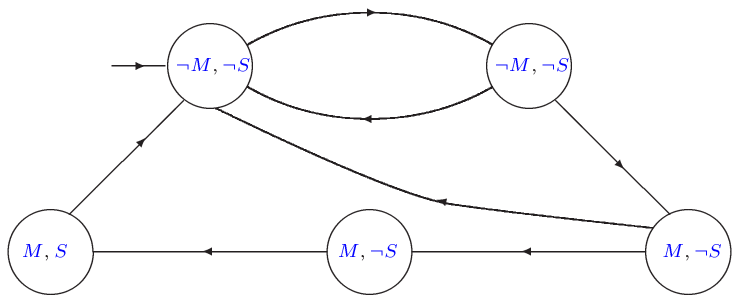

Fig. 2 – Une structure de Kripke finie

2 Structures de Kripke

2.1 Définition

$\mathcal{K} = (S, R, s^0, f, \mathcal{P})$ où $R \subseteq S \times S$, $s^0 \in S$ et $f : S \to \mathcal{P}$. On suppose que, pour tout $s \in S$, il existe au moins un $s' \in S$ tel que $R(s, s')$.

Exemple 3 Si l'on ne tient pas compte des actions et que l'on étiquette chacun des états de l'automate de la figure 1 par un sous-ensemble de $\{M, S\}$, on obtient l'automate et le marquages comme indiqués sur la figure 2 ; M est supposé indiquer que l'on a déjà mis deux francs dans la machine et S que le café est servi.

Exemple 4 Si l'on complique un peu le modèle de la machine à café en acceptant de pouvoir mettre un nombre arbitraire de pièces, on obtient le modèle suivant : $S = \{s^i \mid i \in \mathbb{N}\} \cup \{t^i \mid i \geq 2\} \cup \{u^i \mid i \geq 2\}$,

$$\begin{array}{lcll} s^i & \to & s^{i+1} & \text{pour } i \in \mathbb{N} \\ s^i & \to & s^0 & \text{pour } i \in \mathbb{N} \\ s^i & \to & t^i & \text{pour } i \geq 2 \\ t^i & \to & u^i & \text{pour } i \geq 2 \\ u^i & \to & s^{i-2} & \text{pour } i \geq 2 \end{array}$$

et $f(t^i) = \{M\}$, $f(u^i) = \{S, M\}$ et $f(s^i) = \emptyset$.

2.2 Plusieurs modalités

Si l'on veut tenir compte des émissions/réceptions de messages, on peut considérer le modèle plus général (plusieurs modalités) : on considère un nombre fini d'actions $A = \{a_1, \ldots, a_n\}$ et une relation binaire par action. Par exemple, si on reprend l'automate de la figure 1, on peut considérer les relations binaires $R_{?franc}, R_{?back}, \ldots$.

Elargir le modèle à plusieurs modalités ne présente pas beaucoup d'intérêt car on peut "traduire" le cas multimodal en une seule modalité en remplaçant S par $S \times A$ (soit en devinant à chaque étape l'action qui va suivre, soit en se rappelant de l'action qui précède).

2.3 Executions et dépliage

Une *exécution* de la structure de Kripke est une séquence infinie $w \in S^\omega$ telle que pour tout $i \in \mathbb{N}$, $w(i) R w(i+1)$ et $w(0) = s^0$

Soit E un ensemble. Un sous-ensemble D de E^* est un *domaine d'arbre* s' il est clos par préfixe.

Un *arbre* est une application T d'un domaine d'arbre D dans un ensemble d'étiquettes A.

Si $\mathcal{K}$ est une structure de Kripke, son *dépliage* est un arbre (éventuellement infini) étiqueté par S, défini comme suit : $\epsilon \in D$ et $T(\epsilon) = s^0$. Si $p \in D$ et $T(p) = s$ et E est l'ensemble des éléments s_i de S tels que sRs_i alors $p \cdot (s, s_i) \in D$ pour tout $s_i \in E$ et, pour tout i, $T(p \cdot (s, s_i)) = s_i$.

Le dépliage à partir de $s \in S$ s'obtient de la même façon en remplaçant s^0 par s.

3 Logique du temps linéaire (LTL)

3.1 Syntaxe

Etant donné un ensemble de variables propositionnelles $\mathcal{P}$, l'ensemble des formules de LTL est le plus petit ensemble tel que, si ϕ, ψ sont des formules, alors

$\mathbf{true}$, $\mathbf{false}$

P $(P \in \mathcal{P})$

$\mathbf{X}\phi$

$\phi \wedge \psi$, $\phi \vee \psi$

$\neg\phi$

$\phi\,\mathcal{U}\psi$

sont des formules.

On utilise couramment les abréviations suivantes :

$\Diamond\phi \stackrel{\text{def}}{=} \mathbf{true}\,\mathcal{U}\phi$

$\Box\phi \stackrel{\text{def}}{=} \neg(\mathbf{true}\,\mathcal{U}\neg\phi)$

3.2 Sémantique

Soit $\mathcal{K}$ une structure de Kripke. Par définition, $\mathcal{K}$ satisfait ϕ si, pour toute exécution w de $\mathcal{K}$, $(w, 0) \models \phi$, où $\models$ est définie par :

$w, t \models \mathbf{true}$ et $w, t \not\models \mathbf{false}$

$w, t \models P$ ssi $P \in f(w(t))$

$w, t \models \mathbf{X}\phi$ ssi $w, t+1 \models \phi$

$w, t \models \phi \wedge \psi$ ssi $w, t \models \phi$ et $w, t \models \psi$

$w, t \models \phi \vee \psi$ ssi $w, t \models \phi$ ou $w, t \models \psi$

$w, t \models \neg\phi$ ssi $w, t \not\models \phi$.

$w, t \models \phi\,\mathcal{U}\psi$ ssi
- il existe $t' \geq t$ tel que $w, t' \models \psi$
- et pour tout $t'' \in [t, t'[$ on a $w, t'' \models \phi$.

Exercice 2 On note $\phi \equiv \psi$ lorsque ϕ et ψ ont les mêmes modèles. Démontrer ou invalider les équivalences suivantes :

1. $\mathbf{X}\neg\phi \equiv \neg\mathbf{X}\phi$
2. $\mathbf{X}\Diamond\phi \equiv \Diamond\mathbf{X}\phi$
3. $\Box\mathbf{X}\phi \equiv \mathbf{X}\Box\phi$
4. $\mathbf{X}\phi \wedge \mathbf{X}\psi \equiv \mathbf{X}(\phi \wedge \psi)$
5. $\mathbf{X}\phi \vee \mathbf{X}\psi \equiv \mathbf{X}(\phi \vee \psi)$
6. $(\mathbf{X}\phi)\,\mathcal{U}\psi \equiv \psi \vee \mathbf{X}(\phi\,\mathcal{U}(\phi \wedge \psi))$
7. $\neg\Diamond\phi \equiv \Box\neg\phi$
8. $\neg\Box\phi \equiv \Diamond\neg\phi$

Exercice 3 Montrer que $\mathcal{U}$ n'est pas associatif :

$$(\phi_1\,\mathcal{U}\phi_2)\,\mathcal{U}\phi_3 \not\equiv \phi_1\,\mathcal{U}(\phi_2\,\mathcal{U}\phi_3)$$

Il suffira d' exhiber des formules ϕ_1, ϕ_2 et ϕ_3 bien choisies.

3.3 Langage défini par une formule de LTL

Un *mot infini* sur l'alphabet A est une application de $\mathbb{N}$ dans A. L'ensemble des mots infinis construits sur A est noté A^ω. Si $w \in A^\omega$, $w(i)$ pour $i \in \mathbb{N}$ est la $i$ème lettre de w.

L'ensemble des variables propositionnelles étant ordonné de manière arbitraire, une affectation de ces variables est identifiée à un mot de $\{0,1\}^n$. La notion de satisfaction d'une formule ϕ de LTL pour un mot $w \in (\{0,1\}^k)^\omega$ est définie comme dans la section 3.2, la fonction f étant définie par $f(\alpha) = \{P_i \mid \alpha(i) = 1\}$.

Le *langage* L_ϕ défini par une formule LTL ϕ est l'ensemble des mots $w \in (\{0,1\}^k)^\omega$ qui satisfont ϕ.

Exemple 5 Supposant qu'il y a une seule proposition P, si ϕ est la formule $\Box\Diamond P$, $L_\phi = (0^*1)^\omega$, c'est à dire l'ensemble des mots contenant une infinité de 1.

De même, si $\mathcal{K}$ est une structure de Kripke, on note $L_{\mathcal{K}}$ le langage qu'elle définit sur l'alphabet $\{0,1\}^k$: $L_{\mathcal{K}}$ est l'ensemble des images par f des exécutions de $\mathcal{K}$.

On peut alors reformuler les problèmes de satisfaisabilité et de vérification de modèles dans ce contexte :

$$\boxed{\phi \text{ est satisfaisable} \quad \Leftrightarrow \quad L_\phi \neq \emptyset}$$

$$\boxed{\mathcal{K} \models \phi \quad \Leftrightarrow \quad L_{\mathcal{K}} \subseteq L_\phi}$$

3.4 logique monadique du premier ordre à un successeur

On considère la logique du premier ordre *monadique*, à un successeur construite sur un ensemble fini de prédicats unaires $\mathcal{Q}$: les termes sont construits à partir de $0, S$ et des variables. Les formules atomiques sont de la forme $t \geq t'$ ou bien $t = t'$ ou bien $P(t)$, $P \in \mathcal{Q}$, t, t' étant des termes.

On interprète $0, S, \geq, =$ dans les entiers naturels : S est le successeur. Si $\mathcal{Q} = \{P_1, \ldots, P_k\}$, une interprétation est représentée par k ensembles d'entiers, ou, si l'on superpose leurs fonctions caractéristiques, par des mots de $(\{0,1\}^k)^\omega$.

Par exemple, si $\mathcal{Q} = \{P_1, P_2\}$, et $\phi \stackrel{\text{def}}{=} \forall t. P_1(0) \wedge P_1(t) \Rightarrow P_2(S(t)) \wedge P_2(t) \Rightarrow P_1(S(t))$, alors $w \stackrel{\text{def}}{=} {}^{1}_{0}\,{}^{0}_{1}\,{}^{1}_{0}\,{}^{0}_{1} \ldots = ({}^{1}_{0}\,{}^{0}_{1})^\omega$ satisfait ϕ.

Exercice 4 Donner d'autres modèles de la formule ϕ ci-dessus.

3.5 Traduction de LTL en logique du premier ordre

Si ϕ est une formule de LTL construite sur les variables propositionnelles $\mathcal{P}$, on construit une formule $\tilde{\phi}(t)$ de la logique monadique de premier ordre sur les prédicats unaires $\{Q_P \mid P \in \mathcal{P}\}$, avec une variable libre t de telle manière que, pour toute structure de Kripke $\mathcal{K}$ (sur l'ensemble de variables propositionnelles $\mathcal{P}$) :

$$\mathcal{K} \models \phi \quad \Leftrightarrow \quad L_{\mathcal{K}} = \{w \mid w \models \tilde{\phi}(0)\}$$

$\widetilde{\mathbf{true}} \stackrel{\text{def}}{=} \mathbf{true}$

$\widetilde{\mathbf{false}} \stackrel{\text{def}}{=} \mathbf{false}$

$\widetilde{P}(t) \stackrel{\text{def}}{=} Q_P(t)$.

$\widetilde{\phi \wedge \psi}(t) \stackrel{\text{def}}{=} \widetilde{\phi}(t) \wedge \widetilde{\psi}(t)$

$\widetilde{\phi \vee \psi}(t) \stackrel{\text{def}}{=} \widetilde{\phi}(t) \vee \widetilde{\psi}(t)$

$\widetilde{\neg\phi}(t) \stackrel{\text{def}}{=} \neg\widetilde{\phi}(t)$

$\widetilde{\mathbf{X}\phi}(t) \stackrel{\text{def}}{=} \widetilde{\phi}(t+1)$

$\widetilde{\phi\, \mathcal{U} \psi}(t) \stackrel{\text{def}}{=} \exists t'. t' \geq t \wedge \widetilde{\psi}(t') \wedge \forall t''. (t \leq t'' < t' \Rightarrow \widetilde{\phi}(t''))$

Theorème 6 (Kamp) *L est définissable en logique monadique du premier ordre si et seulement si L est définissable en LTL.*

Plus précisément : on peut traduire les formules de LTL en logique monadique du premier ordre (c'est ce qui est fait dans le paragraphe précédent) et réciproquement (on l'admettra), il est possible de traduire toute formule du premier ordre monadique en une formule de LTL qui définit le même langage. Cette dernière traduction est, en fait, non élémentaire.