CONCEPTION ET VÉRIFICATION DE SYSTÈMES CRITIQUES

LA SPÉCIFICATION DES PROPRIÉTÉS AVEC LA LOGIQUE LTL

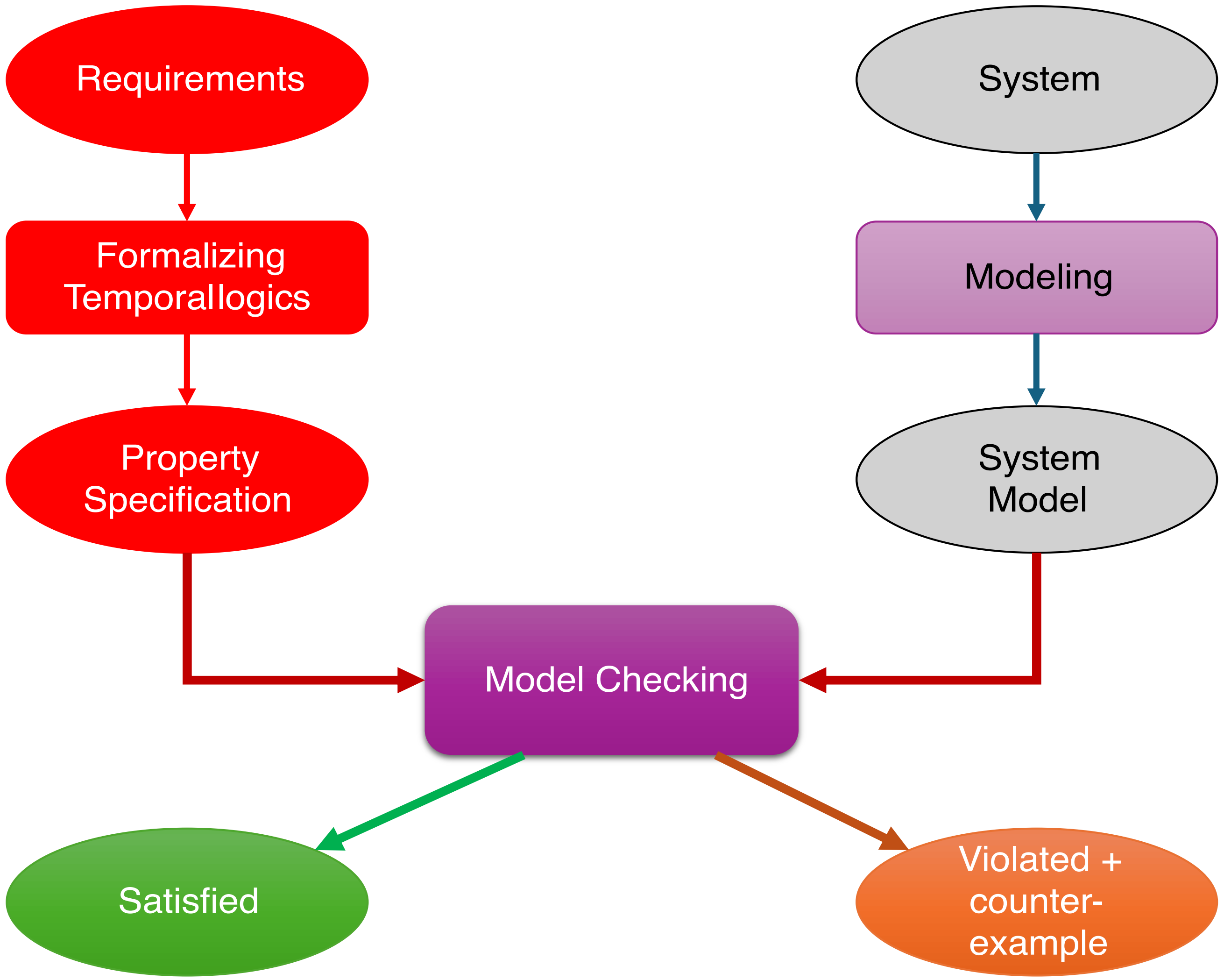
2A Cursus Ingénieurs - ST5 : Modélisation fonctionnelle et régulation
CentraleSupelec - Université Paris-Saclay - 2024/2025

Idir AIT SADOUNE
idir.aitsadoune@centralesupelec.fr

OUTLINE

Back to the outline - Back to the begin

PRINCIPLE OF MODEL-CHECKING



OUTLINE

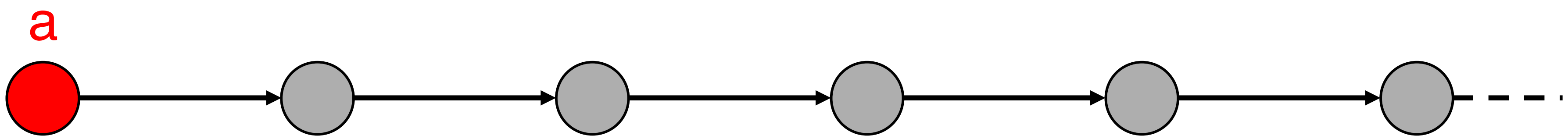
Back to the outline - Back to the begin

PROPOSITIONAL LINEAR TEMPORAL LOGIC (LTL)

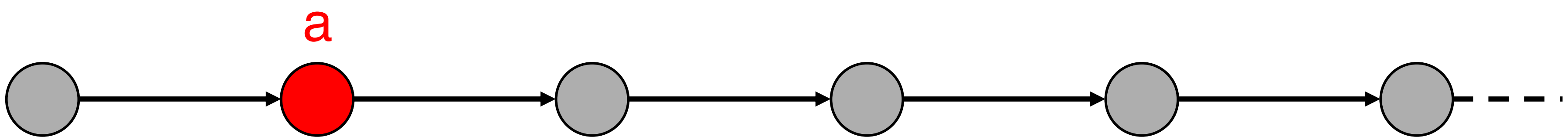
$$\phi \quad ::= \quad true \mid a \mid \phi \wedge \phi \mid \neg \phi \mid \bigcirc \phi \mid \Box \phi$$

where $a \in AP$ $\bigcirc$ (next) $\Box$ (always)

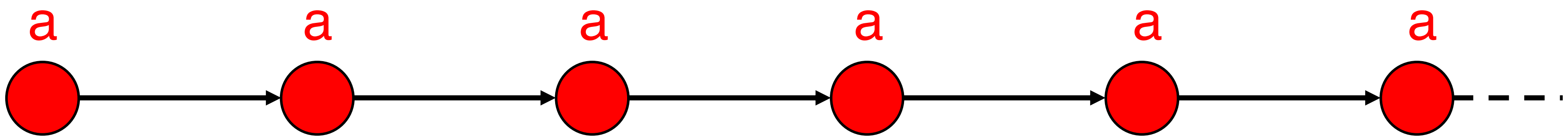
$\sigma \models a$



$\sigma \models \bigcirc a$



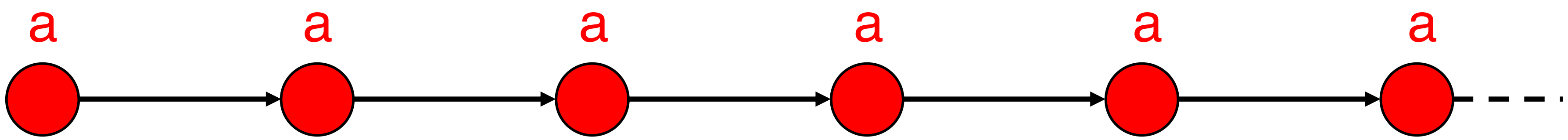
$\sigma \models \Box a$



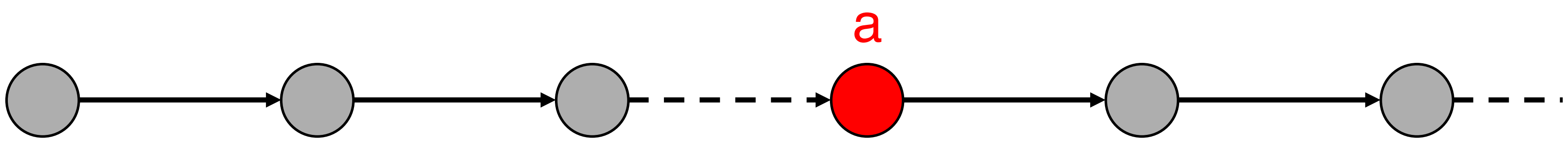
DERIVED TEMPORAL OPERATORS

$\Box\ \phi$	$\Diamond\ \phi \equiv \neg\Box\neg\phi$	$\Diamond\Box\ \phi$	$\Box\Diamond\ \phi \equiv \neg\Diamond\Box\neg\phi$
(always)	(eventually)	(persistence)	(infinitely many)

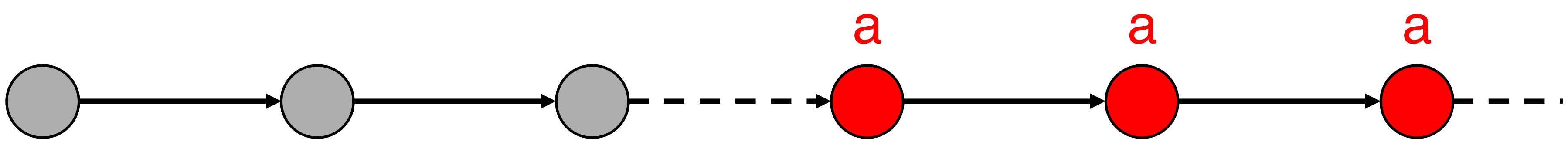
$\sigma \models \Box\ a$



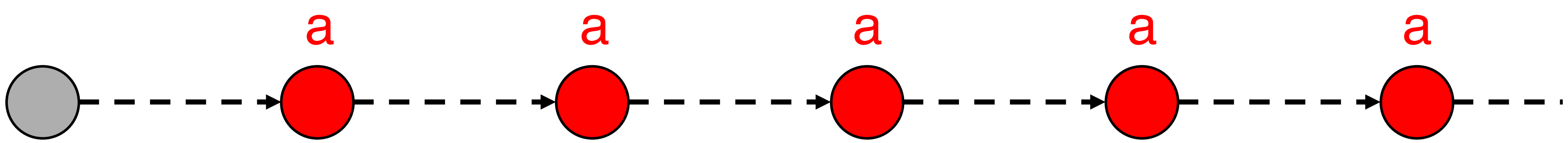
$\sigma \models \Diamond\ a$



$\sigma \models \Diamond\Box\ a$



$\sigma \models \Box\Diamond\ a$



EXAMPLE OF TEMPORAL PROPERTIES

- **Safety**
  - mutual exclusion : $\Box\neg(crit_1 \wedge crit_2)$
  - elevator : $\Box(moving \Rightarrow doors_{closed})$
  - traffic light : $\Box(yellow \Rightarrow \bigcirc red)$

- **Liveness**
  - progress : $\Diamond\ progress$
  - response : $\Box(try\_to\_send \Rightarrow \Diamond\ delivered)$
  - termination : $\Diamond\Box\ terminated$

EXAMPLE OF TEMPORAL PROPERTIES

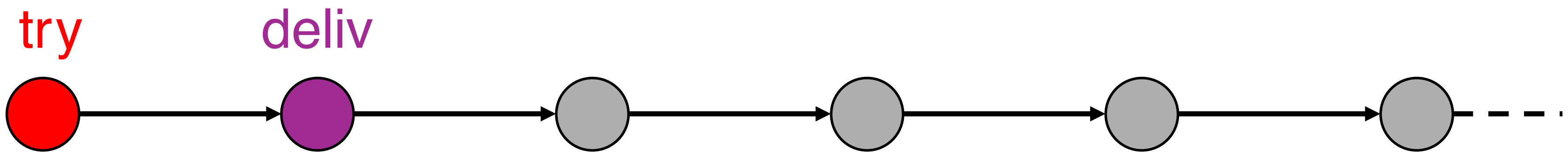
- **Safety** — nuclear plant
  - cooling : $\Box\neg(temp_{high} \land cooling_{low})$
  - alarm : $\Box(temp_{high} \Rightarrow alarm)$
  - saving : $\Box(temp_{high} \Rightarrow \bigcirc react_{low})$

- **Liveness** — nuclear plant
  - reactivity : $\Box\Diamond\, react_{high}$
  - temperature : $\Box(react_{low} \Rightarrow \Diamond temp_{low})$

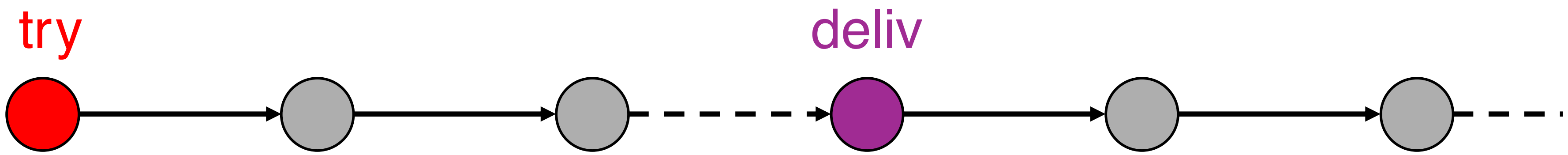
UNTIL OPERATOR

$$\phi \quad ::= \quad true \mid a \mid \phi \wedge \phi \mid \neg\,\phi \mid \bigcirc\,\phi \mid \Box\,\phi \mid \phi \bigcup \phi$$

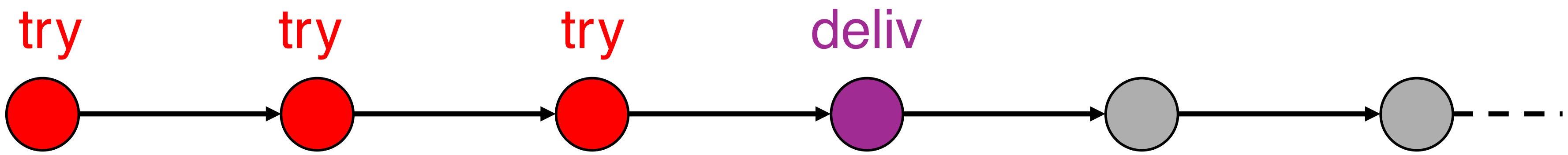
$try \Rightarrow \bigcirc\, deliv$



$try \Rightarrow \Diamond\, deliv$



$try \bigcup deliv$

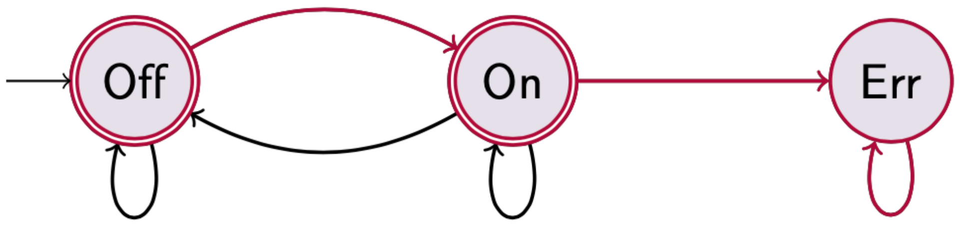


$\Diamond\,\phi \equiv true \cup \phi$ and $\Box\,\phi \equiv \neg\Diamond\neg\phi$

OUTLINE

Back to the outline - Back to the begin

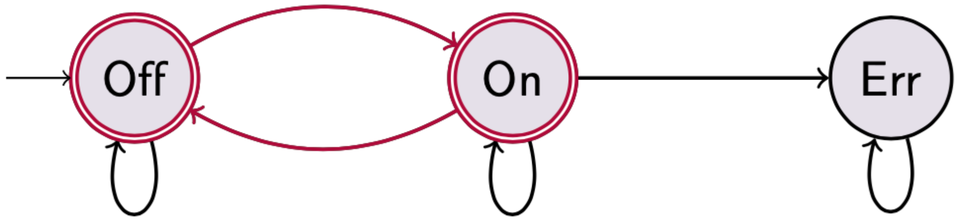
PROPERTIES OF A TRACE



have a path $\pi = \text{Off On Err Err Err} \ldots = \text{Off On Err}^\omega$

- $\pi \models \text{Off}$ but $\pi \not\models \text{On}$ so $\pi \models \neg\text{On}$
- $\pi \models \bigcirc \text{On}$
- $\pi \models \bigcirc\bigcirc \text{Err}$
- $\pi \models (\text{Off} \vee \text{On}) \cup \text{Err}$
- $\pi \models \Box(\text{Err} \Rightarrow \bigcirc \text{Err})$
- $\pi \models \Box(\text{Err} \Rightarrow \Box \text{Err})$
- $\pi \models \Diamond\Box \text{Err}$ (persistence)
- $\pi \models \bigcirc\bigcirc\Box \text{Err}$

PROPERTIES OF A TRACE



have a path $\pi = \text{Off On Off On Off} \; \ldots = (\text{Off On})^\omega$

- $\pi \not\models (\text{Off} \vee \text{On}) \; \mathcal{U} \; \text{Err}$
- $\pi \models \Diamond \text{Err} \Rightarrow ((\text{Off} \vee \text{On}) \; \mathcal{U} \; \text{Err})$ as $\pi \not\models \Diamond \text{Err}$
- $\pi \models \Box(\text{On} \vee \text{Off})$
- $\pi \models \Box\Diamond \text{On} \wedge \Box\Diamond \text{Off}$ (infinitely many)
- $\pi \not\models \Diamond\Box \text{On} \vee \Diamond\Box \text{Off}$ (persistence)
- $\pi \models \Box (\text{Off} \Rightarrow \bigcirc \text{On}) \wedge \Box (\text{On} \Rightarrow \bigcirc \text{Off})$

OUTLINE

Back to the outline - Back to the begin

LINEAR TIME PROPERTY

- Linear-Time properties specify the **admissible** behaviour of the system under consideration
  - LT-property specifies the traces that a TS can exhibit

Formal definition

- A Linear Time Property P over AP is a subset of $(2^{AP})^\omega$
- TS **satisfies** P (over AP):
  - $TS \vDash P$ if and only if $Traces(TS) \subseteq P \subseteq (2^{AP})^\omega$

- We will use the **Linear Time Logic** (**LTL**) to formalize P

LTL SEMANTICS (RECALL)

- $\phi \quad ::= \quad true \mid a \mid \phi_1 \ \wedge \ \phi_2 \mid \neg\, \phi \mid \bigcirc\, \phi \mid \Box\, \phi \mid \Diamond\, \phi \mid \phi_1 \ \bigcup \ \phi_2$
- for $\sigma = A_0 A_1 A_2 \cdots \in (2^{AP})^\omega$:

$\sigma \models true$		
$\sigma \models a$	iff	$a \in A_0$
$\sigma \models \phi_1 \wedge \phi_2$	iff	$\sigma \models \phi_1$ and $\sigma \models \phi_2$
$\sigma \models \neg\, \phi$	iff	$\sigma \not\models \phi$
$\sigma \models \bigcirc\, \phi$	iff	$A_1 A_2 A_3 \cdots \models \phi$
$\sigma \models \Box\, \phi$	iff	$\forall\, i \geq 0,\ A_i A_{i+1} A_{i+2} \cdots \models \phi$
$\sigma \models \Diamond\, \phi$	iff	$\exists\, i \geq 0,\ A_i A_{i+1} A_{i+2} \cdots \models \phi$
$\sigma \models \phi_1 \bigcup \phi_2$	iff	$\exists\, j \geq 0,\ A_j A_{j+1} A_{j+2} \cdots \models \phi_2$ and $\forall\, 0 \leq i < j,\ A_i A_{i+1} A_{i+2} \cdots \models \phi_1$

HOW TO SPECIFY MUTUAL EXCLUSION?

Mutual Exclusion

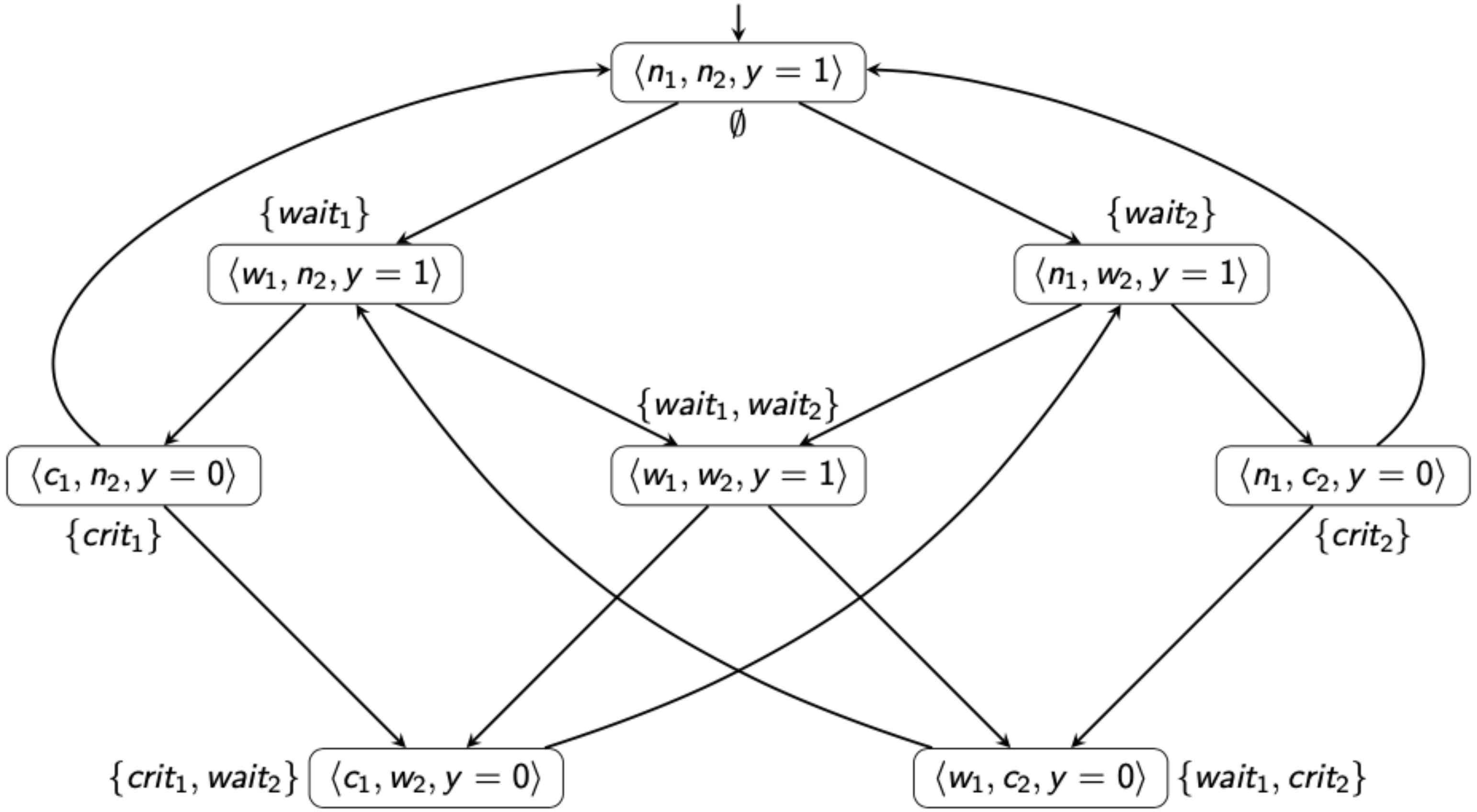
There is at most one process in the critical section

- Let $AP = \{crit_1, crit_2\}$
  - other atomic propositions are not of any relevance for this property
- LTL formalization of the LT property

  $$P_{mutex} = \Box \neg (crit_1 \wedge crit_2)$$

- **Does the semaphore-based algorithm satisfy P_{mutex} ?**

DOES SEMAPHORE-BASED ALGORITHM SATISFY P_{MUTEX}?



YES! as there is no reachable state labeled with $\{crit_1, crit_2\}$

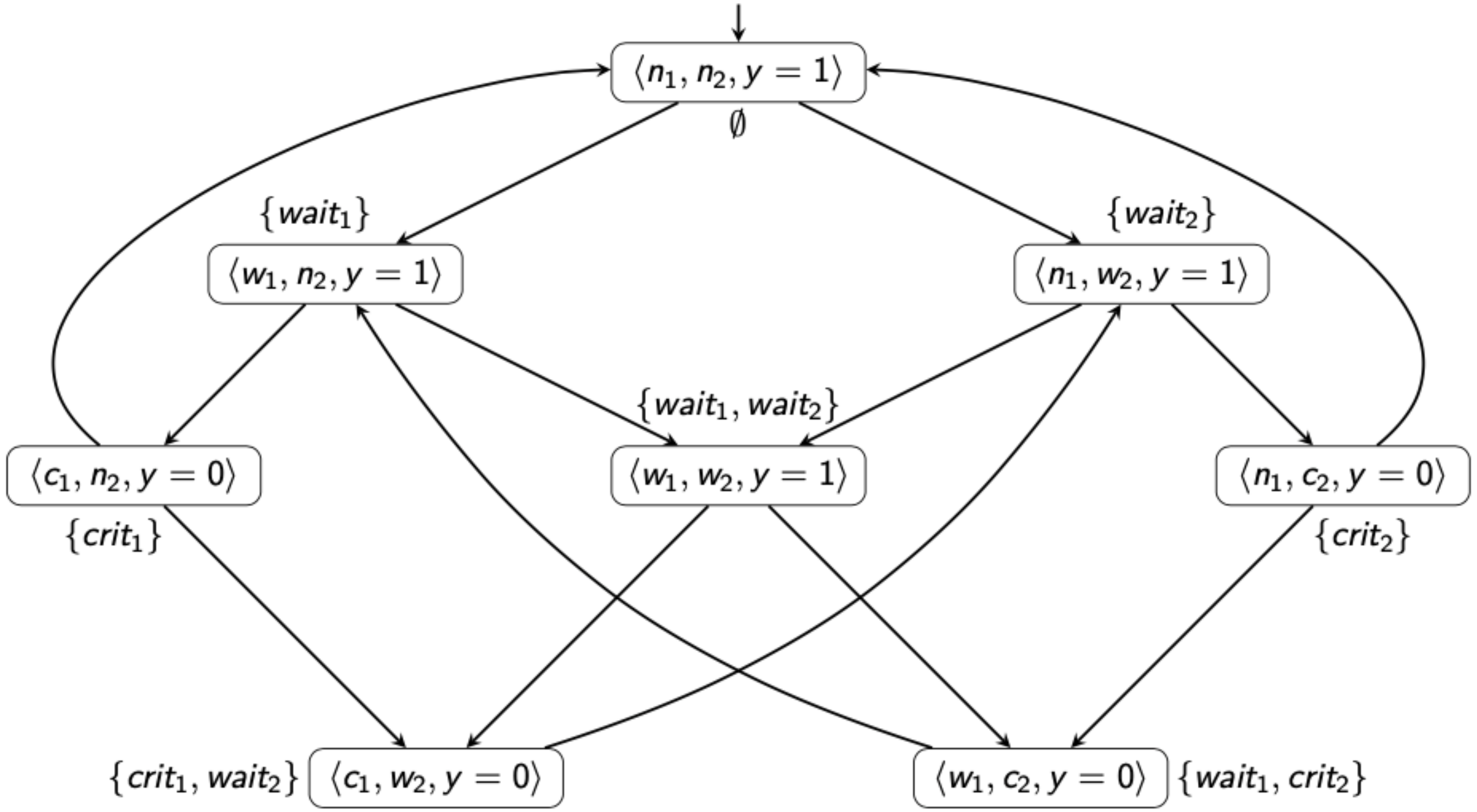
HOW TO SPECIFY STARVATION FREEDOM?

Starvation Freedom

A process that wants to enter the critical section
is eventually able to do so

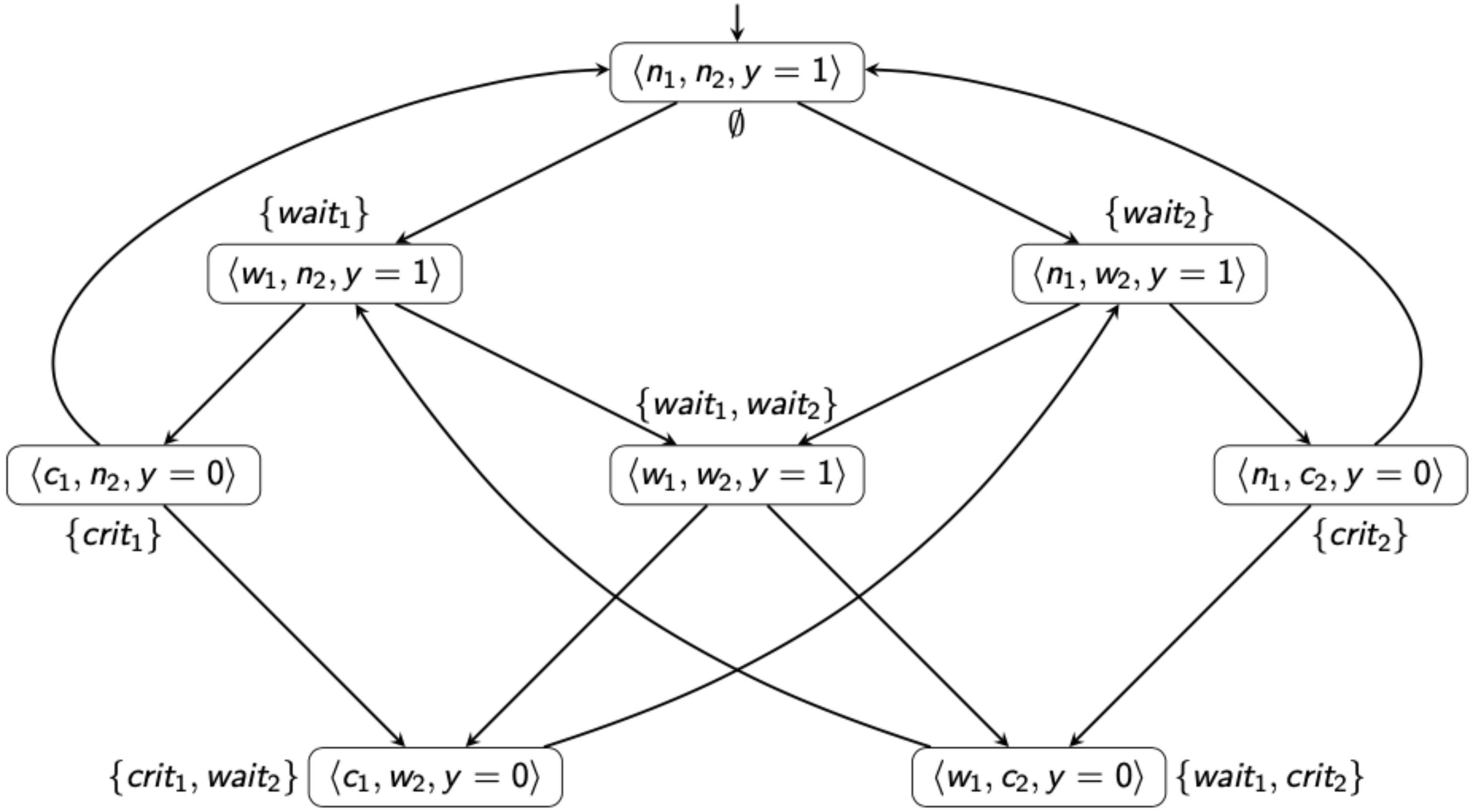
- Let $AP = \{wait_1, crit_1, wait_2, crit_2\}$
- LTL formalization of the LT property
  $P_{nostarve} = \Box\,(wait_1 \Rightarrow \Diamond\, crit_1) \wedge \Box\,(wait_2 \Rightarrow \Diamond\, crit_2)$
- **Does the semaphore-based algorithm satisfy $P_{nostarve}$?**

DOES SEMAPHORE-BASED ALGORITHM SATISFY $P_{NOSTARVE}$?



NO! process one or process two may starve!

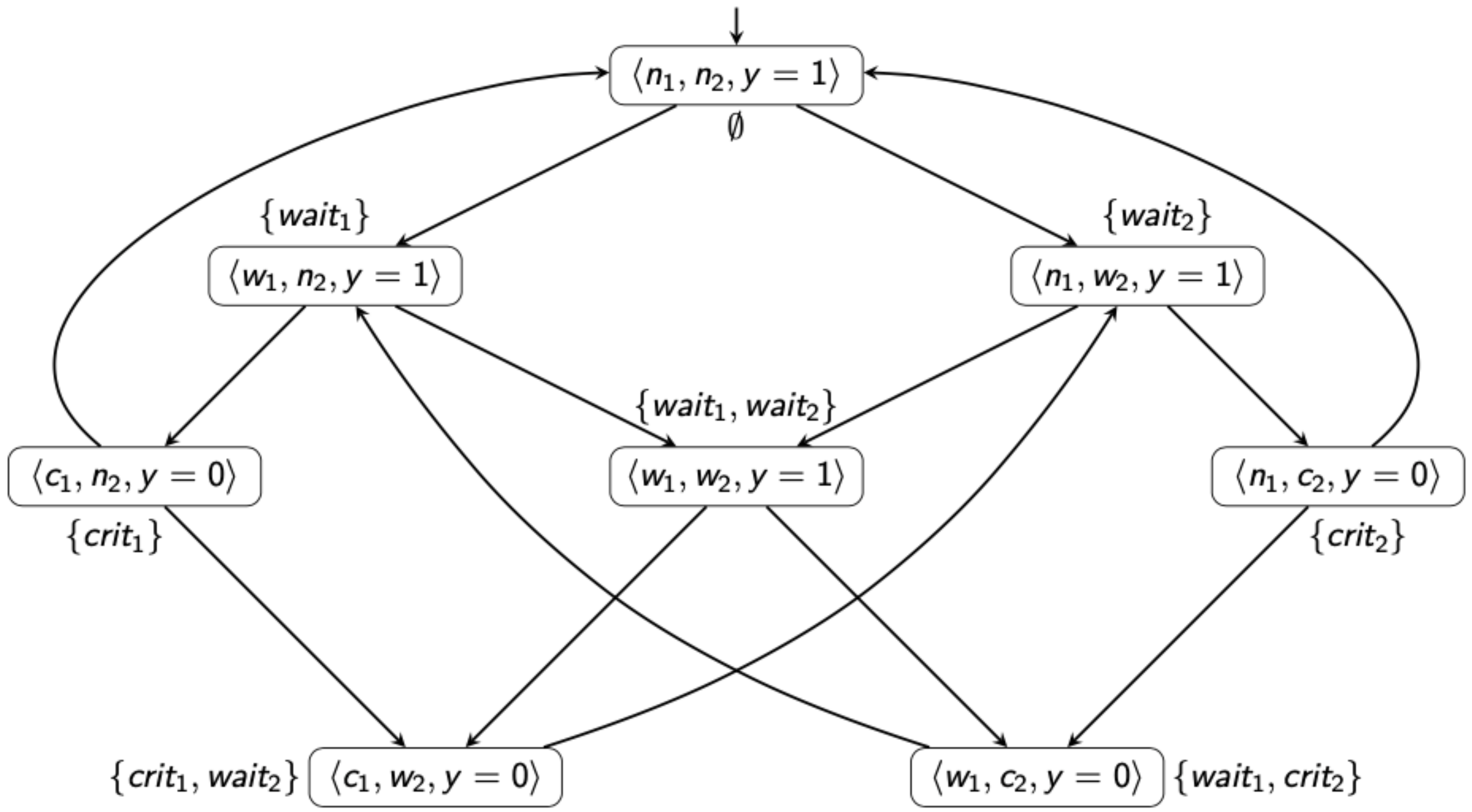
PROCESS ONE STARVES



let $\sigma = \emptyset(\{wait_1\}\{wait_1, wait_2\}\{wait_1, crit_2\})^\omega \in Traces(TS)$

but $\sigma \models \Diamond(wait_1 \wedge \Box\neg\, crit_1) \Rightarrow \sigma \notin P_{nostarve}$

PROCESS TWO STARVES



let $\sigma = \emptyset(\{wait_2\}\{wait_1, wait_2\}\{crit_1, wait_2\})^\omega \in Traces(TS)$

but $\sigma \vDash \Diamond(wait_2 \wedge \Box\neg\, crit_2) \Rightarrow \sigma \notin P_{nostarve}$

INVARIANTS

- Typical safety property: mutual exclusion property
  - the **bad thing** (having > 1 process in the critical section) **never occurs**
- Another typical safety property verifies variable bounds (overflow)

These properties are **Invariants**

- An **Invariant** is an LT property
  - that is given by a **condition** ϕ over AP
  - requires that **condition** ϕ holds **for all states** (reachable ones)
  - e.g. for mutual exclusion property $\phi = \neg(crit_1 \wedge crit_2)$

FORMAL DEFINITION

- An LT property P_{inv} over AP is an **Invariant** if there is a **pure propositional** formula ϕ over AP such that:

$$P_{inv} = \square\, \phi$$

  - ϕ is called an **invariant condition** of P_{inv}

- Note that:
$TS \models P_{inv}$ if and only if $\forall s \in Reach(TS),\ \mathcal{L}(s) \models_{prop} \phi$

- ϕ has to be fulfilled by all initial states and satisfaction of ϕ is invariant under all transitions in the reachable fragment of TS

SAFETY PROPERTIES

- Safety properties: **“nothing bad should happen”**
  - an Invariant property is a **particular** safety property
- Safety properties may impose requirements on **finite** path fragments and cannot be verified by only considering the reachable states
- A safety property which is not an invariant
  - consider a cash dispenser
  - property “money can only be withdrawn once a correct PIN has been provided”
  - **not an invariant, since it is not a state property**
- a typical LTL example: **Bounded Response**

$$\Box(request \Rightarrow \bigvee_{i=n}^{m} \bigcirc^{i} response)$$

LIVENESS PROPERTIES

- Safety properties specify that “something bad never happens”
- Doing nothing easily fulfills a safety property
  - as this will never lead to a “bad” situation
- Safety properties are complemented by **Liveness** properties
  - that require some progress
  - that assert: **”something good” will happen eventually**
- a typical LTL example: $\Diamond\ \phi$

EXAMPLES OF LIVENESS

- Back to our semaphore-based algorithm with

$$AP = \{wait_1, crit_1, wait_2, crit_2\}$$

- Eventually

$$\Diamond\, crit_1 \wedge \Diamond\, crit_2$$

- Repeated eventually

$$\Box\Diamond\, crit_1 \wedge \Box\Diamond\, crit_2$$

- Starvation freedom

$$\Box\,(wait_1 \Rightarrow \Diamond\, crit_1) \wedge \Box\,(wait_2 \Rightarrow \Diamond\, crit_2)$$

THANK YOU

PDF version of the slides

Back to the begin - Back to the outline