COL:750

Foundations of Automatic Verification

Instructor: Priyanka Golia

Course Webpage

https://priyanka-golia.github.io/teaching/COL-750/index.html

LTL Syntax

F = True

= p (atomic proposition)

$= F_1 \wedge F_2, F_1 \vee F_2, F_1 \rightarrow F_2, F_1 \leftrightarrow F_2$

$= \neg F_1$

= **N** F_1 **N** is "Next". F_1 is True at next step. Often represented as **O**, **X** .

= F_1 **U** F_2 **U** is "Until". F_2 is True at "some point, say t", and until then F_1 is True.
At "t", F_1 doesn't have to hold any more!

LTL Syntax

F = $\mathbf{N}\ F_1$ **N** is "Next". F_1 is True at next step. Often represented as **O**, **X** .

If you press the accelerator, the car will move in the next step.

$$accelerate \rightarrow \mathbf{N}\ moving$$

If you shoot the ball, the result will be known in the next step.

$$shoot \rightarrow \mathbf{N}\ (goal \lor miss)$$

F = $F_1\ \mathbf{U}\ F_2$ **U** is "Until". F_2 is True at "some point", and until then F_1 is True.

Mario will keep jumping until he lands.

$$jumping\ \mathbf{U}\ landed$$

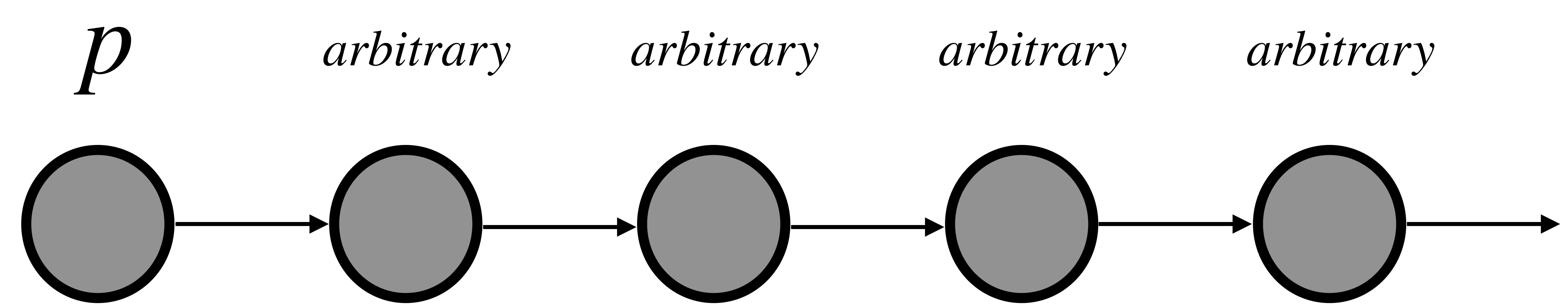
The emergency light will stay on until the power comes back.

$$EmergencyLight\ \mathbf{U}\ PowerRestored$$

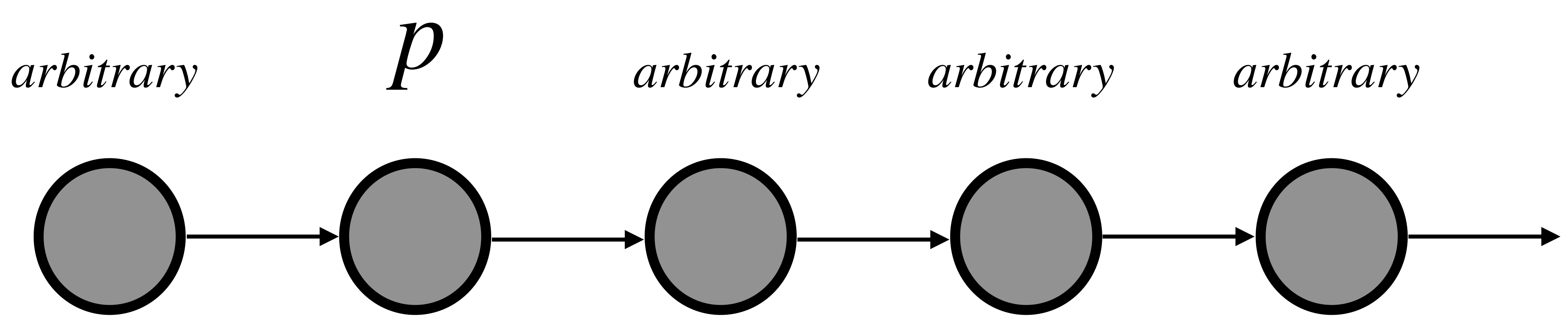
LTL Syntax

Sequence of states (paths).

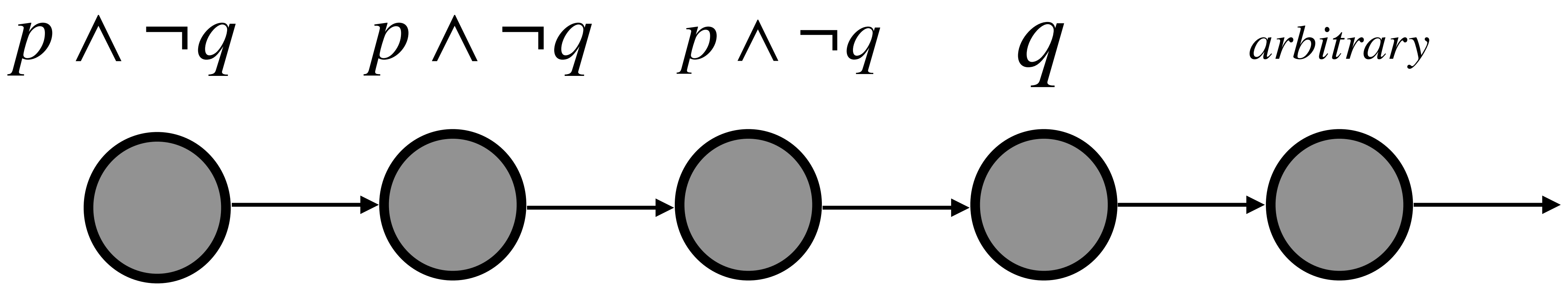
Atomic prop. P



N p



p **U** q

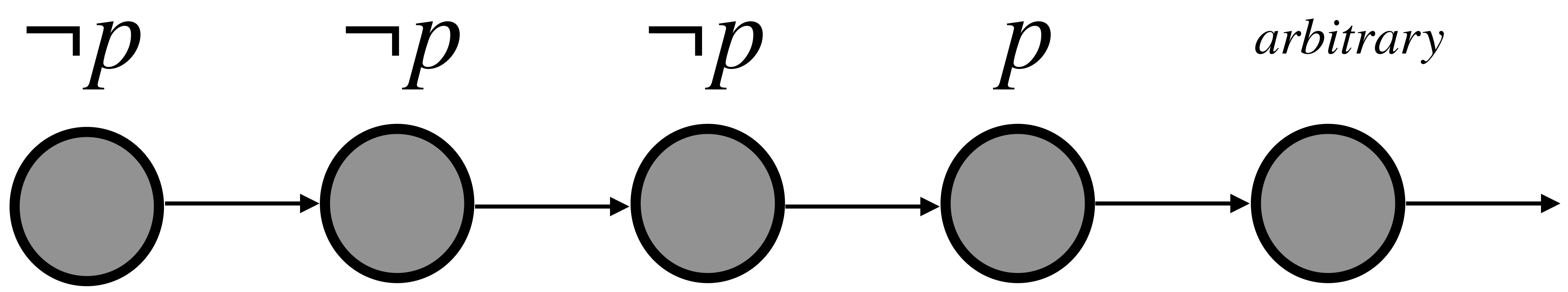


LTL Syntax

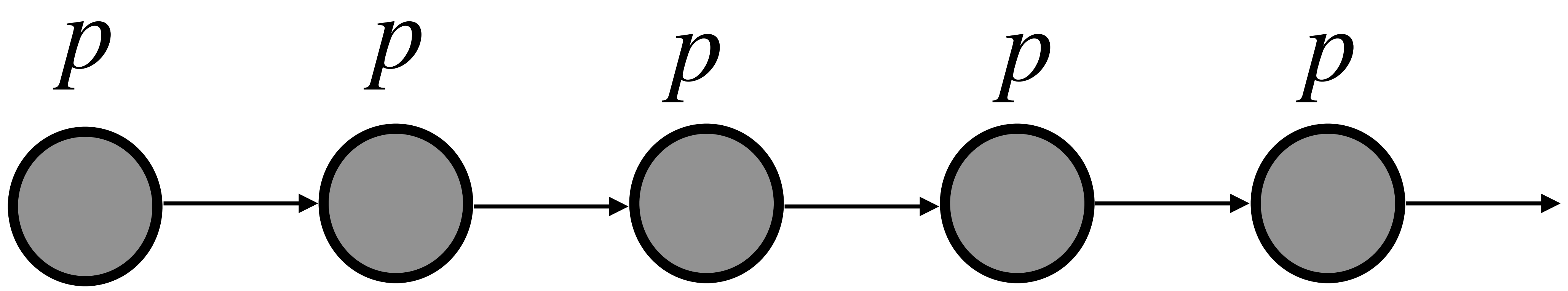
Primary temporal operators: **N** **U**

Eventually $\Diamond$ F F will become true at some point in the future.

$\Diamond F \equiv True\ \mathbf{U}\ F$



Always (valid) $\Box F$ F is always True.



$\Box F \equiv \neg \Diamond \neg F$ (Never (Eventually ($\neg F$))).

LTL Syntax

Primary temporal operators: **N U**

Weak Until $- F_1$ **W** F_2, F_1 must remain true until F_2 becomes true, but F_2 doesn't necessarily need to become true at any point.

$F_1 \mathbf{W} F_2 \equiv (F_1 \mathbf{U} F_2) \vee (\Box F_1)$

It is considered weaker version of **U**, which requires F_2 to eventuallyTrue.

System is in safe mode **W** system is ready

LTL Syntax

Primary temporal operators: **N U**

Release $-F_1$ **R** F_2, F_2 must remain true until and including the point where F_1 first becomes true, but F_1 doesn't necessarily need to become true at any point.

$$F_1 \mathbf{R}\, F_2 \equiv ((F_2 \wedge \neg F_1)\ \mathbf{W}\ (F_2 \wedge F_1))$$

LTL: Operator Precedence

How to read $\mathbf{N}\ p\ \mathbf{U}\ q$?

$\neg, \Diamond, \Box, \mathbf{N}$ Binds stronger than $\mathbf{U}, \wedge, \vee, \rightarrow, \leftrightarrow$

$\mathbf{N}\ p\ \mathbf{U}\ q \equiv ((\mathbf{N}\ p)\ \mathbf{U}\ q)$ The next state must satisfy p, and p must hold until q happens

$\Box\ p \vee q \equiv ((\Box\ p) \vee q)$ Either p always holds or q must hold in the current state.

Binds from right to left: $\neg \mathbf{N} p \equiv \neg(\mathbf{N} p)$

$\mathbf{U}$ Binds stronger than $\wedge, \vee, \rightarrow, \leftrightarrow$ $p \mathbf{U}\ q \vee r \equiv (p \mathbf{U}\ q) \vee r$

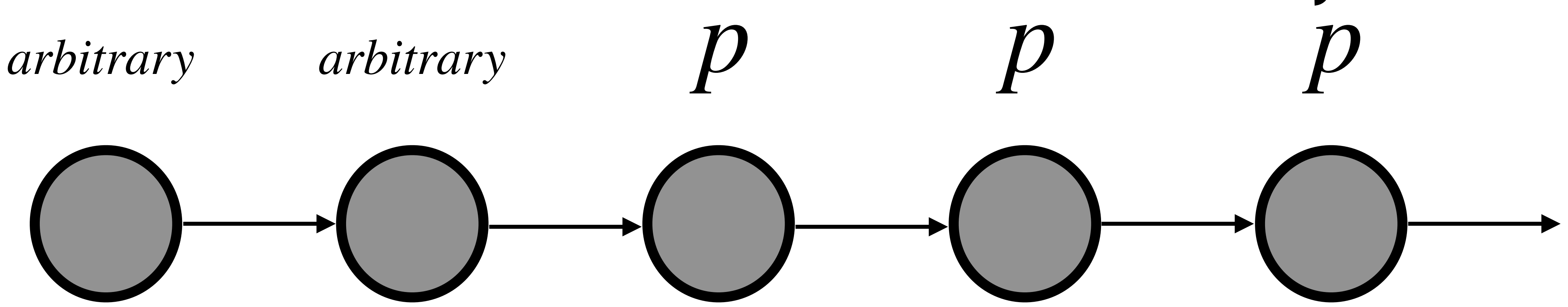
LTL: Common Cases

Response — If p then eventually q. $p \to \Diamond q$

Precedence — If p then q until r. $p \to q\ \mathbf{U}\ r \quad \equiv p \to (q\ \mathbf{U}\ r)$

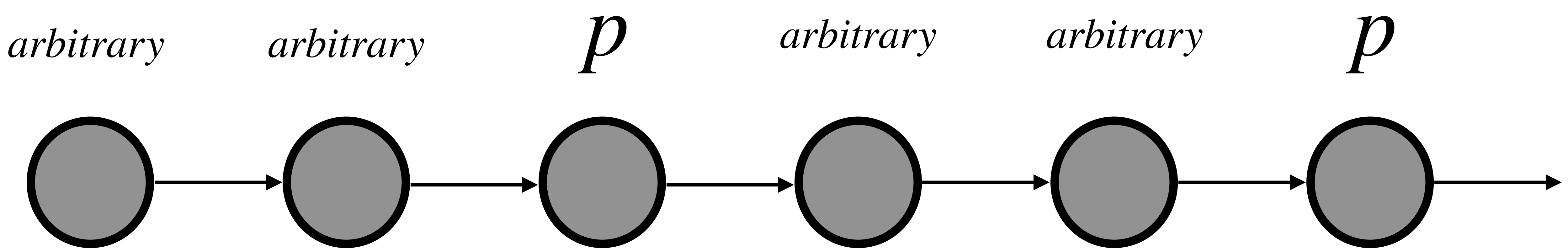
Stability — Once we reach the stable state, we will always be in stable state.

$\Diamond \Box p$



Progress — We will always reach the stable state or desired state.

$\Box \Diamond p$



Correlation — Eventually p implies eventually q. $\Diamond p \to \Diamond q$

LTL: Formulas

Duality Law

$\neg\, \mathbf{N}\, p \equiv \mathbf{N}\, \neg p$ $\qquad \neg \Diamond p \equiv \Box \neg p$ $\qquad \neg \Box p \equiv \Diamond \neg p$

Absorption Law

$\Diamond \Box \Diamond P \equiv \Box \Diamond p$ $\qquad \Box \Diamond \Box P \equiv \Diamond \Box p$

Distributive Law

$\mathbf{N}(p\ \mathbf{U}\ q) \equiv ((\mathbf{N}\, p)\ \mathbf{U}\ (\mathbf{N}\, q))$ $\qquad \Diamond(p \vee q) \equiv \Diamond p \vee \Diamond q$

$\Diamond(p \wedge q) \not\equiv \Diamond p \wedge \Diamond q$ $\qquad \Box(p \wedge q) \equiv \Box p \wedge \Box q$

Expansion Law

$p\ \mathbf{U}\ q \equiv q \vee (p \wedge (\mathbf{N}\, (p\ \mathbf{U}\ q)))$ $\qquad \Box p \equiv p \wedge (\mathbf{N}\, (\Box p))$

$\Diamond p \equiv p \vee (\mathbf{N}\, (\Diamond p))$

LTL: Examples

Traffic light is green infinitely often. $\square \Diamond\, green$

Once red, the light can't become green immediately. $\square (\, red \;\rightarrow \neg \mathbf{N}\; green)$

Once red, the light always becomes green eventually after being yellow for some time.

$\square (red \rightarrow (\Diamond\; green \;\wedge (\neg green\; \mathbf{U}\; yellow)))$ $\square (red \rightarrow \mathbf{N}\; (red\; \mathbf{U}\; (yellow \;\wedge\; \mathbf{N}\; (yellow\; \mathbf{U}\; green))))$

If an intruder is detected, then an alert must be raised at the 3 step.

$$\square (IntruderDettected \rightarrow (\;\mathbf{N}\; \neg alert \wedge \;\mathbf{N}\;\; \mathbf{N}\; \neg alert \wedge \;\mathbf{N}\;\; \mathbf{N}\;\; \mathbf{N}\; alert))$$

A robot must keep moving until it reaches the charging station, and once charged, it must always eventually move again.

$$(Move\; \mathbf{U}\; AtChargeStation) \wedge \square (Charged \rightarrow \Diamond Move)$$

LTL: Examples

If an intruder is detected, then an alert must be raised at the 3 step.

$$\square(IntruderDettected \rightarrow (\ \mathbf{N}\, \neg alert \wedge\ \mathbf{N}\ \ \mathbf{N}\, \neg alert \wedge\ \mathbf{N}\ \ \mathbf{N}\ \ \mathbf{N}\, alert))$$

A robot must keep moving until it reaches the charging station, and once charged, it must always eventually move again.

$$(Move\ \mathbf{U}\ AtChargeStation) \wedge \square(Charged \rightarrow \lozenge Move)$$

LTL: Semantics

We interpret our temporal formulae in a discrete, linear model of time.

$M = < N, I >$, where N is a set of Natural number and $I : N \mapsto 2^{\Sigma}$

I maps each Natural number (representing a moment in time) to a set of propositions

Let $\pi = a_0, a_1, a_2, \ldots$ $\quad$ $\pi(i) = a_i$ $\quad$ AP at i^{th} level.

$\pi^i = a_i, a_{i+1}, a_{i+2}, \ldots$ $\quad$ Suffix of π

LTL: Semantics

Semantics with respect to a given Trace (or Path) π

Let $\pi = a_0, a_1, a_2, \ldots$ $\pi(i) = a_i$ AP at i^{th} level. $\pi^i = a_i, a_{i+1}, a_{i+2}, \ldots$ Suffix of π

$\pi \vDash p$ Iff $p \in \pi(0)$ $\pi^i \vDash p$ Iff $p \in \pi(i)$

$\pi \vDash \mathbf{N}\ F_1$ Iff $\pi^1 \vDash F_1$ $\pi^i \vDash \mathbf{N}\ F$ Iff $\pi^{i+1} \vDash F_1$

$\pi \vDash F_1\ \mathbf{U}\ F_2$ Iff $\exists j \geq 0,\ \pi^j \vDash F_2$, and $\pi^i \vDash F_1$ for all $0 \leq i < j$

$\pi \vDash \Diamond\ F_1$ Iff $\exists j \geq 0,\ \pi^j \vDash F_1$

$\pi \vDash \Box\ F_1$ Iff $\forall j \geq 0,\ \pi^j \vDash F_1$

$\pi \vDash \Box \Diamond\ F_1$ Iff $\exists^\infty j \geq 0,\ \pi^j \vDash F_1$ $\exists^\infty = \forall i \geq 0, \exists j \geq i$

$\pi \vDash \Diamond \Box\ F_1$ Iff $\forall^\infty j \geq 0,\ \pi^j \vDash F_1$ $\exists^\infty = \exists i \geq 0, \forall j \geq i$

LTL: Semantics Kripke Structure

AP — is a set of atomic propositions (Boolean valued variables, predicates)

Kripke structure over AP as a 4-tuple $M = (S, I, R, L)$

S = a finite set of states.

I = a set of initial states $I \subseteq S$

R = a transition relation $R \subseteq S \times S$

L = a labelling function $L : S \rightarrow 2^{AP}$

LTL: Semantics Kripke Structure

Kripke structure over AP as a 4-tuple $M = (S, I, R, L)$

S = a finite set of states. $S = \{s_1, s_2, s_3\}$

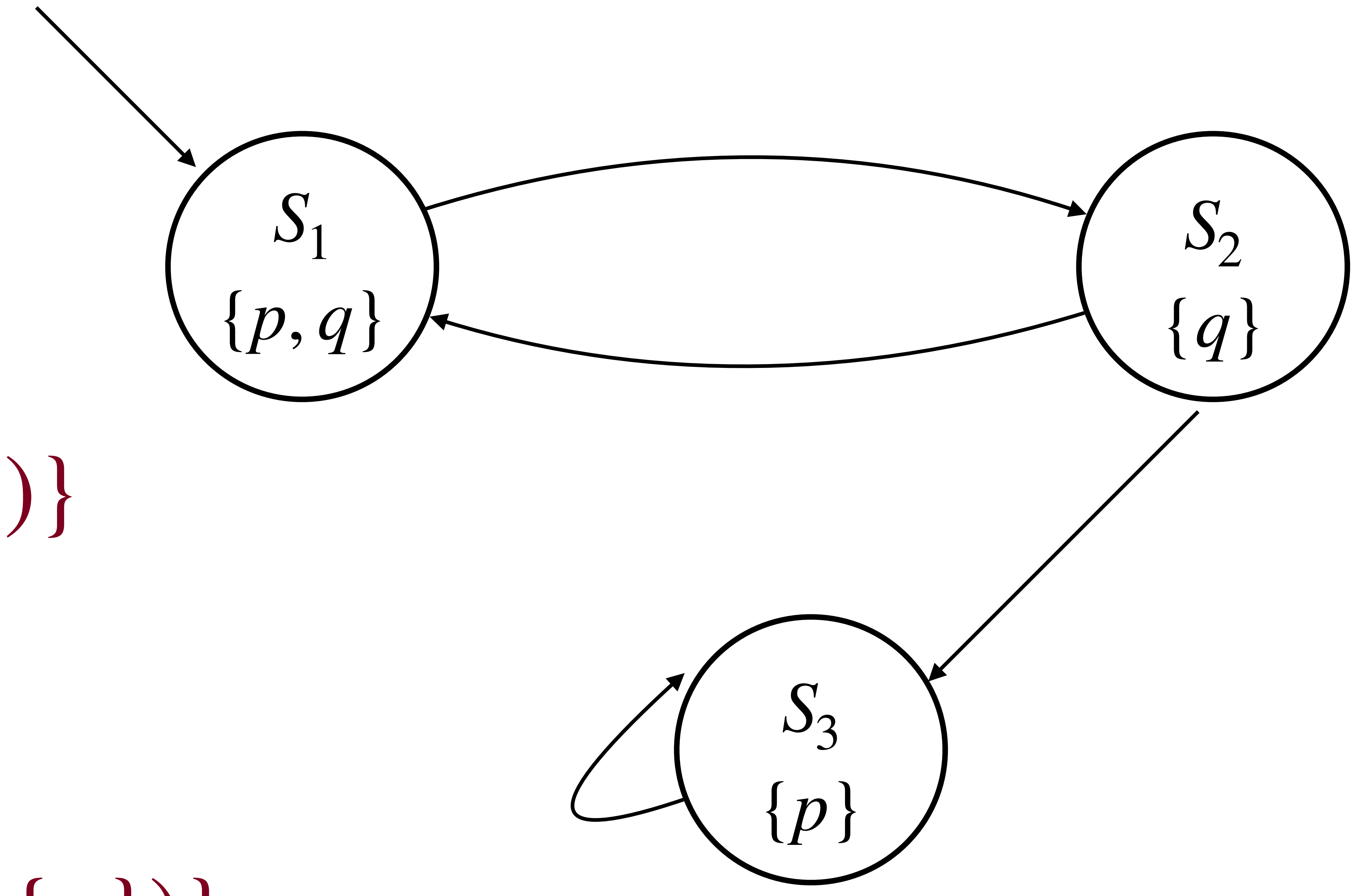
I = a set of initial states $I \subseteq S$ $I = \{s_1\}$

R = a transition relation $R \subseteq S \times S$

$$R = \{(s_1, s_2), (s_2, s_1), (s_2, s_3), (s_3, s_3)\}$$

L = a labelling function $L : S \to 2^{AP}$

$$L = \{(s_1, \{p, q\}), (s_2, \{q\}), (s_3, \{p\})\}$$



AP = $\{p, q\}$

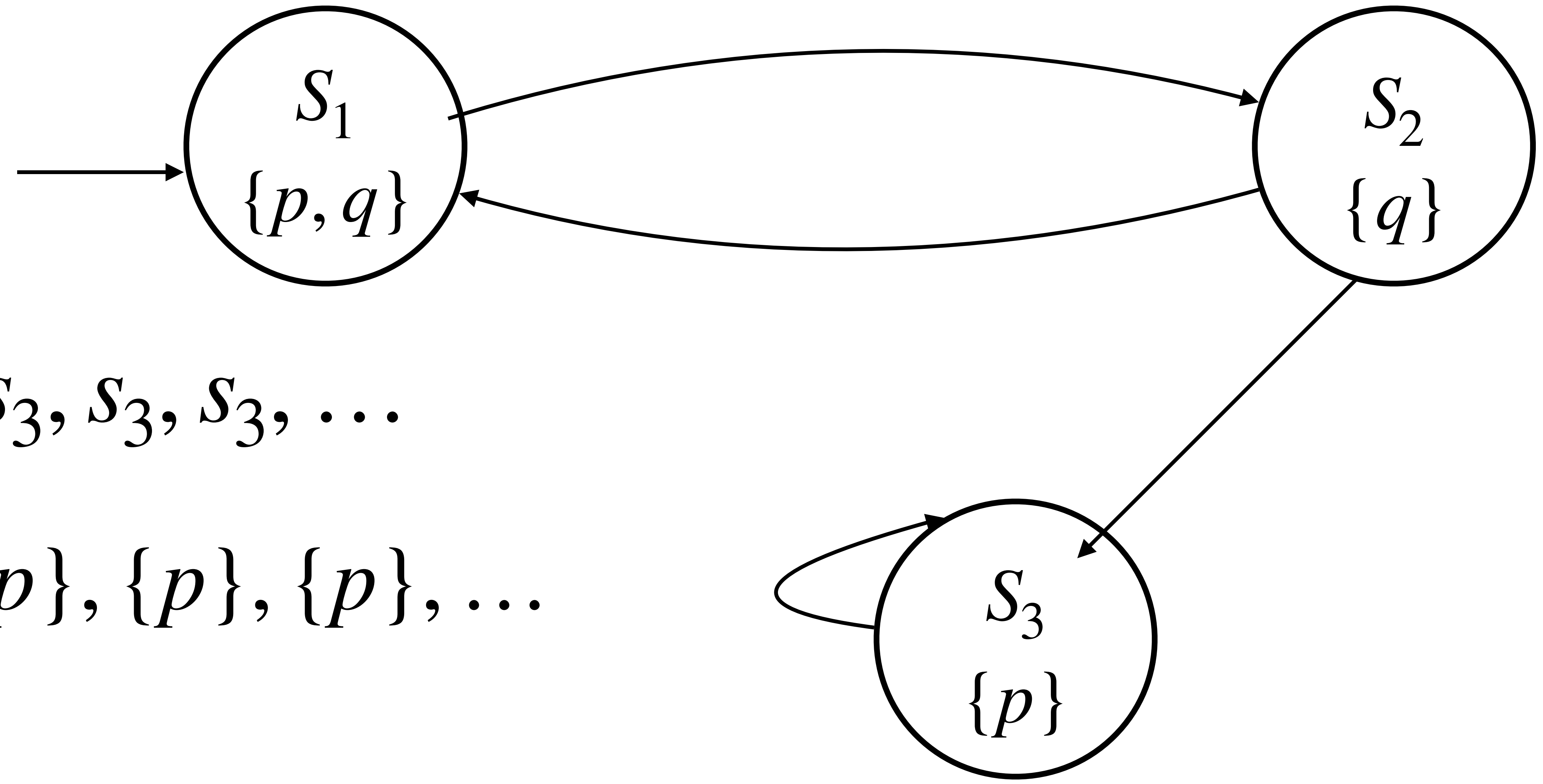
LTL: Semantics

Kripke Structure

Kripke structure over AP as a 4-tuple $M = (S, I, R, L)$ AP $= \{p, q\}$

$S = \{s_1, s_2, s_3\}$ $I = \{s_1\}$ $R = \{(s_1, s_2), (s_2, s_1), (s_2, s_3), (s_3, s_3)\}$

$L = \{(s_1, \{p, q\}), (s_2, \{q\}), (s_3, \{p\})\}$



M may produce a path $w = s_1, s_2, s_1, s_2, s_3, s_3, s_3, s_3, \ldots$

π^{s_1} $\pi = \{p, q\}, \{q\}, \{p, q\}, \{q\}, \{p\}, \{p\}, \{p\}, \ldots$

M can produce words belonging to the language —

$(\{p, q\}\{q\})^*(\{p\})^\omega \cup (\{p, q\}\{q\})^\omega$

LTL: Semantics Kripke Structure

Given a kripke structure M and a path π in M, a state $s \in S$, and an LTL formula F:

1. $< M, \pi > \models F$ iff $\pi^{s_o} \models F$, where s_o is initial state of π
2. $< M, s > \models F$ iff $< M, \pi > \models F$ for paths starting at s_o.
3. $< M > \models F$. iff $< M, s_o > \models F$ for every $s_o \in I$, where I initial states of M.

LTL: Semantics

A formula F is satisfiable if there exists at least one Kripke Structure M, and at least one initial state s_o such that:

$$< M, s_o > \vDash F$$

A formula F is valid if for all Kripke Structures M, and for all initial states s_o:

$$< M, s_o > \vDash F$$

LTL model checking — Given formula F, and Kripke Structure M checks if $< M, s_o > \vDash F$ holds for every initial state $s_o \in I$

Course Webpage

Thanks!