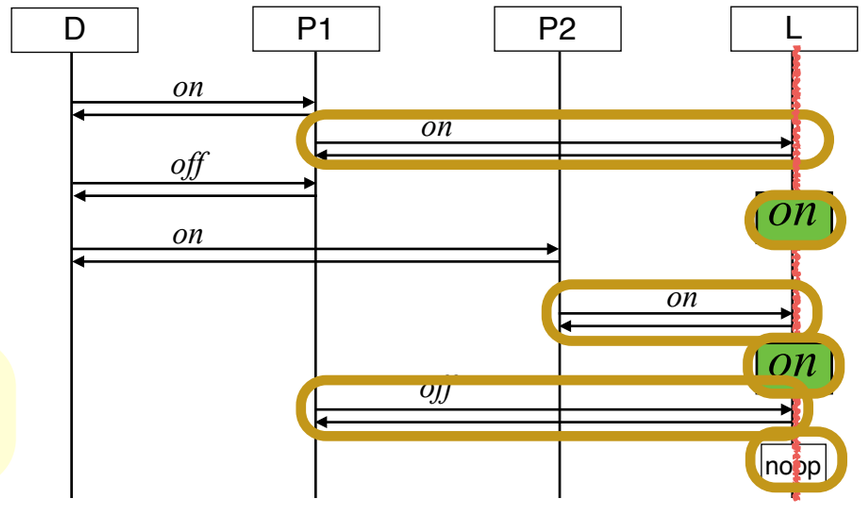
$$(\langle \mathsf{D} \rightleftarrows \mathsf{P1}\rangle \langle \mathsf{P1} \rightleftarrows \mathsf{L}\rangle + \langle \mathsf{D} \overset{on}{\rightleftarrows} \mathsf{P1}\rangle \langle \mathsf{P1} \overset{on}{\rightleftarrows} \mathsf{L}\rangle \langle \mathsf{D} \overset{off}{\rightleftarrows} \mathsf{P1}\rangle \langle \mathsf{P1} \overset{off}{\rightleftarrows} \mathsf{L}\rangle)^* .$$

2. Similarly for P2.

$$L_{spec} \subseteq \Sigma^*$$

$$\begin{aligned}\Sigma = \quad & \{\langle \mathsf{D} \overset{on}{\rightleftarrows} \mathsf{P1}\rangle, \langle \mathsf{D} \overset{on}{\rightleftarrows} \mathsf{P2}\rangle, \langle \mathsf{D} \overset{off}{\rightleftarrows} \mathsf{P1}\rangle, \langle \mathsf{D} \overset{off}{\rightleftarrows} \mathsf{P2}\rangle\} \\ \cup \quad & \{\langle \mathsf{P1} \overset{on}{\rightleftarrows} \mathsf{L}\rangle, \langle \mathsf{P2} \overset{on}{\rightleftarrows} \mathsf{L}\rangle, \langle \mathsf{P1} \overset{off}{\rightleftarrows} \mathsf{L}\rangle, \langle \mathsf{P2} \overset{off}{\rightleftarrows} \mathsf{L}\rangle\} \\ \cup \quad & \{on, off, \mathsf{noop}\}\end{aligned}$$

$$\langle \mathsf{D} \overset{on}{\rightleftarrows} \mathsf{P1}\rangle \langle \mathsf{P1} \overset{on}{\rightleftarrows} \mathsf{L}\rangle \langle \mathsf{D} \overset{off}{\rightleftarrows} \mathsf{P1}\rangle on \langle \mathsf{D} \overset{on}{\rightleftarrows} \mathsf{P2}\rangle \langle \mathsf{P2} \overset{on}{\rightleftarrows} \mathsf{L}\rangle on \langle \mathsf{P1} \overset{off}{\rightleftarrows} \mathsf{L}\rangle \mathsf{noop}$$



1. The projection to P1 is in:

$$(\langle \mathsf{D} \rightleftarrows \mathsf{P1}\rangle \langle \mathsf{P1} \rightleftarrows \mathsf{L}\rangle + \langle \mathsf{D} \rightleftarrows \mathsf{P1}\rangle \langle \mathsf{P1} \rightleftarrows \mathsf{L}\rangle)^* .$$

2. Similarly for P2.

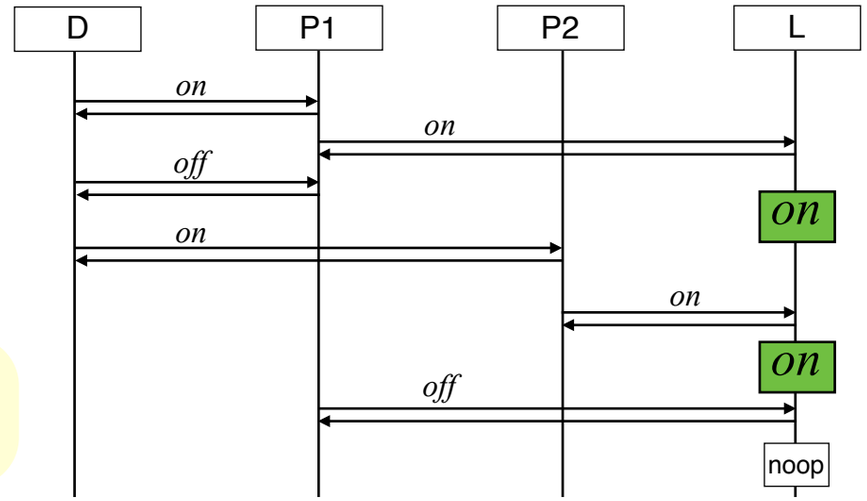
3. The projection to L is in:

$$\Big((\langle \mathsf{P1} \overset{on}{\rightleftarrows} \mathsf{L}\rangle + \langle \mathsf{P2} \overset{on}{\rightleftarrows} \mathsf{L}\rangle + \langle \mathsf{P1} \overset{off}{\rightleftarrows} \mathsf{L}\rangle + \langle \mathsf{P2} \overset{off}{\rightleftarrows} \mathsf{L}\rangle)(on + off + \mathsf{noop})\Big)^* .$$

$$L_{spec} \subseteq \Sigma^*$$

$$\begin{aligned}\Sigma = \quad & \{\langle \mathsf{D} \overset{on}{\rightleftarrows} \mathsf{P1}\rangle, \langle \mathsf{D} \overset{on}{\rightleftarrows} \mathsf{P2}\rangle, \langle \mathsf{D} \overset{off}{\rightleftarrows} \mathsf{P1}\rangle, \langle \mathsf{D} \overset{off}{\rightleftarrows} \mathsf{P2}\rangle\} \\ \cup \quad & \{\langle \mathsf{P1} \overset{on}{\rightleftarrows} \mathsf{L}\rangle, \langle \mathsf{P2} \overset{on}{\rightleftarrows} \mathsf{L}\rangle, \langle \mathsf{P1} \overset{off}{\rightleftarrows} \mathsf{L}\rangle, \langle \mathsf{P2} \overset{off}{\rightleftarrows} \mathsf{L}\rangle\} \\ \cup \quad & \{on, off, \mathsf{noop}\}\end{aligned}$$

$$\langle \mathsf{D} \overset{on}{\rightleftarrows} \mathsf{P1}\rangle \langle \mathsf{P1} \overset{on}{\rightleftarrows} \mathsf{L}\rangle \langle \mathsf{D} \overset{off}{\rightleftarrows} \mathsf{P1}\rangle\, on\, \langle \mathsf{D} \overset{on}{\rightleftarrows} \mathsf{P2}\rangle \langle \mathsf{P2} \overset{on}{\rightleftarrows} \mathsf{L}\rangle\, on\, \langle \mathsf{P1} \overset{off}{\rightleftarrows} \mathsf{L}\rangle\, \mathsf{noop}$$



1. The projection to P1 is in:

$$(\langle \mathsf{D} \rightleftarrows \mathsf{P1}\rangle\langle \mathsf{P1} \rightleftarrows \mathsf{L}\rangle + \langle \mathsf{D} \overset{off}{\rightleftarrows} \mathsf{P1}\rangle\langle \mathsf{P1} \overset{off}{\rightleftarrows} \mathsf{L}\rangle)^* .$$

2. Similarly for P2.

3. The projection to L is in:

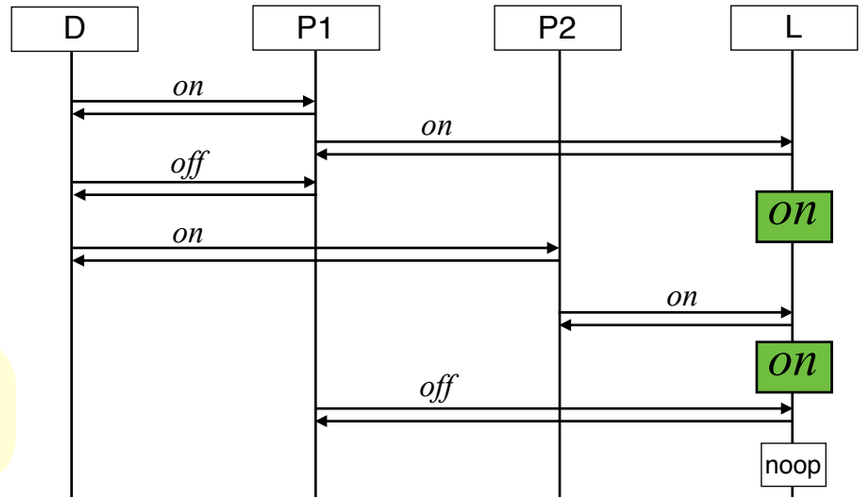
$$\Big((\langle \mathsf{P1} \overset{on}{\rightleftarrows} \mathsf{L}\rangle + \langle \mathsf{P2} \overset{on}{\rightleftarrows} \mathsf{L}\rangle + \langle \mathsf{P1} \overset{off}{\rightleftarrows} \mathsf{L}\rangle + \langle \mathsf{P2} \overset{off}{\rightleftarrows} \mathsf{L}\rangle)(on + off + \mathsf{noop})\Big)^* .$$

4. The display is updated iff the last (previous) status message emitted by D
   that has already been followed by a forward
   was not yet followed by a corresponding update by L.

$$L_{spec} \subseteq \Sigma^*$$

$$\begin{aligned}\Sigma = \quad & \{\langle \mathsf{D} \overset{on}{\rightleftarrows} \mathsf{P1}\rangle, \langle \mathsf{D} \overset{on}{\rightleftarrows} \mathsf{P2}\rangle, \langle \mathsf{D} \overset{off}{\rightleftarrows} \mathsf{P1}\rangle, \langle \mathsf{D} \overset{off}{\rightleftarrows} \mathsf{P2}\rangle\} \\ \cup \quad & \{\langle \mathsf{P1} \overset{on}{\rightleftarrows} \mathsf{L}\rangle, \langle \mathsf{P2} \overset{on}{\rightleftarrows} \mathsf{L}\rangle, \langle \mathsf{P1} \overset{off}{\rightleftarrows} \mathsf{L}\rangle, \langle \mathsf{P2} \overset{off}{\rightleftarrows} \mathsf{L}\rangle\} \\ \cup \quad & \{on, off, \mathsf{noop}\}\end{aligned}$$

$$\langle \mathsf{D} \overset{on}{\rightleftarrows} \mathsf{P1}\rangle \langle \mathsf{P1} \overset{on}{\rightleftarrows} \mathsf{L}\rangle \langle \mathsf{D} \overset{off}{\rightleftarrows} \mathsf{P1}\rangle\, on\, \langle \mathsf{D} \overset{on}{\rightleftarrows} \mathsf{P2}\rangle \langle \mathsf{P2} \overset{on}{\rightleftarrows} \mathsf{L}\rangle\, on\, \langle \mathsf{P1} \overset{off}{\rightleftarrows} \mathsf{L}\rangle\, \mathsf{noop}$$



1. The projection to P1 is in:

$$(\langle \mathsf{D} \rightleftarrows \mathsf{P1}\rangle\langle \mathsf{P1} \rightleftarrows \mathsf{L}\rangle + \langle \mathsf{D} \overset{on}{\rightleftarrows} \mathsf{P1}\rangle\langle \mathsf{P1} \overset{on}{\rightleftarrows} \mathsf{L}\rangle + \langle \mathsf{D} \overset{off}{\rightleftarrows} \mathsf{P1}\rangle\langle \mathsf{P1} \overset{off}{\rightleftarrows} \mathsf{L}\rangle)^*.$$

2. Similarly for P2.

3. The projection to L is in:

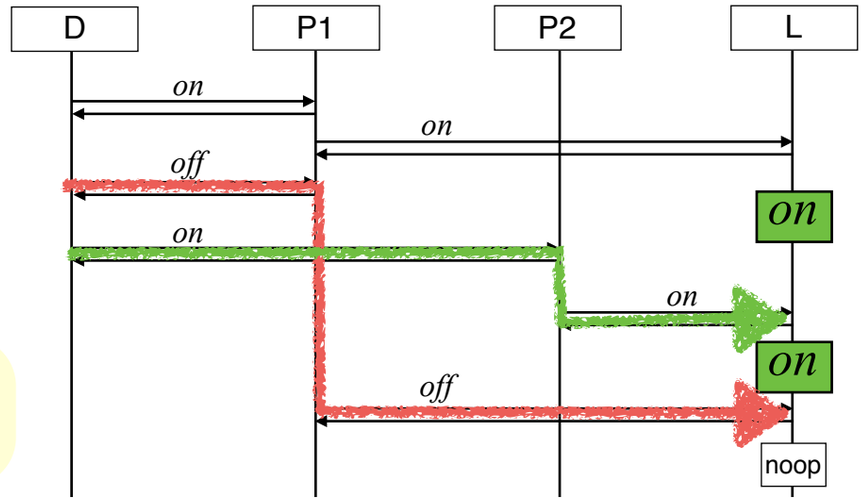
$$\Big((\langle \mathsf{P1} \overset{on}{\rightleftarrows} \mathsf{L}\rangle + \langle \mathsf{P2} \overset{on}{\rightleftarrows} \mathsf{L}\rangle + \langle \mathsf{P1} \overset{off}{\rightleftarrows} \mathsf{L}\rangle + \langle \mathsf{P2} \overset{off}{\rightleftarrows} \mathsf{L}\rangle)(on + off + \mathsf{noop})\Big)^*.$$

4. The display is updated iff the last (previous) status message emitted by D
   that has already been followed by a forward
   was not yet followed by a corresponding update by L.

$$L_{spec} \subseteq \Sigma^*$$

$$\begin{aligned}\Sigma = \quad & \{\langle \mathsf{D} \overset{on}{\rightleftarrows} \mathsf{P1}\rangle, \langle \mathsf{D} \overset{on}{\rightleftarrows} \mathsf{P2}\rangle, \langle \mathsf{D} \overset{off}{\rightleftarrows} \mathsf{P1}\rangle, \langle \mathsf{D} \overset{off}{\rightleftarrows} \mathsf{P2}\rangle\} \\ \cup \quad & \{\langle \mathsf{P1} \overset{on}{\rightleftarrows} \mathsf{L}\rangle, \langle \mathsf{P2} \overset{on}{\rightleftarrows} \mathsf{L}\rangle, \langle \mathsf{P1} \overset{off}{\rightleftarrows} \mathsf{L}\rangle, \langle \mathsf{P2} \overset{off}{\rightleftarrows} \mathsf{L}\rangle\} \\ \cup \quad & \{on, off, \mathsf{noop}\}\end{aligned}$$

$$\langle \mathsf{D} \overset{on}{\rightleftarrows} \mathsf{P1}\rangle \langle \mathsf{P1} \overset{on}{\rightleftarrows} \mathsf{L}\rangle \langle \mathsf{D} \overset{off}{\rightleftarrows} \mathsf{P1}\rangle\, on\, \langle \mathsf{D} \overset{on}{\rightleftarrows} \mathsf{P2}\rangle \langle \mathsf{P2} \overset{on}{\rightleftarrows} \mathsf{L}\rangle\, on\, \langle \mathsf{P1} \overset{off}{\rightleftarrows} \mathsf{L}\rangle\, \mathsf{noop}$$



1. The projection to P1 is in:

$$(\langle \mathsf{D} \overset{on}{\rightleftarrows} \mathsf{P1}\rangle\langle \mathsf{P1} \overset{on}{\rightleftarrows} \mathsf{L}\rangle + \langle \mathsf{D} \overset{off}{\rightleftarrows} \mathsf{P1}\rangle\langle \mathsf{P1} \overset{off}{\rightleftarrows} \mathsf{L}\rangle)^*.$$

2. Similarly for P2.

3. The projection to L is in:

$$\Big((\langle \mathsf{P1} \overset{on}{\rightleftarrows} \mathsf{L}\rangle + \langle \mathsf{P2} \overset{on}{\rightleftarrows} \mathsf{L}\rangle + \langle \mathsf{P1} \overset{off}{\rightleftarrows} \mathsf{L}\rangle + \langle \mathsf{P2} \overset{off}{\rightleftarrows} \mathsf{L}\rangle)(on + off + \mathsf{noop})\Big)^*.$$

4. The display is updated iff the last (previous) status message emitted by D
   that has already been followed by a forward
   was not yet followed by a corresponding update by L.

Exercise: What are the words contained in $L_{spec} \subseteq \Sigma^*$?

- $\langle \mathsf{D} \overset{on}{\rightleftarrows} \mathsf{P1} \rangle \langle \mathsf{D} \overset{off}{\rightleftarrows} \mathsf{P2} \rangle \langle \mathsf{P2} \overset{off}{\rightleftarrows} \mathsf{L} \rangle \mathit{off} \langle \mathsf{P1} \overset{on}{\rightleftarrows} \mathsf{L} \rangle \mathsf{noop}$ ✔

Exercise: What are the words contained in $L_{spec} \subseteq \Sigma^*$?

- $\langle \mathsf{D} \overset{on}{\rightleftarrows} \mathsf{P1} \rangle \langle \mathsf{D} \overset{off}{\rightleftarrows} \mathsf{P2} \rangle \langle \mathsf{P2} \overset{off}{\rightleftarrows} \mathsf{L} \rangle \mathit{off} \langle \mathsf{P1} \overset{on}{\rightleftarrows} \mathsf{L} \rangle \mathsf{noop}$ ✔
- $\langle \mathsf{D} \overset{on}{\rightleftarrows} \mathsf{P1} \rangle \langle \mathsf{P1} \overset{on}{\rightleftarrows} \mathsf{L} \rangle \langle \mathsf{D} \overset{on}{\rightleftarrows} \mathsf{P1} \rangle \mathit{on}$ ✗

Exercise: What are the words contained in $L_{spec} \subseteq \Sigma^*$?

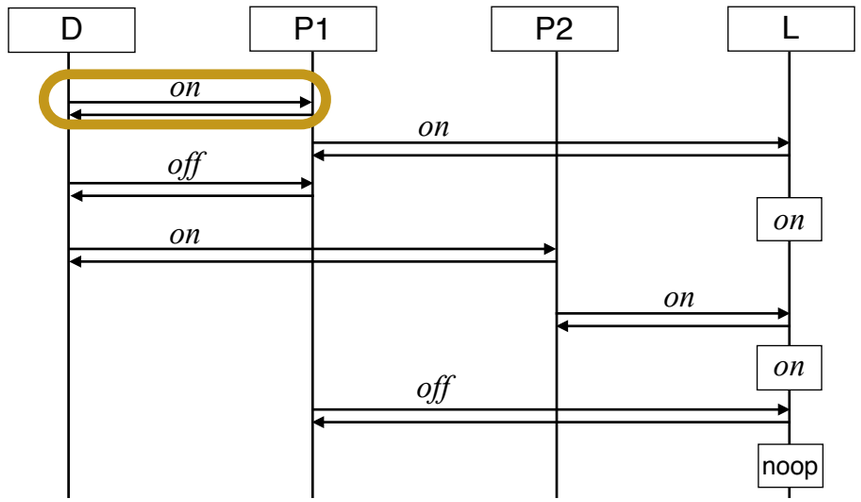
- $\langle \mathsf{D} \overset{on}{\rightleftarrows} \mathsf{P1} \rangle \langle \mathsf{D} \overset{off}{\rightleftarrows} \mathsf{P2} \rangle \langle \mathsf{P2} \overset{off}{\rightleftarrows} \mathsf{L} \rangle off \langle \mathsf{P1} \overset{on}{\rightleftarrows} \mathsf{L} \rangle \mathsf{noop}$ ✔
- $\langle \mathsf{D} \overset{on}{\rightleftarrows} \mathsf{P1} \rangle \langle \mathsf{P1} \overset{on}{\rightleftarrows} \mathsf{L} \rangle \langle \mathsf{D} \overset{on}{\rightleftarrows} \mathsf{P1} \rangle on$ ✗
- $\langle \mathsf{D} \overset{on}{\rightleftarrows} \mathsf{P1} \rangle \langle \mathsf{P1} \overset{on}{\rightleftarrows} \mathsf{L} \rangle \langle \mathsf{D} \overset{off}{\rightleftarrows} \mathsf{P2} \rangle on \langle \mathsf{P2} \overset{off}{\rightleftarrows} \mathsf{L} \rangle off$ ✔

Exercise: What are the words contained in $L_{spec} \subseteq \Sigma^*$?

- $\langle \mathsf{D} \overset{on}{\rightleftarrows} \mathsf{P1} \rangle \langle \mathsf{D} \overset{off}{\rightleftarrows} \mathsf{P2} \rangle \langle \mathsf{P2} \overset{off}{\rightleftarrows} \mathsf{L} \rangle off \langle \mathsf{P1} \overset{on}{\rightleftarrows} \mathsf{L} \rangle \mathsf{noop}$ ✔
- $\langle \mathsf{D} \overset{on}{\rightleftarrows} \mathsf{P1} \rangle \langle \mathsf{P1} \overset{on}{\rightleftarrows} \mathsf{L} \rangle \langle \mathsf{D} \overset{on}{\rightleftarrows} \mathsf{P1} \rangle on$ ✗
- $\langle \mathsf{D} \overset{on}{\rightleftarrows} \mathsf{P1} \rangle \langle \mathsf{P1} \overset{on}{\rightleftarrows} \mathsf{L} \rangle \langle \mathsf{D} \overset{off}{\rightleftarrows} \mathsf{P2} \rangle on \langle \mathsf{P2} \overset{off}{\rightleftarrows} \mathsf{L} \rangle off$ ✔
- $\langle \mathsf{D} \overset{on}{\rightleftarrows} \mathsf{P1} \rangle \langle \mathsf{P1} \overset{on}{\rightleftarrows} \mathsf{L} \rangle on \langle \mathsf{D} \overset{off}{\rightleftarrows} \mathsf{P2} \rangle \langle \mathsf{P2} \overset{off}{\rightleftarrows} \mathsf{L} \rangle off$ ✔

$$L_{spec} \subseteq \Sigma^*$$

$$\begin{aligned}\Sigma = \quad & \{\langle \mathsf{D} \overset{on}{\rightleftarrows} \mathsf{P1}\rangle, \langle \mathsf{D} \overset{on}{\rightleftarrows} \mathsf{P2}\rangle, \langle \mathsf{D} \overset{off}{\rightleftarrows} \mathsf{P1}\rangle, \langle \mathsf{D} \overset{off}{\rightleftarrows} \mathsf{P2}\rangle\} \\ \cup \quad & \{\langle \mathsf{P1} \overset{on}{\rightleftarrows} \mathsf{L}\rangle, \langle \mathsf{P2} \overset{on}{\rightleftarrows} \mathsf{L}\rangle, \langle \mathsf{P1} \overset{off}{\rightleftarrows} \mathsf{L}\rangle, \langle \mathsf{P2} \overset{off}{\rightleftarrows} \mathsf{L}\rangle\} \\ \cup \quad & \{on, off, \mathsf{noop}\}\end{aligned}$$



$\langle \mathsf{D} \overset{on}{\rightleftarrows} \mathsf{P1}\rangle \langle \mathsf{P1} \overset{on}{\rightleftarrows} \mathsf{L}\rangle \langle \mathsf{D} \overset{off}{\rightleftarrows} \mathsf{P1}\rangle\ on\ \langle \mathsf{D} \overset{on}{\rightleftarrows} \mathsf{P2}\rangle \langle \mathsf{P2} \overset{on}{\rightleftarrows} \mathsf{L}\rangle\ on\ \langle \mathsf{P1} \overset{off}{\rightleftarrows} \mathsf{L}\rangle\ \mathsf{noop}$

Observation:

- L_{spec} is regular.
- L_{spec} is closed under permutation rewriting of independent events.

Theorem [Zielonka 1987]:

Let L be a regular set of words that is closed under permutation rewriting of independent events. There is a deterministic finite-state distributed protocol that realizes L.

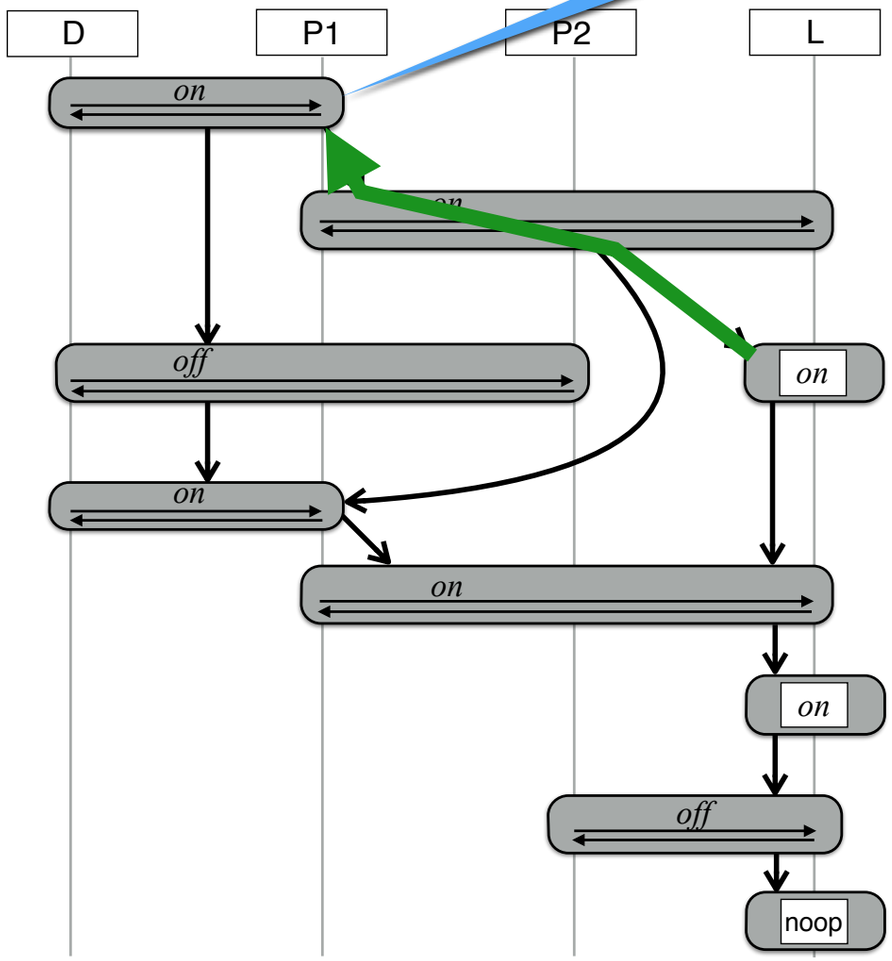
$L = (\langle \mathsf{D} \overset{on}{\rightleftarrows} \mathsf{P1}\rangle\langle \mathsf{P2} \overset{on}{\rightleftarrows} \mathsf{L}\rangle)^*$ is not realizable.

4. The display is updated iff the last (previous) status message emitted by D that has already been followed by a forward was not yet followed by a corresponding update by L.

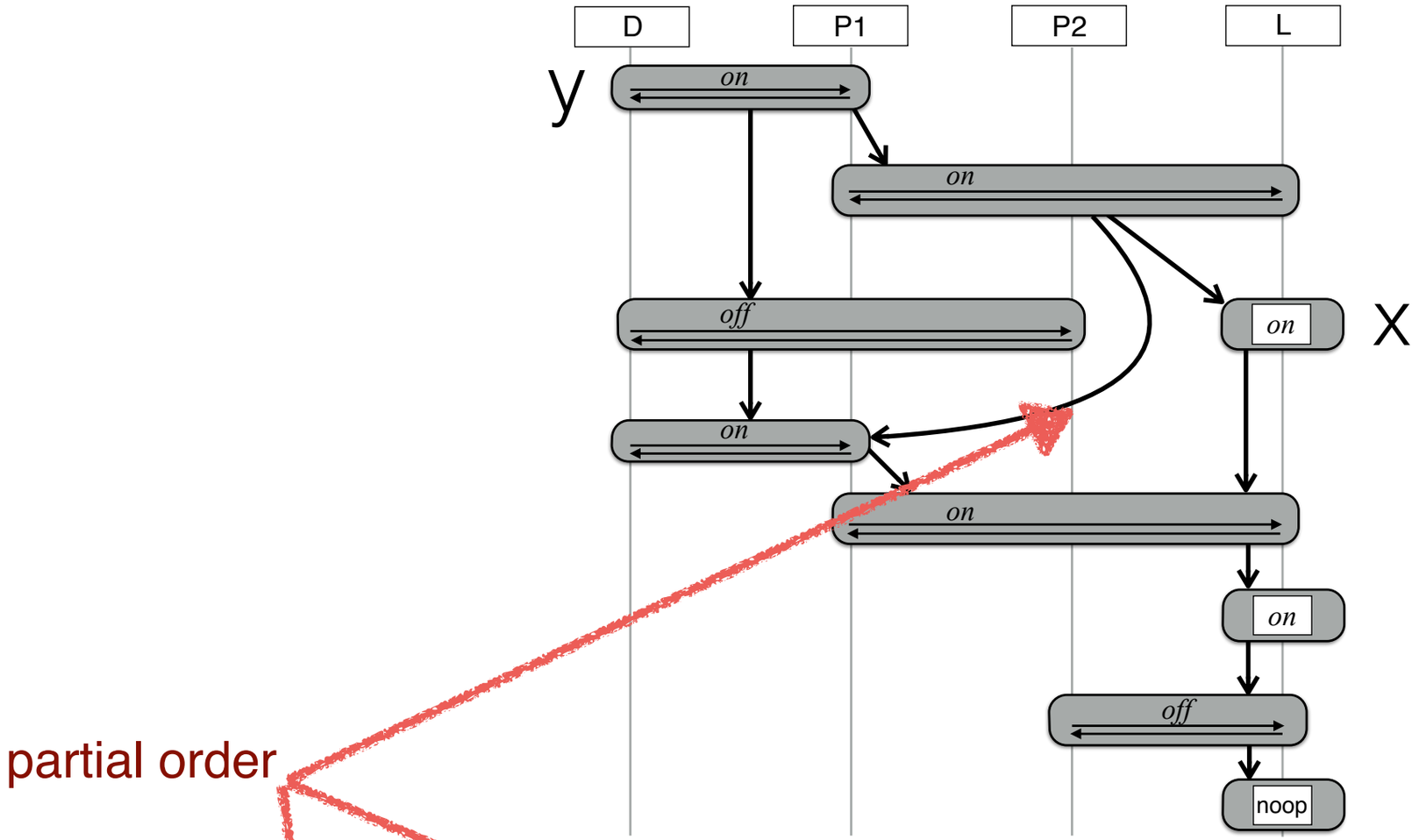
$$\langle \mathsf{D} \overset{on}{\rightleftarrows} \mathsf{P1}\rangle\langle \mathsf{P1} \overset{on}{\rightleftarrows} \mathsf{L}\rangle\langle \mathsf{D} \overset{off}{\rightleftarrows} \mathsf{P2}\rangle on \langle \mathsf{D} \overset{on}{\rightleftarrows} \mathsf{P1}\rangle\langle \mathsf{P1} \overset{on}{\rightleftarrows} \mathsf{L}\rangle on \langle \mathsf{P2} \overset{off}{\rightleftarrows} \mathsf{L}\rangle \mathsf{noop} \in L_{spec}$$

no *causal* dependency

last event in partial order



consider partial orders

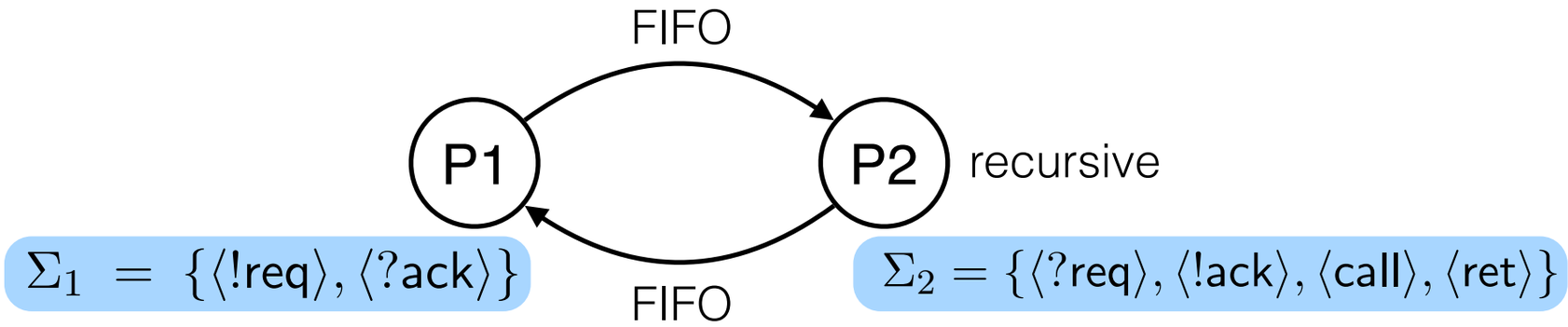


Theorem [Thomas 1990]:

Let L be an MSO-definable set of partial orders. There is a finite-state distributed protocol that realizes L.

"When L performs *on* (*off*, respectively), then the last (wrt. $\leq$) status message sent by D should also be *on* (*off*, respectively). Moreover, a display operation should be noop iff there has already been an acknowledgement between the latest status message and that operation."

Reasonig about recursive processes



Consider the following protocol:

- From time to time, P1 sends requests to P2.
- When receiving a request, P2 calls a procedure, which itself may call sub-procedures.
- When returning from the (outermost) procedure, P2 sends an acknowledgment to P1.

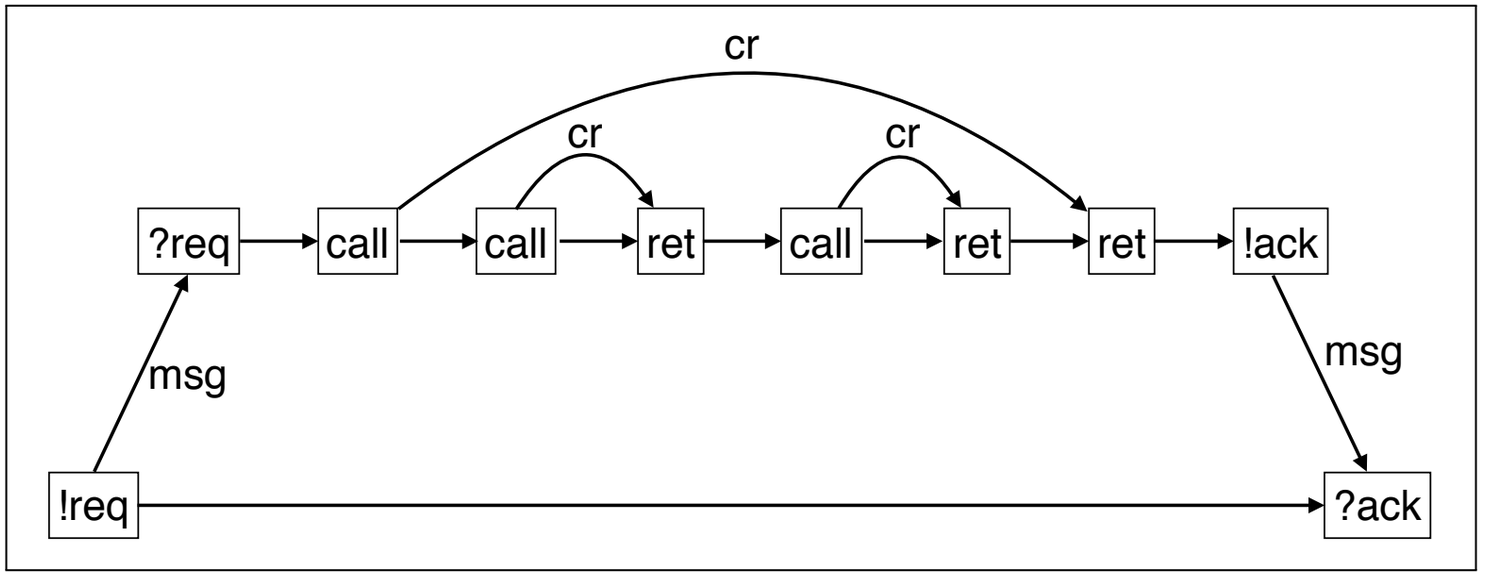
$\langle !\mathsf{req}\rangle\ \underline{\langle ?\mathsf{req}\rangle}\ \langle \mathsf{call}\rangle\ \langle \mathsf{call}\rangle\ \langle \mathsf{ret}\rangle\ \langle \mathsf{call}\rangle\ \langle \mathsf{ret}\rangle\ \langle \mathsf{ret}\rangle\ \underline{\langle !\mathsf{ack}\rangle}\ \langle ?\mathsf{ack}\rangle$ ✔

$\langle !\mathsf{req}\rangle\ \underline{\langle ?\mathsf{req}\rangle}\ \langle \mathsf{call}\rangle\ \langle \mathsf{call}\rangle\ \langle \mathsf{ret}\rangle\ \langle \mathsf{call}\rangle\ \langle \mathsf{ret}\rangle\ \underline{\langle !\mathsf{ack}\rangle}\ \langle ?\mathsf{ack}\rangle\ \langle \mathsf{ret}\rangle$ ✗

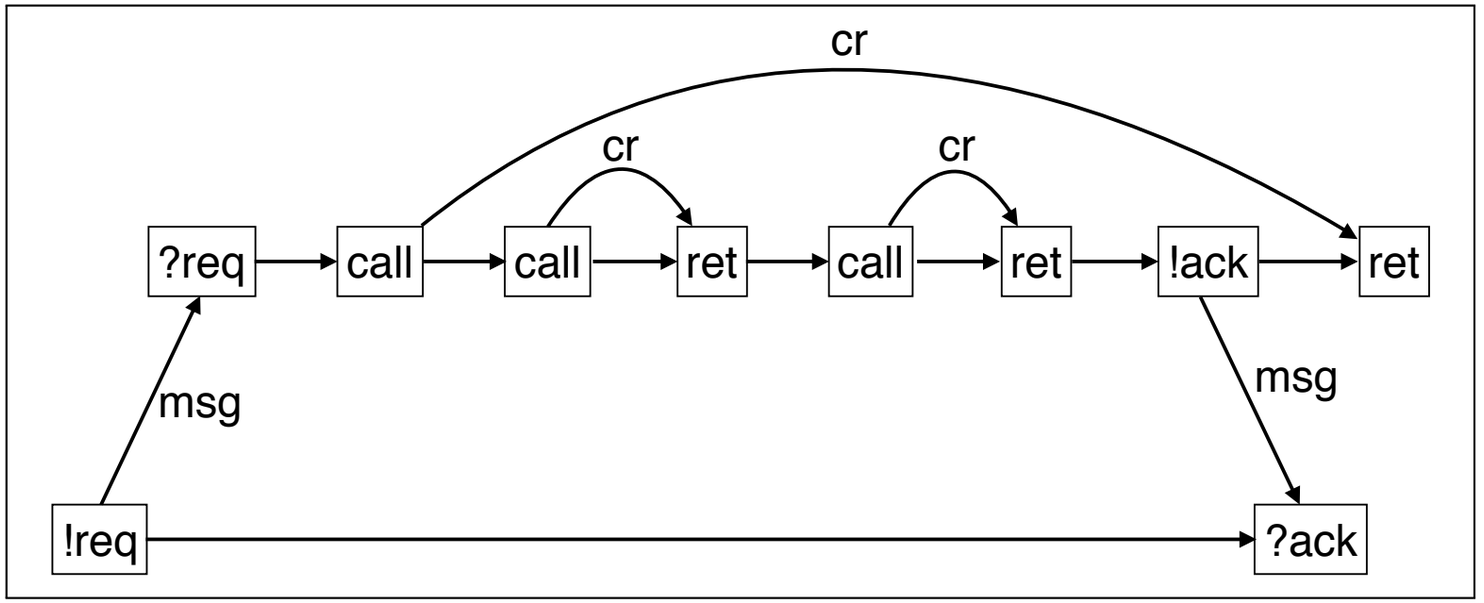
Is this a regular language? No!

So, what is a good specification language?

$\langle !\mathsf{req}\rangle\ \underline{\langle ?\mathsf{req}\rangle}\ \langle \mathsf{call}\rangle\ \langle \mathsf{call}\rangle\ \langle \mathsf{ret}\rangle\ \langle \mathsf{call}\rangle\ \langle \mathsf{ret}\rangle\ \langle \mathsf{ret}\rangle\ \underline{\langle !\mathsf{ack}\rangle}\ \langle ?\mathsf{ack}\rangle$ ✔



$\langle !\mathsf{req}\rangle\ \underline{\langle ?\mathsf{req}\rangle}\ \langle \mathsf{call}\rangle\ \langle \mathsf{call}\rangle\ \langle \mathsf{ret}\rangle\ \langle \mathsf{call}\rangle\ \langle \mathsf{ret}\rangle\ \underline{\langle !\mathsf{ack}\rangle}\ \langle ?\mathsf{ack}\rangle\ \langle \mathsf{ret}\rangle$ ✗



MSO formula over graphs

$$\forall x\ \langle ?\mathsf{req}\rangle(x) \Rightarrow \exists y_1, y_2, z \Big(x \to y_1 \wedge \mathsf{cr}(y_1, y_2) \wedge y_2 \to z \wedge \langle !\mathsf{ack}\rangle(z) \Big)$$

Let's be more formal ...