des étiquettes des positions de longueur i de ρ. Si $\rho(p) = q$, $w(|p|) = a$ et $r(|p|) = (S, S')$, $r(|p|+1) = (S_1, S'_1)$, par définition, il existe un sous-ensemble de $S_1 \cup S'_1$, soit S_q, tel que $\bigwedge_{q' \in S_q} q' \models \delta(q, a)$. De plus, si $q \in S_i \setminus F$, on peut supposer $S_q \subseteq S_{i+1}$. Alors, on ajoute à ρ les positions $p \cdot 1 \ldots p \cdot m$ ($m = |S_q|$) étiquetées par les éléments de S_q. ρ est bien un calcul par construction. Il reste à montrer que chaque branche contient une infinité d'états finaux. Par l'absurde : supposons qu'il existe un chemin infini C de ρ qui ne contient qu'un nombre fini d'états finaux : pour $p \in C$ et $|p| > n$, $\rho(p) \notin F$. Par récurence sur la longueur de p, on montre $\rho(p) \in S_i \setminus F$ puisqu'alors $S_{\rho(p)} \subseteq S_{i+1}$ et $S_{\rho(p)} \cap C \neq \emptyset$. On obtient ainsi que, pour $i > n$, $S_i \neq \emptyset$, ce qui est absurde. □

5.2 LTL et automates alternants

La correspondance entre automates alternants et formules temporelles est donné par le résultat très simple :

Theorème 17 *Soit ϕ une formule de LTL. Il existe un automate de Büchi alternant $\mathcal{A}_\phi$ dont le nombre d'état est $O(|\phi|)$ qui accepte exactement les mots de $(\{0,1\}^k)^\omega$ qui satisfont ϕ.*

Preuve: Les états seront les sous-formules de ϕ et leur négation. Les états finaux sont les formules de la forme $\neg(\phi_1 \,\mathcal{U} \psi_1)$. L'état initial est ϕ.

La relation de transition est donnée par :

$\delta(p, a) \stackrel{\text{def}}{=} \mathbf{true}$ si $p \in a$

$\delta(p, a) \stackrel{\text{def}}{=} \mathbf{false}$ si $p \notin a$

$\delta(\neg p, a) \stackrel{\text{def}}{=} \mathbf{true}$ si $p \notin a$

$\delta(\neg p, a) \stackrel{\text{def}}{=} \mathbf{false}$ si $p \in a$

$\delta(\phi_1 \wedge \phi_2, a) \stackrel{\text{def}}{=} \delta(\phi_1, a) \wedge \delta(\phi_2, a)$

$\delta(\phi_1 \vee \phi_2, a) \stackrel{\text{def}}{=} \delta(\phi_1, a) \vee \delta(\phi_2, a)$

$\delta(\neg\phi_1, a) \stackrel{\text{def}}{=} \overline{\neg\delta(\phi_1, a)}$

$\delta(\mathbf{X}\phi_1, a) \stackrel{\text{def}}{=} \phi_1$

$\delta(\phi_1 \,\mathcal{U} \phi_2, a) \stackrel{\text{def}}{=} \delta(\phi_2, a) \vee (\delta(\phi_1, a) \wedge \phi_1 \,\mathcal{U} \phi_2)$

où $\overline{\psi}$ est la normalisation de ψ par les règles : $\neg(\phi \vee \psi) \to \neg\phi \wedge \neg\psi$, $\neg(\phi \wedge \psi) \to \neg\phi \vee \neg\psi$, $\neg\neg\phi \to \phi$.

La preuve du théorème s'effectue alors par récurrence sur la taille de la formule : on montre par récurrence sur ϕ que $\mathcal{A}_\phi$ accepte les mots sur $(\{0,1\}^k)^\omega$ qui satisfont ϕ.

Cas de base. Si $\phi = \mathbf{true}$ (resp. $\mathbf{false}$), la transition par toutes les lettres est $\mathbf{true}$ (resp. $\mathbf{false}$) et tous les mots sont acceptés (resp. non acceptés).

Si ϕ est une variable propositionnelle p (resp. sa négation $\neg p$), $w \models \phi$ si et seulement si $p \in w(0)$ (resp. $p \notin w(0)$). Un calcul ρ de $\mathcal{A}_\phi$ sur w est un mot d'une lettre : ϕ lui-même et tous les calculs sont réussis. Par définition de δ, il existe un calcul sur w ssi $p \in w(0)$.

récurrence On envisage tous les cas de construction de ϕ :

$\mathbf{X}\phi_1$ Par définition de $\mathcal{A}_\phi$, l'automate est identique à $\mathcal{A}_{\phi_1}$, avec un état en plus (qui devient l'état initial) : ϕ, et une transition supplémentaire $\delta(\phi, a) \stackrel{\text{def}}{=} \phi_1$ pour tout a.

$w \models \phi$ ssi $w|_1 \models \phi_1$ ssi (par hypothèse de récurrence) il existe un calcul réussi ρ_1 de $\mathcal{A}_{\phi_1}$ sur $w|_1$ ssi le calcul $\rho \stackrel{\text{def}}{=} (\phi) \cdot \rho_1$ est un calcul réussi de $\mathcal{A}_\phi$ sur w.

$\phi_1 \wedge \phi_2$ $w \models \phi_1 \wedge \phi_2$ ssi $w \models \phi_1$ et $w \models \phi_2$, ssi, par hypothèse de récurrence, il existe des calculs réussis ρ_1 et ρ_2 de $\mathcal{A}_{\phi_1}$ et $\mathcal{A}_{\phi_2}$ sur w.

Etant donnés deux calculs ρ_1 de $\mathcal{A}_{\phi_1}$ et ρ_2 de $\mathcal{A}_{\phi_2}$ respectivement sur w, $\rho_1 \oplus \rho_2$ est l'arbre dont la racine est étiquetée par $\phi_1 \wedge \phi_2$, les positions sont $Pos(\rho_1) \cup \{(k+i) \cdot q \mid i \cdot q \in Pos(\rho_2)\}$ si k est le branchement à la racine de ρ_1. De plus $(\rho_1 \oplus \rho_2)(i \cdot q) = \rho_1(i \cdot q)$ si $i \leq k$ et $(\rho_1 \oplus \rho_2)(i \cdot q) = \rho_2((i-k) \cdot q)$ sinon. Par définition de δ, $\rho_1 \oplus \rho_2$ est un calcul de $\mathcal{A}$ sur w et, inversement, tout calcul ρ de $\mathcal{A}_\phi$ sur w se décompose en $\rho_1 \oplus \rho_2$. Enfin, $\rho_1 \oplus \rho_2$ est réussi si et seulement si tous les chemins de $\rho_1 \oplus \rho_2$ contiennent une infinité d'états finaux, si et seulement si tous les chemins de ρ_1 et tous les chemins de ρ_2 contiennent une infinité d'états finaux si et seulement si ρ_1 et ρ_2 sont des calculs réussis.

Ainsi, $w \models \phi_1 \wedge \phi_2$ ssi il existe des calculs réussis ρ_1 et ρ_2 sur w ssi il existe un calcul réussi $\rho = \rho_1 \oplus \rho_2$ de $\mathcal{A}_{\phi_1 \wedge \phi_2}$ sur w.

$\phi_1 \vee \phi_2$ $w \models \phi_1 \vee \phi_2$ si et seulement si $w \models \phi_1$ ou $w \models \phi_2$ ssi il existe un calcul réussi de $\mathcal{A}_{\phi_1}$ sur w ou il existe un calcul réussi de $\mathcal{A}_{\phi_2}$ sur w. Or les calculs réussis de $\mathcal{A}_{\phi_1 \vee \phi_2}$ sont ceux de $\mathcal{A}_{\phi_1}$ et ceux de $\mathcal{A}_{\phi_2}$, remplaçant seulement l'étiquetage de la racine par $\phi_1 \vee \phi_2$. DOnc $w \models \phi_1 \vee \phi_2$ ssi il existe un calcul réussi de $\mathcal{A}_{\phi_1 \vee \phi_2}$ sur w.

$\phi\, \mathcal{U} \phi_2$ Supposons d'abord que $w \models \phi_1\, \mathcal{U} \phi_2$. Alors, par définition de la relation de satisfaction, il existe un entier n tel que $w|_n \models \phi_2$ et, pour tout $0 \leq i < n$, $w|_i \models \phi_1$. Montrons par récurrence sur n qu'il existe un calcul réussi de $\mathcal{A}_\phi$ sur w.

Si $n = 0$ alors $w \models \phi_2$ et, par la première hypothèse de récurrence, il existe un calcul réussi ρ_2 de $\mathcal{A}_{\phi_2}$ sur w. Ce calcul, à l'étiquetage de la racine près, est aussi un calcul de $\mathcal{A}_\phi$, par définition de $\delta(\phi_1\, \mathcal{U} \phi_2)$.

Si $n > 0$ par définition de la relation de satisfaction, $w \models \phi_1$ et $w|_1 \models \phi_1\, \mathcal{U} \phi_2$. Alors, par la première hypothèse de récurrence, il existe un calcul réussi ρ_1 de $\mathcal{A}_{\phi_1}$ sur w et, par la deuxième hypothèse de récurrence, il existe un calcul réussi ρ_2 de $\mathcal{A}_\phi$ sur $w|_1$. On obtient alors un calcul réussi $(1 \cdot \rho_2) \oplus \rho_1$ de $\mathcal{A}_\phi$ sur w.

Réciproquement, s'il existe un calcul réussi ρ de $\mathcal{A}_\phi$ sur w, par définition des états finaux, ρ ne contient aucun chemin qui soit étiqueté par ϕ exclusivement. Soit alors n la longueur maximale d'une position de ρ étiquetée par ϕ. Montrons, par récurrence sur n, que $w \models \phi$.

Si $n = 0$ alors, par définition de δ sur $\phi_1\, \mathcal{U} \phi_2$, en remplaçant ϕ par ϕ_2 à la racine de ρ, on obtient encore un calcul réussi sur w. Par hypothèse de récurrence on a alors $w \models \phi_2$ et donc $w \models \phi_1\, \mathcal{U} \phi_2$.

Si $n > 0$ alors, par définition de δ, ρ se décompose $\rho = 1 \cdot \rho_1 \oplus \rho_2$ où ρ_1 est un calcul réussi de $\mathcal{A}_\phi$ sur $w|_1$ et ρ_2 est un calcul réussi de $\mathcal{A}_{\phi_1}$ sur w. Par la première hypothèse de récurrence on a alors $w \models \phi_1$ et, par la deuxième hypothèse de récurrence, $w|_1 \models \phi_1\, \mathcal{U} \phi_2$. D'où $w \models \phi_1\, \mathcal{U} \phi_2$.

$\neg(\phi_1\, \mathcal{U} \phi_2)$

Si $w \models \neg(\phi_1\, \mathcal{U} \phi_2)$, alors ou bien , pour tout i, $w|_i \models \neg\phi_2$, ou bien il existe un n tel que $w|_n \models \neg\phi_1 \wedge \neg\phi_2$ et, pour tout $i < n$, $w \models \neg\phi_2$.

Dans le premier cas, par hypothèse de récurrence, pour tout i, il existe un calcul réussi ρ_i de $\mathcal{A}_{\neg\phi_2}$ sur $w|_i$ et on construit ρ de la manière suivante : $Pos(\rho) = \bigcup_{i \in \mathbb{N}} \{1^i \cdot (j+1) \cdot q \mid j \cdot q \in Pos(\rho_i)\}$ et $\rho(1^i \cdot (j+1 \cdot q)) = \rho_i(j \cdot q)$, $\rho(1^i) = \phi$. ρ est bien un calcul réussi, par définition des états finaux et puisque $\delta(\neg(\phi_1\, \mathcal{U} \phi_2), a) = \delta(\neg\phi_2, a) \wedge (\delta(\neg\phi_1, a) \vee \neg(\phi_1\, \mathcal{U} \phi_2))$.

Dans le deuxième cas, représenté sur la figure 11, on procède par récurrence sur n pour construire un calcul réussi sur w :

Si $n = 0$, $w \models \neg\phi_1 \wedge \neg\phi_2$ et, par hypothèse de récurrence, il existe un calcul réussi de $\mathcal{A}_{\neg\phi_1}$ sur w et un calcul réussi de $\mathcal{A}_{\neg\phi_2}$ sur w, il est alors facile de construire

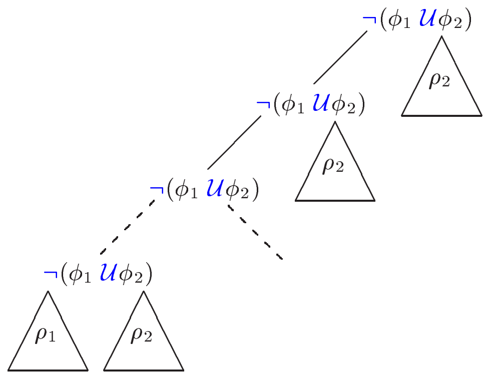

FIG. 11 – Un calcul réussi de $\mathcal{A}_{\neg(\phi_1\,\mathcal{U}\phi_2)}$ sur w

un calcul réussi $\mathcal{A}_\phi$: on procède comme pour l'intersection. (Rappelons aussi que $\delta(\neg\phi_2, a) \wedge \delta(\neg\phi_1, a) \models \delta(\phi, a)$ par définition).

Si $n > 0$ On construit le calcul ρ à partir du calcul réussi sur $w|_1$ (utilisant la deuxième hypothèse de récurrence) et du calcul réussi de $\mathcal{A}_{\neg\phi_2}$ sur w (utilisant la prmière hypothèse de récurrence).

Réciproquement, si l'on suppose que w est accepté par $\mathcal{A}_{\neg(\phi_1\,\mathcal{U}\phi_2)}$, soit ρ un calcul réussi sur w. Par définition de la relation de transition ρ est de la forme représentée sur la figure 11, où ρ_1 peut le cas échéant ne jamais être présent (condition d'acceptance). Dans chacun des cas (cas où il n'y a pas de ρ_1 et cas où il y en a un) on montre que $w \models \neg(\phi_1\,\mathcal{U}\phi_2)$, par récurrence dans le deuxième cas.

Remarque : Selon la manière dont on définit la taille d'un automate alternant, la construction de la preuve qui précède peut conduire ou non à un automate de taille polynômiale. Par exemple, si la taille de l'alphabet fait partie de la taille de l'automate, on passe d'une formule sur un ensemble de n variables propositionnelles à un automate sur un alphabet de taille 2^n, donc exponentiel. Cependant, on peut remarquer que la fonction de transition, telle qu'elle est définie dans la preuve du théorème possède une description polynômiale, indépendemment de l'alphabet, car celui-ci n'est utilisé que pour les cas de base des transitions à partir des variables propositionnelles.

Remarquons, qu'on peut facilement calculer les transitions pour les modalités dérivées $\Box$ et $\Diamond$:

$$\begin{aligned}
\delta(\Box\phi_1, a) &\stackrel{\text{def}}{=} \delta(\phi_1, a) \wedge \Box\phi_1 \\
\delta(\Diamond\phi_1, a) &\stackrel{\text{def}}{=} \delta(\phi_1, a) \vee \Diamond\phi_1
\end{aligned}$$

Exemple 18 Soit $\phi = \Diamond\Box\neg Q$. On obtient l'automate associé suivant, après avoir retiré les états inaccessibles : $Q = \{\phi, \Box\neg Q\}$ avec pour seul état final $\Box\neg Q$ et la relation de transition (on suppose qu'il y a deux variables propositionnelles P et Q, mais comme P ne joue aucun rôle, seules les transitions pour Q sont précisées) :

δ	1	0
$\Box\neg Q$	**false**	$\Box\neg Q$
ϕ	ϕ	$\Box\neg Q \vee \phi$

Dans cet exemple, l'automate est en fait un automate de Büchi non-déterministe.

Exercice 9
Construire les automates alternants acceptant les mots qui satisfont :

1. $(P\,\mathcal{U}Q)\,\mathcal{U}(\neg P)$
2. $(\mathbf{X}\neg P)\,\mathcal{U}Q$

5.3 Satisfaisabilité de LTL et problème du non-vide des automates alternants

Le théorème 17 permet de ramener le problème de la satisfaisabilité d'une formule de LTL au problème du vide des automates alternants ; commençons donc par établir :

Theorème 19 *Le vide des automates de Büchi alternants est dans PSPACE.*

Preuve: En fait, il suffit d'utiliser la proposition 11 le lemme 16 : l'automate du lemme 16 est calculé à la volée (par nécessité) et on utilise l'algorithme de la proposition 11 pour lequel un espace logarithmique suffit. Il faut seulement vérifier que, étant donnés (S, S'), a, (S_1, S_1') (comme dans la preuve du lemme 16) que l'on peut décider en espace polynômial si $((S, S'), a, (S_1, S_1')) \in \delta'$. □

Remarque : le theorème 19, qui ne précise pas dans son énoncé ce qu'est la taille d'un automate alternant, suppose seulement que la fonction de transition δ est donnée sous une forme telle que, étant donné un ensemble d'états S, un état q et une lettre de l'alphabet a, on peut décider en espace polynômial si $S \models \delta(q, a)$. Sauf mention contraire explicite, nous supposerons en fait dans la suite que la *taille* d'un automate alternant est la somme de son nombre d'états et du programme donnant δ. Ce programme sera supposé tel que l'on peut décider en *temps polynômial* si $S \models \delta(q, a)$.

Theorème 20 *La satisfaisabilité de LTL est PSPACE-complète*

Preuve: L'appartenance à PSPACE résulte des théorèmes 17 et 19 (et des remarques sur la taille de δ).

Reste à montrer que le problème est PSPACE-difficile. Il y a plusieurs preuves, plus ou moins compliquées selon les fragments de LTL que l'on veut montrer difficiles pour PSPACE (cf. [DS02]). Nous choisissons ici une preuve peu raffinée mais simple, par codage direct des machines de Turing déterministes en espace polynômial.

Si T est une machine de Turing travaillant en espace polynômial, et w un mot, on calcule en temps polynômial une formule de LTL $\phi_{T,w}$ telle que T accepte w si et seulement si $\phi_{T,w}$ est satisfaisable.

Si l'espace de travail de T est consitué des positions $1, \ldots, n$ du ruban, on considère les variables propositionnelles $R_{a,i}$ pour $i \in \{1, \ldots, n\}$ et a dans l'alphabet d'entrée, ainsi que les variables $P_{q,i}$ pour q état de la machine. Une configuration est alors représentée par une conjonction de variables propositionnelles : $R_{a,i}$ code le fait que la $i$ème lettre du ruban est a et $P_{q,i}$ code le fait que la tête de lecture est en i et la machine dans l'état q.

Chaque position d'un mot infini doit représenter une configuration valide :

- A chaque position du ruban, il y a une et une seule lettre :

$$\Box \bigwedge_{i=1}^{n} (\bigvee_{a \in A} R_{a,i}) \wedge \bigwedge_{a \neq b} \neg R_{a,i} \vee \neg R_{b,i})$$

- Il y a une et une seule position du ruban sur laquelle pointe la tête de lecture et la machine est à chaque instant dans un et un seul état :

$$\Box(\bigvee_{q \in Q} \bigvee_{i=1}^{n} P_{q,i}) \wedge (\bigwedge_{q \neq q'} \bigwedge_{i} \neg P_{q,i} \vee \neg P_{q',i}) \wedge (\bigwedge_{i \neq j} \bigwedge_{q \in Q} \neg P_{q,i} \vee \neg P_{q,j})$$

On exprime ensuite que l'on commence par la configuration initiale :

$$P_{q_0,1} \wedge R_{a_1,1} \wedge \ldots \wedge R_{a_n,n}$$

si $w = a_1 \ldots a_n$.

Un mot satisfaisant la formule doit correspondre à un calcul de la machine. Par exemple, pour les déplacements à droite de la tête de lecture :

$$\Box \bigwedge_{(q,a,q',R,b)\in\Delta} \bigwedge_{i=1}^{n} (P_{q,i} \wedge R_{a,i}) \Rightarrow \mathbf{X}(P_{q',i+1} \wedge R_{b,i})$$

Enfin, on exprime qu'un calcul est réussi :

$$\Diamond \bigvee_{q_f\in Q_f} \bigvee_{i=1}^{n} P_{q_f,i}$$

ce qui termine le codage.

On peut remarquer que, si une formule de LTL est satisfaisable, alors elle admet un modèle fini : *LTL a la propriété de modèles finis.* Par contre :

Exercice 10
Donner deux structures de Kripke distinctes qui satisfont les mêmes formules de LTL.

5.4 LTL :Vérification de modèles. Une technique optimale utilisant les automates

Soit $\mathcal{K}$ une structure de Kripke et ϕ une formule de logique temporelle. À ϕ est associé l'automate $\mathcal{A}_\phi$ comme au théorème 17. $\mathcal{K} \models \phi$ si et seulement si $L_{\mathcal{K}}$, ensemble des executions de $\mathcal{K}$ est contenu dans $L(\mathcal{A}_\phi)$, ou, de façon équivalente, si

$$\mathcal{K} \models \phi \quad \text{ssi} \quad L_{\mathcal{K}} \cap L(\mathcal{A}_{\neg\phi}) = \emptyset$$

Comme l'intersection de deux automates de Büchi $\mathcal{A}_1$ et $\mathcal{A}_2$ peut être calculée en temps $O(|\mathcal{A}_1| \times |\mathcal{A}_2|)$, on obtient ainsi que le problème de model-checking est dans PSPACE. La complétude pour PSPACE s'obtient simplement en considérant pour $\mathcal{K}$ l'automate de Büchi qui accepte tous les mots sur $\{0,1\}^n$: la validité de ϕ (ou l'insatisfaisabilité de $\neg\phi$) est alors équivalente à $\mathcal{K} \models \phi$.

Theorème 21 *Le model-checking pour LTL est PSPACE-complet.*

Si l'on fixe une formule ϕ, savoir si un automate de Büchi $\mathcal{A}$ satisfait ϕ est NLOGSPACE-complet.