Computation Tree Logic

CTL

Formules de CTL

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

avec $P \in AP$

+ Abbréviations :

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

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

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

$\varphi\mathbf{W}\psi = \varphi\mathbf{U}\psi \vee \mathbf{G}\varphi$: "weak until"

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

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

$\mathbf{E}\varphi\mathbf{W}\psi$ $\mathbf{A}\varphi\mathbf{W}\psi$

CTL - sémantique

$\mathbf{S} = (Q, \rightarrow, q_{init}, \ell)$

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: $\rho(i) = q_i \quad \forall\ i \geq 0$

On interprète les formules de CTL sur des états de **S**.

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

$q \models \mathbf{EX}\varphi$ 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. $\exists\ i \geq 0$ t.q. ($\rho(i) \models \psi$ et $(\forall\ 0 \leq j < i: \rho(j) \models \varphi)$)

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

CTL

Définition alternative (équivalente !!):

Formules d'état:

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

$P \in AP$

Formules de chemin:

$\varphi_p,\psi_p ::= \mathbf{X}\,\varphi \mid \varphi\ \mathbf{U}\ \psi$

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

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

CTL - sémantique

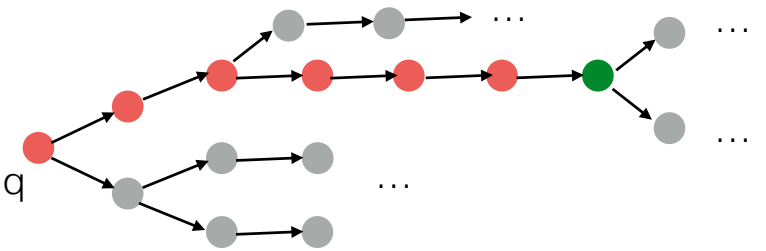
Définition alternative (équivalente !!):

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

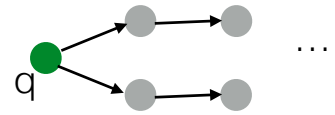
$\rho \models \mathbf{X}\ \varphi$ iff $\rho(1) \models \varphi$
$\rho \models \varphi\ \mathbf{U}\ \psi$ iff $\exists\ i \geq 0$ ($\rho(i) \models \psi$ et $(\forall\ 0 \leq j < i: \rho(j) \models \varphi)$)

CTL - semantics

$q \models \mathbf{E}$ **red** $\mathbf{U}$ **green**

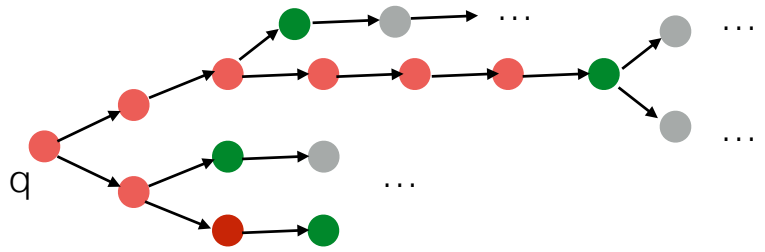


or:

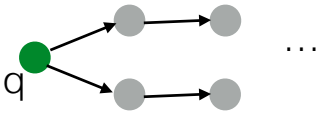


CTL - semantics

$q \models \mathbf{A}$ **red** $\mathbf{U}$ **green**

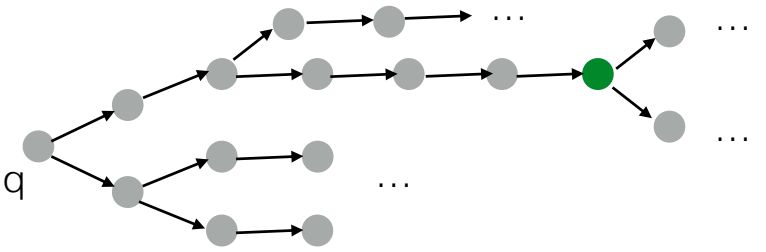


ou:

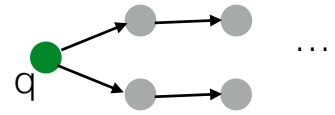


CTL - semantics

$q \models \mathbf{EF}$ **green**

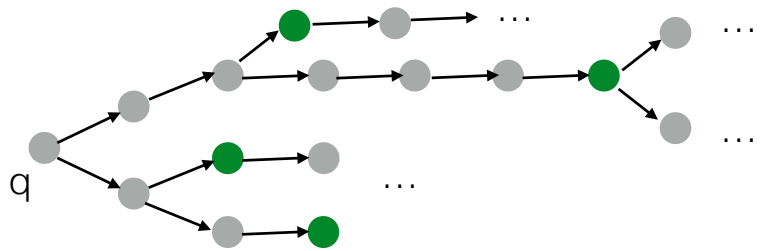


ou:

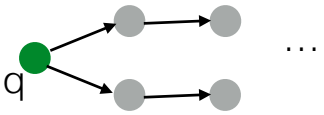


CTL - semantics

q ⊨ **AF green**

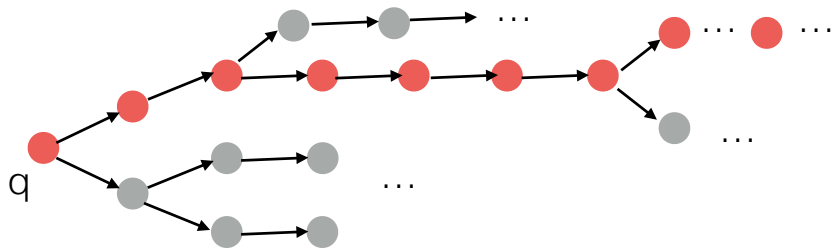


ou:



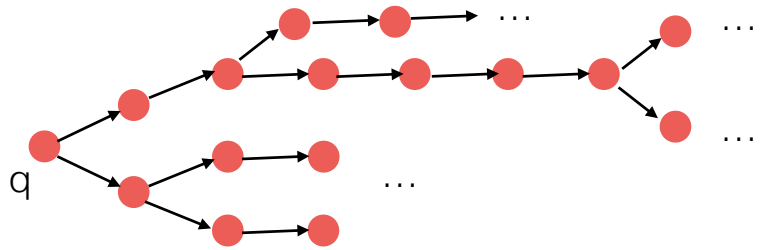
CTL - semantics

q ⊨ **EG red**



CTL - semantics

q ⊨ **AG red**



Every reachable states is red !

Exemples de formules

A G (**E F** accueil)

A F A G ok

AG (request ⇒ **AF** service)

AG AF (a ∧ b) implique **AG AF** a ∧ **AG AF** b

AG AF a ∧ **AG AF** b n'implique pas **AG AF** (a ∧ b)

Algorithme de model-checking pour CTL

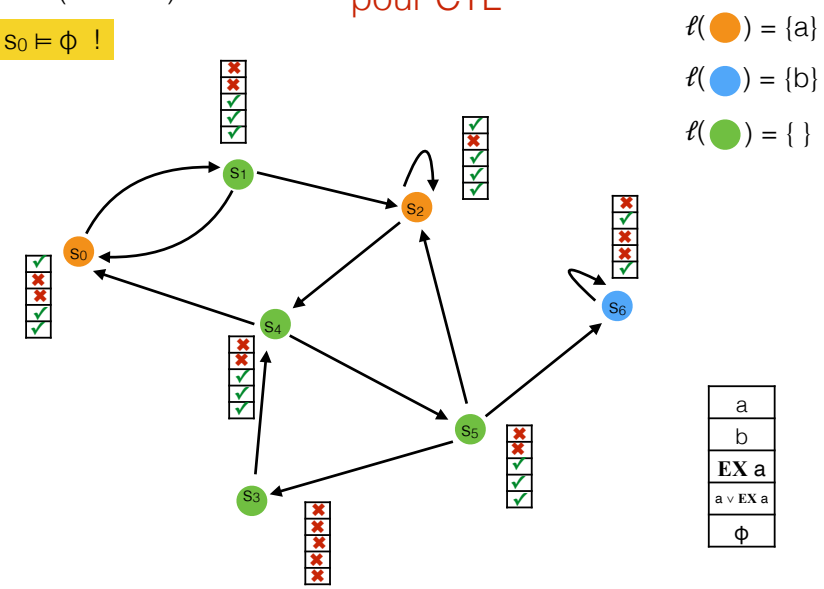
Model-checking:
input: un modèle (une structure de Kripke) $\mathbf{S}=(Q,\rightarrow,q_0,\ell)$ et une formule φ
output: oui ssi $\mathbf{S} \models \varphi$.

Idée de l'algorithme:
Indiquer pour chaque sous-formule de φ les états où elle est vraie.

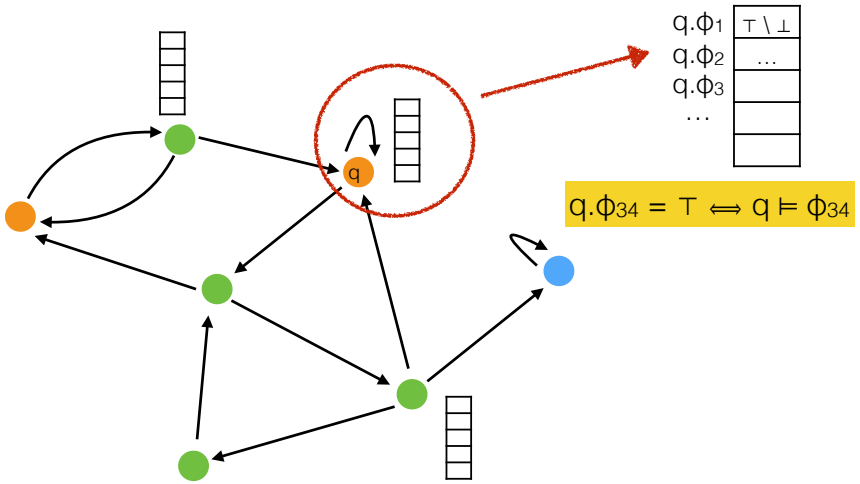
Algorithme de model-checking pour CTL

ϕ = E (a v EX a) U b

$s_0 \models \phi$!



Algorithme de model-checking pour CTL



Objectif: un algorithme de marquage pour décider si $q.\psi = \top$ ou $\bot$ pour toute sous-formule ψ et tout état q.

Algorithme de model-checking pour CTL

$\mathbf{S} = (Q, \rightarrow, q_{init}, \ell)$

Procédure Marquage(ψ):

cas 1: $\psi = P$
- Pour tout $q \in Q$:
  - Si $P \in \ell(q)$ Alors $q.\psi := \top$
  - Sinon: $q.\psi := \bot$

cas 2: $\psi = \neg \psi_1$
- Marquage(ψ_1)
- Pour tout $q \in Q$:
  - $q.\psi := \neg\, q.\psi_1$

cas 3: $\psi = \psi_1 \wedge \psi_2$
- Marquage(ψ_1), Marquage(ψ_2)
- Pour tout $q \in Q$:
  - $q.\psi := q.\psi_1 \wedge q.\psi_2$

Algorithme de model-checking pour CTL

$\mathbf{S} = (Q, \rightarrow, q_{init}, \ell)$

Procédure Marquage(ψ):

cas 4: $\psi = \mathbf{EX}\ \psi_1$
- Marquage(ψ_1)
- Pour tout $q \in Q$:
  - $q.\psi := \bot$
- Pour tout $q \rightarrow q' \in Q$:
  - Si $q'.\psi_1 = \top$ Alors $q.\psi := \top$

cas 3: $\psi = \psi_1 \wedge \psi_2$
- Marquage(ψ_1), Marquage(ψ_2)
- Pour tout $q \in Q$:
  - $q.\psi := q.\psi_1 \wedge q.\psi_2$

Algorithme de model-checking pour CTL

$\mathbf{S} = (Q, \rightarrow, q_{init}, \ell)$

Procédure Marquage(ψ):

cas 4: $\psi = \mathbf{EX}\ \psi_1$
- Marquage(ψ_1)
- Pour tout $q \in Q$:
  - $q.\psi := \bot$
- Pour tout $q \rightarrow q' \in Q$:
  - Si $q'.\psi_1 = \top$ Alors $q.\psi := \top$

Algorithme de model-checking pour CTL

$\mathbf{S} = (Q, \rightarrow, q_{init}, \ell)$

Procédure Marquage(ψ):

cas 5: $\psi = \mathbf{E}\ \psi_1\ \mathbf{U}\ \psi_2$
- Marquage(ψ_1)
- Marquage(ψ_2)
- **Pour tout** $q \in Q$:
  - $q.\psi := \bot$, $q.\text{déjàvu} := \bot$
- $L = \{\}$ // liste vide
- **Pour tout** $q \in Q$:
  - **Si** $q.\psi_2$ **Alors** $L := L + \{q\}$, $q.\text{déjàvu} := \top$
- **Tant que** $L \neq \varnothing$:
  - **Piocher** q dans L
  - $q.\psi := \top$
  - **Pour tout** $q' \rightarrow q$:
    - **Si** $\neg\, q'.\text{déjàvu}$ **Alors**
      - $q'.\text{déjàvu} := \top$
      - **Si** $q'.\psi 1$ **Alors** $L := L + \{q'\}$

Algorithme de model-checking pour CTL

$\mathbf{S} = (Q, \rightarrow, q_{init}, \ell)$

Procédure Marquage(ψ):

cas 6: $\psi = \mathbf{A}\ \psi_1\ \mathbf{U}\ \psi_2$

```
Marquage(ψ1)
Marquage(ψ2)
Pour tout q ∈ Q:
    q.ψ := ⊥ , q.nb := degré(q)
L = { }
Pour tout q ∈ Q:
    Si q.ψ2 Alors L := L + {q}
Tant que L ≠ ∅ :
    Piocher q dans L
    q.ψ := ⊤
    Pour tout q' →q :
        q'.nb := q'.nb -1
        Si q'.nb ==0 ∧ q'.ψ1 ∧ ¬ q'.ψ2 Alors L := L + {q'}
```

Algorithme de model-checking pour CTL

$|\mathbf{S}| = |Q|+|\rightarrow|$

Complexité:
→ Algorithme en $O(|\phi| \cdot |\mathbf{S}|)$
[le model-checking de CTL est un problème P-complet.]

cas 5: $\psi = \mathbf{E}\ \psi_1\ \mathbf{U}\ \psi_2$

```
Marquage(ψ1)
Marquage(ψ2)
Pour tout q ∈ Q:
    q.ψ := ⊥ , q.déjàvu := ⊥
L = { }        // liste vide
Pour tout q ∈ Q:
    Si q.ψ2 Alors L := L + {q} , q.déjàvu := ⊤
Tant que L ≠ ∅ :
    Piocher q dans L
    q.ψ := ⊤
    Pour tout q' →q :
        Si ¬ q'.— Alors
                q'.déjàvu := ⊤
                Si q'.ψ1 Alors L := L + {q'}
```

cas 6: $\psi = \mathbf{A}\ \psi_1\ \mathbf{U}\ \psi_2$

```
Marquage(ψ1)
Marquage(ψ2)
Pour tout q ∈ Q:
    q.ψ := ⊥ , q.nb := degré(q)
L = { }
Pour tout q ∈ Q:
    Si q.ψ2 Alors L := L + {q}
Tant que L ≠ ∅ :
    Piocher q dans L
    q.ψ := ⊤
    Pour tout q' →q :
        q'.nb := q'.nb -1
        Si q'.nb ==0 ∧ q'.ψ1 ∧ ¬ q'.ψ2 Alors L := L + {q'}
```