TU Graz

SCIENCE
PASSION
TECHNOLOGY

Temporal Logic

Bettina Könighofer, Stefan Pranger
bettina.koenighofer@tugraz.at

Plan for next 5 Weeks

1. Intro to Temporal Logics: CTL*, LTL, CTL
2. CTL Model Checking – Part 1
3. MC for timed properties – Florian
4. CTL Model Checking – Part 2
5. LTL Model Checking

- Next: Model Checking of Probabilistic Systems (Stefan's Part)

Plan for Today

- Motivating Example
  - Informal Explanation of Syntax and Semantics
- CTL*
  - Syntax
  - Semantics
- Sublogics: CTL, LTL

Please interrupt at any time!

Warm Up - Model Sentences in Propositional Logic

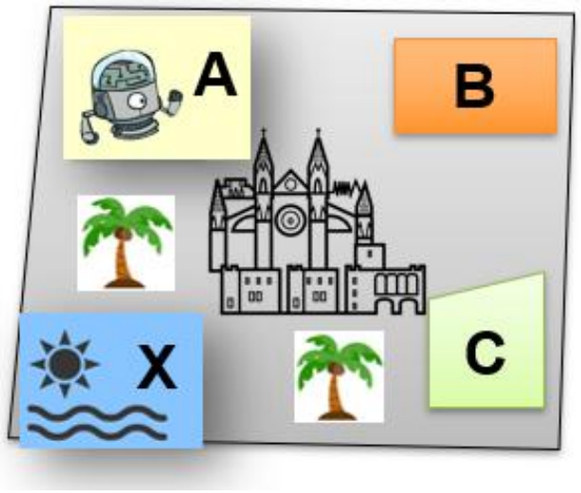
- "**If** a sentence has a truth value **and** is not a question, **then** it is a declarative sentence."
  - t … a sentence has a truth value
  - q … a sentence is a question
  - d … a sentence is a declarative sentence
  - $(\boldsymbol{t} \wedge \boldsymbol{q}) \rightarrow \boldsymbol{d}$

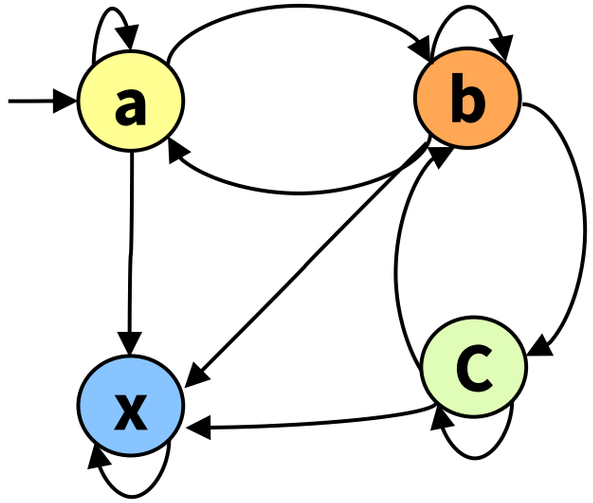
- "**If** two integers are **either** both odd **or** both even, **then** their sum is even."
  - o … two integers are both odd
  - e … two integers are both even
  - s … the sum of the integers is even
  - $(\boldsymbol{o} \oplus \boldsymbol{e}) \rightarrow \boldsymbol{s}$

Properties of Kripke Structures

- **For any run**, it is **always** the case that **if** the robot visits **A**, **then** it visits **C** within the **next two steps**.
- **There exists an run**, in which the robot **always** visits **C** within the **next two steps** after visiting **A.**

- For detailed modelling, we need:
  - **temporal operators**, and
  - **path quantifiers.**

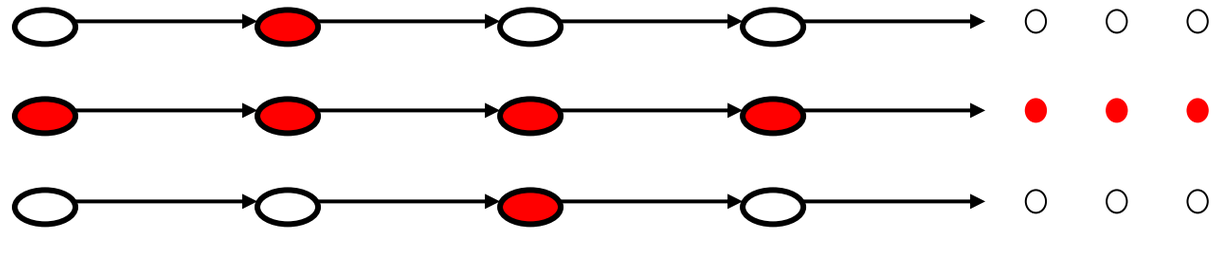




Temporal Operators

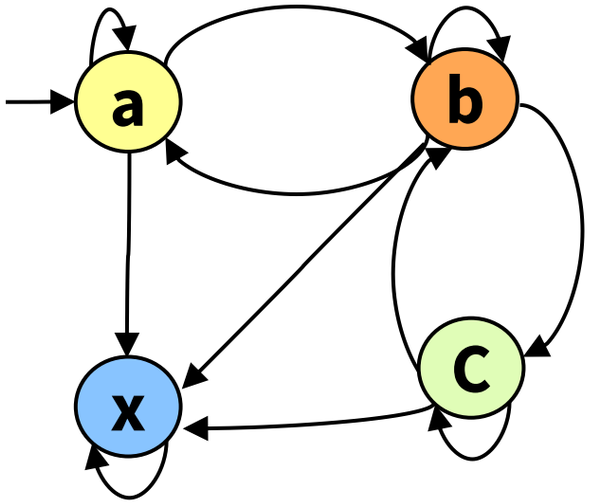
- Temporal operators
  - Describe **properties** along a given **path/execution**
- AP: a set of atomic propositions, $p \in AP$

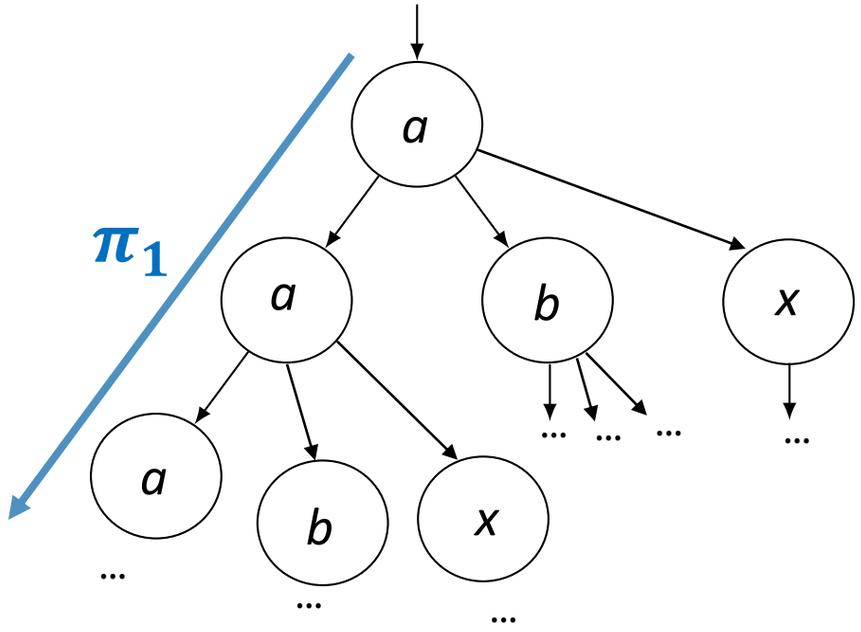
- **Next:** Xp
- **Globally:** Gp
- **Eventually:** Fp



Path Quantifiers

- **For all paths:** $A\varphi$
- **There exists a path:** $E\varphi$





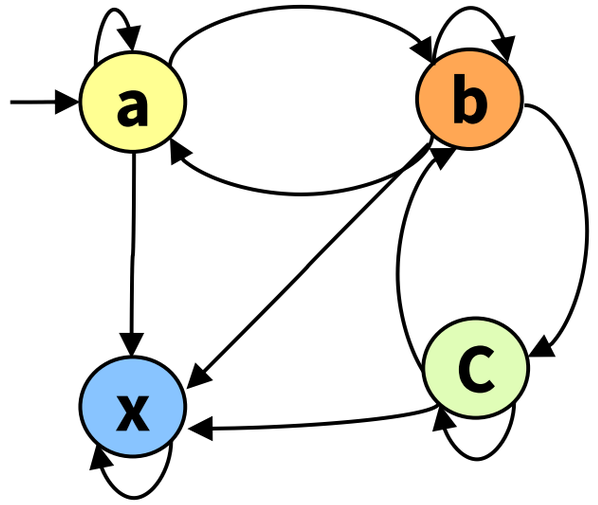
Properties of Kripke Structures

Temporal Operators
X… next
G… globally
F… eventually

Path quantifiers
A for **all** paths
E there **exists** a path

- **For any run**, it is **always** the case that **if** the robot visits **A**, **then** it visits **C** within the **next two steps**.
- $A\ G\ (a \rightarrow Xc \vee XXc)$

- **There exists a run**, it is **always** the case that **if** the robot visits **A**, **then** it visits **C** within the **next two steps**.
- $E\ G\ (a \rightarrow Xc \vee XXc)$



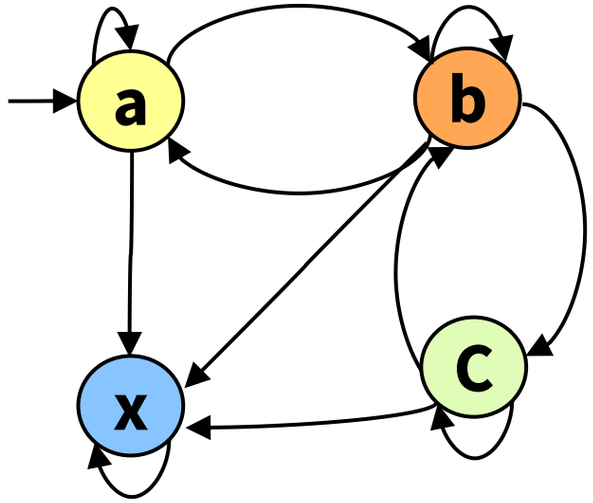
Properties of Kripke Structures

Temporal Operators
X… next
G… globally
F… eventually

Path quantifiers
A for **all** paths
E there **exists** a path

- **For any execution,** it holds that the robot **never** visits **X**.
- $A\ G\ (\neg x)$

- **There exists an execution,** in which it holds that the robot **never** visits **X.**
- $E\ G\ (\neg x)$



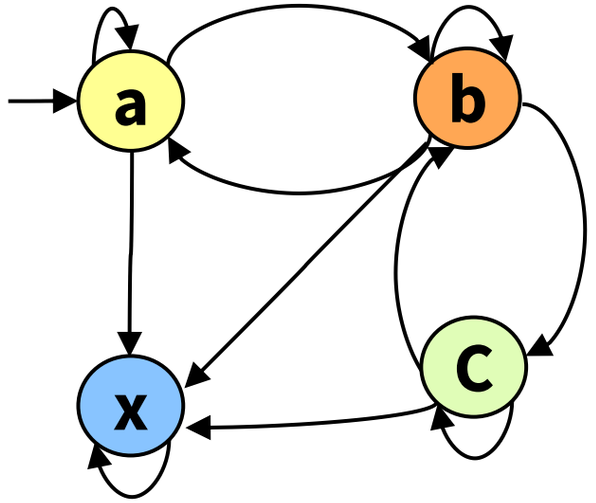
Properties of Kripke Structures

Temporal Operators
X… next
G… globally
F… eventually

Path quantifiers
A for **all** paths
E there **exists** a path

- **There exists an execution** in which it holds that the robot visits **A infinitely often and C infinitely often**.
- $E\ (GF\ a \wedge GF\ c)$

- **For any execution** it holds that **if** the robot visits **A infinitely often then C** is visited only **finitely often**.
- $A\ (GF\ a \rightarrow FG\neg c)$



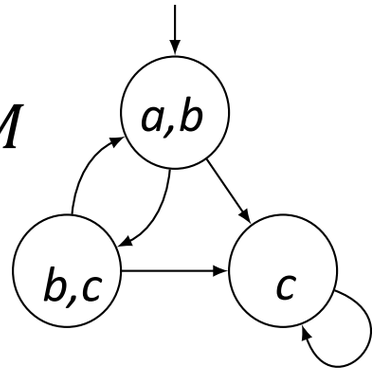
Plan for Today

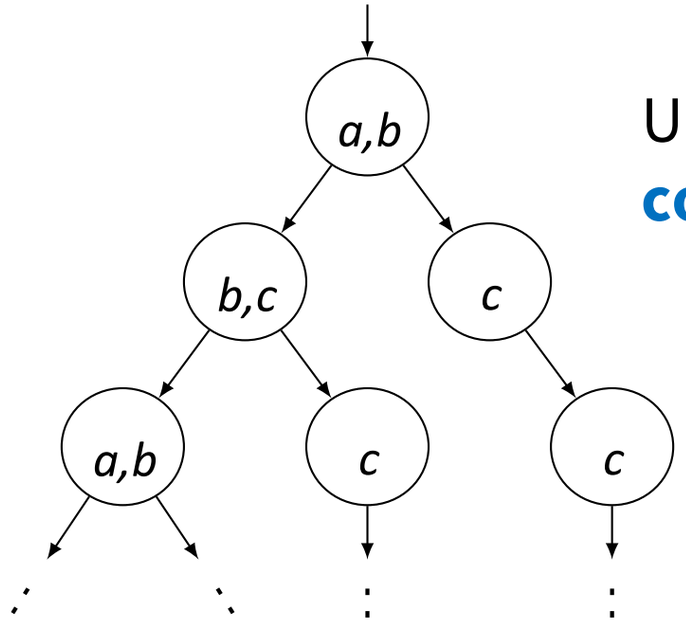
- Motivating Example
  - Informal Explanation of Syntax and Semantics
- CTL*
  - Syntax
  - Semantics
- Sublogics: CTL, LTL

Computation Tree Logic - CTL*

- Defines properties of **Computation Trees** of **Kripke structures**.
- Computation Tree
  - Shows **all possible executions** starting form **initial state**.
  - All branches of the tree are infinite.

Kripke structure M



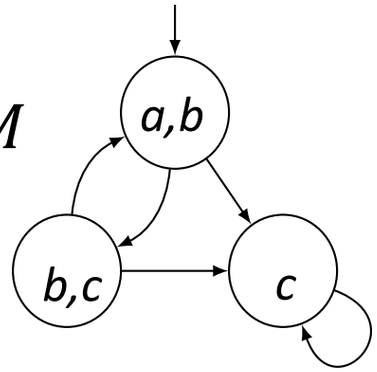


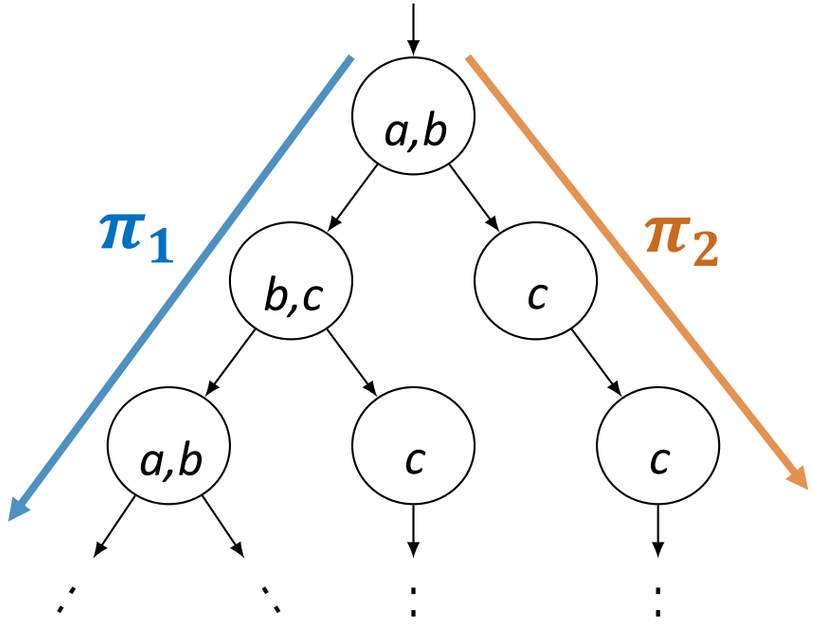
Unwinding of M into **computation tree**

CTL* - Path Quantifiers

- *Infinite* **path** $\pi = s_0, s_1, \ldots$with
  - s_0 is an initial state, and
  - for all $i \geq 0$, $(s_i, s_{i+1}) \in R$.
- Path quantifiers: **Aφ, Eφ**
  - They specify that **all paths** or **some paths** starting from a state s have property φ.

Kripke structure M

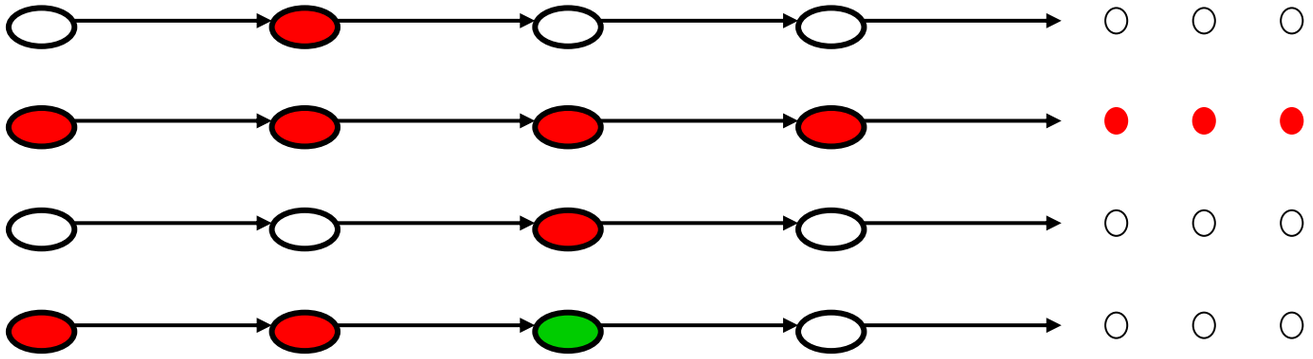




Temporal Operators

- Temporal operators
  - Describe **properties** along a given **path/execution**
- AP: a set of atomic propositions, $\boldsymbol{p}, \boldsymbol{q} \in \boldsymbol{AP}$

- **Next:** $\boldsymbol{Xp}$
- **Globally:** $\boldsymbol{Gp}$
- **Eventually:** $\boldsymbol{Fp}$
- **Until:** $\boldsymbol{pUq}$

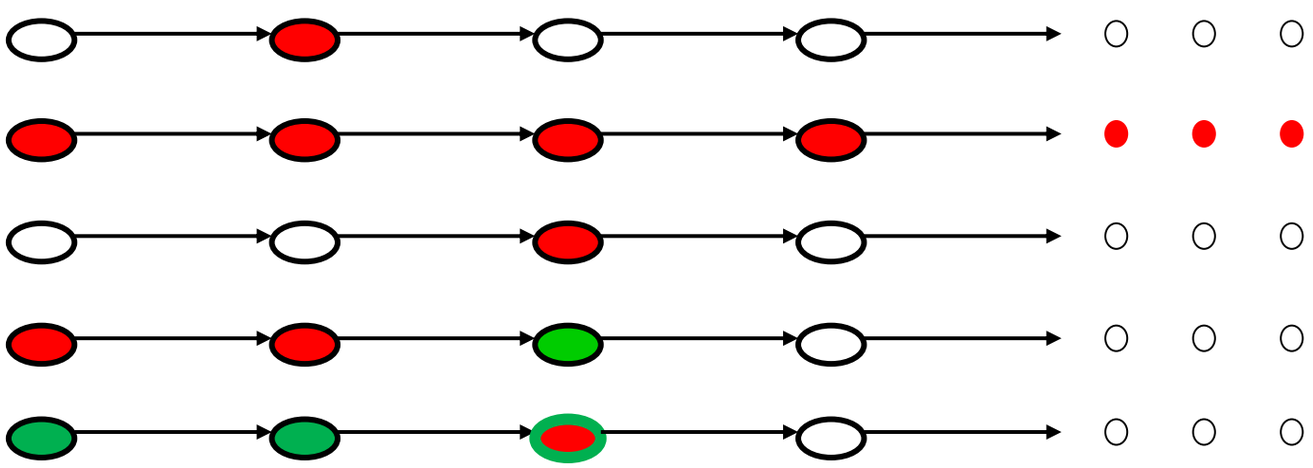


$\boldsymbol{pUq}$ holds if there is a state on π where q holds,
and at every preceding state on π (if it exists), p holds.

Temporal Operators

- Temporal operators
  - Describe **properties** along a given **path/execution**
- AP: a set of atomic propositions, $p, q \in AP$

- **Next:** Xp
- **Globally:** Gp
- **Eventually:** Fp
- **Until:** pUq
- **Release:** pRq

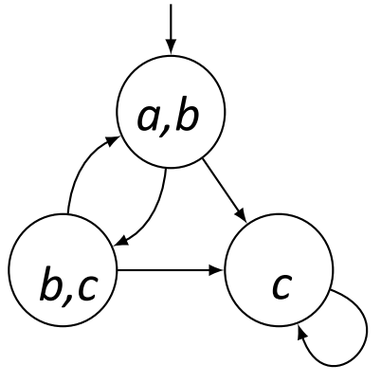


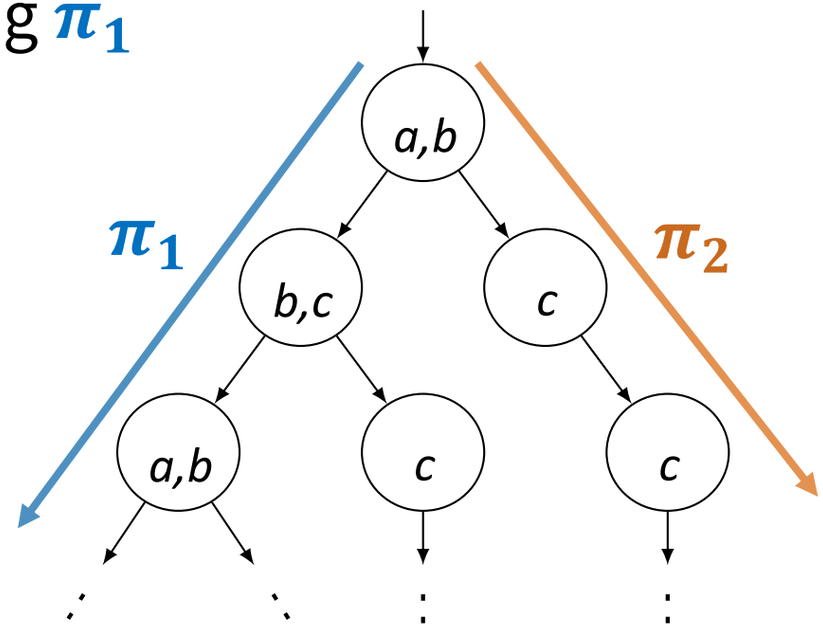
pRq … "p releases q": q has to hold until p holds.
However, p is not required to hold eventually.

State and Path Formulas

- Two types of formulas:
  - **State formulas** …true in a specific **state**
  - **Path formulas** …true along a specific **path**
- CTL* formulas are the **set of all state formulas**

- **Path Formulas:**
  - E.g.: $\pi_1 \models \boldsymbol{Gb}$ since $\boldsymbol{b}$ holds at every state along π_1
  - E.g.: $\pi_2 \not\models \boldsymbol{Gb}$ since $\boldsymbol{b}$ does **not** hold at every state along π_2

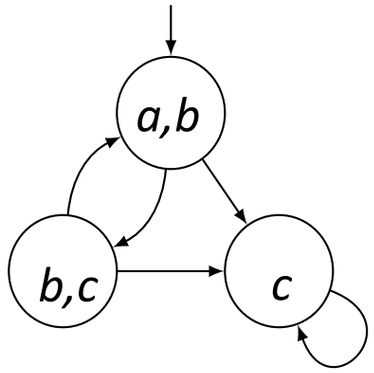


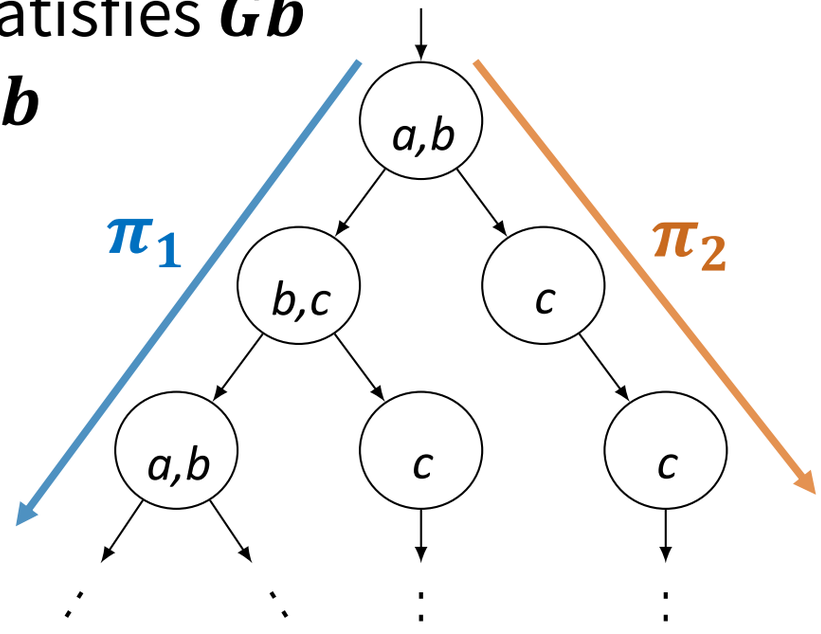


State and Path Formulas

- Two types of formulas:
  - **State formulas** …true in a specific **state**
  - **Path formulas** …true along a specific **path**
- CTL* formulas are the **set of all state formulas**

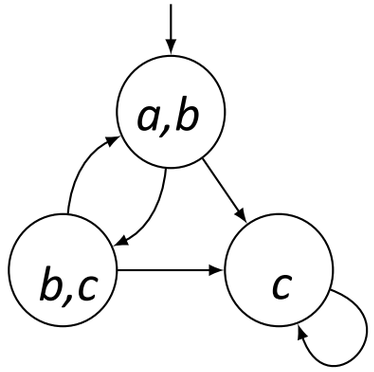
- **State Formulas**:
  - $s_0 \models \mathbf{EG\ b}$ since **there is a path** from s_0 that satisfies Gb
  - $s_0 \not\models \mathbf{AG\ b}$ since **not all paths** from s_0 satisfy Gb

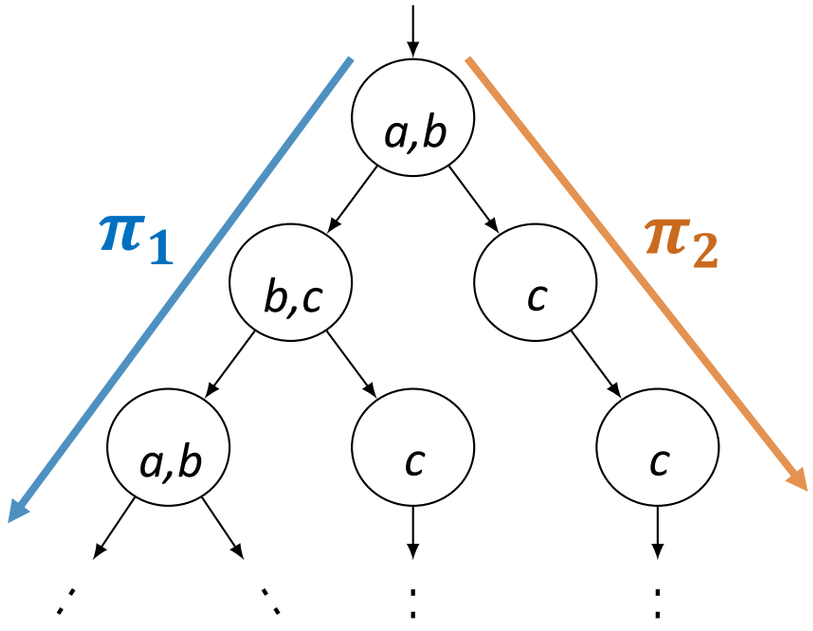




Informal Semantics of State and Path Formulas

- Does s_0 satisfy the following formula?
  - $s_0 \models \mathrm{EXX}\,(\mathrm{a} \wedge b)$
  - $s_0 \not\models \mathrm{EXAX}\,(\mathrm{a} \wedge b)$





Syntax of CTL*

- CTL* formulas are the **set of all state formulas**

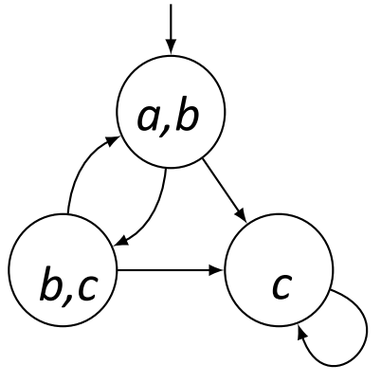
- Inductive definition of **state formulas**:
  - If $\mathbf{p} \in \mathbf{AP}$, then p is a state formula
  - If f_1 and f_2 are state formulas, so are $\neg f_1$, $f_1 \vee f_2$, and $f_1 \wedge f_2$.
  - If g is a path formula, then $\boldsymbol{E}g, \boldsymbol{A}g$ are state formulas

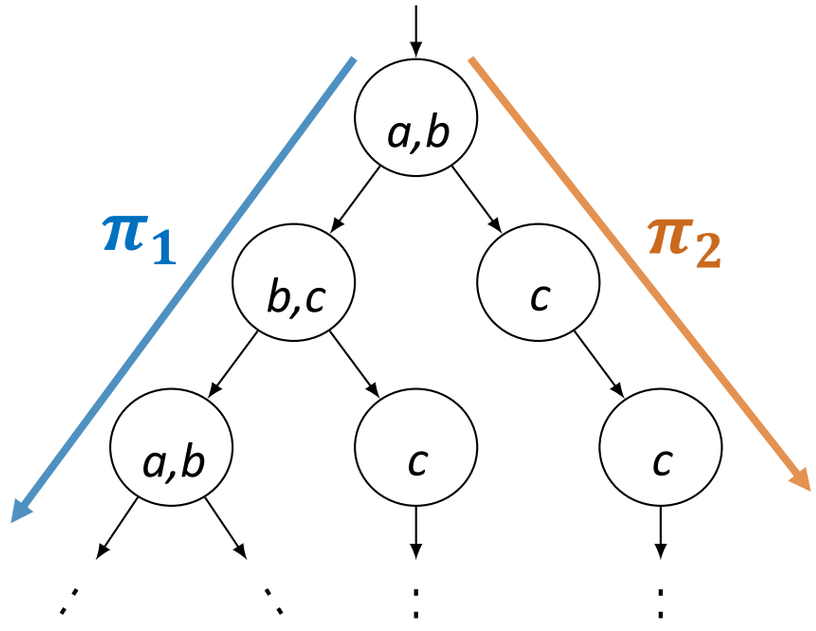
- Inductive definition of **path formulas**:
  - If f is a state formula, then f is also a path formula.
  - If g_1, g_2 are path formulas, then $\neg g_1$, $g_1 \vee g_2, g_1 \wedge g_2$, $\boldsymbol{X}g_1$, $\boldsymbol{G}g_1$, $\boldsymbol{F}g_1$, $g_1 \boldsymbol{U}\, g_2$, $g_1 \boldsymbol{R}\, g_2$ are path formulas.

Semantics of CTL* - Notation

- Kripke Structure $M = (S, S_0, R, AP, L)$
- $\pi = s_1, s_2, s_3 \ldots$ infinite **path** in M
- $\pi^i = s_i, s_{i+1}, s_{i+2} \ldots$ **suffix** of π starting at s_i

- For state formulas:
  - $M, s \models f$ … the **state** formula f holds in state s of M
- For path formulas.
  - $M, \pi \models g$ … the **path** formula g holds along π in M





Semantics of CTL*

- Let g_1 and g_2 be **path formulas** and f_1 and f_2 be **state formulas**
- $\models$ is inductively defined via the structure of the formula

- **State formulas**:
  - $M, s \models p \quad \Leftrightarrow p \in L(s)$ for $p \in AP$
  - $M, s \models \mathbf{E}\, g_1 \Leftrightarrow$ **there is a path** π from s s.t. $M, \pi \models g_1$
  - $M, s \models \mathbf{A}\, g_1 \Leftrightarrow$ **for every path** π from s s.t. $M, \pi \models g_1$
  - Boolean combination ($\wedge$, $\vee$, $\neg$) – the usual semantics

Semantics of CTL*

- Let g_1 and g_2 be **path formulas** and f_1 and f_2 be **state formulas**
- $\models$ is inductively defined via the structure of the formula

- **Path formulas**
  - $M, \pi \models f_1 \quad \Leftrightarrow M, s_0 \models f_1$ and $\pi = s_0, s_1, s_2, s_3$
  - $M, \pi \models \mathbf{X}\ g_1 \quad \Leftrightarrow M, \pi^1 \models g_1$
  - $M, \pi \models \mathbf{G}\ g_1 \quad \Leftrightarrow$ for every $i \geq 0$: $M, \pi^i \models g_1$
  - $M, \pi \models \mathbf{F} g_1 \quad \Leftrightarrow$ there exists $k \geq 0$: $M, \pi^k \models g_1$
  - $M, \pi \models g_1\ \mathbf{U}\ g_2 \quad \Leftrightarrow$ there exists $k \geq 0$: $M, \pi^k \models g_2$ and for every $0 \leq j < k$: $M, \pi^j \models g_1$

Semantics of CTL*

- Let g_1 and g_2 be **path formulas** and f_1 and f_2 be **state formulas**
- $\models$ is inductively defined via the structure of the formula

- $M \models f_1 \Leftrightarrow$ for all initial states $s_0 \in S_0$: $M, s_0 \models f_1$

Properties of CTL*

- The operators $\vee, \neg$, **X**, **U**, **E** are sufficient to express any CTL* formula
- **Your task**:
  - Rewrite the following formulas using $\vee, \neg$, **X**, **U**, **E**

1. $f_1 \wedge f_2 \equiv \neg(\neg f_1 \vee \neg f_2)$

2. $F\ g_1 \equiv true\ U\ g_1$

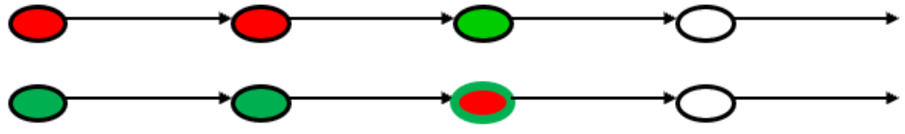
3. $G\ g_1 \equiv \neg\ F\ \neg g_1$

4. $A\ f_1 \equiv \neg\ E\ \neg f_1$

5. $g_1 R\ g_2 \equiv \neg\big(g_2\ U\ (g_1 \wedge g_2)\big) \vee G\ g_2$

Until: pUq

Release: pRq



Negation Normal Form (NNF)

- Formulas in **NNF** are formulas in which **negations** are applied only to **atomic propositions**

- Every CTL* formula is **equivalent** to a CTL* formula in **NNF**
- Negations can be **"pushed" inwards**.

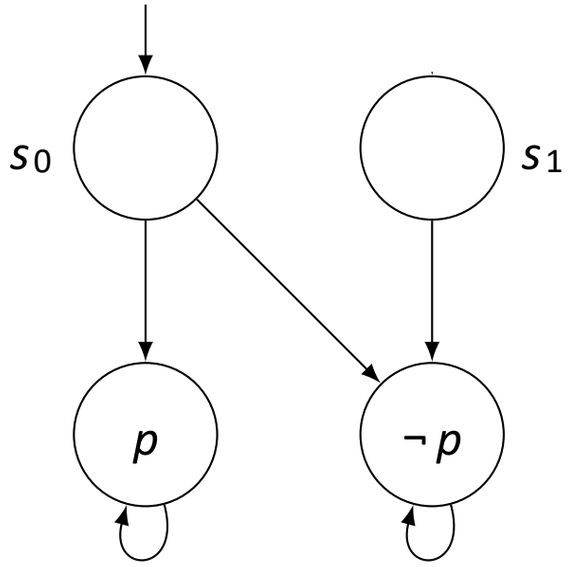
- $\neg\, \mathbf{E}\, f \equiv \mathbf{A}\, \neg f$
- $\neg\, \mathbf{G}\, f \equiv \mathbf{F}\, \neg f$
- $\neg\, (\, f\, \mathbf{U}\, g\,) \equiv (\, \neg f\, \mathbf{R}\, \neg g\,)$

- $\neg\, \mathbf{X}\, f \equiv \mathbf{X}\, \neg f$

Example 1/5: Semantics of CTL*

- $M \models f_1 \Leftrightarrow$ for all initial states $s_0 \in S_0$: $M, s_0 \models f_1$

- Does $M \models EX\ p$ or $M \models \neg EX\ p$?

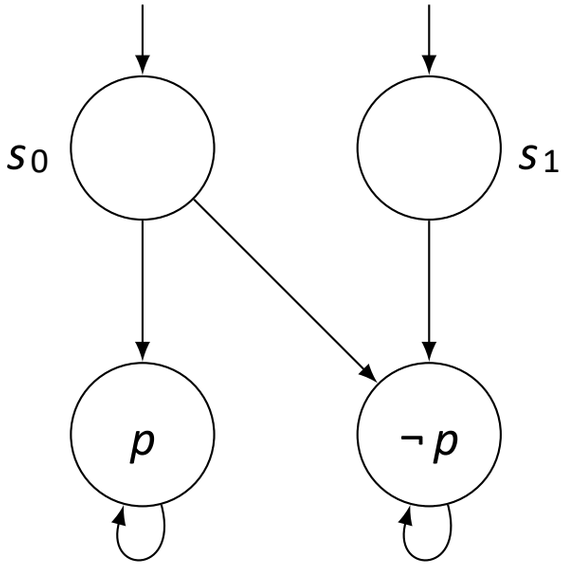


Solution:
$M \models$ **EX** p

Example 2/5: Semantics of CTL*

- $M \models f_1 \Leftrightarrow$ for all initial states $s_0 \in S_0$: $M, s_0 \models f_1$

- Does $M \models EX\ p$ or $M \models \neg EX\ p$?



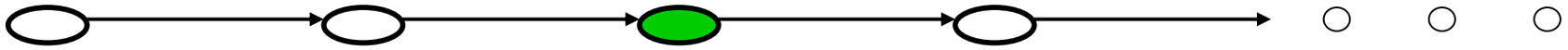
Neither

Such a situation **never** happens when M has a **single initial state.**

Example 3/5: Semantics of CTL*

- Given $a, b \in AP$
  What does a path satisfying $\boldsymbol{F}(a\ \boldsymbol{U}\ b)$ need to satisfy?

$\boldsymbol{F}(a\ \boldsymbol{U}\ b)$



Example 4/5: Semantics of CTL*

- Given $p \in AP$, what is the meaning of the following formulas?
  - $\pi \models GF\, p$ **Infinitely** often p along π
  - $\pi \models FG\, p$ **Finitely** often $\neg p$ along π

Example 5/5: Semantics of CTL*

- Given $p \in AP$, what is the meaning of the following formulas?
  - $\pi \models \boldsymbol{EGF}\ p$ There exists a path that satisfies p infinitely often
  - $\pi \models \boldsymbol{EG}\ \boldsymbol{EF}\ p$ There exists a path in which it is possible to reach p from any state

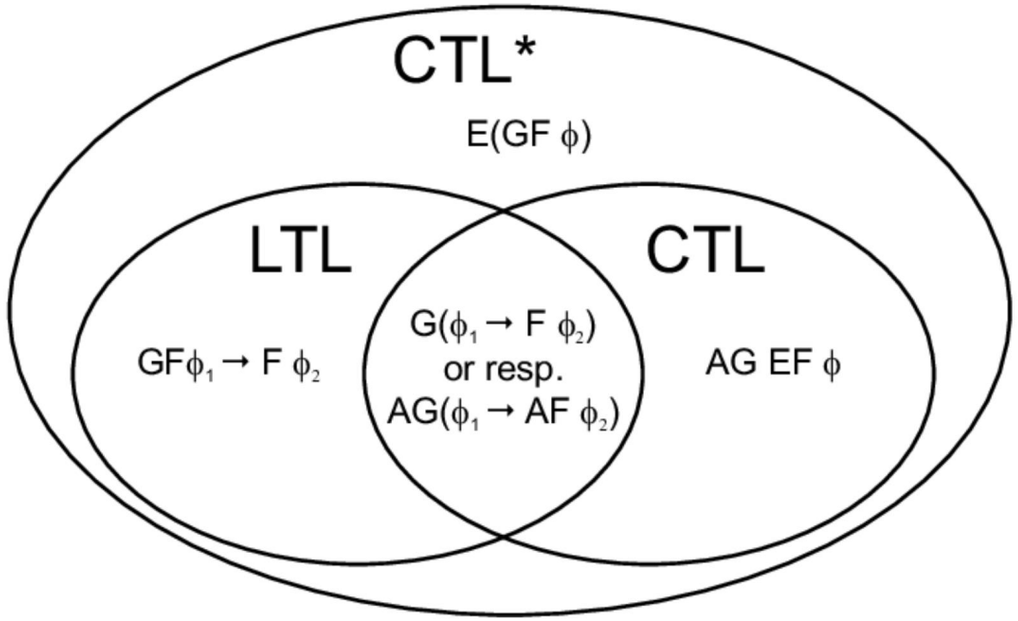
Plan for Today

- Motivating Example
  - Informal Explanation of Syntax and Semantics
- CTL*
  - Syntax
  - Semantics
- **Sublogics: CTL, LTL**

Sublogics of CTL*

- **CTL** and **LTL** are the two most used sub-logics of **CTL***
  - Restriction on **allowed combination** of **temporal operators** and **path quantifiers**

- **CTL*** allows **any combination** of temporal operators and path quantifiers.



Linear Temporal Logic (LTL)

- **LTL** consists of state formulas of the form **Ag**, where g is a **path formula**, containing **no path quantifiers**.

→ Formulas have only **one, outermost universal quantification**

→ Typically in LTL, the path quantifier is omitted.

- Examples:
  - $GF\ \varphi$
  - $G(\varphi \rightarrow F\ \psi)$
  - $G(\varphi \rightarrow XXX\ \psi)$
  - …

LTL - Syntax

- **LTL** is the set of all state formulas as defined below.

- **State formulas**
  - $\boldsymbol{A}g$ where g is a path formula

- **Path formulas**
  - $\boldsymbol{p} \in \boldsymbol{AP}$
  - $\neg g_1$, $g_1 \vee g_2$, $g_1 \wedge g_2$, $\boldsymbol{X} g_1$, $\boldsymbol{G} g_1$, $g_1 \boldsymbol{U} g_2$, $g_1 \boldsymbol{R} g_2$ where g_1 and g_2 are path formulas

Computation Tree (CTL)

- **LTL** consists of state formulas, where **path quantifiers** and **temporal operators** come in **pairs**: **AG**, **AU**, **AX**, **AF**, **AR**, **EG**, **EU**, **EX**, **EF**, **ER**

- Examples:
  - $E(\varphi U \psi)$
  - $EF(\varphi) \wedge EG\psi$
  - $AF\ AG\ \varphi$ ...

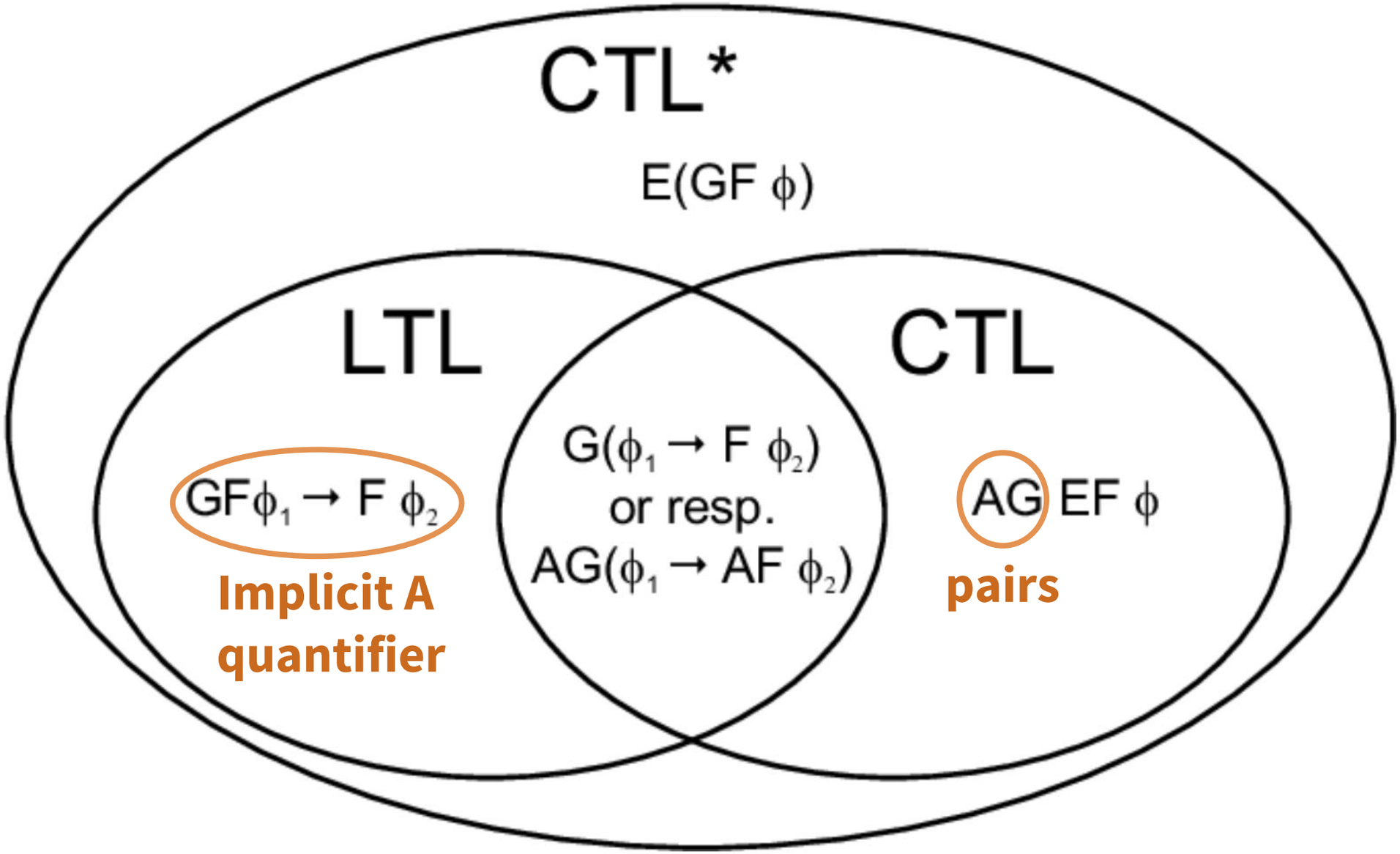
CTL - Syntax

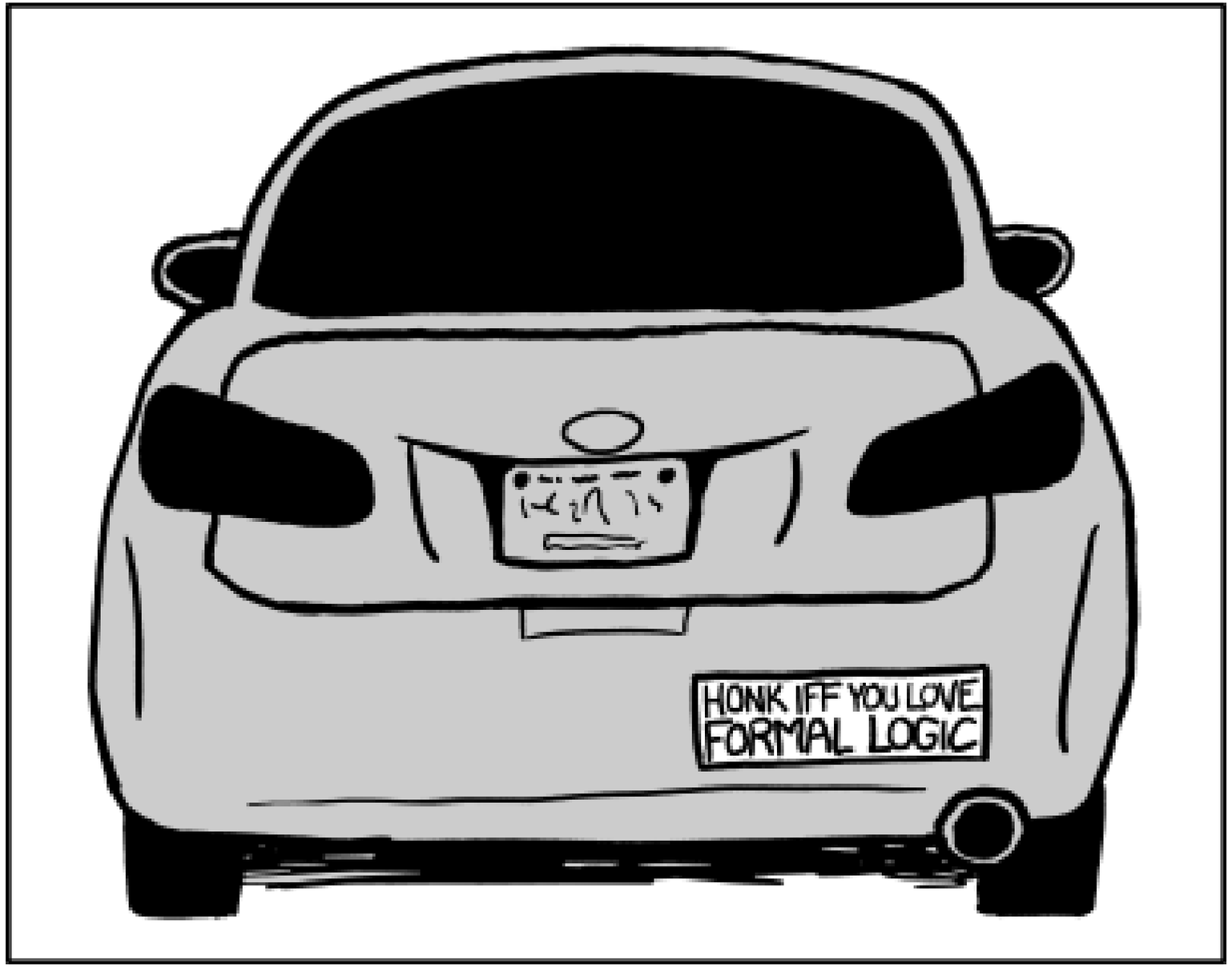
- **CTL** is the set of all state formulas as defined below.

- **State formulas**
  - $\boldsymbol{p} \in \boldsymbol{AP}$
  - $\neg f_1, \quad f_1 \vee f_2, \quad f_1 \wedge f_2$
  - $\boldsymbol{AX} f_1, \quad \boldsymbol{AG} f_1, \quad \boldsymbol{A}(f_1 \boldsymbol{U} f_2), \quad \boldsymbol{A}(f_1 \boldsymbol{R} f_2)$
  - $\mathbf{E}\boldsymbol{X} f_1, \quad \boldsymbol{EG} f_1, \quad \boldsymbol{E}(f_1 \boldsymbol{U} f_2), \quad \boldsymbol{E}(f_1 \boldsymbol{R} f_2)$

where f_1 and f_2 are path formulas

LTL/CTL/CTL*


CTL*
E(GF ϕ)
LTL
CTL
GFϕ1 → F ϕ2
Implicit A quantifier
G(ϕ1 → F ϕ2)
or resp.
AG(ϕ1 → AF ϕ2)
AG EF ϕ
pairs


HONK IFF YOU LOVE
FORMAL LOGIC


ANY
QUESTIONS
?