CTL
Computation Tree Logic

CTL

Formules de CTL

φ,ψ ::= P | $\neg\varphi$ | $\varphi\vee\psi$ | $\mathbf{EX}\varphi$ | $\mathbf{AX}\varphi$ | $\mathbf{E}\varphi\mathbf{U}\psi$ | $\mathbf{A}\varphi\mathbf{U}\psi$

avec P $\in$ AP

+ Abbréviations :

$\top$, $\bot$, $\wedge$, $\Rightarrow$

$\mathbf{EF}\varphi$ $\mathbf{AF}\varphi$

$\mathbf{F}\varphi = \top\ \mathbf{U}\ \varphi$: “eventually“,

$\mathbf{EG}\varphi$ $\mathbf{AG}\varphi$

$\mathbf{G}\ \varphi = \neg\ \mathbf{F}\ \neg\varphi$: “always“

$\varphi\mathbf{W}\ \psi = \varphi\mathbf{U}\ \psi \vee \mathbf{G}\ \varphi$: “weak until” $\mathbf{E}\varphi\mathbf{W}\psi$ $\mathbf{A}\varphi\mathbf{W}\psi$

CTL - sémantique

$\mathbf{S}$ = (Q, Act, $\rightarrow$, q_{init}, AP, L)

Exec(q) = ens. des exécutions infinies partant de q.

$\rho \in$ Exec(q): $\rho = q_0 q_1 q_2 q_3 q_4 \ldots$ avec q_0=q et $q_i \rightarrow q_{i+1}$

Notation: $\rho(i) = q_i$ $\forall$ i≥0

On interprète les formules de CTL sur des états de $\mathbf{S}$.

$q \models P$ iff $P \in L(q)$

$q \models \mathbf{EX}\varphi$ iff $\exists\ q \rightarrow q'$ t.q. $q' \models \varphi$

$q \models \mathbf{AX}\ \varphi$ iff $\forall\ q \rightarrow q'$, on a: $q' \models \varphi$

$q \models \mathbf{E}\varphi\mathbf{U}\psi$ iff $\exists \rho \in$ Exec(q) t.q. $\exists$ i ≥ 0 t.q. ($\rho(i) \models \psi$ et ($\forall$ 0 ≤j<i: $\rho(j) \models \varphi$))

$q \models \mathbf{A}\varphi\mathbf{U}\psi$ iff $\forall\ \rho \in$ Exec(q), $\exists$ i ≥ 0 t.q. ($\rho(i) \models \psi$ et ($\forall$ 0 ≤j<i: $\rho(j) \models \varphi$))

CTL

Définition alternative (équivalente !!):

Formules d’état:

φ,ψ ::= P | $\neg\varphi$ | $\varphi\vee\psi$ | $\mathbf{E}\ \varphi_p$ | $\mathbf{A}\ \varphi_p$

P $\in$ AP

Formules de chemin:

φ_p,ψ_p ::= $\mathbf{X}\ \varphi$ | $\varphi\ \mathbf{U}\ \psi$

$\mathbf{E}\ \varphi_p$ = « il existe un chemin vérifiant φ_p »

$\mathbf{A}\ \varphi_p$ = « tous les chemins vérifient φ_p »

CTL - sémantique

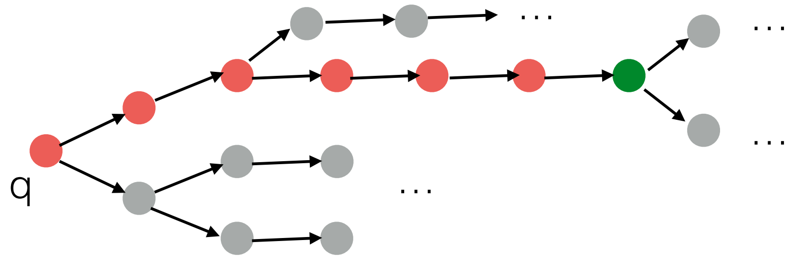
Définition alternative (équivalente !!):

$q \models P$ iff $P \in L(q)$
$q \models \mathbf{E}\, \varphi_p$ iff $\exists\, \rho \in Exec(q)$ t.q. $\rho \models \varphi_p$
$q \models \mathbf{A}\, \varphi_p$ iff $\forall\, \rho \in Exec(q)$, $\rho \models \varphi_p$

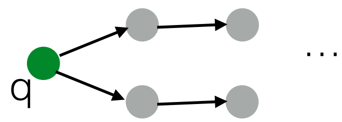
$\rho \models \mathbf{X}\, \varphi$ iff $\rho(1) \models \varphi$
$\rho \models \varphi\, \mathbf{U}\, \psi$ iff $\exists\, i \geq 0$ ($\rho(i) \models \psi$ et ($\forall\, 0 \leq j < i$: $\rho(j) \models \varphi$))

CTL - sémantique

$q \models$ **E rouge U vert**

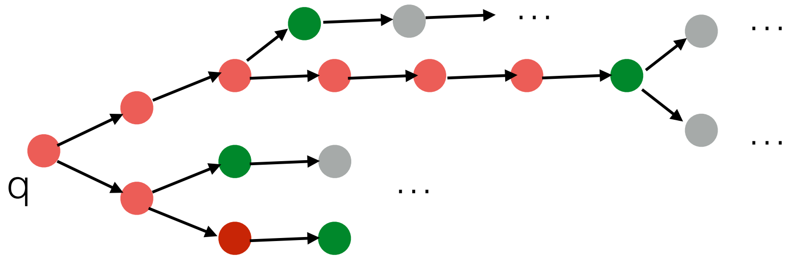


ou:

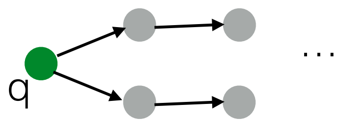


CTL - sémantique

$q \models$ **A rouge U vert**

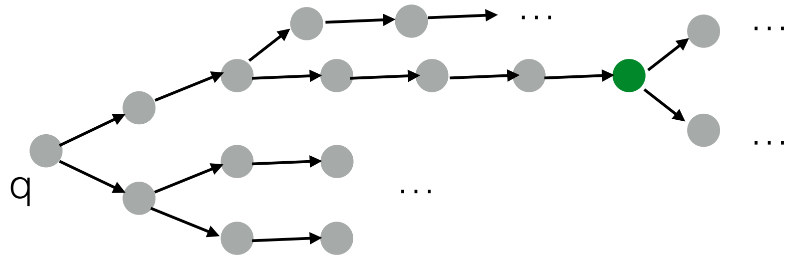


ou:

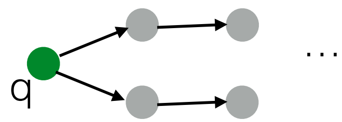


CTL - sémantique

$q \models$ **EF vert**

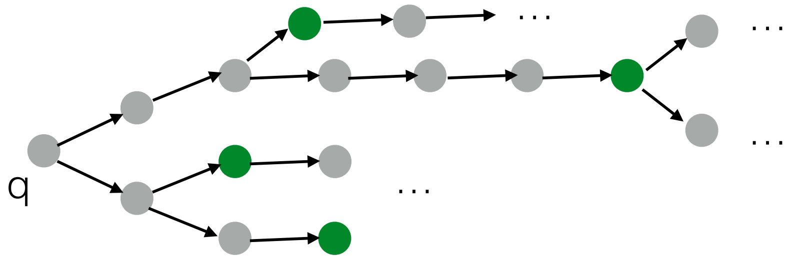


ou:

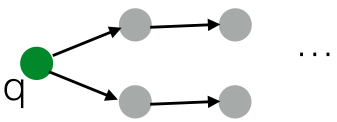


CTL - sémantique

q ⊨ **AF** vert

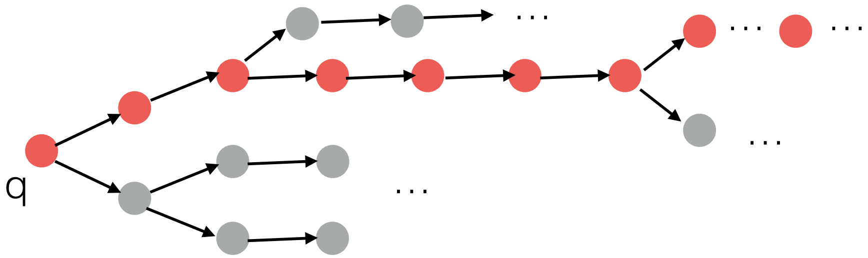


ou:



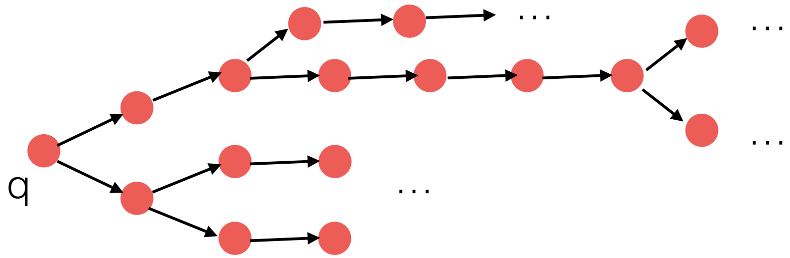
CTL - sémantique

q ⊨ **EG** **rouge**



CTL - sémantique

q ⊨ **AG** **rouge**



Tout ce qui est accessible depuis q est rouge.

Exemples

AG (problème ⇒ **AF** alarme)

AG (**EX** a)

E (**EX** a) **U** b = EU (EX a, b)

AG (**EF** a)

Which logic should we choose ?

Which is the best one ? CTL* ? LTL ? CTL ? …

There are several criteria:

- the expressiveness
- the complexity of decision procedures
- the existence of tools
- …

Expressivité

Expressiveness

3 different notions:

- **Distinguishing power**
  - $\mathcal{L}$ is at least as distinguishing as $\mathcal{L}'$ ($\mathcal{L} \geq \mathcal{L}'$) iff for any $\mathbf{S}$ and $\mathbf{S}'$, $\mathbf{S} \equiv_{\mathcal{L}} \mathbf{S}' \Rightarrow \mathbf{S} \equiv_{\mathcal{L}'} \mathbf{S}'$
  - with: $\mathbf{S} \equiv_{\mathcal{L}} \mathbf{S}'$ iff $(\forall\ \varphi \in \mathcal{L},\ \mathbf{S} \models \varphi \Leftrightarrow \mathbf{S}' \models \varphi)$
- **Expressive power**
  - $\mathcal{L}$ is at least as expressive as $\mathcal{L}'$ ($\mathcal{L} \succcurlyeq \mathcal{L}'$) iff for $\forall\ \varphi' \in \mathcal{L}'$, $\exists\ \varphi \in \mathcal{L}$ s.t. $\varphi \equiv \varphi'$
- **Succinctness**
  when 2 logics $\mathcal{L}$ and $\mathcal{L}'$ are equally expressive, one can be more succinct (w.r.t. the size of the formula)…

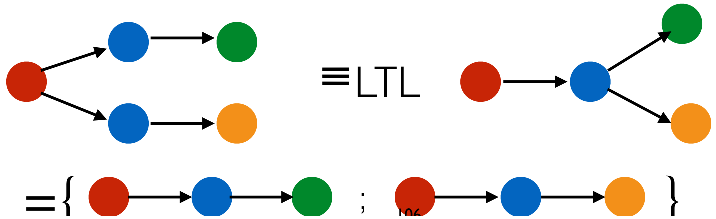
Distinguishing power

CTL and CTL* formulas are interpreted over state of KS, or equivalently over the nodes of its <u>execution tree</u>.

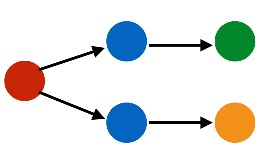
LTL formulas are interpreted over paths.
With LTL, a system is viewed as a <u>set of executions</u>.

<u>Convention:</u>
for a KS $\mathbf{S}$ and $\varphi \in$ LTL, we write $\mathbf{S} \models \varphi$ when $q_0 \models \mathbf{A}\ \varphi$

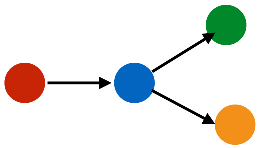
→ Two Kripke structures satisfy the same LTL formulas iff they have the same set of executions (*ie* they are trace-equivalent).



Distinguishing power



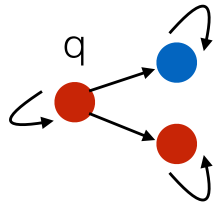
$\not\equiv$CTL



$\models \mathbf{EX}\,(\mathbf{EX}\ \mathbf{P} \;\wedge\; \mathbf{EX}\ \mathbf{P})$

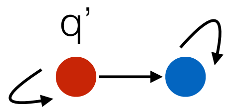
donc LTL ne distingue pas autant que CTL !

Autre exemple:



$\equiv$LTL

$\not\equiv$CTL



mêmes exécutions !

$q' \models \mathbf{AG}\,(\mathbf{EF}\ \mathbf{P})$
$q \not\models \mathbf{AG}\,(\mathbf{EF}\ \mathbf{P})$

Distinguishing power

→ Two (<u>finitely branching</u>) Kripke structures satisfy the same CTL (or CTL*) formulas iff they are bisimilar.
(Hennessy, 1980)

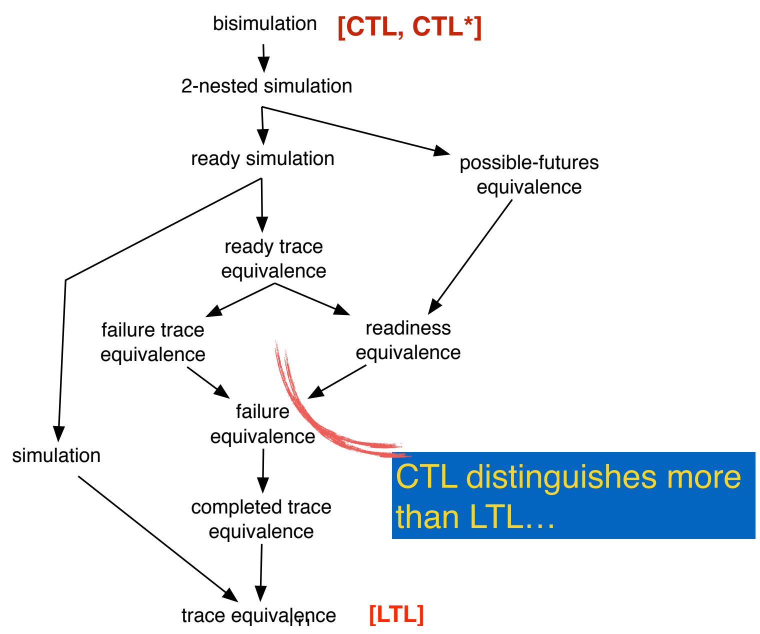
(strong) bisimulation

Let $\mathbf{S}_1 = \langle Q_1, q^0_1, R_1, \ell_1 \rangle$ and $\mathbf{S}_2 = \langle Q_2, q^0_2, R_2, \ell_2 \rangle$
A relation $\mathcal{R} \subseteq Q_1 \times Q_2$ is a bisimulation iff $\forall\, (q_1,q_2) \in \mathcal{R}$ we have:

- $\ell_1(q_1) = \ell_2(q_2)$
- $\forall\, q_1 \rightarrow_{R1} q_1'$, $\exists\, q_2 \rightarrow_{R2} q_2'$ such that $(q_1',q_2') \in \mathcal{R}$
- $\forall\, q_2 \rightarrow_{R2} q_2'$, $\exists\, q_1 \rightarrow_{R1} q_1'$ such that $(q_1',q_2') \in \mathcal{R}$

$\mathbf{S}_1$ and $\mathbf{S}_2$ are bisimilar ($\mathbf{S}_1 \approx \mathbf{S}_2$) iff there exists a bisimulation $\mathcal{R}$ such that $(q^0_1, q^0_2) \in \mathcal{R}$.

Behavioral equivalences (Van Glabbeek, 1990)



Distinguishing power

LTL distinguishing power coincides with trace equivalence.
CTL distinguinshing power coincides with strong bisimulation.

→ CTL (or CTL*) distinguish more than LTL
CTL > LTL

Characteristic formulas

Given a finite Kripke structure **S**, there exists a CTL formula Φ_S such that for any **S**', we have:

$$S' \models \Phi_S \text{ iff } S \approx S'$$

(Browne, 1988)

1) Describe the tree of depth n rooted in q:

$$\Psi^0(q) \stackrel{\text{def}}{=} \bigwedge_{P\in I(q)} P \wedge \bigwedge_{P\notin I(q)} \neg P$$

$$\Psi^{n+1}(q) \stackrel{\text{def}}{=} \Psi^0(q) \wedge \bigwedge_{q\to q'} \left(\mathbf{EX}\ \Psi^n(q')\right) \wedge \mathbf{AX}\left(\bigvee_{q\to q'} \Psi^n(q')\right)$$

2) Find c for **S** such that :

$$\Phi_S \stackrel{\text{def}}{=} \Psi^c(q_{\text{init}}) \wedge \bigwedge_{q\in Q} \mathbf{AG}\Big(\Psi^c(q) \Rightarrow \bigwedge_{q\to q'} \mathbf{EX}\ \Psi^c(q') \wedge \mathbf{AX} \bigvee_{q\to q'} \Psi^c(q')\Big)$$

Expressive power

- LTL is not as expressive as CTL.
  EX (**EX** P ∧ **EX** P') has no equivalent in LTL.
  or **AG** (**EF** init) …

- CTL is not as expressive as LTL.
  AFG P has no equivalent in CTL.
  [= $\mathbf{E}\overset{\infty}{\mathbf{F}}$ P']
  (Emerson, 1986)

LTL and CTL are uncomparable.
CTL* is strictly more expressive than CTL or LTL.