# 9 Logiques du temps arborescent CTL$^\star$et CTL

## 9.1 Syntaxe

On se donne un ensemble fini de variables propositionnelles $\mathcal{P}$.

Les ensembles de formules d'états et de formules de chemins sont les plus petits ensembles tels que, pour toutes formules d'états $\phi_s, \psi_s$ et toutes formules de chemins $\phi_p, \psi_p$ alors

**les formules suivantes sont des formules d'états** :

- **true**, **false**
- $P$ pour $P \in \mathcal{P}$
- $\phi_s \wedge \psi_s$, $\phi_s \vee \psi_s$, $\neg\phi_s$
- $A\phi_p$, $E\phi_p$

**les formules suivantes sont des formules de chemins** :

- $\phi_s$
- $\phi_p \wedge \psi_p$, $\phi_p \vee \psi_p$, $\neg\phi_p$
- $\mathbf{X}\phi_p$
- $\phi_p\, \mathcal{U} \psi_p$

On utilise aussi comme en LTL les abréviations $\Diamond\phi \stackrel{\text{def}}{=} \mathbf{true}\, \mathcal{U}\phi$ et $\Box\phi \stackrel{\text{def}}{=} \neg(\mathbf{true}\, \mathcal{U}\neg\phi)$.

Les formules de CTL$^\star$sont les formules d'états comme défini ci-dessus. CTL est le fragment dans lequel toute sous-formule de chemin $\mathbf{X}\phi_p$ ou $\phi_p\, \mathcal{U}\psi_p$ est immédiatement précédée d'un quantificateur de chemins $A$ ou $E$.

**Exemple 43** $\Box\Diamond P$ est une formule de CTL$^\star$ mais pas de CTL.

## 9.2 Sémantique

Soit $\mathcal{K}$ une structure de Kripke. $T_{\mathcal{K}}$ est le dépliage de $\mathcal{K}$ : c'est un arbre infini étiqueté par les positions de $\mathcal{K}$ . Un *chemin* d'origine $s$ de $T_{\mathcal{K}}$ est une suite de positions de $\mathcal{K}$ : c'est une éxecution de $\mathcal{K}$ à partir de $s$.

$\mathcal{K} \models \phi_s$ ssi $T_{\mathcal{K}}, \epsilon \models \phi_s$.

Dans la suite, $w$ désigne une position de $T_{\mathcal{K}}$ et $n$ un entier.

**formules d'états** $\phi_s, \psi_s$

- $T, w \models$ **true**, et $T, w \not\models$ **false**
- $T, w \models P$ ssi $P \in f(T(w))$
- $T, w \models \phi_s \wedge \psi_s$ ssi $T, w \models \phi_s$ et $T, w \models \psi_s$, etc...
- $T, w \models A\phi_p$, ssi pour tout chemin $\pi$ de $T$ d'origine $w$, $\pi, 0 \models_w \phi_p$
- $T, w \models E\phi_p$ ssi il existe un chemin $\pi$ de $T$ d'origine $w$ tel que $\pi, 0 \models_w \phi_p$

**formules de chemins** $\phi_p, \psi_p$

- $\pi, n \models_{w_0} \phi_s$ ssi $T, w \models \phi_s$ où $w = w_0 \cdot w_1$ et $w_1$ est le préfixe de $\pi$ de longueur $n$.
- $\pi, n \models_{w_0} \phi_p \wedge \psi_p$ ssi $\pi, n \models_{w_0} \phi_p$ et $\pi, n \models_{w_0} \psi_p$, etc ...
- $\pi, n \models_{w_0} \mathbf{X}\phi_p$ ssi $\pi, n+1 \models_{w_0} \phi_p$
- $\pi, n \models_{w_0} \phi_p\, \mathcal{U}\psi_p$ ssi il existe $n' \geq n$ tel que
  - $\pi, n' \models_{w_0} \psi_p$
  - pour tout $n'', n \leq n' < n''$, $\pi, n'' \models_{w_0} \phi_p$

**Exercice 36**
Soit $\mathcal{K}$ la structure de Kripke donnée dans la figure 20. Représenter visuellement son dépliage et dire (en le justifiant) si $\mathcal{K}$ vérifie les formules suivantes :

1. $A\Diamond A\Box P$.
2. $A\Diamond\Box P$
3. $E\Box P$
4. $EA\Box P$

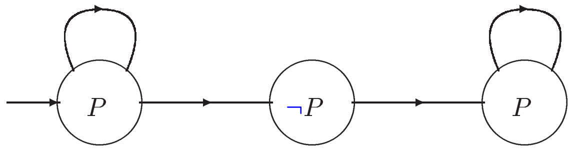


FIG. 20 – Une structure de Kripke

5. $A\mathbf{X}E\mathbf{X}\Box P$

CTL$^\star$est au moins aussi expressive que LTL qu'elle contient (ce sont les formules de chemins précédées du quantificateur $A$) mais il y a nombre de propriétés qui sont expressibles en CTL$^\star$et pas en LTL. Par exemple les propriétés de vivacité comme

$$A\Box(P \Rightarrow E\Diamond Q)$$

"Quand $P$ est vrai, il est possible d'obtenir $Q$ ensuite".

**Exercice 37**

1. Montrer que $A\Box\Diamond P$ n'a pas les mêmes modèles que $A\Box E\Diamond P$.
2. Montrer que $A\Box\Diamond P$ est logiquement équivalente à une formule de CTL que l'on précisera.

Par contre, $A\Diamond\Box P$ n'est équivalente à aucune formule de CTL (voir exercice 43).

## 9.3 Traduction en SkS

On considère seulement ici les structures de Kripke à branchement borné par un nombre fixe $k$. Nous verrons (lemme 49) que ce n'est pas une restriction. Par abus de notation, si K est une structure de Kripke, on note encore $t_\mathcal{K}$ le dépliage de K dans lequel les étiquettes $s \in S$ sont remplacées par $f(s) \in 2^\mathcal{P}$. On ordonne de plus arbitrairement les variables propositionnelles : $\mathcal{P} = \{P_1, \ldots, P_n\}$.

**Theorème 44** *Pour toute structure de Kripke de branchement au plus $k$ et toute formule $\phi$ de* CTL$^\star$*, il existe une formule $\widetilde{\phi}(t, X_1, \ldots, X_n)$ de SkS (effectivement calculable en temps linéaire) telle que $\mathcal{K} \models \phi$ si et seulement si $\epsilon, t_\mathcal{K} \models \widetilde{\phi}(t, X_1, \ldots, X_n)$*

**Preuve**: On peut toujours coder un nombre arbitraire (fini) de successeurs à l'aide de deux successeurs, l'image étant définissable en S2S. On se limite ainsi à deux successeurs.

On code les formules d'états comme des formules de S2S ayant une variable libre de premier ordre $t$ et les formules de chemins comme des formules ayant une variable libre de second ordre $C$. ($X_1, \ldots, X_n$ qui sont libres dans toutes les formules sont sous-entendus).

On utilise aussi les prédicats Origine,Suc et Chaîne, qui sont définissables en S2S, comme nous l'avons vu en exercice.

**formules d'états** $\phi_s, \psi_s$

- $\widetilde{\mathbf{true}}(t) \stackrel{\text{def}}{=} \mathbf{true}$, $\widetilde{\mathbf{false}}(t) \stackrel{\text{def}}{=} \mathbf{false}$
- $\widetilde{P_i}(t) \stackrel{\text{def}}{=} t \in X_i$
- $\widetilde{\phi_s \wedge \psi_s}(t) \stackrel{\text{def}}{=} \widetilde{\phi_s}(t) \wedge \widetilde{\psi_s}(t)$, etc...
- $\widetilde{A\phi_p}(t) \stackrel{\text{def}}{=} \forall C.\text{Chaîne}(C, t) \Rightarrow \widetilde{\phi_p}(C)$, etc...

**formules de chemins** $\phi_p, \psi_p$

- $\widetilde{\phi_s}(C) \stackrel{\text{def}}{=} \exists t.\text{Origine}(t, C) \wedge \widetilde{\phi_s}(t)$

- $\widetilde{\phi_p \wedge \psi_p}(C) \stackrel{\text{def}}{=} \widetilde{\phi_p}(C) \wedge \widetilde{\psi_p}(C)$, etc ...
- $\widetilde{\mathbf{X}\phi_p}(C) \stackrel{\text{def}}{=} \exists C'.\mathrm{Suc}(C, C') \wedge \widetilde{\phi_p}(C')$
- $\widetilde{\phi_p\, \mathcal{U} \psi_p}(C) \stackrel{\text{def}}{=} \exists t.t \in C \wedge (\exists C'.\mathrm{Origine}(t, C') \wedge \widetilde{\psi_p}(C') \wedge \forall t' \in C, \exists C''.t' \geq t \vee (\mathrm{Origine}(t', C'') \wedge \widetilde{\phi_p}(C'')))$

La vérification de l'équivalence est alors routinière. $\square$