

# Software Testing and Validation

A.A. 2023/2024

Corso di Laurea in Informatica

## Logics in Model Checking

Igor Melatti

Università degli Studi dell'Aquila

Dipartimento di Ingegneria e Scienze dell'Informazione e Matematica



UNIVERSITÀ  
DEGLI STUDI  
DELL'AQUILA



DISIM  
Dipartimento di Ingegneria  
e Scienze dell'Informazione  
e Matematica

# Beyond Invariants

- Invariants represent a huge share of properties to be verified on a system
- For many systems, one may be happy with invariants only
  - “nothing bad happens”, that’s all folks
- However, it is not always sufficient: a non-running system of course satisfies invariants
  - no starting states, thus no reachable states...



UNIVERSITÀ  
DEGLI STUDI  
DELL'AQUILA



DISIM  
Dipartimento di Ingegneria  
e Scienze dell'Informazione  
e Matematica

# Safety vs. Liveness

- **Safety** properties: something bad must never happen
  - example: in the Peterson's protocol, it must not happen that both processes are accessing the resource (L3 in the Murphi model)
- Invariants are a special case of safety properties
  - there are some safety properties which are not invariants
  - however, they can be expressed with invariants by adding variables to the Kripke Structure
  - in the following, we will consider "invariants" and "safety properties" as synonyms
- **Liveness** properties: something good will eventually happen
  - example: in the Peterson's protocol, both processes will eventually access the resource
  - not at the same time!
  - cannot be expressed with invariants



UNIVERSITÀ  
DEGLI STUDI  
DELL'AQUILA



DISIM  
Dipartimento di Ingegneria  
e Scienze dell'Informazione  
e Matematica

# Safety vs. Liveness

- Notation: let  $\mathcal{S}$  be a KS and  $\varphi$  be a formula in any logic
- $\mathcal{S} \models \varphi$  is true iff  $\varphi$  is true in  $\mathcal{S}$ 
  - what this means depends on the logic, as we will see
- For most properties  $\varphi$ , if  $\mathcal{S} \not\models \varphi$  then there exists a path  $\pi \in \text{Path}(\mathcal{S})$  which is a *counterexample*
- For safety properties,  $|\pi| < \infty$ 
  - $\mathcal{S}$  arrives to an *unsafe* state and that's it
- For liveness properties,  $|\pi| = \infty$ 
  - since  $\mathcal{S}$  is finite, this implies that  $\pi$  contains a loop (*lasso*) in its final part



UNIVERSITÀ  
DEGLI STUDI  
DELL'AQUILA



DISIM  
Dipartimento di Ingegneria  
e Scienze dell'Informazione  
e Matematica

# Safety vs. Liveness

- Equivalent definition for a safety formula: given a finite counterexample, every extension still contains the error
- There is one formula which is both safety and liveness: the true invariant
  - it cannot have a counterexample...
- There are formulas which are neither safety nor liveness
  - their counterexample is not a path
- For typically used formulas, they are either safety or liveness properties



UNIVERSITÀ  
DEGLI STUDI  
DELL'AQUILA



DISIM  
Dipartimento di Ingegneria  
e Scienze dell'Informazione  
e Matematica

# Safety vs. Liveness: Mathematical Definition

- Let a *model*  $\sigma$  be an infinite sequence of truth assignments to all  $p \in AP$ 
  - $\sigma \in (2^{AP})^\omega$
  - could also be seen as a sequence of sets  $P \subseteq AP$
  - given a path  $\pi$  of a KS  $\mathcal{S}$ , we can always obtain a model from  $\pi$  by replacing each  $\pi(i)$  with  $L(\pi(i))$
- It is possible to define if  $\sigma \models \varphi$ , for a given formula  $\varphi$
- $\varphi$  is a safety property if, for all  $\sigma, \sigma'$  s.t.  $\sigma \not\models \varphi$  and  $\exists j : \sigma|_j = \sigma'|_j, \sigma' \models \varphi$
- $\varphi$  is a liveness property if, for each prefix  $w_0 \dots w_i$ , there exists  $\sigma$  s.t.  $\sigma|_i = w_0 \dots w_i$  and  $\sigma \models \varphi$



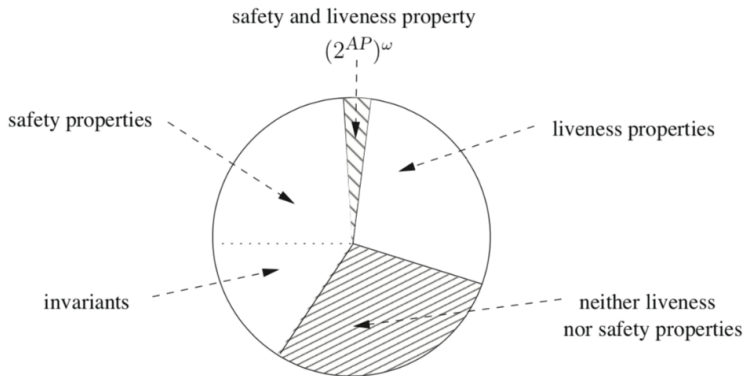
UNIVERSITÀ  
DEGLI STUDI  
DELL'AQUILA



DISIM  
Dipartimento di Ingegneria  
e Scienze dell'Informazione  
e Matematica

# Safety vs. Liveness: Mathematical Definition

If we identify a property by the set of its models ( $\varphi = \{\sigma \mid \sigma \models \varphi\}$ )



# Model Checking Logics: Preliminaries

- Model Checking logics are based on the concept of *execution* of a Kripke structure  $\mathcal{S}$ 
  - thus, on  $\pi \in \text{Path}$
- Often, paths are directly viewed as a sequence of atomic propositions, rather than states
  - from  $\pi = s_1, s_2, \dots$  to  $AP(\pi) = L(s_1), L(s_2), \dots$
- Focusing on executions allows to model *time*
  - property on paths, especially useful for liveness properties
  - time in the sense that we have something coming before of something else (in a path...)
- Trade-off between
  - logics expressiveness: interesting properties can be written
  - logics efficiency: there is an efficient model checking algorithm to compute if  $\mathcal{S} \models \varphi$



UNIVERSITÀ  
DEGLI STUDI  
DELL'AQUILA



DISIM  
Dipartimento di Ingegneria  
e Scienze dell'Informatica  
e Matematica



# Model Checking Logics: Preliminaries

- We will focus on the two leading Model Checking logics: LTL and CTL
  - with some hints on CTL\*
  - LTL (Linear-time Temporal Logic) established by Pnueli in 1977
  - CTL (Computation Tree Logic) established by Clarke and Emerson in 1981
  - used for IEEE standards:
    - PSL (Property Specification Language, IEEE Standard 1850)
    - SVA (SystemVerilog Assertions, IEEE Standard 1800).
- We will see syntax and semantics of both logics
  - syntax: how a valid formula is written
  - semantics: what a valid formula “means”
  - that is, when  $\mathcal{S} \models \varphi$  holds



UNIVERSITÀ  
DEGLI STUDI  
DELL'AQUILA



DISIM  
Dipartimento di Ingegneria  
e Scienze dell'Informazione  
e Matematica

# LTL Syntax

$$\Phi ::= p \mid \Phi_1 \wedge \Phi_2 \mid \neg \Phi \mid (\Phi) \mid \mathbf{X}\Phi \mid \Phi_1 \mathbf{U} \Phi_2$$

- Other derived operators:
  - of course true, false, OR and other propositional logic connectors
  - future (or eventually):  $\mathbf{F}\Phi = \text{true} \mathbf{U} \Phi$
  - globally:  $\mathbf{G}\Phi = \neg(\text{true} \mathbf{U} \neg\Phi)$
  - release:  $\Phi_1 \mathbf{R} \Phi_2 = \neg(\neg\Phi_1 \mathbf{U} \neg\Phi_2)$
  - weak until:  $\Phi_1 \mathbf{W} \Phi_2 = (\Phi_1 \mathbf{U} \Phi_2) \vee \mathbf{G}\Phi_1$
- Other notations:
  - next:  $\mathbf{X}\Phi = \bigcirc\Phi$
  - $\mathbf{G}\Phi = \square\Phi$
  - $\mathbf{F}\Phi = \diamond\Phi$
- We are dropping *past operators*, thus this is *pure future LTL*



UNIVERSITÀ  
DELL'AQUILA



DISIM  
Dipartimento di Ingegneria  
e Scienze dell'Informazione  
e Matematica

# LTL Semantics

- Goal: formally defining when  $\mathcal{S} \models \varphi$ , being  $\mathcal{S}$  a KS and  $\varphi$  an LTL formula
  - we say that  $\mathcal{S}$  *satisfies*  $\varphi$ , or  $\varphi$  *holds in*  $\mathcal{S}$
- This is true when, for all paths  $\pi$  of  $\mathcal{S}$ ,  $\pi$  satisfies  $\varphi$ 
  - i.e.,  $\forall \pi \in \text{Path}(\mathcal{S}). \pi \models \varphi$
  - symbol  $\models$  is overloaded...
- For a given  $\pi$ ,  $\pi \models \varphi$  iff  $\pi, 0 \models \varphi$
- Finally, to define when  $\pi, i \models \varphi$ , a recursive definition over the recursive syntax of LTL is provided
  - $\pi \in \text{Path}(\mathcal{S}), i \in \mathbb{N}$



UNIVERSITÀ  
DEGLI STUDI  
DELL'AQUILA



DISIM  
Dipartimento di Ingegneria  
e Scienze dell'Informazione  
e Matematica

# LTL Semantics for $\pi, i \models \varphi$

- $\forall \pi \in \text{Path}(\mathcal{S}), i \in \mathbb{N}. \pi, i \models \text{true}$
- $\pi, i \models p$  iff  $p \in L(\pi(i))$
- $\pi, i \models \Phi_1 \wedge \Phi_2$  iff  $\pi, i \models \Phi_1 \wedge \pi, i \models \Phi_2$
- $\pi, i \models \neg \Phi$  iff  $\pi, i \not\models \Phi$
- $\pi, i \models \mathbf{X}\Phi$  iff  $\pi, i + 1 \models \Phi$
- $\pi, i \models \Phi_1 \mathbf{U} \Phi_2$  iff  $\exists k \geq i: \pi, k \models \Phi_2 \wedge \forall i \leq j < k. \pi, j \models \Phi_1$



UNIVERSITÀ  
DEGLI STUDI  
DELL'AQUILA



DISIM  
Dipartimento di Ingegneria  
e Scienze dell'Informazione  
e Matematica

# LTL Semantics for Added Operators

- It is easy to prove that:
  - $\pi, i \models \mathbf{G}\Phi$  iff  $\forall j \geq i. \pi, j \models \Phi$
  - $\pi, i \models \mathbf{F}\Phi$  iff  $\exists j \geq i. \pi, j \models \Phi$
  - $\pi, i \models \Phi_1 \mathbf{R} \Phi_2$  iff  $\forall j \geq i. (\forall k < j. \pi, k \models \Phi_1) \rightarrow \pi, j \models \Phi_2$
  - $\pi, i \models \Phi_1 \mathbf{W} \Phi_2$  iff  $(\forall j \geq i. \pi, j \models \Phi_1) \vee (\exists k \geq i: \pi, k \models \Phi_2 \wedge \forall i \leq j < k. \pi, j \models \Phi_1)$
- For many formulas, it is silently required that paths are infinite
- That's why transition relations in KSs must be total



UNIVERSITÀ  
DEGLI STUDI  
DELL'AQUILA



DISIM  
Dipartimento di Ingegneria  
e Scienze dell'Informazione  
e Matematica

# Safety and Liveness Properties in LTL

- Given an LTL formula  $\varphi$ ,  $\varphi$  is a safety formula iff
$$\forall \mathcal{S}. (\exists \pi \in \text{Path}(\mathcal{S}) : \pi \not\models \varphi) \rightarrow \exists k : \pi|_k \not\models \varphi$$
- Given an LTL formula  $\varphi$ ,  $\varphi$  is a liveness formula iff
$$\forall \mathcal{S}. (\exists \pi \in \text{Path}(\mathcal{S}) : \pi \not\models \varphi) \rightarrow |\pi| = \infty$$
- All LTL formulas are either safety, liveness, or the AND of a safety and a liveness
  - being defined on paths, the counterexample is always a path
- Safety properties are those involving only **G**, **X**, true and atomic propositions
- Liveness are all those involving an **F**, or a **U** where the first formula is not the constant true
- Some formulas are both safety and liveness, like true, **G** true and so on

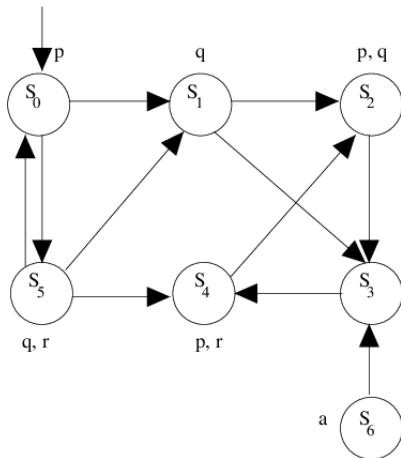


UNIVERSITÀ  
DEGLI STUDI  
DELL'AQUILA



DISIM  
Dipartimento di Ingegneria  
e Scienze dell'Informazione  
e Matematica

# LTL Examples



$\mathcal{S} \models \mathbf{F}p$  since  $p$  holds in the first state

For full: let  $\pi \in \text{Path}(\mathcal{S})$

$\pi, 0 \models \mathbf{F}p$  with  $j = 0$

recall:  $\pi, i \models \mathbf{F}\Phi$  iff

$\exists j \geq i. \pi, j \models \Phi$

$\pi, i \models p$  iff  $p \in L(\pi(i))$

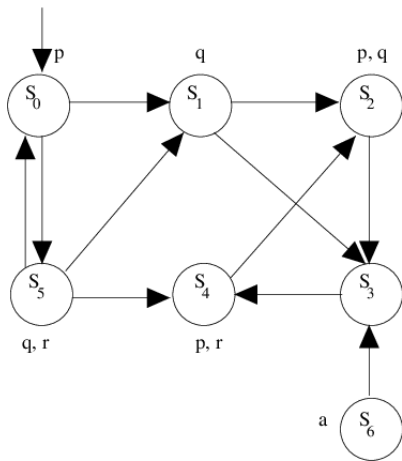


UNIVERSITÀ  
DEGLI STUDI  
DELL'AQUILA



DISIM  
Dipartimento di Ingegneria  
e Scienze dell'Informazione  
e Matematica

# LTl Examples



$\mathcal{S} \not\models \mathbf{F}a$  since  $s_6$  is not reachable from  $s_0$

counterexample:  $\pi = s_0 s_5 s_0 s_5 \dots$

For full:  $\pi, 0 \not\models \mathbf{F}a$  as, for all  $j \geq 0$ ,  $a \notin L(\pi(j))$

Counterexample is infinite, thus this is a liveness property  
Any finite prefix of  $\pi$  is not a counterexample



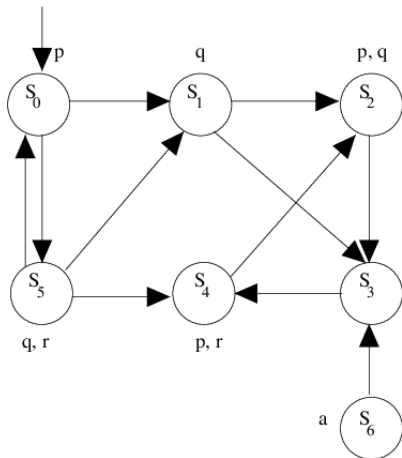
UNIVERSITÀ  
DEGLI STUDI  
DELL'AQUILA



DISIM  
Dipartimento di Ingegneria  
e Scienze dell'Informazione  
e Matematica



# LTL Examples



$\mathcal{S} \not\models \mathbf{G}p$  since there are many counterexamples, here is one:

$\pi = s_0 s_5 s_0 s_5 \dots$

For full:  $\pi, 0 \not\models \mathbf{G}p$  with  $j = 1$

recall:  $\pi, i \models \mathbf{G}\Phi$  iff

$\forall j \geq i. \pi, j \models \Phi$

$\pi, i \models p$  iff  $p \in L(\pi(i))$

Safety property, actually  $\pi|_2$  is enough

Every path having  $\pi|_2$  as a prefix is a counterexample

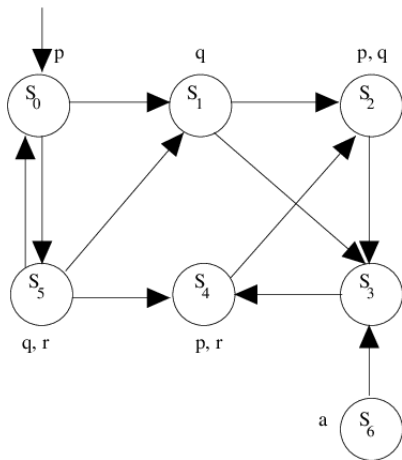


UNIVERSITÀ  
DEGLI STUDI  
FEDERICO II



DISIM  
Dipartimento di Ingegneria  
e Scienze dell'Informazione  
e Matematica

# LTL Examples



$\mathcal{S} \models \mathbf{G}\neg a$  since  $s_6$  is not reachable from  $s_0$

For full: let  $\pi \in \text{Path}(\mathcal{S})$   
 $\pi, 0 \models \mathbf{G}\neg a$  as the only state  $s$  with  $a \in L(s)$  is  $s_6$ , which is not reachable from  $s_0$

recall:  $\pi \in \text{Path}(\mathcal{S})$  implies  $\pi(0) \in I$ , thus  $\pi(0) = s_0$  here

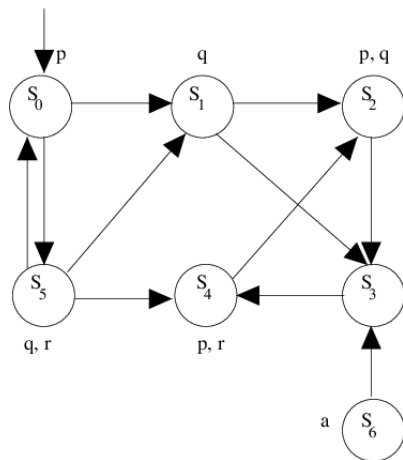


UNIVERSITÀ  
DEGLI STUDI  
DELL'AQUILA



DISIM  
Dipartimento di Ingegneria  
e Scienze dell'Informazione  
e Matematica

# LTL Examples



$\mathcal{S} \models p \text{ U } q$  since  $p \in L(s_0)$ ,  
 $\text{next}(s_0) = \{s_1, s_5\}$  and  $q \in L(s_1) \wedge q \in L(s_5)$

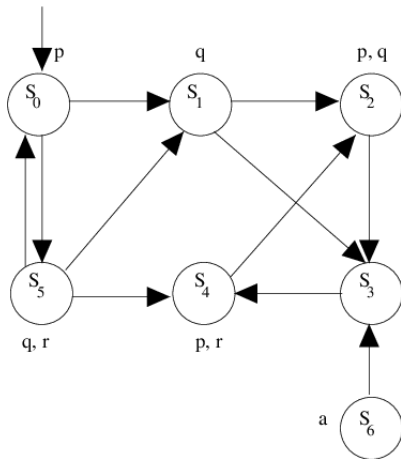


UNIVERSITÀ  
DEGLI STUDI  
DELL'AQUILA



DISIM  
Dipartimento di Ingegneria  
e Scienze dell'Informazione  
e Matematica

# LTL Examples



$\mathcal{S} \not\models p \mathbf{U} r$ , a counterexample is  $\pi = s_0 s_1 (s_2 s_3 s_4)$

Again this is a liveness formula, even if  $\pi|_1$  would have been enough

In fact, you have to consider all possible KSs...

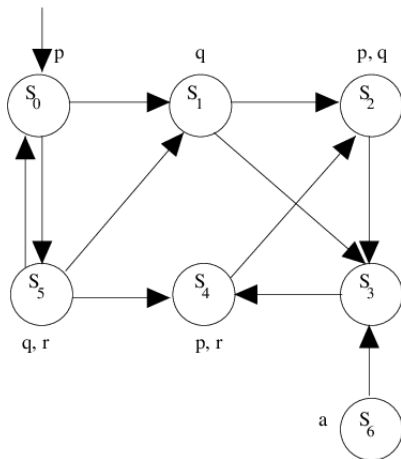


UNIVERSITÀ  
DEGLI STUDI  
DELL'AQUILA



DISIM  
Dipartimento di Ingegneria  
e Scienze dell'Informazione  
e Matematica

# LTl Examples



$\mathcal{S} \not\models \neg(p \mathbf{U} r)$ , a counterexample is  $\pi = (s_0 s_5)$

In fact,  $(s_0 s_5), 0 \models p \mathbf{U} r$

Thus it may happen that  $\mathcal{S} \not\models \Phi$  and  $\mathcal{S} \not\models \neg(\Phi)$

Instead, it is impossible that  $\mathcal{S} \models \Phi$  and  $\mathcal{S} \models \neg(\Phi)$

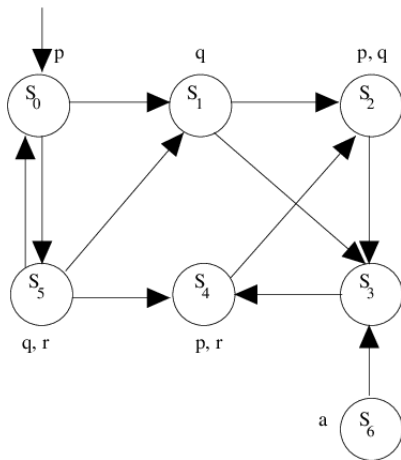


UNIVERSITÀ  
DEGLI STUDI  
DELL'AQUILA



DISIM  
Dipartimento di Ingegneria  
e Scienze dell'Informazione  
e Matematica

# LTl Examples



$\mathcal{S} \not\models \mathbf{FG}p$ , a counterexample is  
 $\pi = s_0 s_1 (s_2 s_3 s_4)$   
Again this is a liveness formula

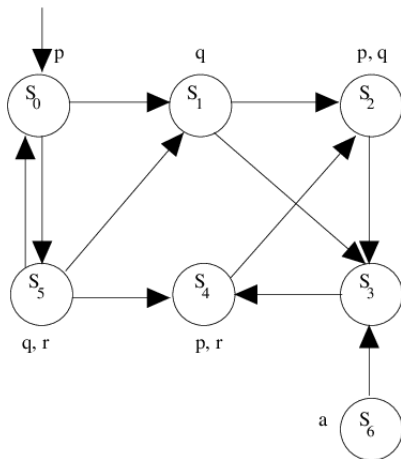


UNIVERSITÀ  
DEGLI STUDI  
DELL'AQUILA



DISIM  
Dipartimento di Ingegneria  
e Scienze dell'Informazione  
e Matematica

# LTL Examples



$\mathcal{S} \models \mathbf{GF}p$

All lassos are  $s_0s_5$  or  $s_2s_3s_4$

In both such lassos, there are states in which  $p$  holds

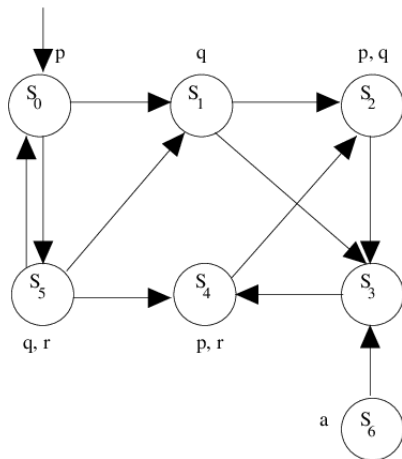


UNIVERSITÀ  
DEGLI STUDI  
DELL'AQUILA



DISIM  
Dipartimento di Ingegneria  
e Scienze dell'Informazione  
e Matematica

# LTl Examples



$\mathcal{S} \models \mathbf{GF}p \vee \mathbf{FG}p$

Consequence of the two previous slides



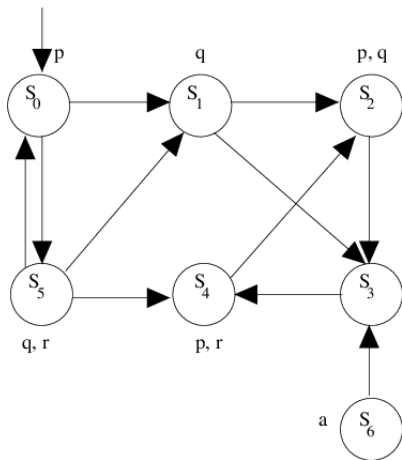
UNIVERSITÀ  
DEGLI STUDI  
DELL'AQUILA



DISIM  
Dipartimento di Ingegneria  
e Scienze dell'Informazione  
e Matematica



# LTL Examples



$\mathcal{S} \not\models \mathbf{G}(p \mathbf{U} q)$ , a counterexample is  $\pi = s_0 s_1 (s_2 s_3 s_4)$   
 $(p \mathbf{U} q)$  must hold at any reachable state  
Ok in  $s_0, s_1, s_2$ , but not in  $s_3$



UNIVERSITÀ  
DEGLI STUDI  
DELL'AQUILA



DISIM  
Dipartimento di Ingegneria  
e Scienze dell'Informazione  
e Matematica

# LTL Non-Toy Examples

- Recall the Peterson's protocol: checking mutual exclusion is  $\mathbf{G}(\neg(p \wedge q))$ , being  $p = P[1] = L3$ ,  $q = P[2] = L3$ 
  - all invariants are of the form  $\mathbf{G}P$ , where  $P$  does not contain modal operators  $\mathbf{X}$ ,  $\mathbf{U}$  or  $\mathbf{F}$
- Checking that both processes access to the critical section *infinitely often* is  $\mathbf{GF} P[1] = L3 \wedge \mathbf{GF} P[2] = L3$ 
  - liveness property: no process is infinitely banned to access the critical section
- Even better:  $\mathbf{G} (P[1] = L2 \rightarrow \mathbf{F} P[1] = L3)$ 
  - the same for the other process
  - since it is symmetric, this is actually enough



# Equivalence Between LTL Properties

- Definition of equivalence between LTL properties:  
 $\varphi_1 \equiv \varphi_2 \text{ iff } \forall \mathcal{S}. \mathcal{S} \models \varphi_1 \Leftrightarrow \mathcal{S} \models \varphi_2$ 
  - equivalent:  $\forall \sigma \dots$
- Idempotency:
  - $\mathbf{FF}p \equiv \mathbf{F}p$
  - $\mathbf{GG}p \equiv \mathbf{G}p$
  - $p \mathbf{U} (p \mathbf{U} q) \equiv (p \mathbf{U} q) \mathbf{U} q \equiv p \mathbf{U} q$
- Absorption:
  - $\mathbf{GFG}p \equiv \mathbf{FG}p$
  - $\mathbf{FGF}p \equiv \mathbf{GF}p$
- Expansion (used by LTL Model Checking algorithms!):
  - $p \mathbf{U} q \equiv q \vee (p \wedge \mathbf{X}(p \mathbf{U} q))$
  - $\mathbf{F}p \equiv p \vee \mathbf{XF}p$
  - $\mathbf{G}p \equiv p \wedge \mathbf{XG}p$



UNIVERSITÀ  
DEGLI STUDI  
DELL'AQUILA



DISIM  
Dipartimento di Ingegneria  
e Scienze dell'Informazione  
e Matematica

# CTL Syntax

$$\Phi ::= p \mid \Phi_1 \wedge \Phi_2 \mid \neg \Phi \mid (\Phi) \mid \mathbf{EX}\Phi \mid \mathbf{EG}\Phi \mid \mathbf{E}\Phi_1 \mathbf{U} \Phi_2$$

- Other derived operators (besides true, false, OR, etc):
  - $\mathbf{EF}\Phi = \mathbf{Etrue} \mathbf{U} \Phi$ 
    - cannot be defined using  $\mathbf{E}\neg\mathbf{G}\neg\Phi$ , as this is not a CTL formula
    - actually, it is a CTL\* formula (see later)
  - $\mathbf{AF}\Phi = \neg\mathbf{EG}\neg\Phi$ ,  $\mathbf{AG}\Phi = \neg\mathbf{EF}\neg\Phi$ ,  $\mathbf{AX}\Phi = \neg\mathbf{EX}\neg\Phi$
  - $\mathbf{A}\Phi_1 \mathbf{U} \Phi_2 = (\neg\mathbf{E}\neg\Phi_2 \mathbf{U} (\neg\Phi_1 \wedge \neg\Phi_1)) \wedge \neg\mathbf{EG}\neg\Phi_2$
  - $\Phi_1 \mathbf{AU}\Phi_2 = \mathbf{A}\Phi_1 \mathbf{U}\Phi_2$ ,  $\Phi_1 \mathbf{EU}\Phi_2 = \mathbf{E}\Phi_1 \mathbf{U}\Phi_2$



# Comparison with LTL Syntax

$$\Phi ::= \text{true} \mid p \mid \Phi_1 \wedge \Phi_2 \mid \neg \Phi \mid (\Phi) \mid \mathbf{X}\Phi \mid \Phi_1 \mathbf{U} \Phi_2$$

- Essentially, all temporal operators are preceded by either **E** or **G**
  - with some care for **U**



UNIVERSITÀ  
DEGLI STUDI  
DELL'AQUILA



DISIM  
Dipartimento di Ingegneria  
e Scienze dell'Informazione  
e Matematica

# CTL Semantics

- Goal: formally defining when  $\mathcal{S} \models \varphi$ , being  $\mathcal{S}$  a KS and  $\varphi$  a CTL formula
- This is true when, for all initial states  $s \in I$  of  $\mathcal{S}$ ,  $s \models \varphi$ 
  - thus, CTL is made of *state* formulas
  - LTL has *path* formulas
- To define when  $s \models \varphi$ , a recursive definition over the recursive syntax of CTL is provided
  - no need of an additional integer as for LTL syntax



UNIVERSITÀ  
DEGLI STUDI  
DELL'AQUILA



DISIM  
Dipartimento di Ingegneria  
e Scienze dell'Informazione  
e Matematica

# CTL Semantics for $s, i \models \varphi$

- $\forall s \in S. s, i \models \text{true}$
- $s \models p$  iff  $p \in L(s)$
- $s \models \Phi_1 \wedge \Phi_2$  iff  $s \models \Phi_1 \wedge s \models \Phi_2$
- $s \models \neg\Phi$  iff  $s \not\models \Phi$
- $s \models \mathbf{EX}\Phi$  iff  $\exists \pi \in \text{Path}(\mathcal{S}, s). \pi(1) \models \Phi$
- $s \models \mathbf{EG}\Phi$  iff  $\exists \pi \in \text{Path}(\mathcal{S}, s). \forall j. \pi(j) \models \Phi$
- $s \models \mathbf{E}\Phi_1 \mathbf{U} \Phi_2$  iff  
 $\exists \pi \in \text{Path}(\mathcal{S}, s) \exists k : \pi(k) \models \Phi_2 \wedge \forall j < k. \pi(j) \models \Phi_1$



# CTL Semantics for Added Operators

- It is easy to prove that:
  - $s \models \mathbf{AG}\Phi$  iff  $\forall \pi \in \text{Path}(\mathcal{S}, s). \forall j. \pi(j) \models \Phi$
  - $s \models \mathbf{AF}\Phi$  iff  $\forall \pi \in \text{Path}(\mathcal{S}, s). \exists j. \pi(j) \models \Phi$
  - analogously for **AU**, **AR**, **AW**
  - just replace  $\forall$  with  $\exists$  for **EF**, **ER**, **EW**
- As for CTL, for many formulas, it is silently required that paths are infinite
- So again transition relations in KSs must be total



UNIVERSITÀ  
DEGLI STUDI  
DELL'AQUILA



DISIM  
Dipartimento di Ingegneria  
e Scienze dell'Informazione  
e Matematica



# Safety and Liveness Properties in CTL

- Some CTL formulas may be neither safety nor liveness
  - being defined on states, the counterexample may be an entire computation tree
- Safety properties are those involving only **AG**, **AX**, true and atomic propositions
- Some formulas are both safety and liveness, like true, **G** true and so on
- Liveness are formulas like **AF**, **AFAG**, **AU**
- **EF** or **EG** are neither liveness nor safety

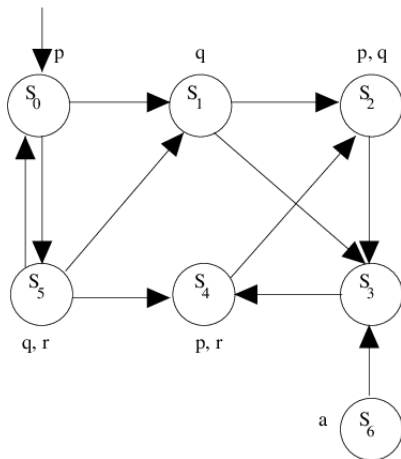


UNIVERSITÀ  
DEGLI STUDI  
DELL'AQUILA



DISIM  
Dipartimento di Ingegneria  
e Scienze dell'Informazione  
e Matematica

# CTL Examples



$\mathcal{S} \models \mathbf{AF}p$  since  $p$  holds in the first state

For full:  $s_0 \models \mathbf{F}p$  since  $p \in L(s_0)$ , thus, for all paths starting in  $s_0$ ,  $p$  holds in the first state, so it holds eventually

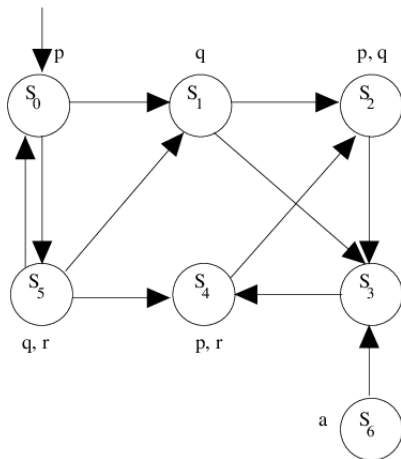


UNIVERSITÀ  
DEGLI STUDI  
DELL'AQUILA



DISIM  
Dipartimento di Ingegneria  
e Scienze dell'Informazione  
e Matematica

# CTL Examples



$\mathcal{S} \models \mathbf{EF}p$  for the same reason as above

If it holds for all paths, then it holds for one path

$\mathbf{AF}\phi \rightarrow \mathbf{EF}\phi$

The same holds for the other temporal operators **G**, **U** etc

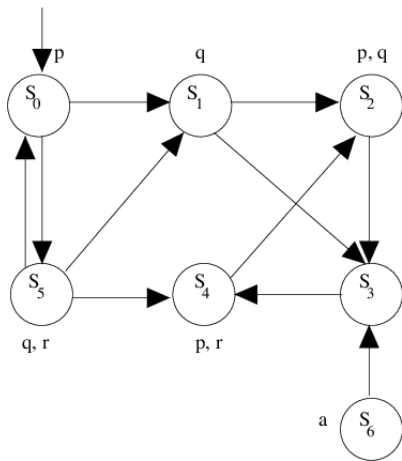


UNIVERSITÀ  
DEGLI STUDI  
DELL'AQUILA



DISIM  
Dipartimento di Ingegneria  
e Scienze dell'Informazione  
e Matematica

# CTL Examples



$\mathcal{S} \not\models \mathbf{EF}a$  since  $s_6$  is not reachable

Note that the counterexample cannot be a single path

Since it would not enough to disprove existence

The full reachable graph must be provided

One could also show the tree of all paths

Neither safety nor liveness

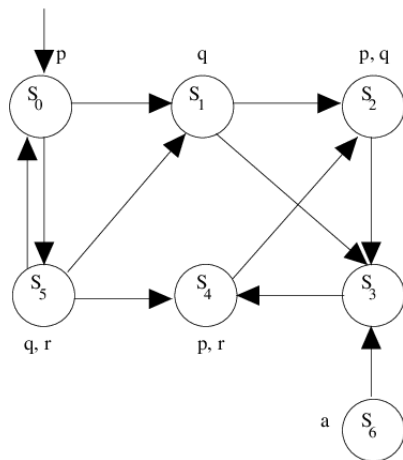


UNIVERSITÀ  
DEGLI STUDI  
DELL'AQUILA



DISIM  
Dipartimento di Ingegneria  
e Scienze dell'Informazione  
e Matematica

# CTL Examples



$\mathcal{S} \models \mathbf{A}(p \mathbf{U} q)$  since  $p \in L(s_0)$ ,  
 $\text{next}(s_0) = \{s_1, s_5\}$  and  $q \in L(s_1) \wedge q \in L(s_5)$

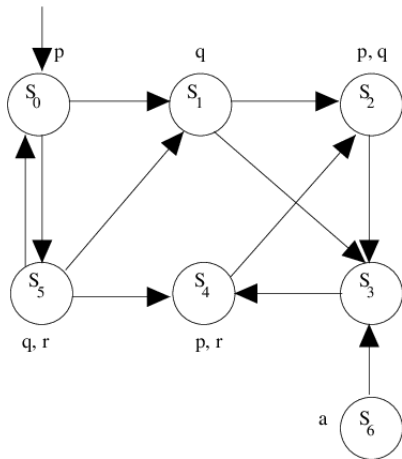


UNIVERSITÀ  
DEGLI STUDI  
DELL'AQUILA



DISIM  
Dipartimento di Ingegneria  
e Scienze dell'Informazione  
e Matematica

# CTL Examples



$\mathcal{S} \not\models \mathbf{A}(p \mathbf{U} r)$ , a counterexample is  $\pi = s_0 s_1 (s_2 s_3 s_4)$

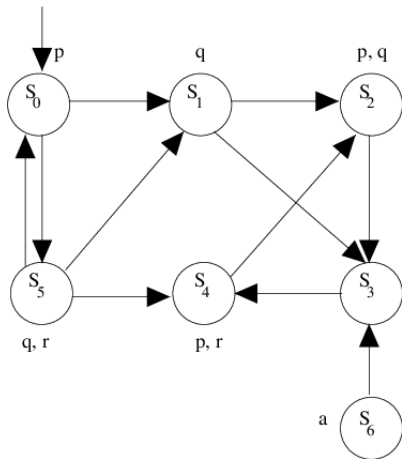


UNIVERSITÀ  
DEGLI STUDI  
DELL'AQUILA



DISIM  
Dipartimento di Ingegneria  
e Scienze dell'Informazione  
e Matematica

# CTL Examples



$\mathcal{S} \models \mathbf{E}(p \mathbf{U} r)$ , an example is  
 $\pi = (s_0 s_5)$

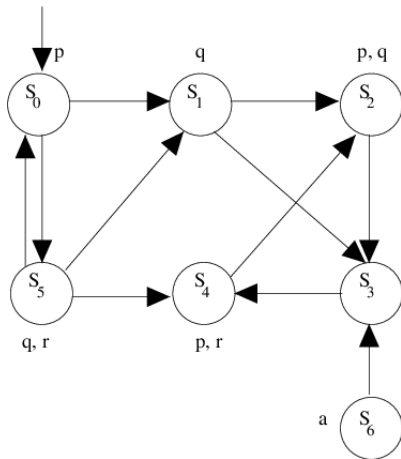


UNIVERSITÀ  
DEGLI STUDI  
DELL'AQUILA



DISIM  
Dipartimento di Ingegneria  
e Scienze dell'Informazione  
e Matematica

# CTL Examples



$\mathcal{S} \not\models \neg \mathbf{E}(p \mathbf{U} r)$ , a counterexample is  $\pi = (s_0 s_5)$

In fact,  $\mathcal{S} \not\models \Phi$  iff  $\mathcal{S} \models \neg(\Phi)$

Because here we have a single initial state



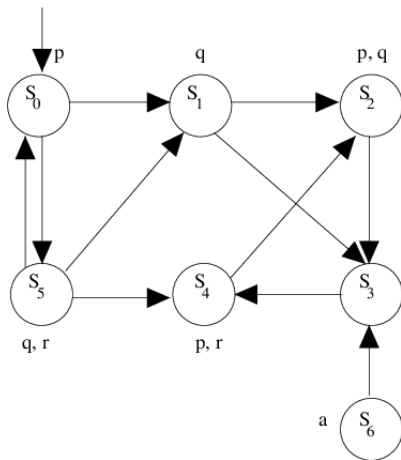
UNIVERSITÀ  
DEGLI STUDI  
DELL'AQUILA



DISIM  
Dipartimento di Ingegneria  
e Scienze dell'Informazione  
e Matematica



# CTL Examples



$\mathcal{S} \not\models \mathbf{AFAG}p$ , a counterexample is  $\pi = s_0s_1(s_2s_3s_4)$   
This is a liveness formula

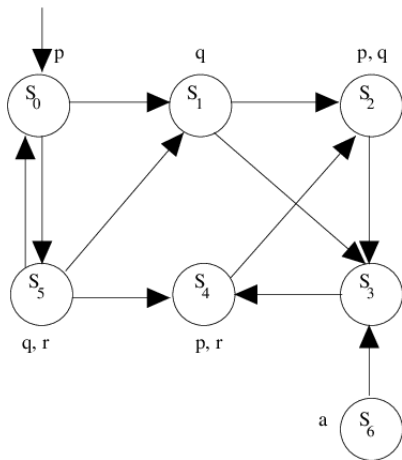


UNIVERSITÀ  
DEGLI STUDI  
DELL'AQUILA



DISIM  
Dipartimento di Ingegneria  
e Scienze dell'Informazione  
e Matematica

# CTL Examples



$\mathcal{S} \not\models \mathbf{EFEG}p$ , a counterexample is again a computation tree

All lassos are  $s_0s_5$  or  $s_2s_3s_4$

In both such lassos, there are states in which  $p$  does not hold

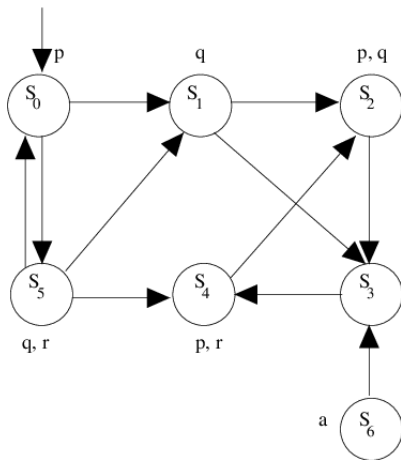


UNIVERSITÀ  
DEGLI STUDI  
DELL'AQUILA



DISIM  
Dipartimento di Ingegneria  
e Scienze dell'Informazione  
e Matematica

# CTL Examples



$\mathcal{S} \not\models \mathbf{AFEG}p$ , a counterexample is again a computation tree  
Since  $\mathcal{S} \not\models \mathbf{EFEG}p \dots$

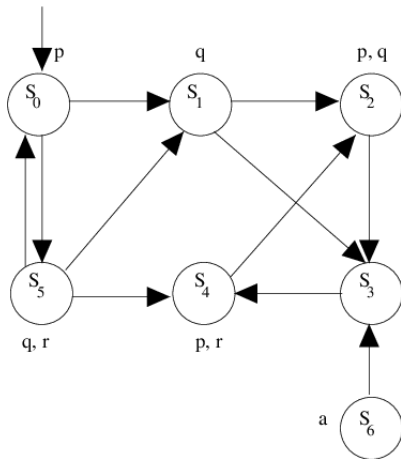


UNIVERSITÀ  
DEGLI STUDI  
DELL'AQUILA



DISIM  
Dipartimento di Ingegneria  
e Scienze dell'Informazione  
e Matematica

# CTL Examples



$\mathcal{S} \not\models \mathbf{EFAG}p$ , a counterexample is again a computation tree  
Since  $\mathcal{S} \not\models \mathbf{EFEG}p$ ...



UNIVERSITÀ  
DEGLI STUDI  
DELL'AQUILA



DISIM  
Dipartimento di Ingegneria  
e Scienze dell'Informazione  
e Matematica

# CTL Non-Toy Examples

- Recall the Peterson's protocol: checking mutual exclusion is **AG**( $\neg(p \wedge q)$ ), being  $p = P[1] = L3, q = P[2] = L3$ 
  - equivalent to LTL **G** $p$
- It is always possible to restart:  
**AGEF**  $P[1] = L0 \wedge \mathbf{AGEF} P[2] = L0$



UNIVERSITÀ  
DEGLI STUDI  
DELL'AQUILA



DISIM  
Dipartimento di Ingegneria  
e Scienze dell'Informazione  
e Matematica

# CTL vs. LTL: a Comparison

- Recall that  $\varphi_1 \equiv \varphi_2$  iff  $\forall \mathcal{S}. \mathcal{S} \models \varphi_1 \Leftrightarrow \mathcal{S} \models \varphi_2$ 
  - also holds (w.l.g.) when  $\varphi_1$  is LTL and  $\varphi_2$  is CTL
- Of course, some CTL formulas cannot be expressed in LTL
  - it is enough to put an **E**, since LTL always universally quantifies paths
  - so, there is not an LTL  $\varphi$  s.t.  $\varphi \equiv \mathbf{EG}p$ 
    - no,  $\mathbf{F}\neg p$  is not the same, why?
- So, one might think: LTL is contained in CTL
  - simply replace each temporal operator **O** with **AO**, that's it
  - let  $\mathcal{T}$  be a translator doing this
  - for any LTL formula  $\varphi$ ,  $\varphi \equiv \mathcal{T}(\varphi)$
  - actually,  $\mathbf{G}p \equiv \mathcal{T}(\mathbf{G}p) = \mathbf{AG}p$



# CTL vs. LTL: a Comparison

- Theorem. Let  $\varphi$  be an LTL formula. Then, either i)  $\varphi \equiv \mathcal{T}(\varphi)$  or ii) there does not exist a CTL formula  $\psi$  s.t.  $\varphi \equiv \psi$ 
  - idea of proof: replacing with **E** is of course not correct, and temporal operators on paths are the same
- Corollary. There exists an LTL formula  $\varphi$  s.t., for all CTL formulas  $\psi$ ,  $\varphi \not\equiv \psi$
- Proof of corollary:
  - by the theorem above and the definitions, we need to find
    - 1 an LTL formula  $\varphi$
    - 2 a KS  $\mathcal{S}$
  - where  $\mathcal{S} \models \varphi$  and  $\mathcal{S} \not\models \mathcal{T}(\varphi)$ 
    - viceversa is not possible



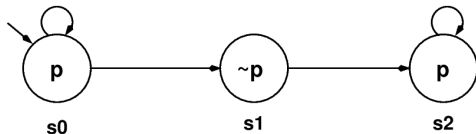
UNIVERSITÀ  
DEGLI STUDI  
DELL'AQUILA



DISIM  
Dipartimento di Ingegneria  
e Scienze dell'Informazione  
e Matematica

# CTL vs. LTL: a Comparison

- For example, as for the LTL formula, we may take  $\varphi = \mathbf{FG}p$ 
  - note instead that  $\mathbf{GF}p \equiv \mathbf{AGAF}p$
- For example, as for the KS  $\mathcal{S}$ , we may take

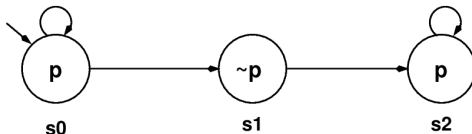


- We have that  $\mathcal{S} \models \mathbf{FG}p$ , but  $\mathcal{S} \not\models \mathbf{AFAG}p$
- Thus, CTL requires “more” than the corresponding LTL





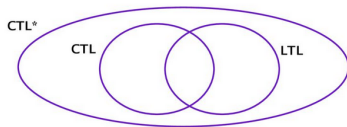
# CTL vs. LTL: a Comparison



- $\mathcal{S} \not\models \mathbf{AFAG}p$  means that
$$\neg(\forall \pi \in \text{Path}(\mathcal{S}). \exists j : \forall \rho \in \text{Path}(\mathcal{S}, \pi(j)). \forall k. p \in \rho(k))$$
$$= \exists \pi \in \text{Path}(\mathcal{S}). \forall j : \exists \rho \in \text{Path}(\mathcal{S}, \pi(j)). \exists k. p \notin \rho(k)$$
  - the path  $\pi$  is a loop on  $s_0 \dots$
- $\mathcal{S} \models \mathbf{FG}p$  means that  $\forall \pi \in \text{Path}(\mathcal{S}). \exists j : \forall k \geq j. p \in \pi(k)$
- Thus, there is not a CTL formula equivalent to  $\mathbf{FG}p$
- Furthermore, there is not an LTL formula equivalent to  $\mathbf{AFAG}p$



# CTL, LTL and CTL\*



- CTL\* introduced in 1986 (Emerson, Halpern) to include both CTL and LTL
- No restrictions on path quantifiers to be 1-1 with temporal operators, as in CTL
- State formulas:  $\Phi ::= \text{true} \mid p \mid \Phi_1 \wedge \Phi_2 \mid \neg \Phi \mid \mathbf{A}\Psi \mid \mathbf{E}\Psi$
- Path formulas:  $\Psi ::= \Phi \mid \Psi_1 \wedge \Psi_2 \mid \neg \Psi \mid \Psi_1 \mathbf{U} \Psi_2 \mid \mathbf{F}\Psi \mid \mathbf{G}\Psi$

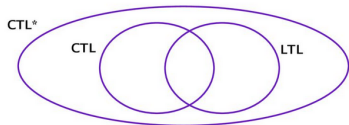


UNIVERSITÀ  
DEGLI STUDI  
DELL'AQUILA



DISIM  
Dipartimento di Ingegneria  
e Scienze dell'Informazione  
e Matematica

# CTL, LTL and CTL\*



- The intersection between CTL and LTL is both syntactic and “semantic”
- Some formulas are both CTL and LTL in syntax: all those involving only boolean combinations of atomic propositions
- “Semantic” intersection: some LTL formulas may be expressed in CTL and vice versa, using different syntax
  - **AGAF** $p$  and **GF** $p$
  - **AG** $p$  and **G** $p$
  - etc



UNIVERSITÀ  
DEGLI STUDI  
DELL'AQUILA



DISIM  
Dipartimento di Ingegneria  
e Scienze dell'Informazione  
e Matematica