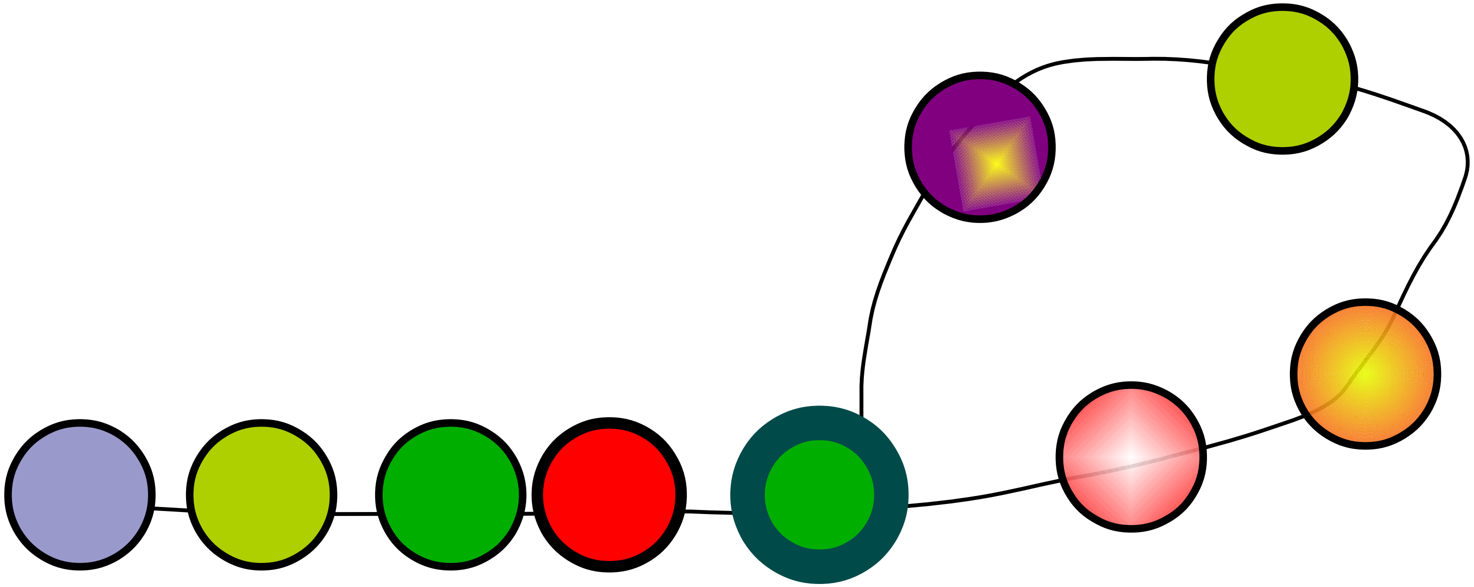
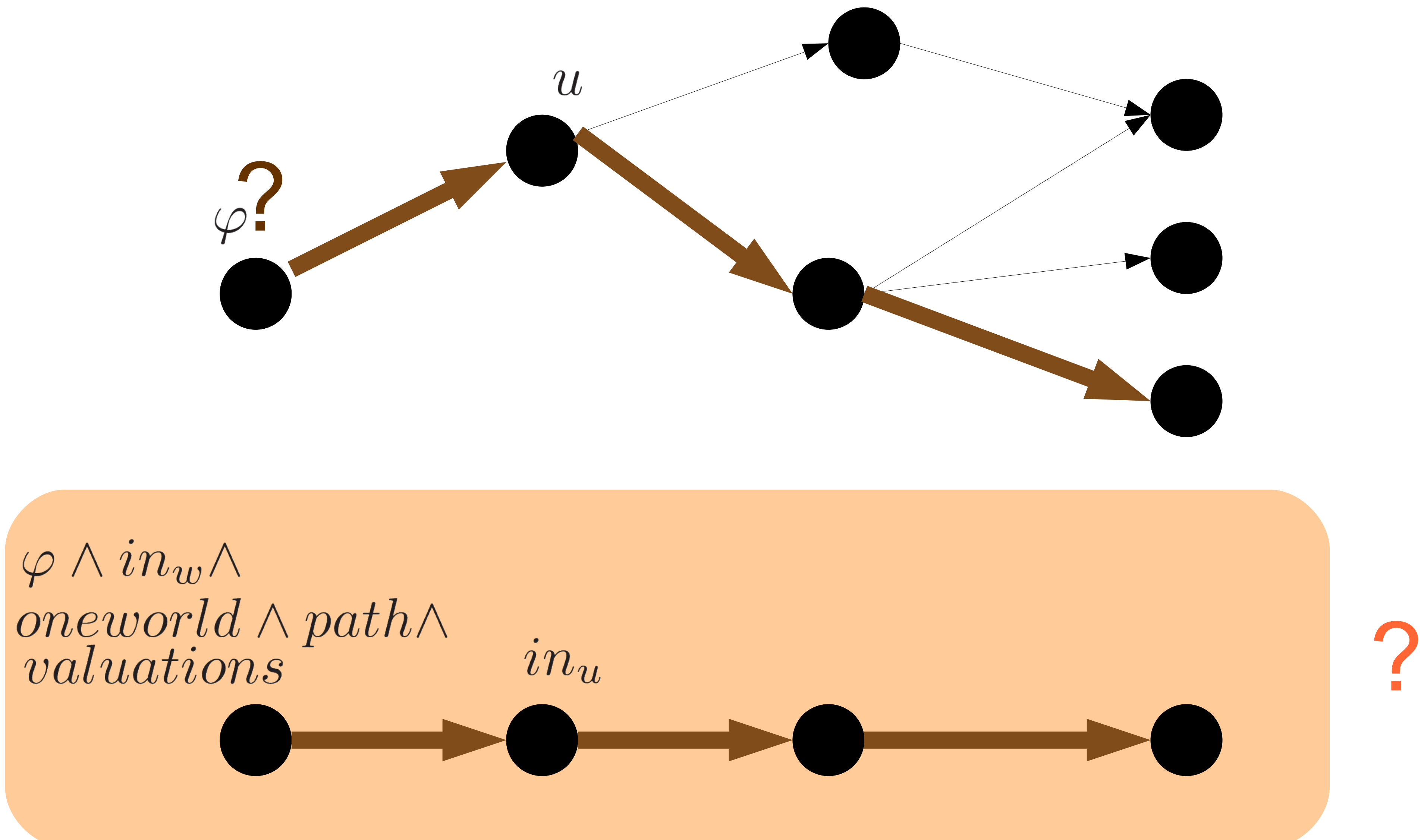
LTL-SAT

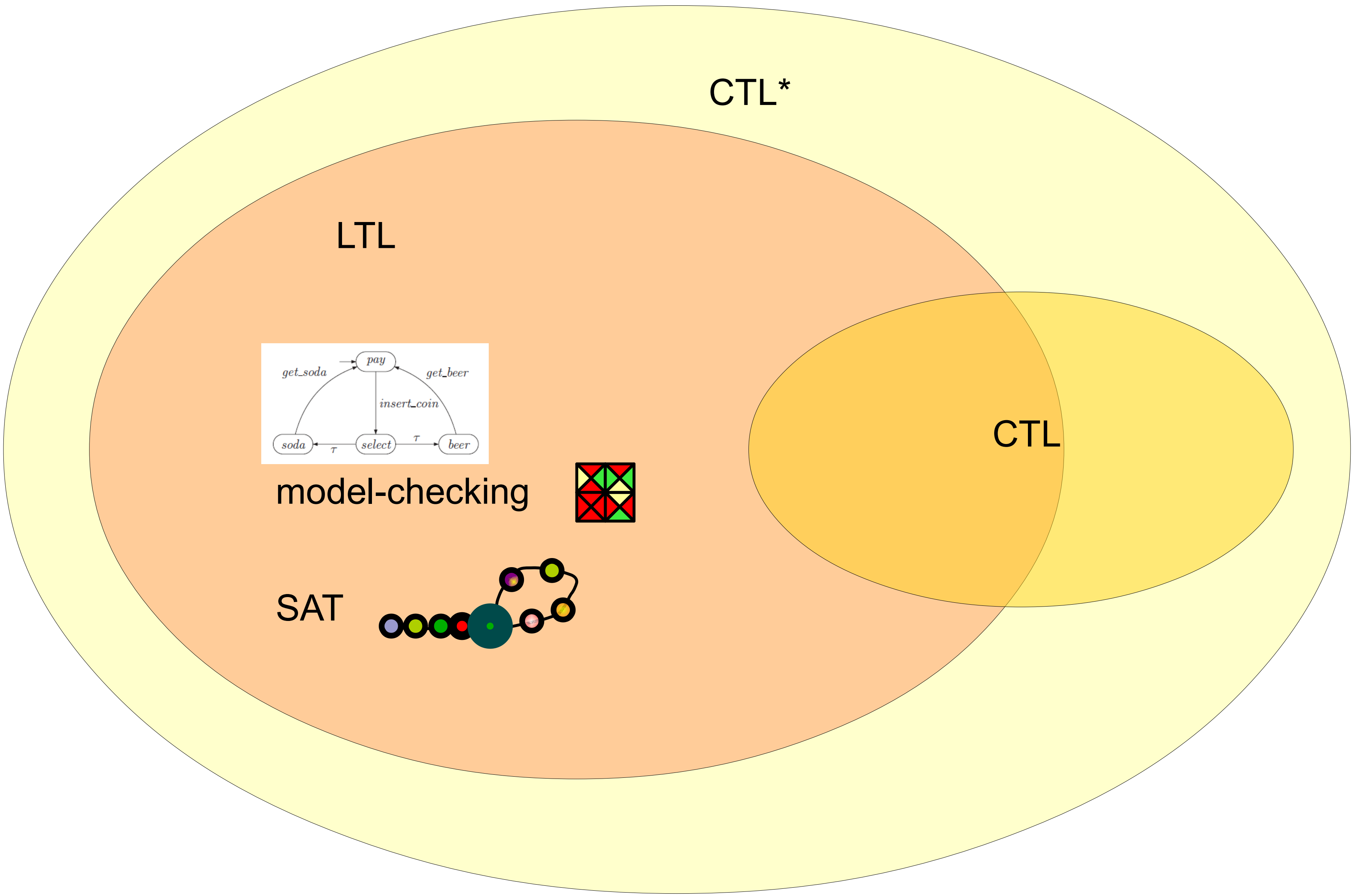


Model-checking de LTL

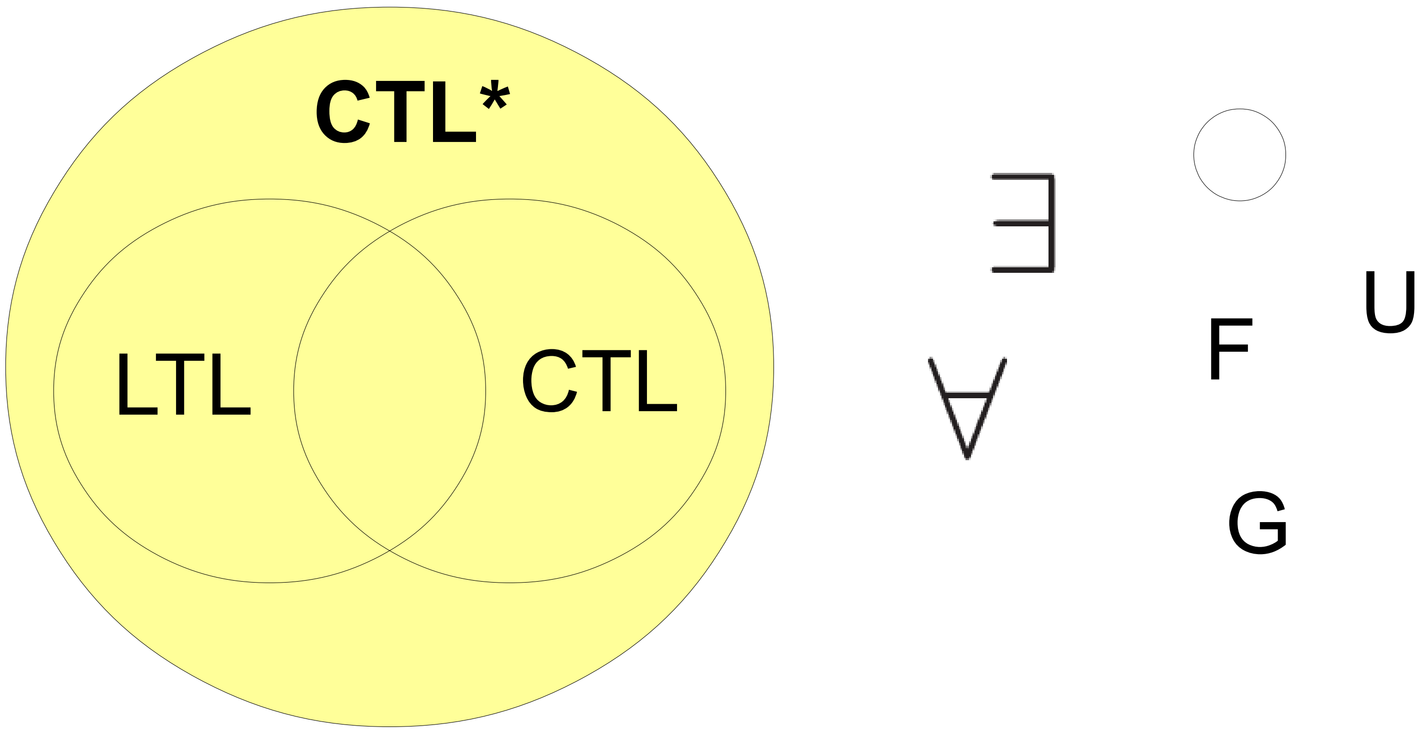


Retour vers CTL* : Computation Tree Logic

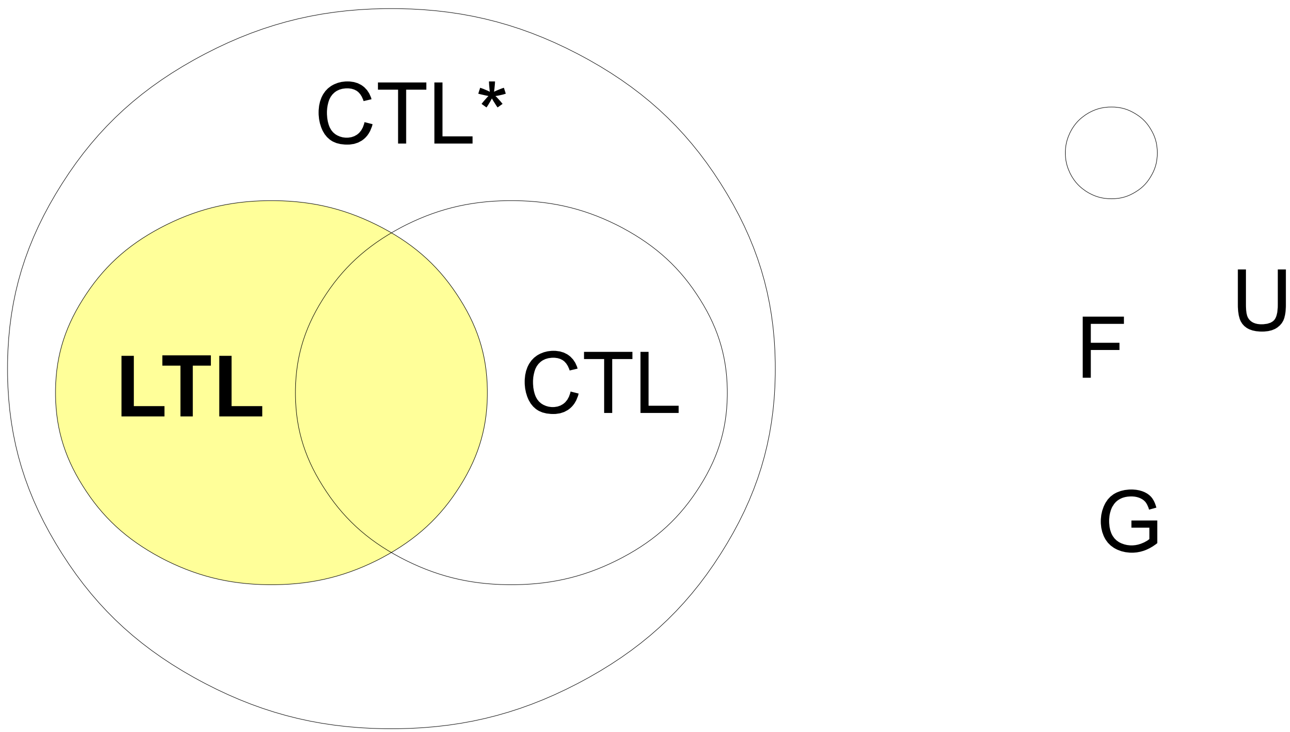
Présentation du cadre général



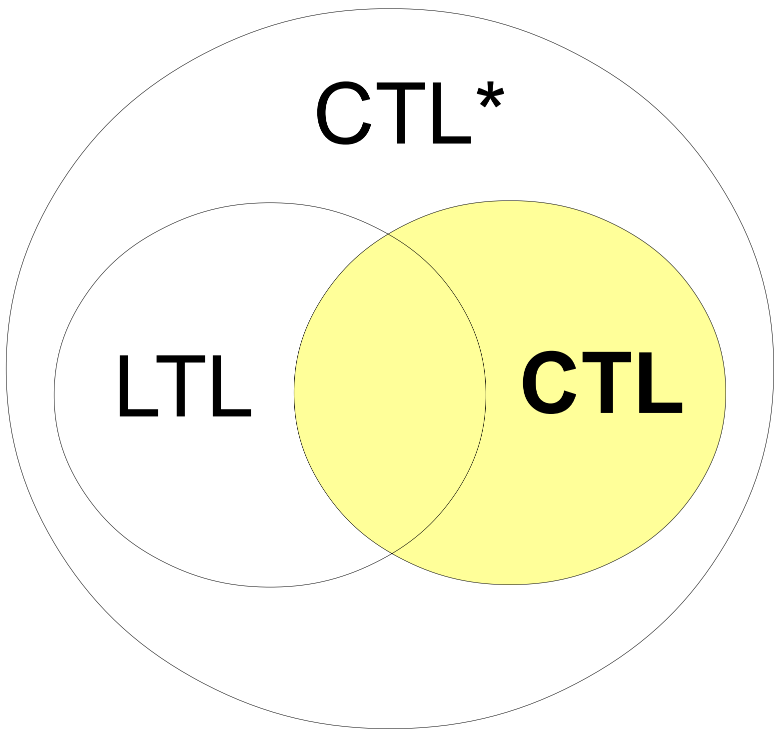
Computation Tree Logic*



Linear Temporal Logic



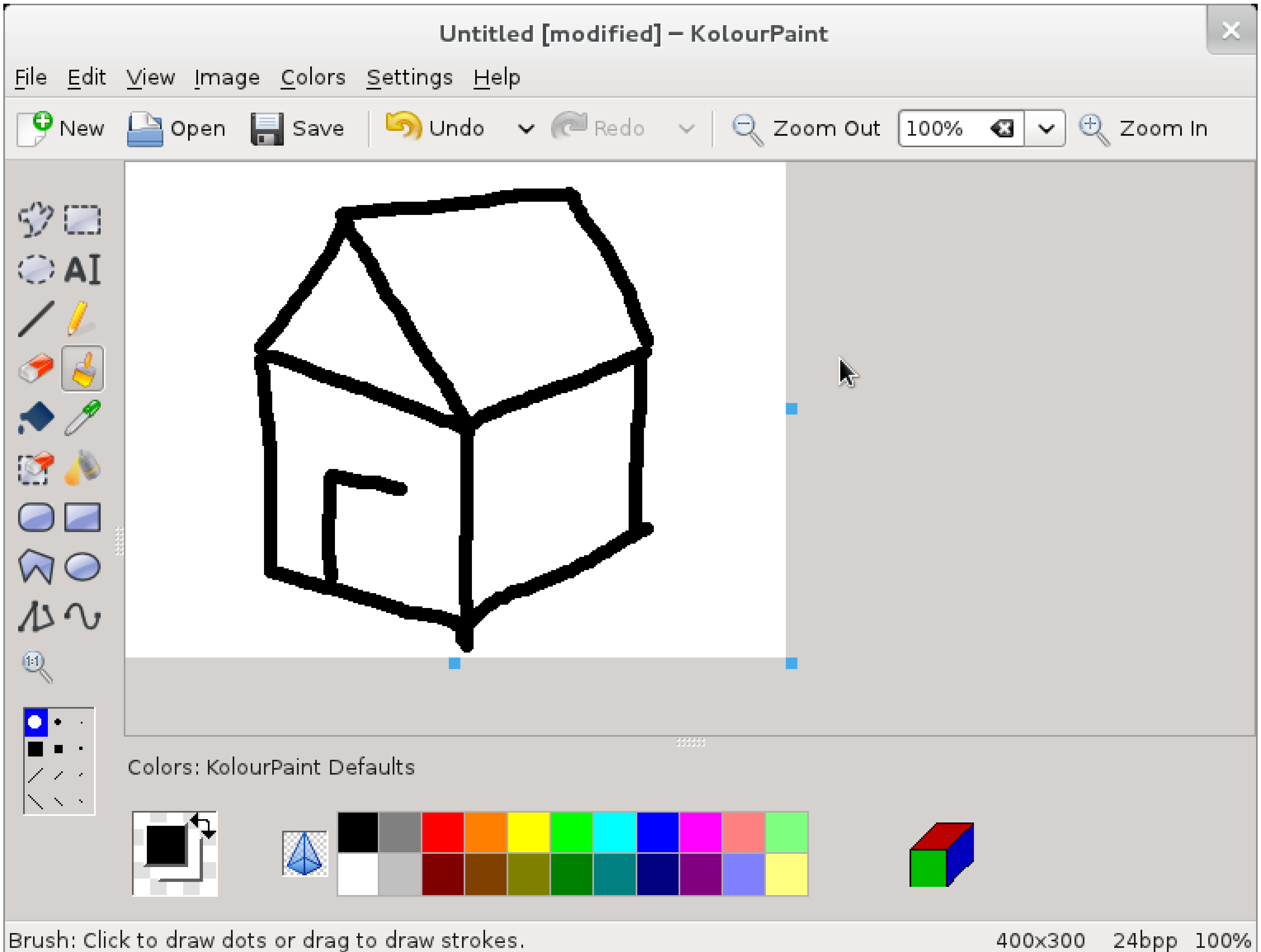
Computation Tree Logic



$\exists\bigcirc$ $\forall F$ $\exists U$ $\forall\bigcirc$ $\exists G$ $\forall F$

Model-checking dans P !

Motivation


Untitled [modified] – KolourPaint
File Edit View Image Colors Settings Help
New
Open
Save
Undo
Redo
Zoom Out
100%
Zoom In
Colors: KolourPaint Defaults
Brush: Click to draw dots or drag to draw strokes.
400x300
24bpp
100%

Propriété LTL

$$\forall G(pipette \wedge sourisappuye \rightarrow F \neg pipette)$$

$$\mathcal{M}, w \models_{\forall} G(pipette \wedge sourisappuye \rightarrow F \neg pipette)$$

Propriété LTL

$$\forall G(pipette \wedge sourisappuye \rightarrow F\neg pipette)$$

$$\mathcal{M}, w \models_{\forall} G(pipette \wedge sourisappuye \rightarrow F\neg pipette)$$

$$\forall G(pipette \wedge sourisappuye \rightarrow \forall F\neg pipette)$$

Parfois LTL ne suffit pas... on a besoin de CTL

~~$\forall G(crayon \longrightarrow F\neg crayon)$~~

$\exists G crayon$

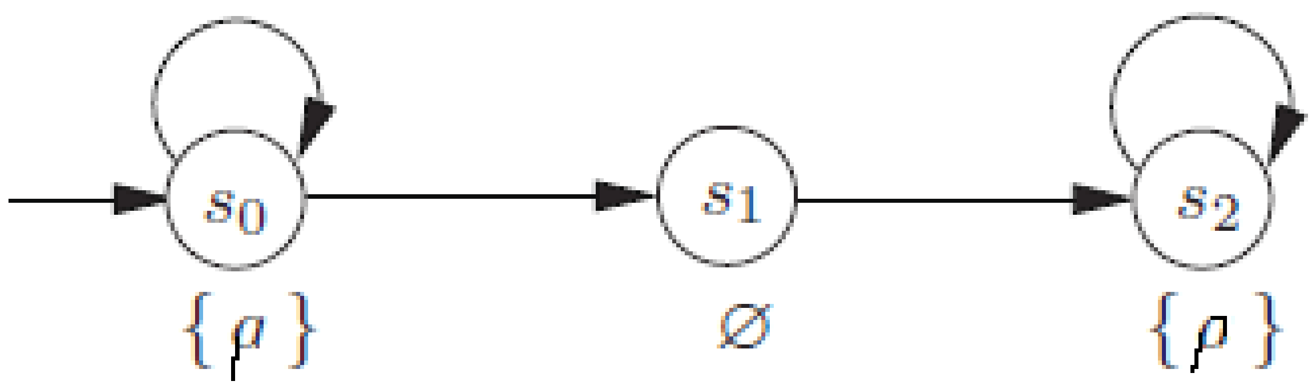
$\forall G(crayon \longrightarrow \exists F\neg crayon)$

$\forall G \exists F crayon$

Parfois LTL suffit mais pas CTL

$\forall F G p$

~~$\forall F \forall G p$~~



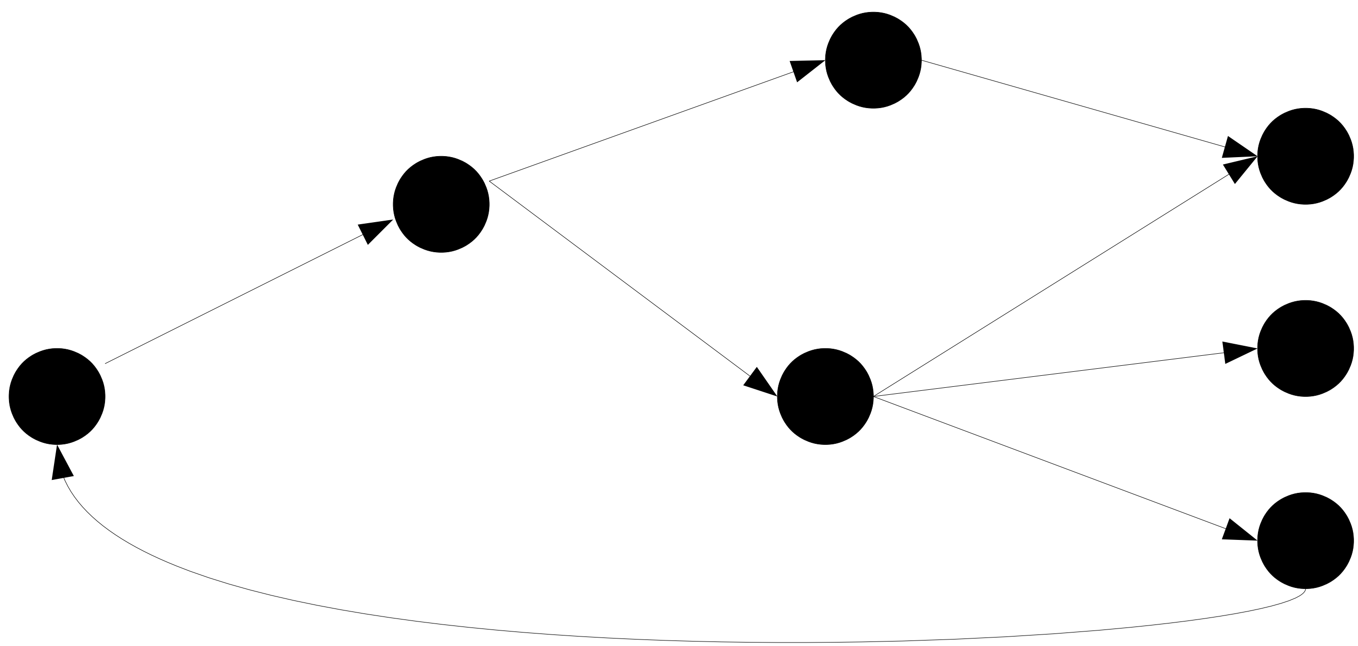
Dans de rare cas, on a besoin de CTL*

$$\forall G \exists X X crayon$$

$$\forall G \exists X \exists X crayon$$

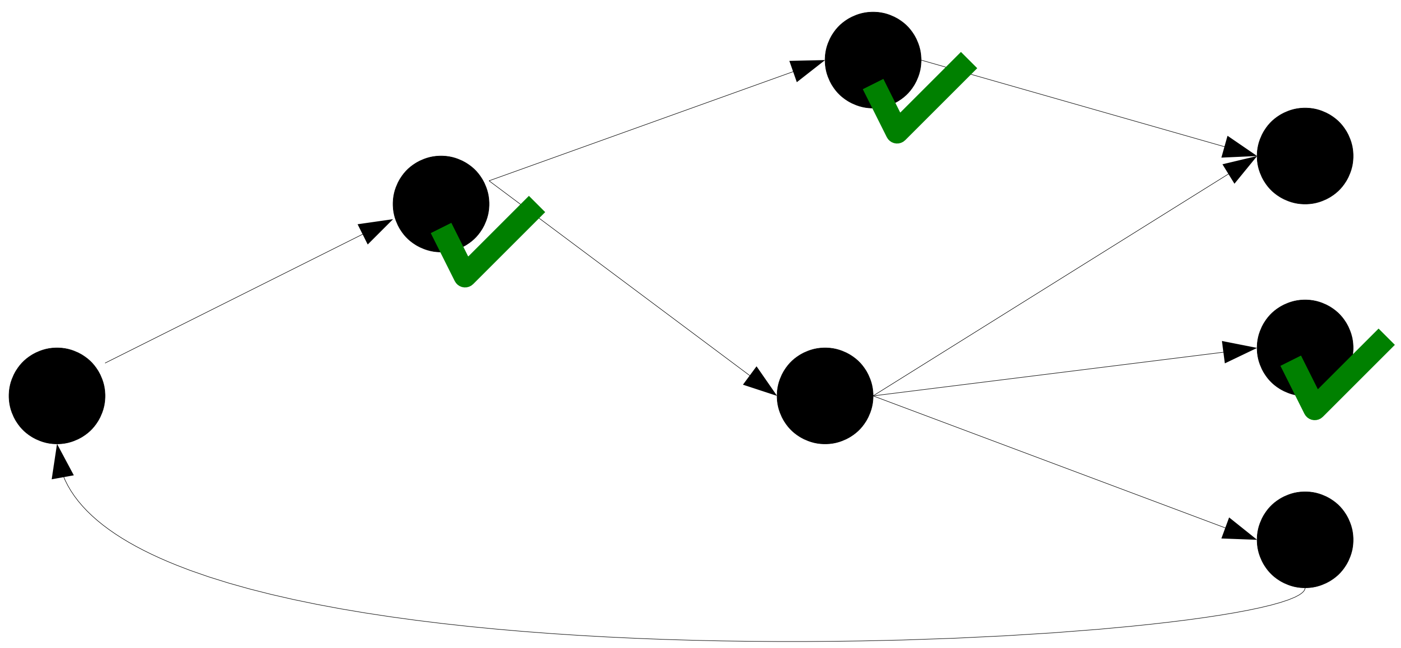
$$\forall X G \neg p \wedge \exists F G (p \vee \forall (q U p))$$

Model-checking de CTL*



$$F(p \to G\neg\exists(q \wedge G\exists FGp)) \qquad ?$$

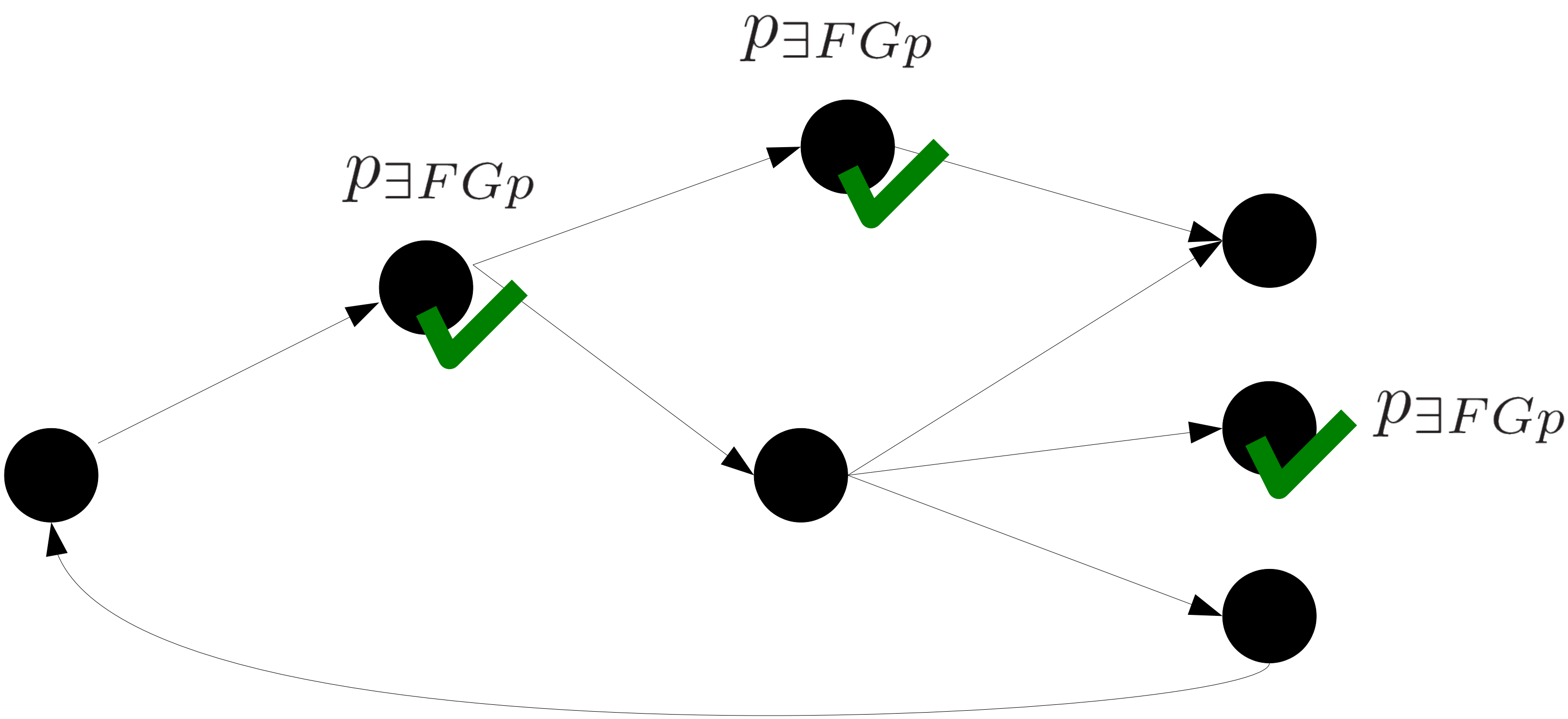
Model-checking de CTL*



$$F(p \to G\neg\exists(q \wedge G\exists FGp))$$

FGp ?

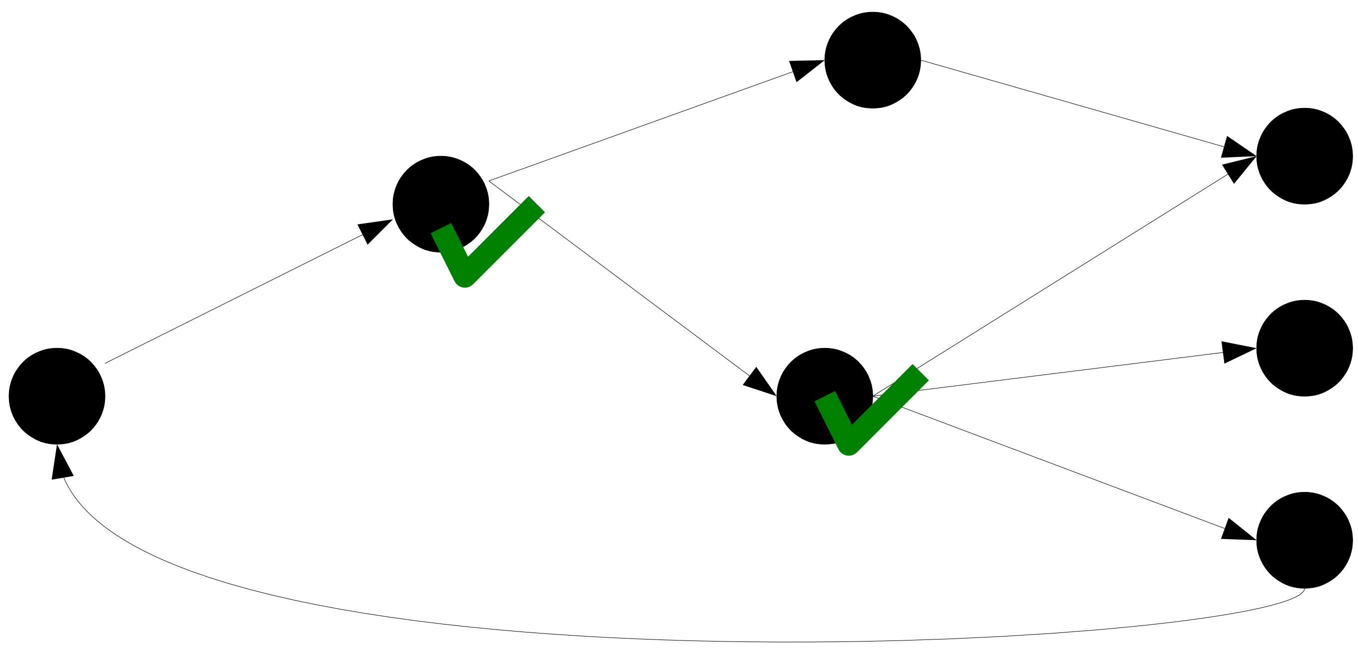
Model-checking de CTL*



$$F(p \rightarrow G\neg\exists(q \wedge Gp_{\exists FGp}))$$

?

Model-checking de CTL



$\exists F \forall (p U q)$

?

model checking CTL de $\exists \psi_1 U \psi_2$

$\exists \psi_1 U \psi_2$:

```
    S_ψ1 := mcCTL(M, ψ1)
    L_totreat := mcCTL(M, ψ2)
    result := L_treated := []
    while L_totreat ≠ ∅
        q := L_totreat.defiler
        result := result ∪ {q}
        for u → q
            if u ∉ L_treated then
                L_treated := L_treated ∪ {u}
                if u ∈ S_ψ1 then
                    L_totreat := L_totreat ∪ {u}
            endIf
        endFor
    endWhile
    return result
```

model checking CTL de $\forall \psi_1 U \psi_2$

$\forall \psi_1 U \psi_2$:

```
S_ψ1 := mcCTL(M, ψ1)
for w ∈ W, deg[w] := number of successors of w
L_totreat := mcCTL(M, ψ2)
result := []
while L_totreat ≠ ∅
    q := L_totreat.defiler
    result := result ∪ {q}
    for u → q
        deg[u] := deg[u] − 1
        if deg[u] = 0 and u ∉ result and u ∈ S_ψ1
            L_totreat := L_totreat ∪ {u}
        endIf
    endFor
endWhile
return result
```