Table des matières

Les logiques temporelles sont apparues il y a trente ans pour spécifier des propriétés sur des systèmes réactifs. Parmi celles-ci, LTL et CTL se sont avérées particulièrement utiles.

Il y a dix ans sont apparues les logiques temporelles alternantes, des logiques temporelles à branchement qui font apparaitre plusieurs antagonismes, que l'on appelle agents, dans le sens où les transitions ne sont que partiellement contrôlées par chaque parti. Ces logiques sont donc particulièrement adaptées à la modélisation de jeux.

Dans les logiques temporelles alternantes couramment employées, qui sont toutes des fragments de ATL^*, la seule notion d'entente entre les joueurs qui puisse être définie est celle de coalition, c'est à dire qu'ils sont invariablement tous unis dans un but commun. L'objectif de ce stage a été de définir une nouvelle logique alternante, dans laquelle apparaissent des notion d'entente plus partielles, telles que le contrôle ou l'arbitrage. Une fois définies, nous avons étudié l'expressivité et la complexité de ces nouveaux langages, en les comparant à ceux qui existent déjà.

Pour cela, nous avons dû considérer les stratégies que chaque joueur est susceptible d'adopter, et les distinguer selon la quantité de mémoire nécéssaire à leur application. Une définition de la taille de la mémoire est ainsi proposée dans le chapitre 4.

Chapitre 1

Rappels

1.1 Les modèles de la logique temporelle alternante

Nous allons introduire une structure permettant de modéliser les jeux à plusieurs joueurs. Les définitions qui suivent proviennent de [1]

Définition 1. *Une* CGS *(de l'anglais Concurrent Game Structure)* $\mathcal{C}$ *est un 6-uple* $(\mathsf{Agt}_\mathcal{C}, \mathsf{Loc}_\mathcal{C}, \mathsf{AP}_\mathcal{C}, \mathsf{Lab}_\mathcal{C}, \mathsf{Mov}_\mathcal{C}, \mathsf{Edg}_\mathcal{C})$ *tel que :*

- $\mathsf{Agt}_\mathcal{C} = \{A_1, ..., A_k\}$ *est un ensemble fini de* joueurs *;*
- $\mathsf{Loc}_\mathcal{C}$ *est un ensemble fini d'*états *;*
- $\mathsf{AP}_\mathcal{C}$ *est un ensemble fini de* propositions atomiques *;*
- $\mathsf{Lab}_\mathcal{C} \colon \mathsf{Loc}_\mathcal{C} \to 2^{\mathsf{AP}_\mathcal{C}}$ *est une fonction qui associe à chaque état l'ensemble des propositions atomiques qui sont vraies dans cet état ;*
- $\mathsf{Mov}_\mathcal{C} \colon \mathsf{Loc}_\mathcal{C} \times \mathsf{Agt}_\mathcal{C} \to \mathbb{N}_{\geq 1}$ *est la fonction de choix. Etant donnés un joueur* A_i *et un état* s, $\mathsf{Mov}(s, A_i)$ *est le nombre de mouvements autorisés du joueur* A_i *depuis* s. *Ces mouvements sont numérotés de* 1 *à* $\mathsf{Mov}(s, A_i)$ *;*
- $\mathsf{Edg}_\mathcal{C} \colon \mathsf{Loc}_\mathcal{C} \times \mathbb{N}^k \to \mathsf{Loc}_\mathcal{C}$, *où* $k = |\mathsf{Agt}_\mathcal{C}|$, *est une fonction partielle qui définit la table des transitions. A chaque état et à chaque uple de mouvement des joueurs, elle associe l'état résultant.*

Si $\mathsf{Agt}_\mathcal{C}$ *est réduit à un singleton (plus informellement, si il n'y a qu'un seul joueur),* $\mathcal{C}$ *est appelé une* Structure de Kripke.

Si chaque état de $\mathcal{C}$ *est contrôlé par un seul joueur (pas forcément le même joueur pour chaque état),* $\mathcal{C}$ *est appelé une* Turn-Based Game Structure, *soit en français une structure de jeu à tours. Une Structure de Kripke est donc un cas particulier de Turn-Based Game Structure.*

Définition 2. *Soit* $\mathcal{C}$ *une* CGS, $A \subseteq \mathsf{Agt}_\mathcal{C}, s \in \mathsf{Loc}_\mathcal{C}$ *et* $m_A = (m_{A_i})_{A_i \in A} \in \prod_{A_i \in A}\{1, ..., \mathsf{Mov}_\mathcal{C}(s, A_i)\}$. *On note* $\mathsf{Next}_\mathcal{C}(s, A, m_A)$ *l'ensemble des successeurs de* s *lorsque chacun des joueurs* $A_i \in A$ *choisit le mouvement* m_{A_i}. *Formellement, on définit :*

$\mathsf{Next}_{\mathcal{C}}(s, A, m_A) = \{q' \in \mathit{Loc}_{\mathcal{C}} \mid \exists m' = (m'_{A_i})_{A_i \in \mathit{Agt}_{\mathcal{C}}}$ *tel que* $\mathit{Edg}_{\mathcal{C}}(q, m') = q'$ *et* $\forall A_i \in A, m'_{A_i} = m_{A_i}\}$.

Dans le cas particulier $A = \emptyset$*, on a nécessairement* $m_A = \emptyset$*, et* $\mathsf{Next}_{\mathcal{C}}(s, A, m_A)$ *sera alors noté plus simplement* $\mathsf{Next}_{\mathcal{C}}(s)$ *: il s'agit de l'ensemble des successeurs de* s.

Définition 3. *Une* exécution *de* $\mathcal{C}$ *est une suite infinie* $\lambda = s_0 s_1 ...$ *d'états telle que pour tout* $k \in \mathbb{N}$, $s_{k+1} \in \mathsf{Next}_{\mathcal{C}}(s_k)$. *On peut utiliser les notations standards de préfixes et de suffixes pour les exécutions ;* $\lambda[i]$ *désigne le i-ème état* s_i *de* λ. *Si* $\lambda[0] = s$*, on dit que* λ *est une* s*-exécution. On note* $\mathsf{Exec}_{\mathcal{C}}(s)$ *l'ensemble des* s*-exécutions de* $\mathcal{C}$.

Définition 4. *Une* stratégie *avec mémoire infinie pour un joueur* $A_i \in \mathit{Agt}_{\mathcal{C}}$ *est une application* F *qui associe un mouvement du joueur* $A_i \in A$ *à chaque préfixe d'exécution. On dit qu'une stratégie est* sans mémoire *si le mouvement associé ne dépend que de l'état courant. Autrement dit, une* stratégie sans mémoire *(ou* stratégie positionnelle*) pour un joueur* $A_i \in \mathit{Agt}_{\mathcal{C}}$ *est une application* F *qui associe à chaque état de* $\mathit{Loc}_{\mathcal{C}}$ *un mouvement du joueur* $A_i \in A$ *disponible depuis cet état. On note respectivement* $\Sigma^{\infty}_{A_i,\mathcal{C}}$ *et* $\Sigma^{0}_{A_i,\mathcal{C}}$ *les ensembles des stratégies à mémoire infinie et des stratégies sans mémoire de* $\mathcal{C}$ *pour le joueur* A_i [1].

Formellement, on écrit donc :

– $\Sigma^{\infty}_{A_i,\mathcal{C}} = \{F \in \mathbb{N}^{\bigcup_{j \in \mathbb{N}^*} \mathit{Loc}^j_{\mathcal{C}}} \mid \forall j \in \mathbb{N}^*, \forall \lambda \in \mathit{Loc}^j_{\mathcal{C}}, F(\lambda) \in \{1, ..., \mathit{Mov}_{\mathcal{C}}(\lambda[j], A_i)\}\}$,

– $\Sigma^{0}_{A_i,\mathcal{C}} = \{F \in \mathbb{N}^{\mathit{Loc}_{\mathcal{C}}} \mid \forall q \in \mathit{Loc}_{\mathcal{C}}, F(q) \in \{1, ..., \mathit{Mov}_{\mathcal{C}}(q, A_i)\}\}$.

Si $n \in \{0, \infty\}$ *et* $A \subseteq \mathit{Agt}_{\mathcal{C}}$*, on note* $\Sigma^n_{A,\mathcal{C}} = \prod_{A_i \in A} \Sigma^n_{A_i,\mathcal{C}}$.

Si de plus $F = (F_{A_i})_{A_i \in A} \in \Sigma^n_{A,\mathcal{C}}$ *et* $\lambda \in \bigcup_{j=1}^{n+1} \mathit{Loc}^j_{\mathcal{C}}$*, on note* $F(\lambda) = (F_{A_i}(\lambda))_{A_i \in A}$.

Une stratégie (avec ou sans mémoire) F *pour la coalition* A *induit un sous-ensemble de* $\mathsf{Exec}_{\mathcal{C}}(s)$*, appelé ensemble des* issues de F *depuis l'état* s*, constitué des* s*-exécutions que les joueurs* $A_i \in A$ *contraignent lorsqu'ils jouent selon la stratégie* F.

Formellement, on écrit donc :

– *Si* $F \in \Sigma^{\infty}_{A,\mathcal{C}}$*, soit* $s \in \mathit{Loc}_{\mathcal{C}}$ *;* $\lambda \in \mathsf{Out}_{\mathcal{C}}(s, F)$ *ssi* $\lambda[0] = s$*, et pour tout* $k \in \mathbb{N}$ *on a* $\lambda[k+1] \in \mathsf{Next}_{\mathcal{C}}(\lambda[k], A, F(\lambda[0,k]))$.

– *Si* $F \in \Sigma^{0}_{A,\mathcal{C}}$*, soit* $s \in \mathit{Loc}_{\mathcal{C}}$ *;* $\lambda \in \mathsf{Out}_{\mathcal{C}}(s, F)$ *ssi* $\lambda[0] = s$*, et pour tout* $k \in \mathbb{N}$ *on a* $\lambda[k+1] \in \mathsf{Next}_{\mathcal{C}}(\lambda[k], A, F(\lambda[k]))$.

Remarques :

– Dans le cas particulier $A = \emptyset, \Sigma^{\infty}_{A,\mathcal{C}} = \Sigma^{0}_{A,\mathcal{C}} = \{\emptyset\}$. L'unique stratégie pour la coalition vide est donc la stratégie vide, et on a $\forall s \in \mathsf{Loc}_{\mathcal{C}}, \mathsf{Out}_{\mathcal{C}}(s, \emptyset) = \mathsf{Exec}_{\mathcal{C}}(s)$.

– Dans le cas particulier $A = \mathsf{Agt}$, si $n \in \{0, \infty\}, F \in \Sigma^n_{A,\mathcal{C}}$ et $s \in \mathsf{Loc}_{\mathcal{C}}$ sont fixés, l'ensemble $\mathsf{Out}_{\mathcal{C}}(s, F)$ est réduit à une s-exécution. Dans le cas où $n = \infty$, toutes les s-exécutions sont ainsi atteintes.

Lorsque le contexte sera clair, on omettra les indices correspondant à une *CGS*.

[1] On parlera de stratégies à mémoire bornée dans la partie 4

1.2 La logique ATL^* et quelques-uns de ses fragments

Au long de ce rapport, on va introduire des logiques temporelles qui permettent d'énoncer des propriétés sur les *CGS*. La première d'entre elles est la logique ATL^*, de l'anglais Alternating-time Temporal Logic.

Définition 5. *La syntaxe de ATL^* est définie par la grammaire suivante :*

$$\begin{aligned} \mathsf{ATL}^* \ni \varphi_s, \psi_s \quad &::= \quad p \mid \neg\varphi_s \mid \varphi_s \vee \psi_s \mid \langle\!\langle A \rangle\!\rangle\, \varphi_p \\ \mathsf{ATL}^*_p \ni \varphi_p \quad &::= \quad \varphi_s \mid \neg\varphi_p \mid \varphi_p \vee \psi_p \mid \mathbf{X}\, \varphi_p \mid \varphi_p\, \mathbf{U}\, \psi_p. \end{aligned}$$

où p décrit l'ensemble AP et A l'ensemble des parties de Agt.

La sémantique de ATL^* est définie comme suit :
Soient $A \subseteq \mathsf{Agt}$ et $q \in \mathsf{Loc}$.

$$\begin{array}{rcl} q \models p & \text{ssi} & p \in \mathsf{Lab}(q), \\ q \models \neg\varphi_s & \text{ssi} & q \not\models \varphi_s, \\ q \models \varphi_s \vee \psi_s & \text{ssi} & q \models \varphi_s \text{ ou } q \models \psi_s, \\ q \models \langle\!\langle A \rangle\!\rangle\, \varphi_p & \text{ssi} & \exists F \in \Sigma_A^\infty \text{ tel que } \forall \lambda \in \mathsf{Out}(q, F), \lambda \models \varphi_p, \\ \lambda \models \varphi_s & \text{ssi} & \lambda[0] \models \varphi_s, \\ \lambda \models \neg\varphi_p & \text{ssi} & \lambda \not\models \varphi_p, \\ \lambda \models \varphi_p \vee \psi_p & \text{ssi} & \lambda \models \varphi_p \text{ ou } \lambda \models \psi_p, \\ \lambda \models \mathbf{X}\, \varphi_p & \text{ssi} & \lambda[1, \infty] \models \varphi_p, \\ \lambda \models \varphi_p\, \mathbf{U}\, \psi_p & \text{ssi} & \exists i \geq 0.\ \lambda[i, \infty] \models \psi_p \text{ et } \forall 0 \leq j < i.\ \lambda[j, \infty] \models \varphi_p \end{array}$$

Remarques :
- Les opérateurs $\langle\!\langle A \rangle\!\rangle$ seront appelés des quantificateurs de stratégie, et les opérateurs $\mathbf{X}$ et $\mathbf{U}$ seront appelés des opérateurs temporels.
- $\mathbf{F}\, \varphi$ sera l'abréviation de true $\mathbf{U}\, \varphi$ (c'est-à-dire "on aura un jour φ").
- $\mathbf{G}\, \varphi$ sera l'abréviation de $\neg\mathbf{F}\, \neg\varphi$ (c'est-à-dire "on a toujours φ").

Nous allons à présent définir un fragment standard de ATL^* que l'on nomme CTL^*. Il s'agit du fragment dans lequel les seuls quantificateurs de stratégie sont $\langle\!\langle \emptyset \rangle\!\rangle$ et $\langle\!\langle \mathsf{Agt} \rangle\!\rangle$. Il n'y a plus besoin de considérer de stratégies pour définir la sémantique dans ce cadre restreint. La remarque précédente sur les ensembles de stratégies pour les coalitions $\emptyset$ et Agt met en évidence les faits suivants :
- $q \models \langle\!\langle \emptyset \rangle\!\rangle\, \varphi_p$ ssi $\forall \lambda \in \mathsf{Exec}(q), \lambda \models \varphi_p$,
- $q \models \langle\!\langle \mathsf{Agt} \rangle\!\rangle\, \varphi_p$ ssi $\exists \lambda \in \mathsf{Exec}(q), \lambda \models \varphi_p$.

Ceci explique que les quantificateurs $\langle\!\langle \emptyset \rangle\!\rangle$ et $\langle\!\langle \mathsf{Agt} \rangle\!\rangle$ soient respectivement notés $\mathbf{A}$ (pour all) et $\mathbf{E}$ (pour exists). Dans le cas particulier où la structure est une Structure de Kripke, les logiques ATL^* et CTL^* sont confondues.

Définition 6. *La syntaxe de CTL^* est définie par la grammaire suivante :*

$$\begin{aligned} \mathsf{CTL}^* \ni \varphi_s, \psi_s \quad &::= \quad p \mid \neg\varphi_s \mid \varphi_s \vee \psi_s \mid \mathbf{A}\varphi_p, \mid \mathbf{E}\varphi_p \\ \mathsf{CTL}^*_p \ni \varphi_p \quad &::= \quad \varphi_s \mid \neg\varphi_p \mid \varphi_p \vee \psi_p \mid \mathbf{X}\, \varphi_p \mid \varphi_p\, \mathbf{U}\, \psi_p. \end{aligned}$$

où p décrit l'ensemble AP.

La sémantique de CTL^* est définie comme suit :
Soit $q \in \mathsf{Loc}$.

$$
\begin{array}{rcl}
q \models p & \text{ssi} & p \in \mathsf{Lab}(q), \\
q \models \neg\varphi_s & \text{ssi} & q \not\models \varphi_s, \\
q \models \varphi_s \vee \psi_s & \text{ssi} & q \models \varphi_s \text{ ou } q \models \psi_s, \\
q \models \mathbf{A}\varphi_p & \text{ssi} & \forall\lambda \in \mathsf{Exec}(q), \lambda \models \varphi_p, \\
q \models \mathbf{E}\varphi_p & \text{ssi} & \exists\lambda \in \mathsf{Exec}(q), \lambda \models \varphi_p, \\
\lambda \models \varphi_s & \text{ssi} & \lambda[0] \models \varphi_s, \\
\lambda \models \neg\varphi_p & \text{ssi} & \lambda \not\models \varphi_p, \\
\lambda \models \varphi_p \vee \psi_p & \text{ssi} & \lambda \models \varphi_p \text{ ou } \lambda \models \psi_p, \\
\lambda \models \mathbf{X}\,\varphi_p & \text{ssi} & \lambda[1,\infty] \models \varphi_p, \\
\lambda \models \varphi_p\,\mathbf{U}\,\psi_p & \text{ssi} & \exists i \geq 0.\ \lambda[i,\infty] \models \psi_p \text{ et } \forall 0 \leq j < i.\ \lambda[j,\infty] \models \varphi_p
\end{array}
$$

Nous allons à présent définir un fragment standard de ATL^*, et même de CTL^* que l'on nomme LTL. Il s'agit du fragment dans lequel le seul quantificateur de stratégie, $\mathbf{A}$, est toujours placé en tête de la formule.

Définition 7. *La syntaxe de LTL est définie par la grammaire suivante :*

$$
\begin{array}{rcl}
\mathsf{LTL} \ni \varphi_s, \psi_s & ::= & \mathbf{A}\varphi_p \\
\mathsf{LTL}_p \ni \varphi_p & ::= & p \mid \neg\varphi_p \mid \varphi_p \vee \psi_p \mid \mathbf{X}\,\varphi_p \mid \varphi_p\,\mathbf{U}\,\psi_p.
\end{array}
$$

où p décrit l'ensemble AP.

La sémantique de LTL est définie comme suit :
Soit $q \in \mathsf{Loc}$.

$$
\begin{array}{rcl}
q \models \mathbf{A}\varphi_p & \text{ssi} & \forall\lambda \in \mathsf{Exec}(q), \lambda \models \varphi_p, \\
\lambda \models p & \text{ssi} & p \in \mathsf{Lab}(\lambda[0]), \\
\lambda \models \neg\varphi_p & \text{ssi} & \lambda \not\models \varphi_p, \\
\lambda \models \varphi_p \vee \psi_p & \text{ssi} & \lambda \models \varphi_p \text{ ou } \lambda \models \psi_p, \\
\lambda \models \mathbf{X}\,\varphi_p & \text{ssi} & \lambda[1,\infty] \models \varphi_p, \\
\lambda \models \varphi_p\,\mathbf{U}\,\psi_p & \text{ssi} & \exists i \geq 0.\ \lambda[i,\infty] \models \psi_p \text{ et } \forall 0 \leq j < i.\ \lambda[j,\infty] \models \varphi_p
\end{array}
$$

La définition classique de LTL correspond plutôt à celle de LTL_p, le quantificateur $\mathbf{A}$ restant implicite [**?**].

Nous allons à présent définir un autre fragment standard de ATL^*, d'une nature complètement différente, que l'on nomme ATL. Il s'agit du fragment dans lequel tout opérateur de chemin est précédé d'un quantificateur de stratégie.

Définition 8. *La syntaxe de ATL est définie par la grammaire suivante :*

$$
\begin{array}{rcl}
\mathsf{ATL} \ni \varphi_s, \psi_s & ::= & p \mid \neg\varphi_s \mid \varphi_s \vee \psi_s \mid \langle\!\langle A \rangle\!\rangle\,\varphi_p \\
\mathsf{ATL}_p \ni \varphi_p & ::= & \varphi_s \mid \mathbf{X}\,\varphi_s \mid \varphi_s\,\mathbf{U}\,\psi_s, \mid \varphi_s\,\mathbf{W}\,\psi_s.
\end{array}
$$

où p décrit l'ensemble AP *et A l'ensemble des parties de* Agt*.*

La sémantique de ATL est définie comme suit :
Soient $A \subseteq \mathsf{Agt}$ et $q \in \mathsf{Loc}$

$$
\begin{array}{rcl}
q \models p & \text{ssi} & p \in \mathsf{Lab}(q), \\
q \models \neg\varphi_s & \text{ssi} & q \not\models \varphi_s, \\
q \models \varphi_s \vee \psi_s & \text{ssi} & q \models \varphi_s \text{ ou } q \models \psi_s, \\
q \models \langle\!\langle A \rangle\!\rangle\, \varphi_p & \text{ssi} & \exists F \in \Sigma_A^{\infty} \text{ tel que } \forall\lambda \in \mathsf{Out}(q,F), \lambda \models \varphi_p, \\
\lambda \models \varphi_s & \text{ssi} & \lambda[0] \models \varphi_s, \\
\lambda \models \mathbf{X}\,\varphi_s & \text{ssi} & \lambda[1] \models \varphi_s, \\
\lambda \models \varphi_s\,\mathbf{U}\,\psi_s & \text{ssi} & \exists i \geq 0.\ \lambda[i] \models \psi_s \text{ et } \forall 0 \leq j < i.\ \lambda[j] \models \varphi_s, \\
\lambda \models \varphi_s\,\mathbf{W}\,\psi_s & \text{ssi} & \lambda \models \varphi_s\,\mathbf{U}\,\psi_s \text{ ou } \forall i \geq 0.\ \lambda[i] \models \psi_s
\end{array}
$$

Remarque :
- Il est bien connu que dans ce nouveau cadre, on peut se restreindre aux stratégies sans mémoire[1], c'est-à-dire que :
  $q \models \langle\!\langle A \rangle\!\rangle\, \varphi_p$ ssi $\exists F \in \Sigma_A^0$ tel que $\forall\lambda \in \mathsf{Out}(q,F), \lambda \models \varphi_p$.

Enfin, voici un dernier fragment standard de ATL^*, et même de CTL^*, que l'on nomme CTL. Il s'agit du fragment de CTL^* dans lequel tout opérateur de chemin est précédé d'un quantificateur de stratégie.

Définition 9. *La syntaxe de* CTL *est définie par la grammaire suivante :*

$$
\begin{array}{rcl}
\mathsf{CTL} \ni \varphi_s, \psi_s & ::= & p \mid \neg\varphi_s \mid \varphi_s \vee \psi_s \mid \mathbf{A}\varphi_p, \mid \mathbf{E}\varphi_p \\
\mathsf{CTL}_p \ni \varphi_p & ::= & \varphi_s \mid \mathbf{X}\,\varphi_s \mid \varphi_s\,\mathbf{U}\,\psi_s.
\end{array}
$$

où p décrit l'ensemble AP*.*

La sémantique de CTL est définie comme suit :
Soit $q \in \mathsf{Loc}$

$$
\begin{array}{rcl}
q \models p & \text{ssi} & p \in \mathsf{Lab}(q), \\
q \models \neg\varphi_s & \text{ssi} & q \not\models \varphi_s, \\
q \models \varphi_s \vee \psi_s & \text{ssi} & q \models \varphi_s \text{ ou } q \models \psi_s, \\
q \models \mathbf{A}\varphi_p & \text{ssi} & \forall\lambda \in \mathsf{Exec}(q), \lambda \models \varphi_p, \\
q \models \mathbf{E}\varphi_p & \text{ssi} & \exists\lambda \in \mathsf{Exec}(q), \lambda \models \varphi_p, \\
\lambda \models \varphi_s & \text{ssi} & \lambda[0] \models \varphi_s, \\
\lambda \models \mathbf{X}\,\varphi_s & \text{ssi} & \lambda[1] \models \varphi_s, \\
\lambda \models \varphi_s\,\mathbf{U}\,\psi_s & \text{ssi} & \exists i \geq 0.\ \lambda[i] \models \psi_s \text{ et } \forall 0 \leq j < i.\ \lambda[j] \models \varphi_s.
\end{array}
$$

Remarque : l'opérateur temporel $\mathbf{W}$ dans le cadre de CTL peut s'exprimer en fonction de l'opérateur $\mathbf{U}$, ce qui n'est pas le cas dans le cadre de ATL[2]. C'est ce qui justifie la différence de définition entre ces deux logiques.

1.3 Deux notions théoriques : expressivité et complexité d'une logique temporelle alternante

Définition 10. *Soient $\mathcal{L}$ et $\mathcal{L}'$ deux logiques alternantes. On définit informellement les deux relations d'ordre $\leq_e$ et $\leq_d$ permettant de comparer leur expressivité :*

- *Deux formules φ et φ' sont équivalentes ssi pour toute* CGS *$\mathcal{C}$,* $\forall q \in \mathsf{Loc}, (\mathcal{C}, q) \models \varphi \leftrightarrow (\mathcal{C}, q) \models \varphi'$.
- *On écrit que $\mathcal{L} \leq_e \mathcal{L}'$ ssi toute formule de $\mathcal{L}$ est équivalente à une formule de $\mathcal{L}'$.*
- *Deux* CGS *$\mathcal{C}$ et $\mathcal{C}'$ sont équivalents vis-à-vis d'une logique $\mathcal{L}$ ssi ils satisfont le même ensemble de formules de $\mathcal{L}$.*
- *On écrit que $\mathcal{L} \leq_d \mathcal{L}'$ ssi deux* CGS *$\mathcal{C}$ et $\mathcal{C}'$ équivalents vis-à-vis de $\mathcal{L}'$ le sont aussi vis-à-vis de $\mathcal{L}$.*

Remarques :

- *$\leq_e$ compare ce que l'on appelle le* pouvoir d'expression *de deux logiques.*
- *$\leq_d$ compare ce que l'on appelle le* pouvoir de distinction *de deux logiques.*
- *Notons que si $\mathcal{L} \leq_e \mathcal{L}'$, alors $\mathcal{L} \leq_d \mathcal{L}'$, c'est-à dire que la relation $\leq_e$ et plus fine que leq_d.*

Définition 11. *Soit $\mathcal{L}$ une logique. Le problème de* model-checking *de $\mathcal{L}$, que l'on notera MC $\mathcal{L}$, est le problème de décision suivant :*

Etant donnés une CGS *$\mathcal{C}$, un état $q \in \mathsf{Loc}$ et une formule φ de $\mathcal{L}$, a-t-on $(\mathcal{C}, q) \models \varphi$?*

On sait que les problèmes MC ATL et MC CTL sont PTIME-complets, que MC LTL et MC CTL^* sont PSPACE-complets et que MC ATL^* est 2EXPTIME-complet. Pour les définitions des classes de complexité, voir [4]

Chapitre 2

Préservation de stratégies dans le cadre de logiques sans mémoire

Dans toutes les logiques introduites jusqu'à présent, les stratégies induites par un quantificateur de stratégie sont utilisées ponctuellement, c'est-à-dire qu'elles ne sont plus prises en compte dès que l'on a affaire à un autre quantificateur de stratégie.

Ainsi, par exemple, si l'on veut exprimer le fait, pour un ascenseur, que le contrôleur dispose d'une stratégie pour que chaque utilisateur dispose toujours d'une stratégie lui permettant d'atteindre chaque étage, on le fera dans ATL par la formule suivante :

$\langle\!\langle cont \rangle\!\rangle\, \mathbf{G}\, (\bigwedge_{i,j}\ \langle\!\langle utilisateur_j \rangle\!\rangle\, \mathbf{F}\, etage_i)$

Le problème ici est que selon la sémantique d'ATL, la recherche d'une stratégie pour un utilisateur ne se fait pas dans le contexte restreint par la stratégie du contrôleur : cette dernière a été oubliée dès la première requantification.

Le premier objectif de ce stage a été de définir une logique qui n'utilise plus ponctuellement des stratégies, mais qui les assimile à des contextes que l'on peut choisir de préserver ou d'oublier tout au long d'une formule. Nous allons dans un premier temps exploiter cette idée en ne considérant que des stratégies sans mémoire.

2.1 Emboîter des stratégies sans mémoire

Pour cela, nous allons définir une opération d'emboitement de stratégies sans mémoire, notée $\circ_0$.

Définition 12. *Soient $\mathcal{C}$ une* CGS, $A, B \subseteq \mathsf{Agt}$*,* $F = (F_{A_i})_{A_i \in A} \in \Sigma_A^0$ *et* $F' = (F'_{A_i})_{A_i \in B} \in \Sigma_B^0$.

On définit :

$F' \circ_0 F = (F''_{A_i})_{A_i \in A \cup B}$,
où $F''_{A_i} = \begin{cases} F'_{A_i} & \text{si } A_i \in B \\ F_{A_i} & \text{sinon.} \end{cases}$
On a donc $F' \circ_0 F \in \Sigma^0_{A \cup B}$.

La loi $\circ_0$ n'est pas commutative, mais elle est associative, et l'on pourra donc omettre les parenthèses. Son élément neutre est la stratégie $\emptyset$.

2.1.1 La logique $\mathcal{L}_0$

On définit d'abord une logique $\mathcal{L}_0$ inspirée de ATL, c'est-à-dire où chaque opérateur temporel est immédiatement précédé d'un quantificateur de stratégie. De plus, dans cette logique, on ne s'autorise pas encore à oublier des stratégies, mais juste à les emboîter par un nouveau quantificateur de stratégie noté $\lessdot\gtrdot_0$.

Définition 13. *La syntaxe de $\mathcal{L}_0$ est définie par la grammaire suivante :*

$$\begin{aligned} \mathcal{L}_0 \ni \varphi_s, \psi_s \quad &::= \quad p \mid \neg\varphi_s \mid \varphi_s \vee \psi_s \mid \lessdot A \gtrdot_0 \varphi_p \\ \mathcal{L}_{0,p} \ni \varphi_p \quad &::= \quad \varphi_s \mid \mathbf{X}\,\varphi_s \mid \varphi_s\,\mathbf{U}\,\psi_s, \mid \varphi_s\,\mathbf{W}\,\psi_s. \end{aligned}$$

où p décrit l'ensemble AP et A l'ensemble des parties de Agt.

La sémantique de $\mathcal{L}_0$ est définie comme suit :
Soient $A, B \subseteq \mathsf{Agt}$, $q \in \mathsf{Loc}$ et $F \in \Sigma^0_A$

$$\begin{array}{rcl} q \models_F p & \text{ssi} & p \in \mathsf{Lab}(q), \\ q \models_F \neg\varphi_s & \text{ssi} & q \not\models_F \varphi_s, \\ q \models_F \varphi_s \vee \psi_s & \text{ssi} & q \models_F \varphi_s \text{ ou } q \models_F \psi_s, \\ q \models_F \lessdot B \gtrdot_0 \varphi_p & \text{ssi} & \exists F' \in \Sigma^0_B \text{ tel que } \forall \lambda \in \mathsf{Out}(q, F' \circ_0 F), \lambda \models_{F' \circ_0 F} \varphi_p, \\ \lambda \models_F \varphi_s & \text{ssi} & \lambda[0] \models_F \varphi_s, \\ \lambda \models_F \mathbf{X}\,\varphi_s & \text{ssi} & \lambda[1] \models_F \varphi_s, \\ \lambda \models_F \varphi_s\,\mathbf{U}\,\psi_s & \text{ssi} & \exists i \geq 0.\ \lambda[i] \models_F \psi_s \text{ et } \forall 0 \leq j < i.\ \lambda[j] \models_F \varphi_s, \\ \lambda \models_F \varphi_s\,\mathbf{W}\,\psi_s & \text{ssi} & \lambda \models_F \varphi_s\,\mathbf{U}\,\psi_s \text{ ou } \forall i \geq 0.\ \lambda[i] \models_F \psi_s. \end{array}$$

Remarques :
- Le symbole $\models_\emptyset$ sera plus simplement noté $\models$.
- Si φ est une formule de $\mathcal{L}_0$, alors $q \models_F \neg \lessdot B \gtrdot_0 \neg\varphi$ signifie $\forall F' \in \Sigma^0_B, q \models_{F' \circ_0 F} \varphi$.
- $q \models_F \lessdot\emptyset\gtrdot_0 \varphi_p$ signifie $\forall \lambda \in \mathsf{Out}(q, F), \lambda \models_F \varphi_p$.

2.1.2 Expressivité de la logique $\mathcal{L}_0$

On va montrer que $\neg(\mathcal{L}_0 \leq_d \mathsf{ATL}^*)$, c'est-à-dire que la logique $\mathcal{L}_0$ permet de distinguer des structures qui sont indistingables par ATL^*.

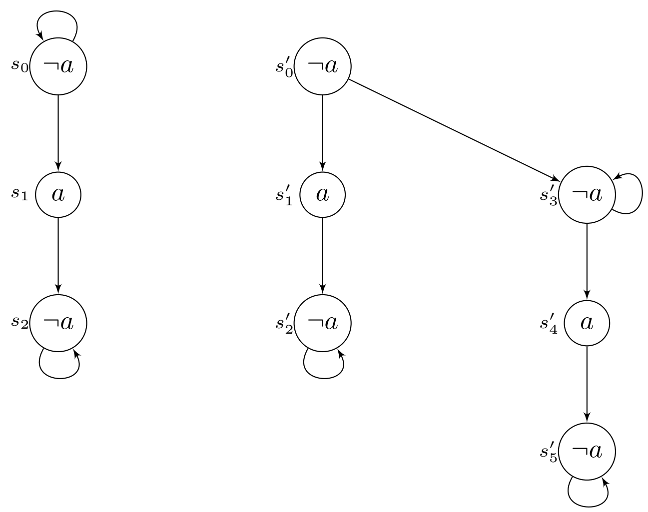

FIG. 2.1 – $\mathcal{C}_1$ et $\mathcal{C}_2$, deux structures de Kripke indistingables par ATL^*

Lemme 1. *On prouve d'abord que les deux structures de Kripke représentées sur la Fig. 1., avec pour états initiaux respectifs s_0 et s'_0, sont indistingables par ATL^*.*

Preuve : La relation $R = (s_0, s'_0), (s_0, s'_3), (s_1, s'_1), (s_1, s'_4), (s_2, s'_2), (s_2, s'_5)$ est manifestement une bisimulation qui lie les deux états initiaux. Donc les deux structures considérées sont bisimilaires, et par conséquent indistingables par CTL^*. Les formules de ATL^* étant les mêmes que celles de CTL^* sur les structures de Kripke, on en déduit que ces modèles sont indistingables par ATL^*. $\square$

Il s'agit maintenant d'exhiber une formule de $\mathcal{L}_0$ qui permette de distinguer ces deux structures.

Lemme 2. *On a* $s'_0 \models \lessdot \mathit{Agt} \gtrdot_0 \mathbf{X} \lessdot \emptyset \gtrdot_0 \mathbf{X}\, a$ *mais* $s_0 \not\models \lessdot \mathit{Agt} \gtrdot_0 \mathbf{X} \lessdot \emptyset \gtrdot_0 \mathbf{X}\, a$

Preuve : Remarquons que l'opérateur $\lessdot \mathsf{Agt} \gtrdot_0$, que l'on pourrait noter $\lessdot \mathbf{E} \gtrdot_0$, a pour effet de transformer la structure de Kripke en l'un de ses fragments déterministe (ce qui correspond à une exécution sans mémoire). Dans chaque état, on supprime toutes les flèches sauf une, et on obtient alors une structure linéaire. Dans $\mathcal{C}_2$, la stratégie pour le joueur consiste donc à transformer le modèle comme sur la Fig. 2. Cette structure linéaire vérifie bien la formule de LTL $\mathbf{X}\,\mathbf{X}\, a$.

Dans $\mathcal{C}_1$, les deux seules stratégies possibles transforment la structure comme sur la Fig. 3. Aucune de ces structures linéaires ne vérifie $\mathbf{X}\,\mathbf{X}\, a$, donc la formule $\lessdot \mathsf{Agt} \gtrdot_0 \mathbf{X} \lessdot \emptyset \gtrdot_0 \mathbf{X}\, a$ est fausse en s_0.

$\square$

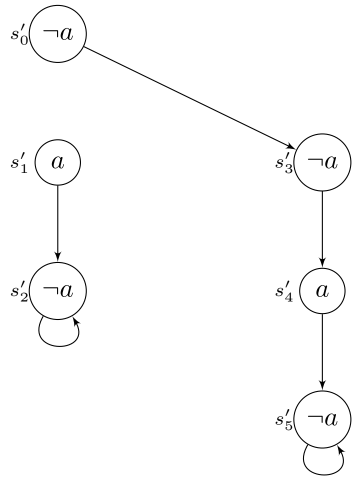

Fig. 2.2 – La stratégie gagnante du joueur depuis $\mathcal{C}_2$

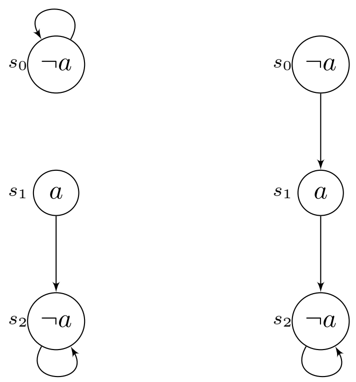

Fig. 2.3 – Les stratégies possibles du joueur depuis $\mathcal{C}_1$

Lemme 3. *$\mathsf{CTL} \leq_e \mathcal{L}_0$, avec une traduction en temps linéaire.*

Preuve : On définit par induction l'application $\sigma\colon \mathsf{CTL}_p \to \mathcal{L}_{0,p}$ telle que :

- Si $\varphi = a \in \mathsf{AP}$, alors $\sigma(\varphi) = a$,
- Si $\varphi = \neg\psi$, alors $\sigma(\varphi) = \neg\sigma(\psi)$,
- Si $\varphi = \psi_1 \vee \psi_2$, alors $\sigma(\varphi) = \sigma(\psi_1) \vee \sigma(\psi_2)$,
- Si $\varphi = \mathbf{X}\,\psi$, alors $\sigma(\varphi) = \mathbf{X}\,\sigma(\psi)$,
- Si $\varphi = \psi_1\,\mathbf{U}\,\psi_2$, alors $\sigma(\varphi) = \sigma(\psi_1)\,\mathbf{U}\,\sigma(\psi_2)$,
- Si $\varphi = \mathbf{A}\psi$, alors $\sigma(\varphi) = \lessdot\emptyset\gtrdot_0\ \sigma(\psi)$,
- Si $\varphi = \mathbf{E}\psi$, alors $\sigma(\varphi) = \neg \lessdot\emptyset\gtrdot_0 \neg\sigma(\psi)$.

On remarque d'abord que $\sigma(\mathsf{CTL}) \subset \mathcal{L}_0$. On va prouver que toute formule φ de CTL est équivalente à $\sigma(\varphi)$ sur l'ensemble des CGS :

Soit $\mathcal{C}$ une CGS. On montre par induction sur les formules de CTL_p que :
$\forall \varphi \in \mathsf{CTL}_p, \forall s \in \mathsf{Loc}, \forall \lambda \in \mathsf{Exec}(s), \lambda \models \varphi$ ssi $\lambda \models \sigma(\varphi)$
- Si $\varphi = a \in \mathsf{AP}$. Soient $s \in \mathsf{Loc}$ et $\lambda \in \mathsf{Exec}(s)$. Les deux formules sont vraies ssi $a \in \mathsf{Loc}(s)$,
- Si $\varphi = \neg\psi$, alors c'est évident,
- Si $\varphi = \psi_1 \vee \psi_2$, alors c'est évident,
- Si $\varphi = \mathbf{X}\,\psi$, où $\psi \in \mathsf{CTL}$. Soient $s \in \mathsf{Loc}$ et $\lambda \in \mathsf{Exec}(s)$. $\lambda \models \sigma(\varphi)$ est équivalent à $\lambda[1] \models \sigma(\psi)$, ou à $\lambda[1,\infty] \models \sigma(\psi)$ puisque $\sigma(\psi) \in \mathcal{L}_0$. On applique HI à la formule ψ, à l'état $\lambda[1]$ et à la $\lambda[1]$-exécution $\lambda[1,\infty]$. Donc $\lambda[1,\infty] \models \sigma(\psi)$ est équivalent à $\lambda[1,\infty] \models \psi$, et donc à $\lambda \models \mathbf{X}\,\psi$.
- Si $\varphi = \psi_1\,\mathbf{U}\,\psi_2$, où $\psi_1, \psi_2 \in \mathsf{CTL}$, cela ne présente pas d'autres difficultés.
- Si $\varphi = \mathbf{A}\psi$. Soient $s \in \mathsf{Loc}$ et $\lambda \in \mathsf{Exec}(s)$. $\lambda \models \sigma(\varphi)$ est équivalent à $s \models \lessdot\emptyset\gtrdot_0\ \sigma(\psi)$, c'est-à-dire à $\forall \lambda' \in \mathsf{Exec}(s), \lambda' \models \sigma(\psi)$. Il reste à appliquer HI à chaque λ'.
- Si $\varphi = \mathbf{E}\psi$, le raisonnement est similaire.

En restreignant ce résultat aux formules de CTL, on obtient la propriéte souhaitée. $\square$

2.1.3 Complexité de la logique $\mathcal{L}_0$

Lemme 4. *Le model-checking de* $\mathcal{L}_0$ *est* PSPACE*-dur.*

Preuve : On envisage une réduction de QBF-SAT à ce problème. Soient $x_1, ..., x_n$ des variables propositionnelles. Soit φ une formule propositionnelle sous forme CNF sur les variables $x_1, ..., x_n$. La formule :

$$\theta = \exists x_1 \forall x_2 \exists x_3 ... Q_n x_n \varphi(x_1, ..., x_n),$$

où Q_n est un $\forall$ si n est pair et un $\exists$ sinon, constitue une instance quelconque de QBF-SAT. Le problème de décicsion auquel on doit répondre pour résoudre QBF-SAT est : a-t-on $\{0,1\} \models \theta$, le symbole $\models$ étant ici celui de la logique du premier ordre ?

Supposons $\varphi = \bigwedge_{j=1,...,J} C_j$. On associe alors à cette instance de QBF-SAT la $CGS\,\mathcal{C} = (\mathsf{Agt}, \mathsf{Loc}, \mathsf{AP}, \mathsf{Lab}, \mathsf{Mov}, \mathsf{Edg})$, avec :
- $\mathsf{Agt} = \{A_1, ..., A_n\}$,
- $\mathsf{Loc} = \{x_i \mid\ 1 \le i \le n\} \cup \{\neg x_i \mid\ 1 \le i \le n\} \cup \{v_i \mid\ 1 \le i \le n+1\}$,
- $\mathsf{AP} = \{C_j \mid\ 1 \le j \le J\}$,
- Lab est telle que :
$$\forall 1 \le i \le n, \mathsf{Lab}(x_i) = \{C_j \mid\ x_i \text{ apparaît dans la clause } C_j\},$$
$$\forall 1 \le i \le n, \mathsf{Lab}(\neg x_i) = \{C_j \mid\ \neg x_i \text{ apparaît dans la clause } C_j\},$$
$$\forall 1 \le i \le n+1, \mathsf{Lab}(v_i) = \emptyset.$$
- Mov est telle que :
$$\forall 1 \le i, k \le n, \mathsf{Mov}(x_i, A_k) = 1,$$
$$\forall 1 \le i, k \le n, \mathsf{Mov}(\neg x_i, A_k) = 1,$$
$$\forall 1 \le i \le n+1, \forall 1 \le k \le n, \mathsf{Mov}(v_i, A_k) = \begin{cases} 2 & \text{si } i = k, \\ 1 & \text{sinon.} \end{cases}$$

– Edg relie chaque x_i à v_{i+1}, chaque $\neg x_i$ à v_{i+1}, chaque v_i à la fois à $\neg x_i$ par le mouvement 1 et à x_i par le mouvement 2, et enfin v_{n+1} à lui-même (voir la Fig. 4.).

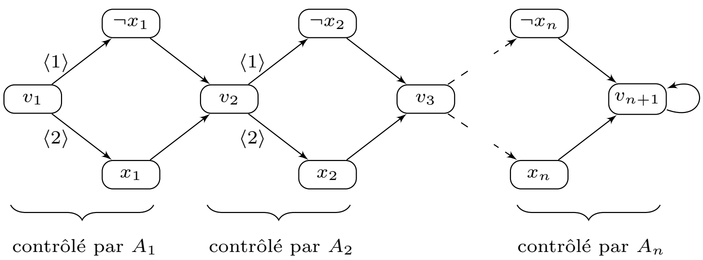

FIG. 2.4 – La CGS $\mathcal{C}$ sans les labels

On définit par récurrence la suite finie $(\varphi_k)_{k\in\{0,..,n\}}$ de formules du premier ordre telle que :

– $\varphi_0 = \varphi$,

– Si $k \in \{0,..,n-1\}, \varphi_{k+1} = \begin{cases} \forall x_{n-k}\varphi_k & \text{si } n-k \text{ est pair ,} \\ \exists x_{n-k}\varphi_k & \text{sinon.} \end{cases}$

En particulier, on a $\varphi_n = \theta$.

On définit par récurrence sur l'ensemble $\{\varphi_k \mid \ k \in \{0,..,n\}\}$ l'application ν à valeurs dans $\mathcal{L}_0$ telle que :

– $\nu(\varphi_0) = \bigwedge_{j=1,\dots,J} \lessdot\emptyset\gtrdot_0 \mathbf{F}\, C_j$,

– Si $k \in \{0,..,n-1\}, \nu(\varphi_{k+1}) = \begin{cases} \neg \lessdot A_{n-k} \gtrdot_0 \neg\nu(\varphi_k) & \text{si } n-k \text{ est pair ,} \\ \lessdot A_{n-k} \gtrdot_0 \nu(\varphi_k) & \text{sinon.} \end{cases}$

La CGS $\mathcal{C}$ et $\nu(\theta)$ sont constructibles en espace logarithmique en fonction de la taille de l'instance θ de QBF-SAT. Il reste donc à établir que cette instance du model-checking de $\mathcal{L}_0$ avec état initial en v_1 donne une réponse positive ssi l'instance de QBF-SAT est vraie, c'est-à-dire que $(\mathcal{C}, v_1) \models \nu(\theta)$ ssi $\{0,1\} \models \theta$.

On va montrer la propriété : $\forall k \in \{0,..,n\}$, $\forall F = (F_{A_i})_{A_i\in\{A_1,\dots,A_{n-k}\}} \in \Sigma^0_{\{A_1,\dots,A_{n-k}\}}$,

$$(\{0,1\}, x_1 \to (F_{A_1}(v_1)-1), \dots, x_{n-k} \to (F_{A_{n-k}}(v_{n-k})-1)) \models \varphi_k \ \text{ ssi}$$

$$(\mathcal{C}, v_1) \models_F \nu(\varphi_k),$$

On prouve cela par récurrence sur k :

– $k = 0$. Soit $F = (F_{A_i})_{A_i\in\mathsf{Agt}} \in \Sigma^0_{\mathsf{Agt}}$. On doit montrer :
$$(\{0,1\}, x_1 \to (F_{A_1}(v_1)-1), \dots, x_n \to (F_{A_n}(v_n)-1)) \models \bigwedge_{j=1,\dots,J} C_j \ \text{ ssi}$$
$$(\mathcal{C}, v_1) \models_F \bigwedge_{j=1,\dots,J} \lessdot\emptyset\gtrdot_0 \mathbf{F}\, C_j.$$
Remarquons que l'ensemble $\mathsf{Out}(v_1, F)$ est un singleton, puisque chaque joueur a une stratégie fixée. On appelle λ l'unique exécution de cet ensemble. On a alors :

$$\lambda[i] = \begin{cases} v_{n+1} & \text{si } i \geq 2n, \\ v_{\frac{i}{2}+1} & \text{si } 0 \leq i \leq 2n-1 \text{ et } i \text{ est pair}, \\ \neg x_{\frac{i+1}{2}} & \text{si } 0 \leq i \leq 2n-1,\ i \text{ est impair et } F_{A_{\frac{i+1}{2}}}(v_{\frac{i+1}{2}}) = 1, \\ x_{\frac{i+1}{2}} & \text{si } 0 \leq i \leq 2n-1,\ i \text{ est impair et } F_{A_{\frac{i+1}{2}}}(v_{\frac{i+1}{2}}) = 2. \end{cases}$$

Soit $j \in \{1,..,J\}$. Supposons $C_j = \bigvee_{l=1,...,L} \epsilon_l x_l$. On va montrer :

$$(\{0,1\}, x_1 \to (F_{A_1}(v_1)-1), ..., x_n \to (F_{A_n}(v_n)-1)) \models C_j \text{ ssi } (\mathcal{C}, \lambda) \models_F \mathbf{F}\, C_j.$$

On a la suite d'équivalences :
$(\mathcal{C}, \lambda) \models_F \mathbf{F}\, C_j$ ssi
$\exists i \geq 0, (\mathcal{C}, \lambda[i]) \models_F C_j$ ssi
$\exists i \geq 0, C_j \in \mathsf{Lab}(\lambda[i])$ ssi
$\exists i \geq 0, \lambda[i] \in C_j$ ssi
$\exists i \geq 0, \exists l \in \{1,...,L\}, \lambda[i] = \epsilon_l x_l$ ssi, en supposant que $\epsilon_l = \emptyset$, le cas $\epsilon_l = \neg$ étant similaire,
$\exists i \geq 0, \exists l \in \{1,...,L\}, \lambda[i] = x_l$ ssi (d'après ce que l'on sait sur λ)
$\exists l \in \{1,...,L\}, F_{A_l}(v_l) = 2$ ssi
$\exists l \in \{1,...,L\}, (\{0,1\}, x_l \to (F_{A_l}(v_l)-1)) \models x_l$ ssi
$\exists l \in \{1,...,L\}, (\{0,1\}, x_1 \to (F_{A_1}(v_1)-1), ..., x_n \to (F_{A_n}(v_n)-1)) \models \epsilon_l x_l$ ssi
$(\{0,1\}, x_1 \to (F_{A_1}(v_1)-1), ..., x_n \to (F_{A_n}(v_n)-1)) \models C_j$.

– Soit $k \in \{0,..,n-1\}$, pour lequel la propriété est vérifiée. Pour achever la récurrence, on doit distinguer deux cas selon que $n-k$ est pair ou non, et montrer que dans les deux cas la propriété est vraie au rang $k+1$. Supposons $n-k$ pair, l'autre cas étant similaire. Soit $F = (F_{A_i})_{A_i \in \{A_1,...,A_{n-k-1}\}} \in \Sigma^0_{\{A_1,...,A_{n-k-1}\}}$. On doit montrer :

$$(\{0,1\}, x_1 \to (F_{A_1}(v_1)-1), ..., x_{n-k-1} \to (F_{A_{n-k-1}}(v_{n-k-1})-1)) \models \forall x_{n-k} \varphi_k \text{ ssi } (\mathcal{C}, v_1) \models_F \neg \lessdot A_{n-k} \gtrdot_0 \neg\nu(\varphi_k).$$

On a la suite d'équivalences :
$(\mathcal{C}, v_1) \models_F \neg \lessdot A_{n-k} \gtrdot_0 \neg\nu(\varphi_k)$ ssi
$\forall f_{n-k} \in \Sigma^0_{A_{n-k}}, (\mathcal{C}, v_1) \models_{f_{n-k} \circ_0 F} \nu(\varphi_k)$ ssi, par HR appliquée à $f_{n-k} \circ_0 F \in \Sigma^0_{\{A_1,...,A_{n-k}\}}$,
$\forall f_{n-k} \in \Sigma^0_{A_{n-k}}, (\{0,1\}, x_1 \to (F_{A_1}(v_1)-1), ..., x_{n-k} \to (f_{n-k}(v_{n-k})-1)) \models \varphi_k$ ssi, puisque $\Sigma^0_{A_{n-k}}$ ne contient que deux éléments, disons f et f', tels que $f(v_{n-k}) = 1$ et $f'(v_{n-k}) = 2$,
$\forall a \in \{0,1\}, (\{0,1\}, x_1 \to (F_{A_1}(v_1)-1), ..., x_{n-k} \to a) \models \varphi_k$ ssi
$(\{0,1\}, x_1 \to (F_{A_1}(v_1)-1), ..., x_{n-k-1} \to (F_{A_{n-k-1}}(v_{n-k-1})-1)) \models \forall x_{n-k} \varphi_k$.

Cette propriété, appliquée au rang n, c'est-à-dire au cas où F est la stratégie $\emptyset$, signifie que l'on a bien construit une réduction entre les deux problèmes considérés. $\square$

2.1.4 Définition et expressivité de la logique $\mathcal{L}_0^*$

En s'inspirant de ATL^*, on définit à présent l'extension naturelle $\mathcal{L}_0^*$.

Définition 14. *La syntaxe de $\mathcal{L}_0^*$ est définie par la grammaire suivante :*

$$\begin{array}{rcl}\mathcal{L}_0^* \ni \varphi_s, \psi_s & ::= & p \mid \neg\varphi_s \mid \varphi_s \vee \psi_s \mid \lessdot A \gtrdot_0 \varphi_p \\ \mathcal{L}_{0,p}^* \ni \varphi_p, \psi_p & ::= & \varphi_s \mid \neg\varphi_p \mid \varphi_p \vee \psi_p \mid \mathbf{X}\,\varphi_p \mid \varphi_p \,\mathbf{U}\, \psi_p.\end{array}$$

où p décrit AP et A les sous-ensembles de Agt.

La sémantique de $\mathcal{L}_0^*$ est définie comme suit :
Soient $A, B \subseteq \mathsf{Agt}, q \in \mathsf{Loc}$ et $F \in \Sigma_A^0$,

$$\begin{array}{rcl}
q \models_F p & \text{ssi} & p \in \mathsf{Lab}(q),\\
q \models_F \neg\varphi_s & \text{ssi} & q \not\models_F \varphi_s,\\
q \models_F \varphi_s \vee \psi_s & \text{ssi} & q \models_F \varphi_s \text{ ou } q \models_F \psi_s,\\
q \models_F \lessdot B \gtrdot_0 \varphi_p & \text{ssi} & \exists F' \in \Sigma_B^0 \text{ tel que } \forall\lambda \in \mathsf{Out}(q, F' \circ_0 F), \lambda \models_{F'\circ_0 F} \varphi_p,\\
\lambda \models_F \varphi_s & \text{ssi} & \lambda[0] \models_F \varphi_s,\\
\lambda \models_F \neg\varphi_p & \text{ssi} & \lambda \not\models_F \varphi_p,\\
\lambda \models_F \varphi_p \vee \psi_p & \text{ssi} & \lambda \models_F \varphi_p \text{ ou } \lambda \models_F \psi_p,\\
\lambda \models_F \mathbf{X}\,\varphi_p & \text{ssi} & \lambda[1,\infty] \models_F \varphi_p,\\
\lambda \models_F \varphi_p \,\mathbf{U}\, \psi_p & \text{ssi} & \exists i \geq 0.\ \lambda[i,\infty] \models_F \psi_p \text{ et } \forall 0 \leq j < i.\ \lambda[j,\infty] \models_F \varphi_p.
\end{array}$$

Lemme 5. *$\mathsf{CTL}^* \leq_e \mathcal{L}_0^*$, avec une traduction en temps linéaire.*

Preuve : On définit par induction l'application $\sigma\colon \mathsf{CTL}_p^* \to \mathcal{L}_{0,p}^*$ telle que :
- Si $\varphi = a \in \mathsf{AP}$, alors $\sigma(\varphi) = a$,
- Si $\varphi = \neg\psi$, alors $\sigma(\varphi) = \neg\sigma(\psi)$,
- Si $\varphi = \psi_1 \vee \psi_2$, alors $\sigma(\varphi) = \sigma(\psi_1) \vee \sigma(\psi_2)$,
- Si $\varphi = \mathbf{X}\,\psi$, alors $\sigma(\varphi) = \mathbf{X}\,\sigma(\psi)$,
- Si $\varphi = \psi_1 \,\mathbf{U}\, \psi_2$, alors $\sigma(\varphi) = \sigma(\psi_1)\,\mathbf{U}\,\sigma(\psi_2)$,
- Si $\varphi = \mathbf{A}\psi$, alors $\sigma(\varphi) = \lessdot\emptyset\gtrdot_0 \sigma(\psi)$,
- Si $\varphi = \mathbf{E}\psi$, alors $\sigma(\varphi) = \neg \lessdot \emptyset \gtrdot_0 \neg\sigma(\psi)$.

On va prouver que toute formule φ de CTL^* est équivalente à $\sigma(\varphi)$ sur l'ensemble des CGS :

Soit $\mathcal{C}$ une CGS. On montre par induction sur les formules de CTL_p^* que :
$\forall\varphi \in \mathsf{CTL}_p^*, \forall s \in \mathsf{Loc}, \forall\lambda \in \mathsf{Exec}(s), \lambda \models \varphi$ ssi $\lambda \models \sigma(\varphi)$
- Si $\varphi = a \in \mathsf{AP}$. Soient $s \in \mathsf{Loc}$ et $\lambda \in \mathsf{Exec}(s)$. Les deux formules sont vraies ssi $a \in \mathsf{Loc}(s)$,
- Si $\varphi = \neg\psi$, alors c'est évident,
- Si $\varphi = \psi_1 \vee \psi_2$, alors c'est évident,
- Si $\varphi = \mathbf{X}\,\psi$. Soient $s \in \mathsf{Loc}$ et $\lambda \in \mathsf{Exec}(s)$. $\lambda \models \sigma(\varphi)$ est équivalent à $\lambda[1,\infty] \models \sigma(\psi)$. On applique HI à la formule ψ, à l'état $\lambda[1]$ et à la $\lambda[1]$-exécution $\lambda[1,\infty]$. Donc $\lambda[1,\infty] \models \sigma(\psi)$ est équivalent à $\lambda[1,\infty] \models \psi$, et donc à $\lambda \models \mathbf{X}\,\psi$.

– Si $\varphi = \psi_1 \mathbf{U} \psi_2$, cela ne présente pas d'autres difficultés.
– Si $\varphi = \mathbf{A}\psi$. Soient $s \in \mathsf{Loc}$ et $\lambda \in \mathsf{Exec}(s)$. $\lambda \models \sigma(\varphi)$ est équivalent à $s \models \lessdot\emptyset\gtrdot_0 \sigma(\psi)$, c'est-à-dire à $\forall \lambda' \in \mathsf{Exec}(s), \lambda' \models \sigma(\psi)$. Il reste à appliquer HI à chaque λ'.
– Si $\varphi = \mathbf{E}\psi$, le raisonnement est similaire.

En restreignant ce résultat aux formules de CTL^*, on obtient la propriéte souhaitée. $\square$

2.2 Emboîter et oublier des stratégies sans mémoire

Nous allons à présent définir une opération d'oubli de stratégie sans mémoire, qui correspond en fait à une restriction, et que l'on note $|_0$.

Définition 15. *Soient $\mathcal{C}$ une* CGS, $A, C \subseteq \mathsf{Agt}$ *tels que $C \subseteq A$ et $F = (F_{A_i})_{A_i \in A} \in \Sigma^0_A$.*

On définit $F|_{0,C} = (F_{A_i})_{A_i \in C}$. On a donc $F|_{0,C} \in \Sigma^0_C$. En particulier, $F|_{0,\emptyset} = \emptyset$

2.2.1 La logique $\mathcal{L}_0^{**}$

On étend à nouveau $\mathcal{L}_0^*$ en $\mathcal{L}_0^{**}$, en s'autorisant maintenant à oublier des stratégies, cela par un nouveau quantificateur de stratégie noté $\gtrdot\lessdot_0$.

Définition 16. *La syntaxe de $\mathcal{L}_0^{**}$ est définie par la grammaire suivante :*

$$\begin{aligned}
\mathcal{L}_0^{**} \ni \varphi_s, \psi_s &::= p \mid \neg\varphi_s \mid \varphi_s \vee \psi_s \mid \lessdot A \gtrdot_0 \varphi_p \mid \gtrdot A \lessdot_0 \varphi_p \\
\mathcal{L}_{0,p}^{**} \ni \varphi_p, \psi_p &::= \varphi_s \mid \neg\varphi_p \mid \varphi_p \vee \psi_p \mid \mathbf{X}\,\varphi_p \mid \varphi_p \,\mathbf{U}\, \psi_p.
\end{aligned}$$

où p décrit AP et A les sous-ensembles de Agt.

La sémantique de $\mathcal{L}_0^{**}$ est définie comme suit :
Soient $A, B \subseteq \mathsf{Agt}, q \in \mathsf{Loc}$ et $F \in \Sigma^0_A$.

$$\begin{array}{rcl}
q \models_F p & \text{ssi} & p \in \mathsf{Lab}(q), \\
q \models_F \neg\varphi_s & \text{ssi} & q \not\models_F \varphi_s, \\
q \models_F \varphi_s \vee \psi_s & \text{ssi} & q \models_F \varphi_s \text{ ou } q \models_F \psi_s, \\
q \models_F \lessdot B \gtrdot_0 \varphi_p & \text{ssi} & \exists F' \in \Sigma^0_B \text{ tel que } \forall \lambda \in \mathsf{Out}(q, F' \circ_0 F), \lambda \models_{F' \circ_0 F} \varphi_p, \\
q \models_F \gtrdot B \lessdot_0 \varphi_p & \text{ssi} & \forall \lambda \in \mathsf{Out}(q, F|_{0,A\setminus B}), \lambda \models_{F|_{0,A\setminus B}} \varphi_p, \\
\lambda \models_F \varphi_s & \text{ssi} & \lambda[0] \models_F \varphi_s, \\
\lambda \models_F \neg\varphi_p & \text{ssi} & \lambda \not\models_F \varphi_p, \\
\lambda \models_F \varphi_p \vee \psi_p & \text{ssi} & \lambda \models_F \varphi_p \text{ ou } \lambda \models_F \psi_p, \\
\lambda \models_F \mathbf{X}\,\varphi_p & \text{ssi} & \lambda[1,\infty] \models_F \varphi_p, \\
\lambda \models_F \varphi_p \,\mathbf{U}\, \psi_p & \text{ssi} & \exists i \geq 0.\ \lambda[i,\infty] \models_F \psi_p \text{ et } \forall 0 \leq j < i.\ \lambda[j,\infty] \models_F \varphi_p.
\end{array}$$

Remarque :

– $q \models_F \ \gtrdot \mathsf{Agt} \lessdot_0 \ \varphi_p$ signifie $\forall \lambda \in \mathsf{Exec}(q), \lambda \models \varphi_p$.

Définition 17. *On peut maintenant définir le rang de quantification d'une formule $\varphi \in \mathcal{L}^{**}_{0,p}$, que l'on notera $rq(\varphi)$. Il s'agit d'une application de $\mathcal{L}^{**}_{0,p}$ dans $\mathbb{N}$ définie par induction de la manière suivante :*

- *Si $\varphi = a \in \mathsf{AP}$, alors $rq(\varphi) = 0$,*
- *Si $\varphi = \neg\psi$, alors $rq(\varphi) = rq(\psi)$,*
- *Si $\varphi = \psi_1 \vee \psi_2$, alors $rq(\varphi) = \mathsf{Max}(rq(\psi_1), rq(\psi_2))$,*
- *Si $\varphi = \ \lessdot A \gtrdot_0 \ \psi$, alors $rq(\varphi) = rq(\psi) + 1$,*
- *Si $\varphi = \ \gtrdot A \lessdot_0 \ \psi$, alors $rq(\varphi) = rq(\psi) + 1$,*
- *Si $\varphi = \mathbf{X}\,\psi$, alors $rq(\varphi) = rq(\psi)$,*
- *Si $\varphi = \psi_1 \,\mathbf{U}\, \psi_2$, alors $rq(\varphi) = \mathsf{Max}(rq(\psi_1), rq(\psi_2))$.*

2.2.2 Expressivité de la logique $\mathcal{L}^{**}_0$

Lemme 6. *$\mathsf{ATL} \leq_e \mathcal{L}^{**}_0$, avec une traduction en temps linéaire.*

Preuve : On définit par induction l'application $\sigma \colon \mathsf{ATL}_p \to \mathcal{L}^{**}_{0,p}$ telle que :

- Si $\varphi = a \in \mathsf{AP}$, alors $\sigma(\varphi) = a$,
- Si $\varphi = \neg\psi$, alors $\sigma(\varphi) = \neg\sigma(\psi)$,
- Si $\varphi = \psi_1 \vee \psi_2$, alors $\sigma(\varphi) = \sigma(\psi_1) \vee \sigma(\psi_2)$,
- Si $\varphi = \langle\!\langle B \rangle\!\rangle\, \psi$, alors $\sigma(\varphi) = \ \gtrdot \mathsf{Agt} \lessdot_0 \ \lessdot B \gtrdot_0 \ \sigma(\psi)$,
- Si $\varphi = \mathbf{X}\,\psi$, alors $\sigma(\varphi) = \mathbf{X}\,\sigma(\psi)$,
- Si $\varphi = \psi_1 \,\mathbf{U}\, \psi_2$, alors $\sigma(\varphi) = \sigma(\psi_1) \,\mathbf{U}\, \sigma(\psi_2)$,
- Si $\varphi = \psi_1 \,\mathbf{W}\, \psi_2$, alors $\sigma(\varphi) = (\sigma(\psi_1) \,\mathbf{U}\, \sigma(\psi_2)) \vee \mathbf{G}\,\sigma(\psi_1)$.

On va prouver que toute formule φ de ATL est équivalente à $\sigma(\varphi)$ sur l'ensemble des CGS :

Soit $\mathcal{C}$ une CGS. On montre par induction sur les formules de ATL_p que :

$\forall \varphi \in \mathsf{ATL}_p, \forall s \in \mathsf{Loc}, \forall \lambda \in \mathsf{Exec}(s), \forall A \subseteq \mathsf{Agt}, \forall F \in \Sigma^0_A, \lambda \models \varphi$ ssi $\lambda \models_F \sigma(\varphi)$

- Si $\varphi = a \in \mathsf{AP}$. Soient $s \in \mathsf{Loc}, \lambda \in \mathsf{Exec}(s), A \subseteq \mathsf{Agt}$ et $F \in \Sigma^0_A$. Les deux formules sont vraies ssi $a \in \mathsf{Loc}(s)$,
- Si $\varphi = \neg\psi$, alors c'est évident,
- Si $\varphi = \psi_1 \vee \psi_2$, alors c'est évident,
- Si $\varphi = \langle\!\langle B \rangle\!\rangle\, \psi$. Soient $s \in \mathsf{Loc}, \lambda \in \mathsf{Exec}(s), A \subseteq \mathsf{Agt}$ et $F \in \Sigma^0_A$. $\lambda \models_F \sigma(\varphi)$ est équivalent à $s \models_F \ \gtrdot \mathsf{Agt} \lessdot_0 \ \lessdot B \gtrdot_0 \ \sigma(\psi)$, c'est-à-dire à $\forall \lambda' \in \mathsf{Exec}(s), \lambda' \models \ \lessdot B \gtrdot_0 \sigma(\psi)$, soit puisque $\lessdot B \gtrdot_0 \sigma(\psi)$ est une formule d'état, $s \models \ \lessdot B \gtrdot_0 \ \sigma(\psi)$. Et cette dernière expression signifie par définition $\exists F' \in \Sigma^0_B$ tel que $\forall \lambda \in \mathsf{Out}(q, F'), \lambda \models_{F'} \sigma(\psi)$. Il reste à appliquer HI à chaque λ pour obtenir l'équivalence avec $\exists F' \in \Sigma^0_B$ tel que $\forall \lambda \in \mathsf{Out}(q, F'), \lambda \models \psi$, qui est exactement $q \models \langle\!\langle B \rangle\!\rangle\, \psi$, donc $\lambda \models \langle\!\langle B \rangle\!\rangle\, \psi$.
- Si $\varphi = \mathbf{X}\,\psi$. Soient $s \in \mathsf{Loc}, \lambda \in \mathsf{Exec}(s), A \subseteq \mathsf{Agt}$ et $F \in \Sigma^0_A$. $\lambda \models_F \sigma(\varphi)$ est équivalent à $\lambda[1, \infty] \models_F \sigma(\psi)$. On applique HI à la formule ψ, à l'état $\lambda[1]$, à la $\lambda[1]$-exécution $\lambda[1, \infty]$, et à la stratégie F. Donc $\lambda[1, \infty] \models_F \sigma(\psi)$ est équivalent à $\lambda[1, \infty] \models \psi$, et donc à $\lambda \models \mathbf{X}\,\psi$.
- Si $\varphi = \psi_1 \,\mathbf{U}\, \psi_2$, cela ne présente pas d'autres difficultés,

– Si $\varphi = \psi_1 \,\mathbf{W}\, \psi_2$, cela ne présente pas d'autres difficultés.
En restreignant ce résultat aux formules de ATL et à la stratégie $\emptyset$, on obtient la propriéte souhaitée. $\square$

2.2.3 Complexité de la logique $\mathcal{L}_0^{**}$

On va maintenant s'attaquer au problème de model-checking de la logique $\mathcal{L}_0^{**}$, que l'on notera MC $\mathcal{L}_0^{**}$. Pour cela, l'idée essentielle est de remarquer que les issues d'une stratégie sans mémoire correspondent toujours aux exécutions d'une *CGS* particulière (c'est ce qui motive la définition et le lemme suivants), puis de recourir au model-checking de LTL. La difficulté principale est alors la préservation des contextes imposés par les stratégies, mais elle est résolue en rajoutant des propositions atomiques en nombre suffisant.

Définition 18. *Soient $\mathcal{C} = (\mathit{Agt}, \mathit{Loc}, \mathit{AP}, \mathit{Lab}, \mathit{Mov}, \mathit{Edg})$ une* CGS, *$A \subseteq \mathit{Agt}$ et $F = (F_{A_i})_{A_i \in A} \in \Sigma^0_A$, avec la notation $|\mathit{Agt}| = k$.*

On définit $(\mathcal{C}, F) = (\mathit{Agt}, \mathit{Loc}, \mathit{AP}, \mathit{Lab}, \mathit{Mov}', \mathit{Edg}')$, la sous-structure de $\mathcal{C}$ induite par la stratégie sans mémoire F, de la manière suivante :

– Mov' : $\begin{cases} \mathit{Loc} \times \mathit{Agt} & \rightarrow \quad \mathbb{N}_{\geq 1} \\ (q, A_i) & \mapsto \quad \mathit{Mov}'(q, A_i) = \begin{cases} 1 & \text{si } A_i \in A \\ \mathit{Mov}(q, A_i) & \text{sinon} \end{cases} \end{cases}$;

– Edg' : $\begin{cases} \mathit{Loc} \times \mathbb{N}^k & \rightarrow \quad \mathit{Loc} \\ (q, m_{A_1}, ..., m_{A_k}) & \mapsto \quad \mathit{Edg}'(q, m_{A_1}, ..., m_{A_k}) = \mathit{Edg}(q, m'_{A_1}, ..., m'_{A_k}) \end{cases}$,

où $m'_{A_i} = \begin{cases} F_{A_i}(q) & \text{si } A_i \in A \\ m_{A_i} & \text{sinon} \end{cases}$;

Remarques :
- Dans le cas particulier $A = \emptyset$, on a $(\mathcal{C}, \emptyset) = \mathcal{C}$
- Intuitivement, cette opération consiste à ne conserver, pour chaque joueur concerné par la stratégie, que le mouvement imposé par cette stratégie (qui devient donc le mouvement 1).
- Lorsque l'on fait cette opération, on peut supprimer des transitions et isoler des points.

Lemme 7. *Pour toute* CGS *$\mathcal{C}$, $\forall A \subseteq \mathit{Agt}, \forall F \in \Sigma^0_A, \forall s \in \mathit{Loc}$,*
$\mathsf{Out}_{\mathcal{C}}(s, F) = \mathsf{Exec}_{(\mathcal{C},F)}(s)$.

Preuve : Soient $\mathcal{C} = (\mathsf{Agt}, \mathsf{Loc}, \mathsf{AP}, \mathsf{Lab}, \mathsf{Mov}, \mathsf{Edg})$ une *CGS*, $A \subseteq \mathsf{Agt}$, $F = (F_{A_i})_{A_i \in A} \in \Sigma^0_A$ et $s \in \mathsf{Loc}$.

Soit $\lambda \in \mathsf{Out}_{\mathcal{C}}(s, F)$. On a tout de suite $\lambda[0] = s$. Soit $k \in \mathbb{N}$. Par hypothèse, $\lambda[k+1] \in \mathsf{Next}_{\mathcal{C}}(\lambda[k], A, F(\lambda[k]))$, donc il existe une suite de mouvements $m_A = (m_{A_i})_{A_i \in \mathsf{Agt}}$ telle que $\mathsf{Edg}(\lambda[k], m_A) = \lambda[k+1]$ et $\forall A_i \in A, m_{A_i} = F_{A_i}(\lambda[k])$. Soit alors la suite de mouvements $m'_A = (m'_{A_i})_{A_i \in \mathsf{Agt}}$ telle que $m'_{A_i} = 1$ si $A_i \in A$, et $m'_{A_i} = m_{A_i}$ sinon. On obtient $\mathsf{Edg}'(\lambda[k], m'_A) = \mathsf{Edg}(\lambda[k], m_A) = \lambda[k+1]$, donc $\lambda[k+1] \in \mathsf{Next}_{(\mathcal{C},F)}(\lambda[k])$ et $\lambda \in \mathsf{Exec}_{(\mathcal{C},F)}(s)$.

Soit $\lambda \in \mathsf{Exec}_{(\mathcal{C},F)}(s)$. On a tout de suite $\lambda[0] = s$. Soit $k \in \mathbb{N}$. Par hypothèse, $\lambda[k+1] \in \mathsf{Next}_{(\mathcal{C},F)}(\lambda[k])$, donc il existe une suite de mouvements $m_A = (m_{A_i})_{A_i \in \mathsf{Agt}}$ telle que $\mathsf{Edg}_{(\mathcal{C},F)}(\lambda[k], m_A) = \lambda[k+1]$. Soit $m'_A = (m'_{A_i})_{A_i \in \mathsf{Agt}}$ tel que $m'_{A_i} = F_{A_i}(\lambda[k])$ si $A_i \in A$, et $m'_{A_i} = m_{A_i}$ sinon. Alors $\mathsf{Edg}'(\lambda[k], m_A) = \mathsf{Edg}(\lambda[k], m'_A) = \lambda[k+1]$. On a ainsi prouvé que $\lambda[k+1] \in \mathsf{Next}_{\mathcal{C}}(\lambda[k], A, F(\lambda[k]))$, donc $\lambda \in \mathsf{Out}_{\mathcal{C}}(s, F)$. $\square$

Lemme 8. *Le model-checking de $\mathcal{L}_0^{**}$ est dans* PSPACE.

Algorithm 1 MC $\mathcal{L}_0^{**}$

Require: une *CGS* $\mathcal{C}$, une stratégie $F \in \Sigma_A^0$, un état $s_0 \in \mathsf{Loc}$ et une formule θ_p de $\mathcal{L}_{0,p}^{**}$
Ensure: OUI ssi $\forall\lambda \in \mathsf{Out}_{\mathcal{C}}(s_0, F), (\mathcal{C}, \lambda) \models_F \theta_p$
$\mathcal{M} := (\mathcal{C}, F)$
$\Phi_1 := \{\psi \in Sub(\theta_p) \mid\ rq(\psi) = rq(\theta_p)$ et $\exists B \subseteq \mathsf{Agt}, \exists \psi' \in \mathcal{L}_0^{**}, \psi = \lessdot B \gtrdot_0 \psi'\}$
$\Phi_2 := \{\psi \in Sub(\theta_p) \mid\ rq(\psi) = rq(\theta_p)$ et $\exists B \subseteq \mathsf{Agt}, \exists \psi' \in \mathcal{L}_0^{**}, \psi = \gtrdot B \lessdot_0 \psi'\}$
for $\psi \in \Phi_1 \cup \Phi_2$ **do**
  $\mathsf{AP}_{\mathcal{M}} := \mathsf{AP}_{\mathcal{M}} \cup \{a_\psi\}$
  for $s \in \mathsf{Loc}$ **do**
    if $\psi \in \Phi_1$ **then**
      for $F' \in \Sigma_B^0$ **do**
        if MC $\mathcal{L}_0^{**}(\mathcal{C}, F' \circ_0 F, s, \psi')$ **then**
          $\mathsf{Lab}_{\mathcal{M}}(s) := \mathsf{Lab}_{\mathcal{M}}(s) \cup \{a_\psi\}$
        end if
      end for
    else
      if MC $\mathcal{L}_0^{**}(\mathcal{C}, F|_{0,A\setminus B}, s, \psi')$ **then**
        $\mathsf{Lab}_{\mathcal{M}}(s) := \mathsf{Lab}_{\mathcal{M}}(s) \cup \{a_\psi\}$
      end if
    end if
  end for
end for
if MC LTL $(\mathcal{M}, s_0, \mathbf{A}\sigma(\theta_p))$ **then**
  retourner OUI
else
  retourner NON
end if

Preuve : On définit par induction sur $\mathcal{L}_{0,p}^{**}$ l'application σ à valeurs dans LTL_p (avec toutefois plus de propositions atomiques) telle que :
- Si $\varphi_p = a \in \mathsf{AP}$, alors $\sigma(\varphi_p) = a$,
- Si $\varphi_p = \neg\psi$, alors $\sigma(\varphi_p) = \neg\sigma(\psi)$,

– Si $\varphi_p = \psi_1 \vee \psi_2$, alors $\sigma(\varphi_p) = \sigma(\psi_1) \vee \sigma(\psi_2)$,
– Si $\varphi_p = \lessdot B \gtrdot_0 \psi$, alors $\sigma(\varphi_p) = a_{\varphi_p}$,
– Si $\varphi_p = \gtrdot B \lessdot_0 \psi$, alors $\sigma(\varphi_p) = a_{\varphi_p}$,
– Si $\varphi_p = \mathbf{X}\,\psi$, alors $\sigma(\varphi_p) = \mathbf{X}\,\sigma(\psi)$,
– Si $\varphi_p = \psi_1 \,\mathbf{U}\, \psi_2$, alors $\sigma(\varphi_p) = \sigma(\psi_1) \,\mathbf{U}\, \sigma(\psi_2)$.

On va montrer que l'algorithme 1 fait bien ce que l'on affirme. On désigne par $\mathcal{M}_{f,(\mathcal{C},F,\theta_p)}$ la structure contenue dans la variable $\mathcal{M}$ en fin d'exécution de l'algorithme 1 sur l'entrée $(\mathcal{C}, F, s_0, \theta_p)$. Cette notation est justifiée par le fait que la valeur de s_0 n'importe pas.

Il est clair que pour toute entrée $(\mathcal{C}, F, s_0, \theta_p)$ de l'algorithme 1, les structures $(\mathcal{C}, F)$ et $\mathcal{M}_{f,(\mathcal{C},F,\theta_p)}$ ont mêmes états et mêmes transitions (elles ne diffèrent que par leurs propositions atomiques et leurs labels), donc $\forall s \in \mathsf{Loc}, \mathsf{Exec}_{(\mathcal{C},F)}(s) = \mathsf{Exec}_{\mathcal{M}_{f,(\mathcal{C},F,\theta_p)}}(s)$.

Soit une *CGS* $\mathcal{C}$. On va montrer par induction que pour toute formule $\varphi_p \in \mathcal{L}^{**}_{0,p}$, on a :

$\forall A \subseteq \mathsf{Agt}, \forall F \in \Sigma^0_A, \forall s \in \mathsf{Loc}, \forall \lambda \in \mathsf{Exec}_{(\mathcal{C},F)}(s)$,
$(\mathcal{C}, \lambda) \models_F \varphi_p$ ssi $(\mathcal{M}_{f,(\mathcal{C},F,\varphi_p)}, \lambda) \models \sigma(\varphi_p)$.

– Si $\varphi_p = a \in \mathsf{AP}$. Soient $A \subseteq \mathsf{Agt}, F \in \Sigma^0_A, s \in \mathsf{Loc}, \lambda \in \mathsf{Exec}_{(\mathcal{C},F)}(s)$. On doit montrer que $a \in \mathsf{Lab}_{\mathcal{M}_{f,(\mathcal{C},F,\varphi_p)}}(s)$ ssi $a \in \mathsf{Lab}_{\mathcal{C}}(s)$. Ceci est vrai, puisque la transformation de $\mathcal{C}$ en $\mathcal{M}_{f,(\mathcal{C},F,\varphi_p)}$ ne modifie pas les labels par des propositions atomiques de $\mathcal{C}$,
– Si $\varphi_p = \neg\psi$, alors c'est trivial puisque l'on cherche à montrer une équivalence,
– Si $\varphi_p = \psi_1 \vee \psi_2$, alors c'est trivial aussi,
– Si $\varphi_p = \lessdot B \gtrdot_0 \psi$. Soient $A \subseteq \mathsf{Agt}, F \in \Sigma^0_A, s \in \mathsf{Loc}, \lambda \in \mathsf{Exec}_{(\mathcal{C},F)}(s)$. On a la suite d'équivalences :
  $(\mathcal{M}_{f,(\mathcal{C},F,\varphi_p)}, \lambda) \models \sigma(\lessdot B \gtrdot_0 \psi)$ ssi
  $(\mathcal{M}_{f,(\mathcal{C},F,\varphi_p)}, \lambda) \models a_{\varphi_p}$ ssi
  $a_{\varphi_p} \in \mathsf{Lab}_{\mathcal{M}_{f,(\mathcal{C},F,\varphi_p)}}(s)$ ssi
  on a pu trouver une stratégie $F' \in \Sigma^0_B$ telle que l'algorithme 1 retourne OUI sur l'entrée $(\mathcal{C}, F' \circ_0 F, s, \psi)$ ssi (par définition de l'algorithme 1)
  $\exists F' \in \Sigma^0_B$, MC LTL $(\mathcal{M}_{f,(\mathcal{C},F' \circ_0 F, \mathbf{A}\psi)}, s, \sigma(\psi))$ ssi
  $\exists F' \in \Sigma^0_B, \forall \lambda' \in \mathsf{Exec}_{(\mathcal{C},F' \circ_0 F)}(s), (\mathcal{M}_{f,(\mathcal{C},F' \circ_0 F,\psi)}, \lambda') \models \sigma(\psi)$ ssi (par HI appliquée à ψ)
  $\exists F' \in \Sigma^0_B, \forall \lambda' \in \mathsf{Exec}_{(\mathcal{C},F' \circ_0 F)}(s), (\mathcal{C}, \lambda') \models_{F' \circ_0 F} \psi$ ssi, par le lemme 25,
  $(\mathcal{C}, s) \models_F \lessdot B \gtrdot_0 \psi$ ssi
  $(\mathcal{C}, \lambda) \models_F \lessdot B \gtrdot_0 \psi$.
– Si $\varphi_p = \gtrdot B \lessdot_0 \psi$. Soient $A \subseteq \mathsf{Agt}, F \in \Sigma^0_A, s \in \mathsf{Loc}, \lambda \in \mathsf{Exec}_{(\mathcal{C},F)}(s)$. On a la suite d'équivalences :
  $(\mathcal{M}_{f,(\mathcal{C},F,\varphi_p)}, \lambda) \models \sigma(\gtrdot B \lessdot_0 \psi)$ ssi
  $(\mathcal{M}_{f,(\mathcal{C},F,\varphi_p)}, \lambda) \models a_{\varphi_p}$ ssi
  $a_{\varphi_p} \in \mathsf{Lab}_{\mathcal{M}_{f,(\mathcal{C},F,\varphi_p)}}(s)$ ssi
  l'algorithme 1 retourne OUI sur l'entrée $(\mathcal{C}, F|_{0,A\setminus B}, s, \psi)$ ssi (par définition de l'algorithme 1)
  MC LTL $(\mathcal{M}_{f,(\mathcal{C},F|_{0,A\setminus B},\mathbf{A}\psi)}, s, \sigma(\psi))$ ssi
  $\forall \lambda' \in \mathsf{Exec}_{(\mathcal{C},F|_{0,A\setminus B})}(s), (\mathcal{M}_{f,(\mathcal{C},F|_{0,A\setminus B},\psi)}, \lambda') \models \sigma(\psi)$ ssi (par HI ap-

pliquée à ψ)
$\forall \lambda' \in \mathsf{Exec}_{(\mathcal{C},F|_{0,A\setminus B})}(s), (\mathcal{C},\lambda') \models_{F|_{0,A\setminus B}} \psi$ ssi, par le lemme 25,
$(\mathcal{C},s) \models_F \ \gg B \ll_0 \psi$ ssi
$(\mathcal{C},\lambda) \models_F \ \gg B \ll_0 \psi$.

- Si $\varphi_p = \mathbf{X}\,\psi$. Soient $A \subseteq \mathsf{Agt}, F \in \Sigma_A^0, s \in \mathsf{Loc}, \lambda \in \mathsf{Exec}_{(\mathcal{C},F)}(s)$. On a la suite d'équivalences :
  $(\mathcal{M}_{f,(\mathcal{C},F,\varphi_p)},\lambda) \models \sigma(\mathbf{X}\,\psi)$ ssi
  $(\mathcal{M}_{f,(\mathcal{C},F,\varphi_p)},\lambda) \models \mathbf{X}\,\sigma(\psi)$ ssi
  $(\mathcal{M}_{f,(\mathcal{C},F,\varphi_p)},\lambda[1,\infty]) \models \sigma(\psi)$ ssi (puisque l'on voit facilement $\mathcal{M}_{f,(\mathcal{C},F,\varphi_p)} = \mathcal{M}_{f,(\mathcal{C},F,\psi)}$)
  $(\mathcal{M}_{f,(\mathcal{C},F,\psi)},\lambda[1,\infty]) \models \sigma(\psi)$ ssi (par HI)
  $((\mathcal{C},F),\lambda[1,\infty]) \models \psi$ ssi
  $((\mathcal{C},F),\lambda) \models \mathbf{X}\,\psi$.
- Si $\varphi_p = \psi_1\,\mathbf{U}\,\psi_2$. Soient $A \subseteq \mathsf{Agt}, F \in \Sigma_A^0, s \in \mathsf{Loc}, \lambda \in \mathsf{Exec}_{(\mathcal{C},F)}(s)$. On a la suite d'équivalences :
  $(\mathcal{M}_{f,(\mathcal{C},F,\varphi_p)},\lambda) \models \sigma(\psi_1\,\mathbf{U}\,\psi_2)$ ssi
  $(\mathcal{M}_{f,(\mathcal{C},F,\varphi_p)},\lambda) \models \sigma(\psi_1)\,\mathbf{U}\,\sigma(\psi_2)$ ssi
  $\exists i \geq 0.\ (\mathcal{M}_{f,(\mathcal{C},F,\theta_p)},\lambda[i,\infty]) \models \sigma(\psi_2)$ et $\forall 0 \leq j < i.\ (\mathcal{M}_{f,(\mathcal{C},F,\theta_p)},\lambda[j,\infty]) \models \sigma(\psi_1)$ ssi (puisque l'on voit facilement $\mathcal{M}_{f,(\mathcal{C},F,\varphi_p)} = \mathcal{M}_{f,(\mathcal{C},F,\psi_1)} = \mathcal{M}_{f,(\mathcal{C},F,\psi_2)}$)
  $\exists i \geq 0.\ (\mathcal{M}_{f,(\mathcal{C},F,\psi_2)},\lambda[i,\infty]) \models \sigma(\psi_2)$ et $\forall 0 \leq j < i.\ (\mathcal{M}_{f,(\mathcal{C},F,\psi_1)},\lambda[j,\infty]) \models \sigma(\psi_1)$ ssi (par HI)
  $\exists i \geq 0.\ ((\mathcal{C},F),\lambda[i,\infty]) \models \psi_2$ et $\forall 0 \leq j < i.\ ((\mathcal{C},F),\lambda[j,\infty]) \models \psi_1$ ssi
  $((\mathcal{C},F),\lambda) \models \psi_1\,\mathbf{U}\,\psi_2$.

On sait que l'algorithme 1 retourne OUI sur l'entrée $(\mathcal{C},F,s_0,\theta_p)$ ssi :

$((\mathcal{M}_{f,(\mathcal{C},F,\theta_p)}),s_0) \models\ \mathbf{A}\sigma(\theta_p)$. La propriété que l'on vient d'établir, le lemme 25 et le fait que $(\mathcal{M}_{f,(\mathcal{C},F,\theta_p)})$ et $(\mathcal{C},F)$ aient les mêmes exécutions permettent de prouver que l'algorithme 1 retourne OUI sur l'entrée $(\mathcal{C},F,s_0,\theta_p)$ ssi $\forall \lambda \in \mathsf{Out}_{\mathcal{C}}(s_0,F), (\mathcal{C},\lambda) \models_F \theta_p$, comme on l'avait annoncé. Cet algorithme est bien un algorithme de model-checking pour $\mathcal{L}_0^{**}$: étant donnés une *CGS* $\mathcal{C}$, un état s_0 de $\mathcal{C}$ et une formule θ de $\mathcal{L}_0^{**}$, si l'on veut savoir si $(\mathcal{C},s_0) \models \theta$, il suffit de lancer l'algorithme 1 sur l'entrée $(\mathcal{C},\emptyset,s_0,\theta)$. En effet, il retournera OUI ssi $\forall \lambda \in \mathsf{Out}_{\mathcal{C}}(s_0,\emptyset), (\mathcal{C},\lambda) \models \theta$, c'est-à-dire ssi $(\mathcal{C},s_0) \models \theta$ car θ est une fornule d'état.

Il s'agit à présent de montrer que l'algorithme 1 est dans PSPACE. Soit une *CGS* $\mathcal{C}$. On va montrer par récurrence sur le rang de quantification que pour toute formule $\theta_p \in \mathcal{L}_{0,p}^{**}$, on a :

$\forall F \in \Sigma_A^0, \forall s_0 \in \mathsf{Loc}$, l'algorithme 1 utilise un espace polynomial en fonction de la taille de l'entrée $(\mathcal{C},F,s_0,\theta_p)$:

Si $rq(\theta_p) = 0$, alors l'algorithme 1 effectue seulement les affectations initiales qui se font facilement et l'opération MC LTL $((\mathcal{C},F),s_0,\ \mathbf{A}\theta_p)$, dont on sait qu'elle utilise un espace polynomial en fonction de son entrée.

Si $rq(\theta_p) = n+1$. La taille de $(\mathcal{C},F)$ est bornée par celle de $\mathcal{C}$. On ajoute un nombre de propositions atomiques borné par le nombre de sous-formules de θ_p à $(\mathcal{C},F)$ pour obtenir la structure $\mathcal{M}_{f,(\mathcal{C},F,\theta_p)}$. Donc le contenu de la variable $\mathcal{M}$ est toujours de longueur polynomiale en fonction de celle de l'entrée.

Les opérations d'assignations sont très faciles : restreindre une structure selon une stratégie, parcourir un arbre de formule et ajouter un élément à un ensemble donné, sont des actions pouvant aisément être effectuées dans PSPACE.

Les énumérations portent toutes sur des objets qui ont une taille polynomiales en fonction de celle de l'entrée : sous-formules de θ_p, stratégies sans mémoire d'une structure avec autant d'états que $\mathcal{C}$, et états de $\mathcal{C}$. On peut donc les énumérer sans problème dans PSPACE.

Enfin, les tests peuvent être effectués dans PSPACE : par HR, les test MC $\mathcal{L}_0^{**}(\mathcal{C}, F' \circ_0 F, s, \psi')$ et MC $\mathcal{L}_0^{**}(\mathcal{C}, F|_{0,A\setminus B}, s, \psi')$ prennent un espace polynomial en fonction de la taille de leur entrée, qui est plus petite que $(\mathcal{C}, F, s_0, \theta_p)$. Enfin, on sait que le test MC LTL $(\mathcal{M}, s_0, \mathbf{A}\sigma(\theta_p))$ prend un espace polynomial en fonction de la taille de son entrée, qui est elle-même polynomiale en fonction de celle de $(\mathcal{C}, F, s_0, \theta_p)$. □

Corollaire 1. *Les problèmes de model-checking de $\mathcal{L}_0$, $\mathcal{L}_0^*$ et $\mathcal{L}_0^{**}$ sont* PSPACE-*complets.*

Chapitre 3

Préservation de stratégies dans le cadre de logiques à mémoire infinie

Il s'agit à présent d'étendre les principes d'emboitement et d'oubli des stratégies au cas où celles-ci sont à mémoire infinie. Pour cela, nous allons simplement redéfinir successivement les mêmes objets que dans la section précédente, mais en considérant cette fois des stratégies avec mémoire infinie.

3.1 Emboîter des stratégies avec mémoire infinie

Voici l'opération d'emboitement de stratégies à mémoire infinie, notée $\circ_\infty$.

Définition 19. *Soient* $\mathcal{C}$ *une* CGS, $A, B \subseteq \mathsf{Agt}$, $F = (F_{A_i})_{A_i \in A} \in \Sigma_A^\infty$ *et* $F' = (F'_{A_i})_{A_i \in B} \in \Sigma_B^\infty$.

On définit :

$F' \circ_\infty F = (F''_{A_i})_{A_i \in A \cup B}$,

où $F''_{A_i} = \begin{cases} F'_{A_i} & \text{si } A_i \in B \\ F_{A_i} & \text{sinon.} \end{cases}$

On a donc $F' \circ_\infty F \in \Sigma_{A \cup B}^\infty$.

La loi $\circ_\infty$ *n'est pas commutative, mais elle est associative, et l'on pourra donc omettre les parenthèses. Son élément neutre est la stratégie* $\emptyset$.

3.1.1 Définition et expressivité de la logique $\mathcal{L}_\infty$

On définit une logique $\mathcal{L}_\infty$ où chaque opérateur temporel est immédiatement précédé d'un quantificateur de stratégie. De plus, dans cette logique, on ne s'autorise pas encore à oublier des stratégies, mais juste à les emboîter. L'unique différence avec la logique $\mathcal{L}_0$ définie précédemment est que les stratégies considérées sont à mémoire infinie.

Définition 20. *La syntaxe de $\mathcal{L}_\infty$ est définie par la grammaire suivante :*

$$\begin{aligned}
\mathcal{L}_\infty \ni \varphi_s, \psi_s \quad &::= \quad p \mid \neg\varphi_s \mid \varphi_s \vee \psi_s \mid \langle\!\langle A \rangle\!\rangle_\infty\ \varphi_p \\
\mathcal{L}_{\infty,p} \ni \varphi_p \quad &::= \quad \varphi_s \mid \mathbf{X}\,\varphi_s \mid \mathbf{G}\,\varphi_s \mid \varphi_s\,\mathbf{U}\,\psi_s.
\end{aligned}$$

où p décrit l'ensemble AP et A l'ensemble des parties de Agt.

La sémantique de $\mathcal{L}_\infty$ est définie comme suit :
Soient $A, B \subseteq \mathsf{Agt}$, $q \in \mathsf{Loc}$ et $F \in \Sigma_A^\infty$

$$\begin{array}{rcl}
q \models_F p & \text{ssi} & p \in \mathsf{Lab}(q), \\
q \models_F \neg\varphi_s & \text{ssi} & q \not\models_F \varphi_s, \\
q \models_F \varphi_s \vee \psi_s & \text{ssi} & q \models_F \varphi_s \text{ ou } q \models_F \psi_s, \\
q \models_F \langle\!\langle B \rangle\!\rangle_\infty\ \varphi_p & \text{ssi} & \exists F' \in \Sigma_B^\infty \ \text{ tel que } \forall \lambda \in \mathsf{Out}(q, F' \circ_\infty F), \lambda \models_{F' \circ_\infty F} \varphi_p, \\
\lambda \models_F \varphi_s & \text{ssi} & \lambda[0] \models_F \varphi_s, \\
\lambda \models_F \mathbf{X}\,\varphi_s & \text{ssi} & \lambda[1] \models_F \varphi_s, \\
\lambda \models_F \varphi_s\,\mathbf{U}\,\psi_s & \text{ssi} & \exists i \geq 0.\ \lambda[i] \models_F \psi_s \text{ et } \forall 0 \leq j < i.\ \lambda[j] \models_F \varphi_s, \\
\lambda \models_F \varphi_s\,\mathbf{W}\,\psi_s & \text{ssi} & \lambda \models_F \varphi_s\,\mathbf{U}\,\psi_s \text{ ou } \forall i \geq 0.\ \lambda[i] \models_F \psi_s.
\end{array}$$

Remarques :
- Les opérateurs $\langle\!\langle\emptyset\rangle\!\rangle_0$ et $\langle\!\langle\emptyset\rangle\!\rangle_\infty$ sont équivalents ; en revanche, si $A \neq \emptyset$, les opérateurs $\langle\!\langle A\rangle\!\rangle_0$ sont différents des opérateurs $\langle\!\langle A\rangle\!\rangle_\infty$.
- Si φ est une formule de $\mathcal{L}_\infty$, alors $q \models_F \neg \langle\!\langle B \rangle\!\rangle_\infty \neg\varphi$ signifie $\forall F' \in \Sigma_B^\infty, q \models_{F' \circ_\infty F} \varphi$.

On va montrer que $\neg(\mathcal{L}_\infty \leq_d \mathsf{ATL}^*)$, c'est-à-dire que la logique $\mathcal{L}_\infty$ permet de distinguer des structures qui sont indistingables par ATL^*.

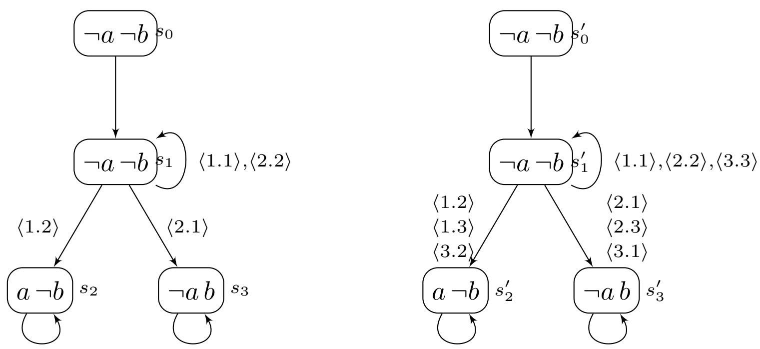

FIG. 3.1 – $\mathcal{C}_1$ et $\mathcal{C}_2$, deux *CGS* indistingables par ATL^*

Lemme 9. *On prouve d'abord que les deux* CGS *représentées sur la Fig. 5., avec pour états initiaux respectifs s_0 et s_0', sont indistingables par ATL^*.*

Preuve : On appelle A_1 et A_2 les deux joueurs de chacune des structures.

L'application σ qui à toute exécution (ou préfixe d'exécution) λ de $\mathcal{C}_1$ fait correspondre l'exécution (ou préfixe d'exécution) λ' de $\mathcal{C}_2$ obtenue en mettant des $'$ à chaque état constituant λ, est manifestement une bijection entre les ensembles des exécutions (ou préfixe d'exécution) des deux structures.

Soit $j \in \{1,2\}$. On définit les applications :

$$\nu_j : \begin{cases} \Sigma^{\infty}_{A_j,\mathcal{C}_1} & \rightarrow & \Sigma^{\infty}_{A_j,\mathcal{C}_2} \\ F & \mapsto & \nu_j(F) : \begin{cases} \bigcup_{j\in\mathbb{N}^*} \mathsf{Loc}^j_{\mathcal{C}_2} & \rightarrow & \mathbb{N} \\ \lambda & \mapsto & (\nu_j(F))(\lambda) = F(\sigma^{-1}(\lambda)) \end{cases} \end{cases}$$

$$\nu'_j : \begin{cases} \Sigma^{\infty}_{A_j,\mathcal{C}_2} & \rightarrow & \Sigma^{\infty}_{A_j,\mathcal{C}_1} \\ F & \mapsto & \nu'_j(F) : \begin{cases} \bigcup_{j\in\mathbb{N}^*} \mathsf{Loc}^j_{\mathcal{C}_1} & \rightarrow & \mathbb{N} \\ \lambda & \mapsto & (\nu'_j(F))(\lambda) = \begin{cases} 1 & \text{si } F(\sigma(\lambda)) = 3 \\ F(\sigma(\lambda)) & \text{sinon} \end{cases} \end{cases} \end{cases}$$

Intuitivement, ν_j transporte une stratégie pour le joueur A_j de $\mathcal{C}_1$ vers $\mathcal{C}_2$, et ν'_j transporte une stratégie pour le joueur A_j de $\mathcal{C}_2$ vers $\mathcal{C}_1$, en remplaçant le mouvement 3 par le mouvement 1.

Les deux résultats suivants sont faciles mais de démonstration laborieuse et sans intérêt : $\forall i \in \{0,1,2,3\}, \forall j \in \{1,2\}$:

- $\forall F \in \Sigma^{\infty}_{A_j,\mathcal{C}_1}, \mathsf{Out}(s'_i, \nu_j(F)) \subseteq \sigma(\mathsf{Out}(s_i, F))$,
- $\forall F \in \Sigma^{\infty}_{A_j,\mathcal{C}_2}, \mathsf{Out}(s_i, \nu'_j(F)) \subseteq \sigma^{-1}(\mathsf{Out}(s'_i, F))$.

On va montrer par induction sur les formules de ATL^*_p que :
$\forall \varphi \in \mathsf{ATL}^*_p, \forall i \in \{0,1,2,3\}, \forall \lambda \in \mathsf{Exec}(s_i), \lambda \models \varphi$ ssi $\sigma(\lambda) \models \varphi$.

- Si $\varphi = a \in \mathsf{AP}$. Soit $i \in \{0,1,2,3\}$. On doit montrer que $a \in \mathsf{Lab}(s_i)$ ssi $a \in \mathsf{Lab}(s'_i)$, ce qui est vrai puisque $\mathsf{Lab}(s_i) = \mathsf{Lab}(s'_i)$,
- Si $\varphi = \neg\psi$, alors c'est évident,
- Si $\varphi = \psi_1 \vee \psi_2$, alors c'est évident,
- Si $\varphi = \mathbf{X}\,\psi$. Soient $i \in \{0,1,2,3\}$ et $\lambda \in \mathsf{Exec}(s_i)$. Alors $\lambda \models \varphi$ signifie $\lambda[1,\infty] \models \psi$. En appliquant HI à la formule ψ, à l'état $\lambda[1]$ et à $\lambda[1,\infty] \in \mathsf{Exec}(\lambda[1])$, on obtient l'équivalence avec $\sigma(\lambda[1,\infty]) \models \psi$. Or $\sigma(\lambda[1,\infty]) = \sigma(\lambda)[1,\infty]$, donc c'est encore équivalent à $\sigma(\lambda) \models \mathbf{X}\,\psi$.
- Si $\varphi = \psi_1\,\mathbf{U}\,\psi_2$, alors un raisonnement similaire donne le résultat.
- Soit $j \in \{1,2\}$. Si $\varphi = \langle\!\langle A_j \rangle\!\rangle\,\psi$. Soient $i \in \{0,1,2,3\}$ et $\lambda \in \mathsf{Exec}(s_i)$.
  Supposons $\lambda \models \varphi$; soit $F \in \Sigma^{\infty}_{A_j,\mathcal{C}_1}$ telle que $\forall \lambda' \in \mathsf{Out}(s_i, F), \lambda' \models \psi$. Soit maintenant $\lambda'' \in \mathsf{Out}(s'_i, \nu_j(F))$. Par le résultat ci-dessus, $\lambda'' \in \sigma(\mathsf{Out}(s_i, F))$. On a donc $\sigma^{-1}(\lambda'') \models \psi$, d'où par HI $\lambda'' \models \psi$, ce qui prouve que $s'_i \models \langle\!\langle A_j \rangle\!\rangle\,\psi$, et finalement que $\sigma(\lambda) \models \langle\!\langle A_j \rangle\!\rangle\,\psi$.
  Supposons maintenant que $\sigma(\lambda) \models \varphi$; soit $F \in \Sigma^{\infty}_{A_j,\mathcal{C}_2}$ telle que $\forall \lambda' \in \mathsf{Out}(s'_i, F), \lambda' \models \psi$. Soit maintenant $\lambda'' \in \mathsf{Out}(s_i, \nu'_j(F))$. Par le résultat ci-dessus, $\lambda'' \in \sigma^{-1}(\mathsf{Out}(s'_i, F))$. On a donc $\sigma(\lambda'') \models \psi$, d'où par HI $\lambda'' \models \psi$, ce qui prouve que $s_i \models \langle\!\langle A_j \rangle\!\rangle\,\psi$, et finalement que $\lambda \models \langle\!\langle A_j \rangle\!\rangle\,\psi$.
- Si $\varphi = \langle\!\langle \emptyset \rangle\!\rangle\,\psi$. Soient $i \in \{0,1,2,3\}$. On doit montrer que $s_i \models \mathbf{E}\psi$ ssi $s'_i \models \mathbf{E}\psi$, ce qui est évident car σ réalise une bijection de $\mathsf{Exec}(s_i)$ dans $\mathsf{Exec}(s'_i)$, et par HI.
- Si $\varphi = \langle\!\langle \mathsf{Agt} \rangle\!\rangle\,\psi$. C'est bon pour les mêmes raisons.

Ceci appliqué dans le cas où $\varphi \in \mathsf{ATL}^*$, établit le résultat cherché. $\square$

Il s'agit maintenant d'exhiber une formule de $\mathcal{L}_\infty$ qui permette de distinguer ces deux structures.

Lemme 10. *Sur la Fig. 4., on a* $s'_0 \models \lessdot A_1 \gtrdot_\infty \mathbf{X}\,(\lessdot A_2 \gtrdot_\infty \mathbf{X}\,a \wedge \lessdot A_2 \gtrdot_\infty \mathbf{X}\,b)$ *mais* $s_0 \not\models \lessdot A_1 \gtrdot_\infty \mathbf{X}\,(\lessdot A_2 \gtrdot_\infty \mathbf{X}\,a \wedge \lessdot A_2 \gtrdot_\infty \mathbf{X}\,b)$

Preuve : Dans $\mathcal{C}_2$, une stratégie f_{A_1} telle que $f_{A_1}(s'_0 s'_1) = 3$ nous met dans un contexte qui vérifie que des stratégies f_{A_2} et f'_{A_2} telles que $f_{A_2}(s'_0 s'_1) = 2$ et $f'_{A_2}(s'_0 s'_1) = 1$ prouvent respectivement $\lessdot A_2 \gtrdot_\infty \mathbf{X}\,a$ et $\lessdot A_2 \gtrdot_\infty \mathbf{X}\,b$.

Dans $\mathcal{C}_1$, une stratégie f_{A_1} telle que $f_{A_1}(s'_0 s'_1) = 1$ nous met dans un contexte tel qu'il sera impossible à A_2 d'obtenir b au coup suivant. De même, une stratégie f_{A_1} telle que $f_{A_1}(s'_0 s'_1) = 2$ nous met dans un contexte tel qu'il sera impossible à A_2 d'obtenir a au coup suivant. $\square$

Voici une définition de GL (la *game logic*) :

Définition 21. *La syntaxe de* GL *est définie par la grammaire suivante :*

$$\begin{aligned}
\mathsf{GL} \ni \varphi_s, \psi_s \quad &::= \quad p \mid \neg\varphi_s \mid \varphi_s \vee \psi_s \mid \exists A \varphi_t \\
\mathsf{GL}_t \ni \varphi_t, \psi_t \quad &::= \quad \varphi_s \mid \neg\varphi_t \mid \varphi_t \vee \psi_t \mid \exists \varphi_p \\
\mathsf{GL}_p \ni \varphi_p, \psi_p \quad &::= \quad \varphi_t \mid \neg\varphi_p \mid \varphi_p \vee \psi_p \mid \mathbf{X}\,\varphi_p \mid \varphi_p \,\mathbf{U}\, \psi_p.
\end{aligned}$$

où p *décrit* AP *et* A *les sous-ensembles de* Agt.

La sémantique de GL est définie comme suit :

Soient $A \subseteq \mathsf{Agt}, q \in \mathsf{Loc}, t$ un arbre d'exécution et λ une exécution, $t[0]$ désignant la racine de l'arbre t et $t(x)$ le sous-arbre complet de t ayant x pour racine.

$$\begin{array}{rcl}
q \models p & \text{ssi} & p \in \mathsf{Lab}(q), \\
q \models \neg\varphi_s & \text{ssi} & q \not\models \varphi_s, \\
q \models \varphi_s \vee \psi_s & \text{ssi} & q \models \varphi_s \text{ ou } q \models \psi_s, \\
q \models \exists A \varphi_t & \text{ssi} & \exists F' \in \Sigma_A^\infty \text{ tel que } \mathsf{Out}(q, F') \models \varphi_t, \\
t \models \varphi_s & \text{ssi} & t[0] \models \varphi_s, \\
t \models \neg\varphi_t & \text{ssi} & t \not\models \varphi_t, \\
t \models \varphi_t \vee \psi_t & \text{ssi} & t \models \varphi_t \text{ ou } t \models \psi_t, \\
t \models \exists \varphi_p & \text{ssi} & \exists \lambda \in t \text{ tel que } (t, \lambda) \models \varphi_p, \\
(t, \lambda) \models \varphi_t & \text{ssi} & t \models \varphi_t, \\
(t, \lambda) \models \neg\varphi_p & \text{ssi} & (t, \lambda) \not\models \varphi_p, \\
(t, \lambda) \models \varphi_p \vee \psi_p & \text{ssi} & (t, \lambda) \models \varphi_p \text{ ou } (t, \lambda) \models \psi_p, \\
(t, \lambda) \models \mathbf{X}\,\varphi_p & \text{ssi} & (t(\lambda[1]), \lambda[1, \infty]) \models \varphi_p, \\
(t, \lambda) \models \varphi_p \,\mathbf{U}\, \psi_p & \text{ssi} & \exists i \geq 0.\ (t(\lambda[i]), \lambda[i, \infty]) \models \psi_p \text{ et } \forall 0 \leq j < i.\ (t(\lambda[j]), \lambda[j, \infty]) \models \varphi_p.
\end{array}$$

On va montrer que $\neg(\mathcal{L}_\infty \leq_d \mathsf{GL})$, c'est-à-dire que la logique $\mathcal{L}_\infty$ permet de distinguer des structures qui sont indistingables par GL.

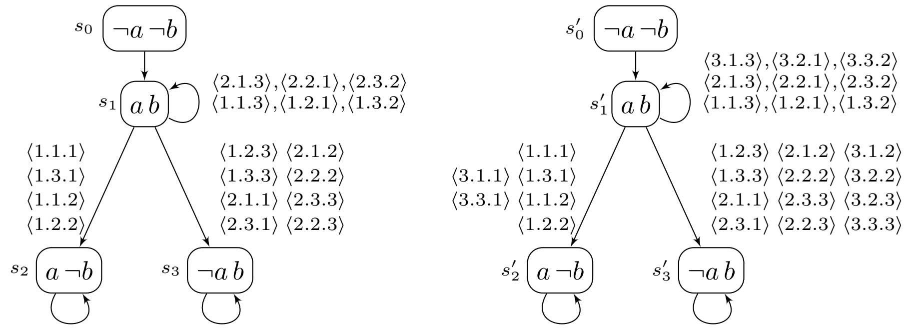

FIG. 3.2 – $\mathcal{C}_1$ et $\mathcal{C}_2$, deux *CGS* indistingables par GL

Lemme 11. *On prouve d'abord que les deux* CGS *représentées sur la Fig. 3.2, avec pour états initiaux respectifs s_0 et s'_0, sont indistingables par* GL.

Il s'agit maintenant d'exhiber une formule de $\mathcal{L}_\infty$ qui permette de distinguer ces deux structures.

Lemme 12. *Sur la Fig. 3.2, on a* $s'_0 \models \lessdot A_1 \gtrdot_\infty \mathbf{X}\,(\lessdot A_2 \gtrdot_\infty \mathbf{X}\,b \wedge \lessdot A_3 \gtrdot_\infty \mathbf{X}\,a)$ *mais* $s_0 \not\models \lessdot A_1 \gtrdot_\infty \mathbf{X}\,(\lessdot A_2 \gtrdot_\infty \mathbf{X}\,b \wedge \lessdot A_3 \gtrdot_\infty \mathbf{X}\,a)$

Lemme 13. $\mathit{CTL}^* \leq_e \mathcal{L}_\infty$*, avec une traduction en temps linéaire.*

Preuve : On définit par induction l'application $\sigma\colon \mathsf{CTL}^*_p \to \mathcal{L}_\infty$ telle que :
- Si $\varphi = a \in \mathsf{AP}$, alors $\sigma(\varphi) = a$
- Si $\varphi = \neg\psi$, alors $\sigma(\varphi) = \neg\sigma(\psi)$
- Si $\varphi = \psi_1 \vee \psi_2$, alors $\sigma(\varphi) = \sigma(\psi_1) \vee \sigma(\psi_2)$
- Si $\varphi = \mathbf{X}\,\psi$, alors $\sigma(\varphi) = \lessdot\emptyset\gtrdot_\infty \mathbf{X}\,\sigma(\varphi)$
- Si $\varphi = \psi_1\,\mathbf{U}\,\psi_2$, alors $\sigma(\varphi) = \lessdot\emptyset\gtrdot_\infty \sigma(\psi_1)\,\mathbf{U}\,\sigma(\psi_2)$
- Si $\varphi = \mathbf{E}\psi$, alors $\sigma(\varphi) = \lessdot\mathsf{Agt}\gtrdot_\infty \sigma(\psi)$
- Si $\varphi = \mathbf{A}\psi$, alors $\sigma(\varphi) = \neg \lessdot \mathsf{Agt} \gtrdot_\infty \neg\sigma(\psi)$

Soit $\mathcal{C}$ une structure de Kripke. On montre par induction sur les formules de CTL^*_p que :

$\forall\varphi \in \mathsf{CTL}^*_p, \forall F \in \Sigma^\infty_{\mathsf{Agt}}, \forall s \in \mathsf{Loc}$, si l'on note $\lambda_{s,F}$ l'unique élément de $\mathsf{Out}(s,F)$, alors $\lambda_{s,F} \models \varphi$ ssi $s \models_F \sigma(\varphi)$.
- Si $\varphi = a \in \mathsf{AP}$, alors les deux formules sont vraies ssi $a \in \mathsf{Lab}(s)$,
- Si $\varphi = \neg\psi$, alors c'est évident,
- Si $\varphi = \psi_1 \vee \psi_2$, alors c'est évident,
- Si $\varphi = \mathbf{X}\,\psi$. Soient $F \in \Sigma^\infty_{\mathsf{Agt}}$ et $s \in \mathsf{Loc}$. Alors $s \models_F \sigma(\varphi)$ est équivalent à $\lambda_{s,F}[1] \models_F \sigma(\psi)$. On applique HI à la même stratégie F et à l'état $\lambda_{s,F}[1]$; on a alors $\lambda_{\lambda_{s,F}[1],F} = \lambda_{s,F}[1,\infty]$. Donc $\lambda_{s,F}[1] \models_F \sigma(\psi)$ est équivalent à $\lambda_{s,F}[1,\infty] \models \psi$, et donc à $\lambda_{s,F} \models \mathbf{X}\,\psi$.
- Si $\varphi = \psi_1\,\mathbf{U}\,\psi_2$, alors un raisonnement similaire donne le résultat.
- Si $\varphi = \mathbf{E}\psi$. Soient $F \in \Sigma^\infty_{\mathsf{Agt}}$ et $s \in \mathsf{Loc}$. Alors $s \models_F \sigma(\varphi)$ signifie $\exists F' \in \Sigma^\infty_{\mathsf{Agt}}$ telle que $\forall\lambda \in \mathsf{Out}(s, F' \circ_\infty F), \lambda \models_{F'\circ_\infty F} \sigma(\psi)$. Mais $F' \circ_\infty F =$

F', donc c'est équivalent à $\exists F' \in \Sigma^{\infty}_{\mathsf{Agt}}$ telle que $\forall \lambda \in \mathsf{Out}(s, F'), \lambda \models_{F'} \sigma(\psi)$, ou encore à $\exists F' \in \Sigma^{\infty}_{\mathsf{Agt}}$ telle que $s \models_{F'} \sigma(\psi)$, puisque $\sigma(\psi) \in \mathcal{L}_{\infty}$. En appliquant HI à chaque stratégie F' et au même état s, on obtient l'équivalence avec $\exists F' \in \Sigma^{\infty}_{\mathsf{Agt}}$ telle que $\lambda_{s,F'} \models \psi$. Or dans tout *CGS* $\mathcal{C}$, si l'on fixe un état s, l'application entre $\Sigma^{\infty}_{\mathsf{Agt}}$ et $\mathsf{Exec}(s)$ qui à F' associe $\lambda_{s,F'}$ est bijective ; l'existence d'une stratégie est donc équivalente à l'existence d'une s-exécution, ce qui implique que tout ceci est enfin équivalent à $s \models \mathbf{E}\psi$, où bien à $\lambda_{s,F} \models \mathbf{E}\psi$.

- Si $\varphi = \mathbf{A}\psi$. Soient $F \in \Sigma^{\infty}_{\mathsf{Agt}}$ et $s \in \mathsf{Loc}$. Alors $s \models_F \sigma(\varphi)$ signifie $\forall F' \in \Sigma^{\infty}_{\mathsf{Agt}}, s \models_{F' \circ_{\infty} F} \sigma(\psi)$. Mais $F' \circ_{\infty} F = F'$, donc c'est équivalent à $\forall F' \in \Sigma^{\infty}_{\mathsf{Agt}}, s \models_{F'} \sigma(\psi)$. En appliquant HI à chaque stratégie F' et au même état s, on obtient l'équivalence avec $\forall F' \in \Sigma^{\infty}_{\mathsf{Agt}}, \lambda_{s,F'} \models \psi$. Toujours en se servant de la bijection, c'est encore équivalent à $s \models \mathbf{A}\psi$, ou bien à $\lambda_{s,F} \models \mathbf{A}\psi$.

On va maintenant montrer par induction sur les formules de CTL^* que :
$\forall \varphi \in \mathsf{CTL}^*, \forall s \in \mathsf{Loc}$, $s \models \varphi$ ssi $s \models \sigma(\varphi)$.

- Si $\varphi = a \in \mathsf{AP}$, alors les deux formules sont vraies ssi $a \in \mathsf{Lab}(s)$,
- Si $\varphi = \neg\psi$, alors c'est évident,
- Si $\varphi = \psi_1 \vee \psi_2$, alors c'est évident,
- Si $\varphi = \mathbf{E}\psi$, où $\psi \in \mathsf{CTL}^*_p$. Soit $s \in \mathsf{Loc}$. Alors $s \models \sigma(\varphi)$ signifie $\exists F \in \Sigma^{\infty}_{\mathsf{Agt}}$ telle que $\forall \lambda \in \mathsf{Out}(s, F), \lambda \models_F \sigma(\psi)$. Mais $\sigma(\psi) \in \mathcal{L}_{\infty}$, donc c'est équivalent à $\exists F \in \Sigma^{\infty}_{\mathsf{Agt}}$ telle que $s \models_F \sigma(\psi)$. En appliquant le résultat ci-dessus à la formule $\psi \in \mathsf{CTL}^*_p$, à l'état s et à chaque stratégie F, on obtient l'équivalence avec $\exists F \in \Sigma^{\infty}_{\mathsf{Agt}}$ telle que $\lambda_{s,F} \models \psi$. Or dans tout *CGS* $\mathcal{C}$, si l'on fixe un état s, l'application entre $\Sigma^{\infty}_{\mathsf{Agt}}$ et $\mathsf{Exec}(s)$ qui à F associe $\lambda_{s,F}$ est bijective ; l'existence d'une stratégie est donc équivalente à l'existence d'une s-exécution, ce qui implique que tout ceci est enfin équivalent à $s \models \mathbf{E}\psi$.
- Si $\varphi = \mathbf{A}\psi$, où $\psi \in \mathsf{CTL}^*_p$. Soit $s \in \mathsf{Loc}$. Alors $s \models \sigma(\varphi)$ signifie $\forall F \in \Sigma^{\infty}_{\mathsf{Agt}}, s \models_F \sigma(\psi)$. En appliquant le résultat ci-dessus à la formule $\psi \in \mathsf{CTL}^*_p$, à l'état s et à chaque stratégie F, on obtient l'équivalence avec $\forall F \in \Sigma^{\infty}_{\mathsf{Agt}}, \lambda_{s,F} \models \psi$. Toujours en se servant de la bijection, c'est encore équivalent à $s \models \mathbf{A}\psi$.

On a prouvé que les formules φ et $\sigma(\varphi)$ sont équivalentes sur les *CGS*. On voit bien que la traduction se fait en temps linéaire. □

3.1.2 Définition et expressivité de la logique $\mathcal{L}^*_{\infty}$

En s'inspirant de ATL^*, on définit à présent l'extension naturelle $\mathcal{L}^*_{\infty}$.

Définition 22. *La syntaxe de $\mathcal{L}^*_{\infty}$ est définie par la grammaire suivante :*

$$\begin{aligned} \mathcal{L}^*_{\infty} \ni \varphi_s, \psi_s \quad &::= \quad p \mid \neg\varphi_s \mid \varphi_s \vee \psi_s \mid \langle\!\langle A \rangle\!\rangle_{\infty}\ \varphi_p \\ \mathcal{L}^*_{\infty,p} \ni \varphi_p, \psi_p \quad &::= \quad \varphi_s \mid \neg\varphi_p \mid \varphi_p \vee \psi_p \mid \mathbf{X}\,\varphi_p \mid \varphi_p\,\mathbf{U}\,\psi_p. \end{aligned}$$

où p décrit AP et A les sous-ensembles de Agt.

La sémantique de $\mathcal{L}^*_\infty$ est définie comme suit :
Soient $A, B \subseteq \mathsf{Agt}, q \in \mathsf{Loc}$ et $F \in \Sigma^\infty_A$,

$$\begin{array}{rcl}
q \models_F p & \text{ssi} & p \in \mathsf{Lab}(q), \\
q \models_F \neg\varphi_s & \text{ssi} & q \not\models_F \varphi_s, \\
q \models_F \varphi_s \vee \psi_s & \text{ssi} & q \models_F \varphi_s \text{ ou } q \models_F \psi_s, \\
q \models_F \lessdot B \gtrdot_\infty \varphi_p & \text{ssi} & \exists F' \in \Sigma^\infty_B \text{ tel que } \forall\lambda \in \mathsf{Out}(q, F' \circ_\infty F), \lambda \models_{F'\circ_\infty F} \varphi_p, \\
\lambda \models_F \varphi_s & \text{ssi} & \lambda[0] \models_F \varphi_s, \\
\lambda \models_F \neg\varphi_p & \text{ssi} & \lambda \not\models_F \varphi_p, \\
\lambda \models_F \varphi_p \vee \psi_p & \text{ssi} & \lambda \models_F \varphi_p \text{ ou } \lambda \models_F \psi_p, \\
\lambda \models_F \mathbf{X}\,\varphi_p & \text{ssi} & \lambda[1,\infty] \models_F \varphi_p, \\
\lambda \models_F \varphi_p\,\mathbf{U}\,\psi_p & \text{ssi} & \exists i \geq 0.\ \lambda[i,\infty] \models_F \psi_p \text{ et } \forall 0 \leq j < i.\ \lambda[j,\infty] \models_F \varphi_p.
\end{array}$$

Lemme 14. *Sur les structures de Kripke, $\mathcal{L}^*_\infty \leq_e \mathsf{CTL}^*$*

Preuve : On définit par induction les applications $\sigma_1 \colon \mathcal{L}^*_{\infty,p} \to \mathsf{CTL}^*$ et $\sigma_2 \colon \mathcal{L}^*_{\infty,p} \to \mathsf{CTL}^*_p$ telles que :

- Si $\varphi = a \in \mathsf{AP}$, alors $\sigma_1(\varphi) = \sigma_2(\varphi) = a$
- Si $\varphi = \neg\psi$, alors $\sigma_1(\varphi) = \neg\sigma_1(\psi)$ et $\sigma_2(\varphi) = \neg\sigma_2(\psi)$
- Si $\varphi = \psi_1 \vee \psi_2$, alors $\sigma_1(\varphi) = \sigma_1(\psi_1) \vee \sigma_1(\psi_2)$ et $\sigma_2(\varphi) = \sigma_2(\psi_1) \vee \sigma_2(\psi_2)$
- Si $\varphi = \mathbf{X}\,\psi$, alors $\sigma_1(\varphi) = \mathbf{X}\,\sigma_1(\psi)$ et $\sigma_2(\varphi) = \mathbf{X}\,\sigma_2(\psi)$
- Si $\varphi = \psi_1\,\mathbf{U}\,\psi_2$, alors $\sigma_1(\varphi) = \sigma_1(\psi_1)\,\mathbf{U}\,\sigma_1(\psi_2)$ et $\sigma_2(\varphi) = \sigma_2(\psi_1)\,\mathbf{U}\,\sigma_2(\psi_2)$
- Si $\varphi = \lessdot \mathsf{Agt} \gtrdot_\infty \psi$, alors $\sigma_1(\varphi) = \mathbf{E}\sigma_2(\psi)$ et $\sigma_2(\varphi) = \mathbf{E}\sigma_2(\psi)$
- Si $\varphi = \lessdot \emptyset \gtrdot_\infty \psi$, alors $\sigma_1(\varphi) = \mathbf{A}\sigma_1(\psi)$ et $\sigma_2(\varphi) = \sigma_2(\psi)$

Soit $\mathcal{C}$ une structure de Kripke. On montre par induction sur les formules de $\mathcal{L}^*_{\infty,p}$ que :

$\forall\varphi \in \mathcal{L}^*_{\infty,p}, \forall F \in \Sigma^\infty_{\mathsf{Agt}}, \forall s \in \mathsf{Loc}$, si l'on note $\lambda_{s,F}$ l'unique élément de $\mathsf{Out}(s, F)$, alors $\lambda_{s,F} \models \sigma_2(\varphi)$ ssi $\lambda_{s,F} \models_F \varphi$.

- Si $\varphi = a \in \mathsf{AP}$, alors les deux formules sont vraies ssi $a \in \mathsf{Lab}(s)$,
- Si $\varphi = \neg\psi$, alors c'est évident,
- Si $\varphi = \psi_1 \vee \psi_2$, alors c'est évident,
- Si $\varphi = \mathbf{X}\,\psi$. Soient $F \in \Sigma^\infty_{\mathsf{Agt}}$ et $s \in \mathsf{Loc}$. Alors $\lambda_{s,F} \models \sigma_2(\varphi)$ est équivalent à $\lambda_{s,F}[1,\infty] \models \sigma_2(\psi)$. On applique HI à la même stratégie F et à l'état $\lambda_{s,F}[1]$; on a alors $\lambda_{\lambda_{s,F}[1],F} = \lambda_{s,F}[1,\infty]$. Donc $\lambda_{s,F}[1,\infty] \models \sigma_2(\psi)$ est équivalent à $\lambda_{s,F}[1,\infty] \models_F \psi$, et donc à $\lambda_{s,F} \models_F \mathbf{X}\,\psi$.
- Si $\varphi = \psi_1\,\mathbf{U}\,\psi_2$, alors un raisonnement similaire donne le résultat.
- Si $\varphi = \lessdot \mathsf{Agt} \gtrdot_\infty \psi$. Soient $F \in \Sigma^\infty_{\mathsf{Agt}}$ et $s \in \mathsf{Loc}$. Alors $\lambda_{s,F} \models \sigma_2(\varphi)$ signifie $\exists\lambda \in \mathsf{Exec}(s)$ telle que $\lambda \models \sigma_2(\psi)$, ou bien $\exists F' \in \Sigma^\infty_{\mathsf{Agt}}$ telle que $\lambda_{s,F'} \models \sigma_2(\psi)$ par notre bijection habituelle. On applique HI et l'on obtient que c'est équivalent à $\exists F' \in \Sigma^\infty_{\mathsf{Agt}}$ telle que $\lambda_{s,F'} \models_{F'} \psi$. Mais $F' = F' \circ_\infty F$, donc c'est équivalent à $\exists F' \in \Sigma^\infty_{\mathsf{Agt}}$ telle que $\lambda_{s,F'\circ_\infty F} \models_{F'\circ_\infty F} \psi$, ou encore à $\exists F' \in \Sigma^\infty_{\mathsf{Agt}}$ telle que $\forall\lambda \in \mathsf{Out}(s, F' \circ_\infty F), \lambda \models_{F'\circ_\infty F} \psi$, c'est-à-dire à $s \models_F \lessdot \mathsf{Agt} \gtrdot_\infty \psi$, et enfin à $\lambda_{s,F} \models_F \lessdot \mathsf{Agt} \gtrdot_\infty \psi$.

– Si $\varphi = \lessdot\emptyset\gtrdot_\infty \psi$. Soient $F \in \Sigma^\infty_{\mathsf{Agt}}$ et $s \in \mathsf{Loc}$. Alors $\lambda_{s,F} \models \sigma_2(\varphi)$ signifie $\lambda_{s,F} \models \sigma_2(\psi)$, soit par HI $\lambda_{s,F} \models_F \psi$. On voit directement que ceci est équivalent à $\lambda_{s,F} \models_F \lessdot\emptyset\gtrdot_\infty \psi$.

On va maintenant montrer par induction sur les formules de $\mathcal{L}^*_{\infty,p}$ que : $\forall\varphi \in \mathcal{L}^*_{\infty,p}, \forall s \in \mathsf{Loc}, \forall\lambda \in \mathsf{Exec}(s), \lambda \models \varphi$ ssi $\lambda \models \sigma_1(\varphi)$.

– Si $\varphi = a \in \mathsf{AP}$, alors les deux formules sont vraies ssi $a \in \mathsf{Lab}(s)$,
– Si $\varphi = \neg\psi$, alors c'est évident,
– Si $\varphi = \psi_1 \vee \psi_2$, alors c'est évident,
– Si $\varphi = \mathbf{X}\,\psi$. Soient $s \in \mathsf{Loc}$ et $\lambda \in \mathsf{Exec}(s)$. Alors $\lambda \models \sigma_1(\varphi)$ est équivalent à $\lambda[1,\infty] \models \sigma_1(\psi)$. On applique HI à l'état $\lambda[1]$ et à l'exécution $\lambda[1,\infty]$; on a alors $\lambda[1,\infty] \models \sigma_1(\psi)$ est équivalent à $\lambda[1,\infty] \models \psi$, et donc à $\lambda \models \mathbf{X}\,\psi$,
– Si $\varphi = \psi_1\,\mathbf{U}\,\psi_2$, alors un raisonnement similaire donne le résultat,
– Si $\varphi = \lessdot\mathsf{Agt}\gtrdot_\infty \psi$, où $\psi \in \mathcal{L}^*_p$. Soit $s \in \mathsf{Loc}$ et $\lambda \in \mathsf{Exec}(s)$. Alors $\lambda \models \sigma_1(\varphi)$ signifie $s \models \sigma_1(\varphi)$ ou $\exists\lambda' \in \mathsf{Exec}(s)$ telle que $\lambda' \models \sigma_2(\psi)$, ou bien $\exists F \in \Sigma_{\mathsf{Agt}}$ telle que $\lambda_{s,F} \models \sigma_2(\psi)$ par notre bijection habituelle. On applique le résultat ci-dessus et l'on obtient que c'est équivalent à $\exists F \in \Sigma^\infty_{\mathsf{Agt}}$ telle que $\lambda_{s,F} \models_F \psi$, ou encore à $\exists F \in \Sigma^\infty_{\mathsf{Agt}}$ telle que $\forall\lambda' \in \mathsf{Out}(s,F), \lambda' \models_F \psi$, c'est-à-dire à $s \models \lessdot\mathsf{Agt}\gtrdot_\infty \psi$ puisque $F \circ_\infty \emptyset = F$.
– Si $\varphi = \lessdot\emptyset\gtrdot_\infty \psi$, où $\psi \in \mathcal{L}^*_p$. Soit $s \in \mathsf{Loc}$ et $\lambda \in \mathsf{Exec}(s)$. Alors $\lambda \models \sigma_1(\varphi)$ signifie $s \models \sigma_1(\varphi)$ ou $\forall\lambda' \in \mathsf{Exec}(s), \lambda' \models \sigma_1(\psi)$. Par HI, c'est équivalent à $\forall\lambda' \in \mathsf{Exec}(s), \lambda' \models \psi$, soit à $s \models \mathbf{A}\psi$, ou enfin $\lambda \models \mathbf{A}\psi$.

En appliquant ceci dans le cas où $\varphi \in \mathcal{L}^*_\infty$, on a prouvé que les formules φ et $\sigma_1(\varphi)$ sont équivalentes sur les structures de Kripke. On voit bien que la traduction se fait en temps linéaire. □

3.2 Emboîter et oublier des stratégies avec mémoire infinie

Nous allons à présent définir une opération d'oubli de stratégie à mémoire infinie, qui correspond en fait à une restriction, et que l'on note $|_\infty$.

Définition 23. *Soient $\mathcal{C}$ une* CGS, A, C $\subseteq$ *Agt tels que $C \subseteq A$ et $F = (F_{A_i})_{A_i \in A} \in \Sigma^\infty_A$.*

On définit $F|_{\infty,C} = (F_{A_i})_{A_i \in C}$. On a donc $F|_{\infty,C} \in \Sigma^\infty_C$. En particulier, $F|_{\infty,\emptyset} = \emptyset$

3.2.1 Définition de la logique $\mathcal{L}^{**}_\infty$

On étend à nouveau $\mathcal{L}^*_\infty$ en $\mathcal{L}^{**}_\infty$, en s'autorisant maintenant à oublier des stratégies, cela par l'opération que l'on vient de définir.

Définition 24. *La syntaxe de $\mathcal{L}^{**}_\infty$ est définie par la grammaire suivante :*

$$\begin{array}{rcl} \mathcal{L}^{**}_\infty \ni \varphi_s, \psi_s & ::= & p \mid \neg\varphi_s \mid \varphi_s \vee \psi_s \mid \lessdot A\gtrdot_\infty \varphi_p \mid \gtrdot A\lessdot_\infty \varphi_p \\ \mathcal{L}^{**}_{\infty,p} \ni \varphi_p, \psi_p & ::= & \varphi_s \mid \neg\varphi_p \mid \varphi_p \vee \psi_p \mid \mathbf{X}\,\varphi_p \mid \varphi_p\,\mathbf{U}\,\psi_p. \end{array}$$

où p décrit AP et A les sous-ensembles de Agt.

La sémantique de $\mathcal{L}_\infty^{**}$ est définie comme suit :
Soient $A, B \subseteq \mathsf{Agt}, q \in \mathsf{Loc}$ et $F \in \Sigma_A^\infty$,

$$\begin{array}{rll}
q \models_F p & \text{ssi} & p \in \mathsf{Lab}(q), \\
q \models_F \neg\varphi_s & \text{ssi} & q \not\models_F \varphi_s, \\
q \models_F \varphi_s \vee \psi_s & \text{ssi} & q \models_F \varphi_s \text{ ou } q \models_F \psi_s, \\
q \models_F \lessdot B \gtrdot_\infty \varphi_p & \text{ssi} & \exists F' \in \Sigma_B^\infty \text{ tel que } \forall \lambda \in \mathsf{Out}(q, F' \circ_\infty F), \lambda \models_{F' \circ_\infty F} \varphi_p, \\
q \models_F \gtrdot B \lessdot_\infty \varphi_p & \text{ssi} & \forall \lambda \in \mathsf{Out}(q, F|_{\infty, A\setminus B}), \lambda \models_{F|_{\infty,A\setminus B}} \varphi_p, \\
\lambda \models_F \varphi_s & \text{ssi} & \lambda[0] \models_F \varphi_s, \\
\lambda \models_F \neg\varphi_p & \text{ssi} & \lambda \not\models_F \varphi_p, \\
\lambda \models_F \varphi_p \vee \psi_p & \text{ssi} & \lambda \models_F \varphi_p \text{ ou } \lambda \models_F \psi_p, \\
\lambda \models_F \mathbf{X}\, \varphi_p & \text{ssi} & \lambda[1, \infty] \models_F \varphi_p, \\
\lambda \models_F \varphi_p \,\mathbf{U}\, \psi_p & \text{ssi} & \exists i \geq 0.\ \lambda[i, \infty] \models_F \psi_p \text{ et } \forall 0 \leq j < i.\ \lambda[j, \infty] \models_F \varphi_p.
\end{array}$$

Lemme 15. $\mathsf{GL} \leq_e \mathcal{L}_\infty^{**}$*, avec une traduction en temps linéaire.*

Preuve : On définit par induction l'application $\sigma\colon \mathsf{GL}_p \to \mathcal{L}_{\infty,p}^{**}$ telle que :

- Si $\varphi = a \in \mathsf{AP}$, alors $\sigma(\varphi) = a$
- Si $\varphi = \neg\psi$, alors $\sigma(\varphi) = \neg\sigma(\psi)$
- Si $\varphi = \psi_1 \vee \psi_2$, alors $\sigma(\varphi) = \sigma(\psi_1) \vee \sigma(\psi_2)$
- Si $\varphi = \mathbf{X}\,\psi$, alors $\sigma(\varphi) = \mathbf{X}\,\sigma(\varphi)$
- Si $\varphi = \psi_1 \,\mathbf{U}\, \psi_2$, alors $\sigma(\varphi) = \sigma(\psi_1) \,\mathbf{U}\, \sigma(\psi_2)$
- Si $\varphi = \exists\psi$, alors $\sigma(\varphi) = \neg \lessdot \emptyset \gtrdot_\infty \neg\sigma(\psi)$
- Si $\varphi = \exists A\psi$, alors $\sigma(\varphi) = \gtrdot \mathsf{Agt} \lessdot_\infty \lessdot A \gtrdot_\infty \sigma(\psi)$

Soit $\mathcal{C}$ une structure de Kripke. On montre par induction sur les formules de GL_p que :

$\forall \varphi \in \mathsf{GL}_p, \forall A \subseteq \mathsf{Agt}, \forall F \in \Sigma_A^\infty, \forall q \in \mathsf{Loc}, \forall \lambda \in \mathsf{Out}(q, F)$,
$(\mathsf{Out}(q, F), \lambda) \models \varphi$ ssi $\lambda \models_F \sigma(\varphi)$.

- Si $\varphi = a \in \mathsf{AP}$, alors les deux formules sont vraies ssi $a \in \mathsf{Lab}(q)$,
- Si $\varphi = \neg\psi$, alors c'est évident,
- Si $\varphi = \psi_1 \vee \psi_2$, alors c'est évident,
- Si $\varphi = \mathbf{X}\,\psi$. Soient $A \subseteq \mathsf{Agt}, F \in \Sigma_A^\infty, q \in \mathsf{Loc}$ et $\lambda \in \mathsf{Out}(q, F)$. Alors $\lambda \models_F \sigma(\varphi)$ est équivalent à $\lambda[1, \infty] \models_F \sigma(\psi)$. On applique HI à la même stratégie F et à l'exécution $\lambda[1, \infty] \in \mathsf{Out}(\lambda[1], F)$; on a alors $\mathsf{Out}(\lambda[1], F) = \mathsf{Out}(q, F)(\lambda[1])$. Donc $\lambda[1, \infty] \models_F \sigma(\psi)$ est équivalent à $(\mathsf{Out}(q, F)(\lambda[1]), \lambda[1, \infty]) \models \psi$, et donc à $(\mathsf{Out}(q, f), \lambda) \models \mathbf{X}\,\psi$.
- Si $\varphi = \psi_1 \,\mathbf{U}\, \psi_2$, alors un raisonnement similaire donne le résultat.
- Si $\varphi = \exists\psi$. Soient $A \subseteq \mathsf{Agt}, F \in \Sigma_A^\infty, q \in \mathsf{Loc}$ et $\lambda \in \mathsf{Out}(q, F)$. Alors on a la suite d'équivalences :
  $\lambda \models_F \sigma(\varphi)$ ssi
  $\lambda \models_F \neg \lessdot \emptyset \gtrdot_\infty \neg\sigma(\psi)$ ssi
  $q \not\models_F \lessdot\emptyset \gtrdot_\infty \neg\sigma(\psi)$ ssi

$\neg(\forall \lambda' \in \mathsf{Out}(q,F), \lambda' \not\models_F \sigma(\psi))$ ssi
$\exists \lambda' \in \mathsf{Out}(q,F), \lambda' \models_F \sigma(\psi)$ ssi, par HI appliquée à chaque λ',
$\exists \lambda' \in \mathsf{Out}(q,F), (\mathsf{Out}(q,F), \lambda') \models \psi$ ssi
$\mathsf{Out}(q,F) \models \exists \psi$ ssi
$(\mathsf{Out}(q,F), \lambda) \models \exists \psi$.

– Si $\varphi = \exists B \psi$. Soient $A \subseteq \mathsf{Agt}, F \in \Sigma_A^\infty, q \in \mathsf{Loc}$ et $\lambda \in \mathsf{Out}(q,F)$. Alors on a la suite d'équivalences :
$\lambda \models_F \sigma(\varphi)$ ssi
$\lambda \models_F \gg\!\mathsf{Agt}\!\ll_\infty \; \ll\! B \!\gg_\infty \sigma(\psi)$ ssi
$q \models_F \gg\!\mathsf{Agt}\!\ll_\infty \; \ll\! B \!\gg_\infty \sigma(\psi)$ ssi
$q \models \; \ll\! B \!\gg_\infty \sigma(\psi)$ ssi
$\exists F' \in \Sigma_B^\infty, \forall \lambda' \in \mathsf{Out}(q,F'), \lambda' \models_{F'} \sigma(\psi)$ ssi, par HI appliquée à chaque λ',
$\exists F' \in \Sigma_B^\infty, \forall \lambda' \in \mathsf{Out}(q,F'), (\mathsf{Out}(q,F'), \lambda') \models \psi$ ssi, puisque ψ est une formule d'arbre,
$\exists F' \in \Sigma_B^\infty, \mathsf{Out}(q,F') \models \psi$ ssi
$q \models \exists B \psi$ ssi, puisque $\exists B \psi$ est une formule d'état,
$(\mathsf{Out}(q,F), \lambda) \models \exists B \psi$.

En appliquant ce résultat à la stratégie $\emptyset$ pour la coalition $\emptyset$, dans le cas où φ est une formule d'état, on a prouvé que les formules φ et $\sigma(\varphi)$ sont équivalentes sur les *CGS*. On voit bien que la traduction se fait en temps linéaire. $\square$

Chapitre 4

Préservation de stratégies dans le cadre de logiques à mémoire bornée

Dans cette quatrième section, on introduit deux définitions de stratégies avec mémoire finie, dont la seconde, moins naturelle, semble plus prometteuse. Partant de ceci, nous allons définir des nouvelles logiques sur le modèle des précédentes, en sauvegardant toujours les principes d'emboitement et d'oubli de stratégies.

4.1 Première définition de la taille de la mémoire d'une stratégie ; les logiques $\mathcal{L}_n^{**}$; limites de cette définition

Une première idée, plus intuitive, consiste à définir les objets suivants :

Définition 25. *Soient* $\mathcal{C}$ *une* CGS, $A \subseteq \mathsf{Agt}_{\mathcal{C}}$ *et* $n \in \mathbb{N} \cup \{\infty\}$*. On note* $\Sigma^n_{A,\mathcal{C}} = \{F \in (\mathbb{N}^{|A|})^{\bigcup_{j=1,\ldots,n+1} \mathsf{Loc}^j_{\mathcal{C}}} \mid \forall j \in \{1, ..., n+1\}, \forall \lambda \in \mathsf{Loc}^j_{\mathcal{C}}, F(\lambda) \in \prod_{A_i \in A}\{1, ..., \mathsf{Mov}_{\mathcal{C}}(\lambda[j], A_i)\}\}$*. Un élément* F *de* $\Sigma^n_{A,\mathcal{C}}$ *associe donc à chaque préfixe d'exécution* λ *de taille* $\leq n+1$*, un mouvement de chaque joueur* $A_i \in A$ *disponible depuis le dernier état constituant* λ*.*

Soient $F \in \Sigma^n_{A,\mathcal{C}}$ *et* $s \in \mathsf{Loc}_{\mathcal{C}}$ *;* F *induit un sous-ensemble de* $\mathsf{Exec}_{\mathcal{C}}(s)$*, appelé ensemble des* issues *de* F *depuis l'état* s*, constitué des* s*-exécutions que les joueurs* $A_i \in A$ *contraignent lorsqu'ils jouent selon la stratégie* F : $\lambda \in \mathsf{Out}_{\mathcal{C}}(s, F)$ *ssi* $\lambda[0] = s$*, et pour tout* $k \in \mathbb{N}$ *on a* $\lambda[k+1] \in \mathsf{Next}_{\mathcal{C}}(\lambda[k], A, F(\lambda[\mathsf{Max}(k-n, 0), k]))$*.*

Remarques :
– Dans les cas $n = 0$ et $n = \infty$, on retrouve les notations déjà introduites.

– Dans le cas particulier $A = \emptyset$, $\forall n \in \mathbb{N} \cup \{\infty\}, \Sigma^n_{A,\mathcal{C}} = \{\emptyset\}$. L'unique stratégie pour la coalition $\emptyset$ est donc la stratégie vide, et on a : $\forall s \in \mathsf{Loc}_{\mathcal{C}}, \mathsf{Out}_{\mathcal{C}}(s, \emptyset) = \mathsf{Exec}_{\mathcal{C}}(s)$.

Lorsque le contexte sera clair, on omettra les indices correspondant à une *CGS*.

Définition 26. *Soient $\mathcal{C}$ une* CGS, $n \in \mathbb{N} \cup \{\infty\}$, $A, B, C \subseteq \mathsf{Agt}$ *tels que* $C \subseteq A$, $F = (F_{A_i})_{A_i \in A} \in \Sigma^n_A$ *et* $F' = (F'_{A_i})_{A_i \in B} \in \Sigma^n_B$.

On définit :

– $F' \circ_n F = (F''_{A_i})_{A_i \in A \cup B}$,

où $F''_{A_i} = \begin{cases} F'_{A_i} & \textit{si } A_i \in B \\ F_{A_i} & \textit{sinon.} \end{cases}$

– $F|_{n,C} = (F_{A_i})_{A_i \in C}$.

On a donc $\forall n \in \mathbb{N} \cup \{\infty\}, F' \circ_n F \in \Sigma^n_{A \cup B}$ *et* $F|_{n,C} \in \Sigma^n_C$.

Les lois $\circ_n$ ne sont pas commutatives, mais elles sont associatives, et l'on pourra donc omettre les parenthèses. Leur élément neutre est la stratégie $\emptyset$.

Remarque :

– Dans les cas $n = 0$ et $n = \infty$, on retrouve les notations déjà introduites.

Soit $n \in \mathbb{N} \cup \{\infty\}$. On définit la logique $\mathcal{L}^{**}_n$ sur le modèle des logiques à deux étoiles déjà introduites.

Définition 27. *La syntaxe de $\mathcal{L}^{**}_n$ est définie par la grammaire suivante :*

$$\begin{aligned} \mathcal{L}^{**}_n \ni \varphi_s, \psi_s \quad &::= \quad p \mid \neg\varphi_s \mid \varphi_s \vee \psi_s \mid \lessdot A \gtrdot_n \varphi_p \mid \gtrdot A \lessdot_n \varphi_p \\ \mathcal{L}^{**}_{n,p} \ni \varphi_p, \psi_p \quad &::= \quad \varphi_s \mid \neg\varphi_p \mid \varphi_p \vee \psi_p \mid \mathbf{X}\,\varphi_p \mid \varphi_p \,\mathbf{U}\, \psi_p. \end{aligned}$$

où p décrit AP et A les sous-ensembles de Agt.

La sémantique de $\mathcal{L}^{**}_n$ est définie comme suit :

Soient $A, B \subseteq \mathsf{Agt}, q \in \mathsf{Loc}$ et $F \in \Sigma^n_A$,

$$\begin{aligned}
q \models_F p \quad & \text{ssi} \quad p \in \mathsf{Lab}(q), \\
q \models_F \neg\varphi_s \quad & \text{ssi} \quad q \not\models_F \varphi_s, \\
q \models_F \varphi_s \vee \psi_s \quad & \text{ssi} \quad q \models_F \varphi_s \text{ ou } q \models_F \psi_s, \\
q \models_F \lessdot B \gtrdot_n \varphi_p \quad & \text{ssi} \quad \exists F' \in \Sigma^n_B \text{ tel que } \forall \lambda \in \mathsf{Out}(q, F' \circ_n F), \lambda \models_{F' \circ_n F} \varphi_p, \\
q \models_F \gtrdot B \lessdot_n \varphi_p \quad & \text{ssi} \quad \forall \lambda \in \mathsf{Out}(q, F|_{n, A \setminus B}), \lambda \models_{F|_{n, A \setminus B}} \varphi_p, \\
\lambda \models_F \varphi_s \quad & \text{ssi} \quad \lambda[0] \models_F \varphi_s, \\
\lambda \models_F \neg\varphi_p \quad & \text{ssi} \quad \lambda \not\models_F \varphi_p, \\
\lambda \models_F \varphi_p \vee \psi_p \quad & \text{ssi} \quad \lambda \models_F \varphi_p \text{ ou } \lambda \models_F \psi_p, \\
\lambda \models_F \mathbf{X}\,\varphi_p \quad & \text{ssi} \quad \lambda[1, \infty] \models_F \varphi_p, \\
\lambda \models_F \varphi_p \,\mathbf{U}\, \psi_p \quad & \text{ssi} \quad \exists i \geq 0.\ \lambda[i, \infty] \models_F \psi_p \text{ et } \forall 0 \leq j < i.\ \lambda[j, \infty] \models_F \varphi_p.
\end{aligned}$$

Remarques :

– Le symbole $\models_\emptyset$ sera plus simplement noté $\models$.

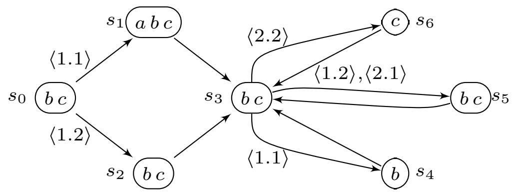

FIG. 4.1 – Les limites de l'approche

– Dans les cas $n = 0$ et $n = \infty$, on retrouve les notations déjà introduites.
On va montrer les limites de cette approche.

Lemme 16. *Sur la* CGS $\mathcal{C}$ *de la Fig. 7., on a :*
$\forall n \in \mathbb{N}, s_0 \models \neg \lessdot A_1 \gtrdot_n ((\mathbf{X}\, a \wedge \mathbf{G}\, b) \vee (\mathbf{X} \neg a \wedge \mathbf{G}\, c))$.

Preuve : Soit $n \in \mathbb{N}$, et $F \in \Sigma^n_{A_1}$. Pour simplifier, on va supposer que n est impair. Considérons alors le préfixe d'exécution $\alpha = s_3 s_5 s_3 s_5 ... s_3$ de taille n. Supposons que $F(\alpha) = 1$; si ce n'est pas le cas, alors $F(\lambda) = 2$, et on peut raisonner de façon similaire. Soit λ l'exécution telle que :
– $\lambda[0] = s_0$,
– $\lambda[1] = s_2$,
– Si $m \geq 2$ est pair, alors $\lambda[m] = s_3$,
– $\lambda[n+2] = s_4$,
– Si $m \geq 3$ est impair et $m \neq n+2$, alors $\lambda[m] = s_5$
On a alors $\lambda \in \mathsf{Out}(s_0, F)$, mais $\lambda \models_F \neg((\mathbf{X}\, a \wedge \mathbf{G}\, b) \vee (\mathbf{X} \neg a \wedge \mathbf{G}\, c))$, puisque $\lambda[1] = s_2$ et $\lambda[n+2] = s_4$, donc $\lambda \models_F \mathbf{X} \neg a$ et $\lambda \models_F \neg \mathbf{G}\, c$. $\square$

4.2 Seconde définition de la taille de la mémoire d'une stratégie ; les logiques $\mathcal{L}'^{**}_n$

Une autre idée plus féconde consiste à définir les objets suivants :

Définition 28. *Soient* $\mathcal{C}$ *une* CGS, $A_i \in \mathsf{Agt}_\mathcal{C}$ *et* $n \in \mathbb{N} \cup \{\infty\}$.
On note $\Sigma'^n_{A_i,\mathcal{C}}$ *l'ensemble des applications de* $\{0, ..., n\} \times \mathsf{Loc}_\mathcal{C}$ *vers* $\{0, ..., n\} \times \mathbb{N}$, *telles que* $\forall (l, s) \in \{0, ..., n\} \times \mathsf{Loc}_\mathcal{C}, \pi_2(F(l, s)) \in \{1, ..., \mathsf{Mov}_\mathcal{C}(s, A_i)\}$. *Si* $A \subseteq \mathsf{Agt}_\mathcal{C}$, *on note* $\Sigma'^n_{A,\mathcal{C}} = \prod_{A_i \in A} \Sigma'^n_{A_i,\mathcal{C}}$.
Soient $F = (F_{A_i})_{A_i \in A} \in \Sigma'^n_{A,\mathcal{C}}$ *et* $s \in \mathsf{Loc}_\mathcal{C}$ *;* F *induit un sous-ensemble de* $\mathsf{Exec}_\mathcal{C}(s)$, *appelé ensemble des* issues *de* F *depuis l'état* s, *constitué des* s-*exécutions que les joueurs* $A_i \in A$ *contraignent lorsqu'ils jouent selon la stratégie* F. *Soit* $\lambda \in (\mathbb{N}^{|A|} \times \mathsf{Loc}_\mathcal{C})^\mathbb{N}$ *une suite d'éléments de* $\mathbb{N}^{|A|} \times \mathsf{Loc}_\mathcal{C}$. *En munissant* A *d'un ordre total, si* A_j *constitue le i-ème joueur de* A, *on désignera par* π_{A_j} *la projection* π_i. *On pose* $\lambda \in \mathsf{Out}_{A,\mathcal{C}}(s, F)$ *ssi*
– $\pi_{|A|+1}(\lambda[0]) = s$,
– $\forall A_i \in A, \pi_{A_i}(\lambda[0]) = 0$,

– $\forall A_i \in A, \forall k \in \mathbb{N}, \pi_{A_i}(\lambda[k+1]) = \pi_1(F_{A_i}(\pi_{A_i}(\lambda[k]), \pi_{|A|+1}(\lambda[k])))$,
– $\forall k \in \mathbb{N}, \pi_{|A|+1}(\lambda[k{+}1]) \in \mathsf{Next}_{\mathcal{C}}(\pi_{|A|+1}(\lambda[k]), A, (\pi_2(F_{A_i}(\pi_{A_i}(\lambda[k]), \pi_{|A|+1}(\lambda[k]))))_{A_i \in A})$.

On notera donc $\mathsf{Out}_{\mathcal{C}}(s,F) = \pi_{|A|+1}(\mathsf{Out}_{A,\mathcal{C}}(s,F))$. *Il s'agit en quelque sorte des traces des exécutions de* $\mathsf{Out}_{A,\mathcal{C}}(s,F)$. *On a bien* $\mathsf{Out}_{\mathcal{C}}(s,F) \subseteq \mathsf{Exec}_{\mathcal{C}}(s)$.

Remarques :
– Lorsque le contexte sera clair, on omettra les indices correspondant à une *CGS*.
– Dans le cas particulier $A = \emptyset$, $\forall n \in \mathbb{N} \cup \{\infty\}, \Sigma'^{n}_{A,\mathcal{C}} = \{\emptyset\}$. L'unique stratégie pour la coalition $\emptyset$ est donc la stratégie vide, et on a : $\forall s \in \mathsf{Loc}_{\mathcal{C}}, \mathsf{Out}_{\mathcal{C}}(s,\emptyset) = \mathsf{Exec}_{\mathcal{C}}(s)$
– Soit $n \in \mathbb{N} \cup \{\infty\}, s \in \mathsf{Loc}$ et $A \in \mathsf{Agt}$. Si $F \in \Sigma'^{n}_{A,\mathcal{C}}$, alors $\mathsf{Out}_A(s,F) \subset ((\{0,...,n\})^{|A|} \times \mathsf{Loc})^{\mathbb{N}}$.
– 0 est choisi pour symboliser l'"origine" des mémoires.

Définition 29. *Soient* $\mathcal{C}$ *une* CGS, $n \in \mathbb{N} \cup \{\infty\}, A_i \in \mathit{Agt}$ *et* $F \in \Sigma'^{n}_{A_i}$. *On définit par récurrence la famille d'applications* $(\nu_{F,j})_{j\in\mathbb{N}}$ *telle que :*

– $\nu_{F,0} : \left\{ \begin{array}{rcl} \mathit{Loc}^0 & \to & \mathbb{N} \\ \emptyset & \mapsto & \nu_{F,0}(\emptyset) = 0 \end{array} \right.$,

– $\nu_{F,j+1} : \left\{ \begin{array}{rcl} \mathit{Loc}^{j+1} & \to & \mathbb{N} \\ \lambda & \mapsto & \nu_{F,j+1}(\lambda) = \pi_1(F(\nu_{F,j}(\lambda[0,j-1]), \lambda[j])) \end{array} \right.$,

On pose $\nu_F = \bigcup_{j\in\mathbb{N}} \nu_{F,j}$. *Intuitivement,* $\nu_F(\lambda)$ *désigne l'état de la cellule mémoire associée à la stratégie* F *lorsque s'est effectué le préfixe d'exécution* λ.

Définition 30. *Soient* $\mathcal{C}$ *une* CGS, $n \in \mathbb{N} \cup \{\infty\}, A, B, C \subseteq \mathit{Agt}$ *tels que* $C \subseteq A$, $F = (F_{A_i})_{A_i \in A} \in \Sigma'^{n}_{A}$ *et* $F' = (F'_{A_i})_{A_i \in B} \in \Sigma'^{n}_{B}$.

On définit :

– $F' \circ'_n F = (F''_{A_i})_{A_i \in A\cup B}$,

où $F''_{A_i} = \begin{cases} F'_{A_i} & \text{si } A_i \in B \\ F_{A_i} & \text{sinon.} \end{cases}$,

– $F|'_{n,C} = (F_{A_i})_{A_i \in C}$.

On a donc $\forall n \in \mathbb{N} \cup \{\infty\}, F' \circ'_n F \in \Sigma'^{n}_{A\cup B}$ *et* $F|'_C \in \Sigma'^{n}_{C}$.

Les lois $\circ'_n$ *ne sont pas commutatives, mais elles sont associatives, et l'on pourra donc omettre les parenthèses. L'élément neutre de la loi* $\circ'_n$ *est la stratégie* $\emptyset$.

Soit $n \in \mathbb{N} \cup \{\infty\}$. On définit la logique $\mathcal{L}'^{**}_{n}$.

Définition 31. *La syntaxe de* $\mathcal{L}'^{**}_{n}$ *est définie par la grammaire suivante :*

$$\begin{array}{rcl} \mathcal{L}'^{**}_{n} \ni \varphi_s, \psi_s & ::= & p \mid \neg\varphi_s \mid \varphi_s \vee \psi_s \mid \lessdot A \gtrdot'_n \varphi_p \mid \gtrdot A \lessdot'_n \varphi_p \\ \mathcal{L}'^{**}_{n,p} \ni \varphi_p, \psi_p & ::= & \varphi_s \mid \neg\varphi_p \mid \varphi_p \vee \psi_p \mid \mathbf{X}\,\varphi_p \mid \varphi_p\,\mathbf{U}\,\psi_p. \end{array}$$

où p *décrit* AP *et* A *les sous-ensembles de* Agt.

La sémantique de $\mathcal{L}'^{**}_{n}$ est définie comme suit :

Soient $A, B \subseteq \mathsf{Agt}, q \in \mathsf{Loc}$ et $F \in \Sigma_A^{\prime n}$,

$$\begin{array}{rcl}
q \models_F p & \text{ssi} & p \in \mathsf{Lab}(q), \\
q \models_F \neg\varphi_s & \text{ssi} & q \not\models_F \varphi_s, \\
q \models_F \varphi_s \vee \psi_s & \text{ssi} & q \models_F \varphi_s \text{ ou } q \models_F \psi_s, \\
q \models_F \ \lessdot B \gtrdot_n' \ \varphi_p & \text{ssi} & \exists F' \in \Sigma_B^{\prime n} \text{ tel que } \forall \lambda \in \mathsf{Out}(q, F' \circ_n' F), \lambda \models_{F' \circ_n' F} \varphi_p, \\
q \models_F \ \gtrdot B \lessdot_n' \ \varphi_p & \text{ssi} & \forall \lambda \in \mathsf{Out}(q, F|_{n, A\setminus B}'), \lambda \models_{F|_{n,A\setminus B}'} \varphi_p, \\
\lambda \models_F \varphi_s & \text{ssi} & \lambda[0] \models_F \varphi_s, \\
\lambda \models_F \neg\varphi_p & \text{ssi} & \lambda \not\models_F \varphi_p, \\
\lambda \models_F \varphi_p \vee \psi_p & \text{ssi} & \lambda \models_F \varphi_p \text{ ou } \lambda \models_F \psi_p, \\
\lambda \models_F \mathbf{X}\,\varphi_p & \text{ssi} & \lambda[1, \infty] \models_F \varphi_p, \\
\lambda \models_F \varphi_p \,\mathbf{U}\, \psi_p & \text{ssi} & \exists i \geq 0.\ \lambda[i, \infty] \models_F \psi_p \text{ et } \forall 0 \leq j < i.\ \lambda[j, \infty] \models_F \varphi_p.
\end{array}$$

Remarque :
- Le symbole $\models_\emptyset$ sera plus simplement noté $\models$.

4.2.1 Cohérence des définitions

Lemme 17. *La nouvelle logique $\mathcal{L}_0^{\prime **}$ est identique à la logique $\mathcal{L}_0^{**}$ définie auparavant.*

Preuve : Soient $\mathcal{C}$ une *CGS*, $A_i \in \mathsf{Agt}$ et $A \subseteq \mathsf{Agt}$. On définit les applications suivantes : $\sigma_{A_i} : \left\{ \begin{array}{ccc} \Sigma_{A_i}^0 & \to & \Sigma_{A_i}^{\prime 0} \\ F & \mapsto & \sigma_{A_i}(F) : \left\{ \begin{array}{ccc} \{0\} \times \mathsf{Loc} & \to & \{0\} \times \mathbb{N} \\ (0, q) & \mapsto & (\sigma_{A_i}(F))(0, q) = (0, F(q)) \end{array} \right. \end{array} \right.$,
puis $\sigma_A : \left\{ \begin{array}{ccc} \Sigma_A^0 & \to & \Sigma_A^{\prime 0} \\ F = (F_{A_i})_{A_i \in A} & \mapsto & \sigma_A(F) = (\sigma_{A_i}(F_{A_i}))_{A_i \in A} \end{array} \right.$,

On définit maintenant par induction l'application $\delta \colon \mathcal{L}_{0,p}^{**} \to \mathcal{L}_{0,p}^{\prime **}$ telle que :
- Si $\varphi = a \in \mathsf{AP}$, alors $\delta(\varphi) = a$,
- Si $\varphi = \neg\psi$, alors $\delta(\varphi) = \neg\delta(\psi)$,
- Si $\varphi = \psi_1 \vee \psi_2$, alors $\delta(\varphi) = \delta(\psi_1) \vee \delta(\psi_2)$,
- Si $\varphi = \mathbf{X}\,\psi$, alors $\delta(\varphi) = \mathbf{X}\,\delta(\psi)$,
- Si $\varphi = \psi_1 \,\mathbf{U}\, \psi_2$, alors $\delta(\varphi) = \delta(\psi_1) \,\mathbf{U}\, \delta(\psi_2)$,
- Si $\varphi = \ \lessdot B \gtrdot_0 \ \psi$, alors $\delta(\varphi) = \ \lessdot B \gtrdot_0' \ \delta(\psi)$,
- Si $\varphi = \ \gtrdot B \lessdot_0 \ \psi$, alors $\delta(\varphi) = \ \gtrdot B \lessdot_0' \ \delta(\psi)$.

Il est évident que σ_A et δ sont des applications bijectives. De plus, dans le cas particulier $A = \emptyset, \sigma_A$ envoie la stratégie $\emptyset$ sur la stratégie $\emptyset$.

Soit $\mathcal{C}$ une *CGS*. On va montrer par induction sur les formules de $\mathcal{L}_{0,p}^{**}$ que :
$\forall \varphi \in \mathcal{L}_{0,p}^{**}, \forall s \in \mathsf{Loc}, \forall \lambda \in \mathsf{Exec}(s), \forall A \subseteq \mathsf{Agt}, \forall F \in \Sigma_A^0$,
$\lambda \models_F \varphi$ ssi $\lambda \models_{\sigma_A(F)} \delta(\varphi)$
- Si $\varphi = a \in \mathsf{AP}$, alors les deux formules sont vraies ssi $a \in \mathsf{Lab}(s)$.
- Si $\varphi = \neg\psi$, alors c'est évident.
- Si $\varphi = \psi_1 \vee \psi_2$, alors c'est évident.

– Si $\varphi = \mathbf{X}\,\psi$. Soient $s \in \mathsf{Loc}, \lambda \in \mathsf{Exec}(s), A \subseteq \mathsf{Agt}$ et $F \in \Sigma^0_A$. On a la suite d'équivalences :
$\lambda \models_{\sigma_A(F)} \delta(\mathbf{X}\,\psi)$ ssi
$\lambda \models_{\sigma_A(F)} \mathbf{X}\,\delta(\psi)$ ssi
$\lambda[1,\infty] \models_{\sigma_A(F)} \delta(\psi)$ ssi, par HI appliquée à l'état $\lambda[1]$, à l'exécution $\lambda[1,\infty]$, à la même coalition A et à la même stratégie F,
$\lambda[1,\infty] \models_F \psi$ ssi
$\lambda \models_F \mathbf{X}\,\psi$.
– Si $\varphi = \psi_1 \,\mathbf{U}\, \psi_2$, alors un raisonnement similaire donne le résultat.
– Si $\varphi = \lessdot B \gtrdot_0 \psi$. Soient $s \in \mathsf{Loc}, \lambda \in \mathsf{Exec}(s), A \subseteq \mathsf{Agt}$ et $F \in \Sigma^0_A$. On a la suite d'équivalences :
$\lambda \models_{\sigma_A(F)} \delta(\lessdot B \gtrdot_0 \psi)$ ssi
$\lambda \models_{\sigma_A(F)} \lessdot B \gtrdot'_0 \delta(\psi)$ ssi
$s \models_{\sigma_A(F)} \lessdot B \gtrdot'_0 \delta(\psi)$ ssi
$\exists F' \in \Sigma'^0_B$ tel que $\forall \lambda' \in \mathsf{Out}(s, F' \circ'_0 \sigma_A(F)), \lambda' \models_{F' \circ'_0 \sigma_A(F)} \delta(\psi)$ ssi, puisque σ_B est une bijection,
$\exists F' \in \Sigma^0_B$ tel que $\forall \lambda' \in \mathsf{Out}(s, \sigma_B(F') \circ'_0 \sigma_A(F)), \lambda' \models_{\sigma_B(F') \circ'_0 \sigma_A(F)} \delta(\psi)$ ssi, puisque $\sigma_B(F') \circ'_0 \sigma_A(F) = \sigma_{A\cup B}(F' \circ_0 F)$,
$\exists F' \in \Sigma^0_B$ tel que $\forall \lambda' \in \mathsf{Out}(s, \sigma_{A\cup B}(F' \circ_0 F)), \lambda' \models_{\sigma_{A\cup B}(F' \circ_0 F)} \delta(\psi)$ ssi, puisque $\mathsf{Out}(s, \sigma_{A\cup B}(F' \circ_0 F)) = \mathsf{Out}(s, F' \circ_0 F)$,
$\exists F' \in \Sigma^0_B$ tel que $\forall \lambda' \in \mathsf{Out}(s, F' \circ_0 F), \lambda' \models_{\sigma_{A\cup B}(F' \circ_0 F)} \delta(\psi)$ ssi, par HI appliquée à l'état s, à l'exécution λ', à la coalition $A \cup B$ et à la stratégie $F' \circ_0 F$,
$\exists F' \in \Sigma^0_B$ tel que $\forall \lambda' \in \mathsf{Out}(s, F' \circ_0 F), \lambda' \models_{F' \circ_0 F} \psi$ ssi
$s \models_F \lessdot B \gtrdot_0 \psi$ ssi
$\lambda \models_F \lessdot B \gtrdot_0 \psi$
– Si $\varphi = \gtrdot B \lessdot_0 \psi$. Soient $s \in \mathsf{Loc}, \lambda \in \mathsf{Exec}(s), A \subseteq \mathsf{Agt}$ et $F \in \Sigma^0_A$. On a la suite d'équivalences :
$\lambda \models_{\sigma_A(F)} \delta(\gtrdot B \lessdot_0 \psi)$ ssi
$\lambda \models_{\sigma_A(F)} \gtrdot B \lessdot'_0 \delta(\psi)$ ssi
$s \models_{\sigma_A(F)} \gtrdot B \lessdot'_0 \delta(\psi)$ ssi
$\forall \lambda' \in \mathsf{Out}(s, \sigma_A(F)|'_{0,A\setminus B}), \lambda' \models_{\sigma_A(F)|'_{0,A\setminus B}} \delta(\psi)$ ssi, puisque $\sigma_A(F)|'_{0,A\setminus B} = \sigma_{A\setminus B}(F|_{0,A\setminus B})$,
$\forall \lambda' \in \mathsf{Out}(s, \sigma_{A\setminus B}(F|_{0,A\setminus B})), \lambda' \models_{\sigma_{A\setminus B}(F|_{0,A\setminus B})} \delta(\psi)$ ssi, puisque $\mathsf{Out}(s, \sigma_{A\setminus B}(F|_{0,A\setminus B})) = \mathsf{Out}(s, F|_{0,A\setminus B})$,
$\forall \lambda' \in \mathsf{Out}(s, F|_{0,A\setminus B}), \lambda' \models_{\sigma_{A\setminus B}(F|_{0,A\setminus B})} \delta(\psi)$ ssi, par HI appliquée à l'état s, à l'exécution λ', à la coalition $A\setminus B$ et à la stratégie $F|_{0,A\setminus B}$,
$\forall \lambda' \in \mathsf{Out}(s, F|_{0,A\setminus B}), \lambda' \models_{F|_{0,A\setminus B}} \psi$ ssi
$s \models_F \gtrdot B \lessdot_0 \psi$ ssi
$\lambda \models_F \gtrdot B \lessdot_0 \psi$

Ceci, appliqué dans la cas où φ est une formule d'état et F est la stratégie $\emptyset$, établit que

$\forall \varphi \in \mathcal{L}_0^{**}, \forall s \in \mathsf{Loc}, s \models \varphi$ ssi $s \models \delta(\varphi)$, c'est-à-dire que les formules φ et $\delta(\varphi)$ sont équivalentes. $\square$

Lemme 18. *La nouvelle logique $\mathcal{L}'^{**}_{\infty}$ est identique à la logique $\mathcal{L}^{**}_{\infty}$ définie auparavant.*

Preuve : Soit $\mathcal{C}$ une *CGS*. L'ensemble $\bigcup_{j\in\mathbb{N}} \mathsf{Loc}^j_{\mathcal{C}}$ est dénombrable, soit donc une bijection α de $\mathbb{N}$ vers cet ensemble, telle que $\alpha(0) = \emptyset$. La longueur d'un préfixe d'exécution λ sera notée $lg(\lambda)$; en particulier, le dernier état constituant λ sera $\lambda[lg(\lambda) - 1]$. On définit les applications :

– $\tau_{A_i} : \left\{ \begin{array}{rcl} \Sigma'^{\infty}_{A_i} & \to & \Sigma^{\infty}_{A_i} \\ F & \mapsto & \tau_{A_i}(F) : \left\{ \begin{array}{rcl} \bigcup_{j\in\mathbb{N}^*} \mathsf{Loc}^j & \to & \mathbb{N} \\ \lambda & \mapsto & (\tau_{A_i}(F))(\lambda) = \pi_2(F(\nu_F(\lambda[0, lg(\lambda)-2]), \lambda[lg(\lambda)-1] \end{array} \right. \end{array} \right.$

– $\sigma_{A_i} : \left\{ \begin{array}{rcl} \Sigma^{\infty}_{A_i} & \to & \Sigma'^{\infty}_{A_i} \\ F & \mapsto & \sigma_{A_i}(F) : \left\{ \begin{array}{rcl} \mathbb{N} \times \mathsf{Loc} & \to & \mathbb{N} \times \mathbb{N} \\ (l, q) & \mapsto & (\sigma_{A_i}(F))(l, q) = (\alpha^{-1}(\alpha(l)q), F(\alpha(l)q)) \end{array} \right. \end{array} \right.$.

Soit $A \subseteq \mathsf{Agt}$. On définit encore les applications :

– $\tau_A : \left\{ \begin{array}{rcl} \Sigma'^{\infty}_{A} & \to & \Sigma^{\infty}_{A} \\ F = (F_{A_i})_{A_i\in A} & \mapsto & \tau_A(F) = (\tau_{A_i}(F_{A_i}))_{A_i\in A} \end{array} \right.$,

– $\sigma_A : \left\{ \begin{array}{rcl} \Sigma^{\infty}_{A} & \to & \Sigma'^{\infty}_{A} \\ F = (F_{A_i})_{A_i\in A} & \mapsto & \sigma_A(F) = (\sigma_{A_i}(F_{A_i}))_{A_i\in A} \end{array} \right.$.

Soit $F \in \Sigma'^{\infty}_{A_i}$. On va montrer par récurrence sur la longueur de λ que $\forall \lambda \in \bigcup_{j\in\mathbb{N}} \mathsf{Loc}^j, \nu_{\sigma_{A_i}(F)}(\lambda) = \alpha^{-1}(\lambda)$.

– Si $lg(\lambda) = 0$, alors $\lambda = \emptyset$, et $\nu_{\sigma_{A_i}(F)}(\lambda) = 0 = \alpha^{-1}(\emptyset)$.
– Si $lg(\lambda) = j+1$, alors $\nu_{\sigma_{A_i}(F)}(\lambda) = \nu_{\sigma_{A_i}(F),j+1}(\lambda) = \pi_1(\sigma_{A_i}(F)(\nu_{\sigma_{A_i}(F),j}(\lambda[0, j-1]), \lambda[j])) = \pi_1(\sigma_{A_i}(F)(\alpha^{-1}(\lambda[0, j-1]), \lambda[j])) = \alpha^{-1}(\alpha(\alpha^{-1}(\lambda[0, j-1]))\lambda[j]) = \alpha^{-1}(\lambda)$.

De cela, on déduit que $\tau_{A_i} \circ \sigma_{A_i} = \mathrm{Id}$, et donc que τ_{A_i} et par conséquent τ_A sont des applications surjectives.

Soit $F = (F_{A_i})_{A_i\in A} \in \Sigma'^{\infty}_{A}$ et $s \in \mathsf{Loc}$. On va prouver que $\mathsf{Out}(s, F) = \mathsf{Out}(s, \tau_A(F))$.

Soit $\lambda \in \mathsf{Out}(s, F)$, et $\lambda' \in \mathsf{Out}_A(s, F)$ tel que $\pi_{|A|+1}(\lambda') = \lambda$. On va montrer par induction sur $k \in \mathbb{N}$ que

$\forall k \in \mathbb{N}, \forall A_i \in A, \nu_{F_{A_i}}(\lambda[0, k-1]) = \pi_{A_i}(\lambda'[k])$.

– Si $k = 0$. Soit $A_i \in A$. Alors $\nu_{F_{A_i}}(\lambda[0, k-1]) = \pi_{A_i}(\lambda'[k]) = 0$
– Soit $k \in \mathbb{N}$ et $A_i \in A$. Alors $\nu_{F_{A_i}}(\lambda[0, k]) = \nu_{F_{A_i},k+1}(\lambda[0, k]) = \pi_1(F_{A_i}(\nu_{F_{A_i},k}(\lambda[0, k-1]), \lambda[k])) = \pi_1(F_{A_i}(\pi_{A_i}(\lambda'[k], \pi_{|A|+1}(\lambda'[k])))) = \pi_{A_i}(\lambda'[k+1])$

Soit $k \in \mathbb{N}$. Puisque $\lambda' \in \mathsf{Out}_A(s, F)$, on a

$\pi_{|A|+1}(\lambda'[k+1]) \in \mathsf{Next}(\pi_{|A|+1}(\lambda'[k]), A, (\pi_2(F_{A_i}(\pi_{A_i}(\lambda'[k]), \pi_{|A|+1}(\lambda'[k]))))_{A_i\in A})$, c'est-à-dire $\lambda[k+1] \in \mathsf{Next}(\lambda[k], A, (\pi_2(F_{A_i}(\nu_{F_{A_i}}(\lambda[0, k-1]), \lambda[k])))_{A_i\in A})$. Or $\pi_2(F_{A_i}(\nu_{F_{A_i}}(\lambda[0, k-1]), \lambda[k])))_{A_i\in A} = \tau_{A_i}(F_{A_i})(\lambda[0, k])$, donc on a bien ce que l'on veut.

On doit à présent montrer l'autre inclusion. Soit $\lambda \in \mathsf{Out}(s, \tau_A(F))$. Soit λ' la suite d'éléments de $\mathbb{N}^{|A|} \times \mathsf{Loc}_{\mathcal{C}}$ telle que :

– $\forall A_i \in A, \forall k \in \mathbb{N}, \pi_{A_i}(\lambda')[k] = \nu_{F_{A_i}}(\lambda[0, k-1])$,
– $\pi_{|A|+1}(\lambda') = \lambda$.

Il reste à établir que $\lambda' \in \mathsf{Out}_A(s,F)$. Soit $A_i \in A$ et $k \in \mathbb{N}$. On a $\pi_{A_i}(\lambda'[k+1]) = \nu_{F_{A_i}}(\lambda[0,k]) = \pi_1(F_{A_i}(\nu_{F_{A_i}}(\lambda[0,k-1]),\lambda[k])) = \pi_1(F_{A_i}(\pi_{A_i}(\lambda')[k],\pi_{|A|+1}(\lambda'[k])))$, et

$\pi_{|A|+1}(\lambda'[k+1]) \in \mathsf{Next}(\pi_{|A|+1}(\lambda[k]),A,(\tau_{A_i}(F_{A_i})(\lambda[0,k]))_{A_i\in A})$. Or $\forall A_i \in A, \tau_{A_i}(F_{A_i})(\lambda[0,k]) = \pi_2(F_{A_i}(\nu_{F_{A_i}}(\lambda[0,k-1]),\lambda[k])) = \pi_2(F_{A_i}(\pi_{A_i}(\lambda'[k]),\pi_{|A|+1}(\lambda'[k])))$.

On définit maintenant par induction l'application $\delta\colon \mathcal{L}'^{**}_{\infty,p} \to \mathcal{L}^{**}_{\infty,p}$ telle que :
- Si $\varphi = a \in \mathsf{AP}$, alors $\delta(\varphi) = a$,
- Si $\varphi = \neg\psi$, alors $\delta(\varphi) = \neg\delta(\psi)$,
- Si $\varphi = \psi_1 \vee \psi_2$, alors $\delta(\varphi) = \delta(\psi_1) \vee \delta(\psi_2)$,
- Si $\varphi = \mathbf{X}\,\psi$, alors $\delta(\varphi) = \mathbf{X}\,\delta(\psi)$,
- Si $\varphi = \psi_1\,\mathbf{U}\,\psi_2$, alors $\delta(\varphi) = \delta(\psi_1)\,\mathbf{U}\,\delta(\psi_2)$,
- Si $\varphi = \lessdot B \gtrdot'_\infty\ \psi$, alors $\delta(\varphi) = \lessdot B \gtrdot_\infty\ \delta(\psi)$,
- Si $\varphi = \gtrdot B \lessdot'_\infty\ \psi$, alors $\delta(\varphi) = \gtrdot B \lessdot_\infty\ \delta(\psi)$.

Il est évident que δ est une application bijective. De plus, dans le cas particulier $A = \emptyset, \tau_A$ envoie la stratégie $\emptyset$ sur la stratégie $\emptyset$.

Soit $\mathcal{C}$ une *CGS*. On va montrer par induction sur les formules de $\mathcal{L}'^{**}_{\infty,p}$ que :
$\forall\varphi \in \mathcal{L}'^{**}_{\infty,p}, \forall s \in \mathsf{Loc}, \forall\lambda \in \mathsf{Exec}(s), \forall A \subseteq \mathsf{Agt}, \forall F \in \Sigma'^{\infty}_A$,
$\lambda \models_F \varphi$ ssi $\lambda \models_{\tau_A(F)} \delta(\varphi)$
- Si $\varphi = a \in \mathsf{AP}$, alors les deux formules sont vraies ssi $a \in \mathsf{Lab}(s)$.
- Si $\varphi = \neg\psi$, alors c'est évident.
- Si $\varphi = \psi_1 \vee \psi_2$, alors c'est évident.
- Si $\varphi = \mathbf{X}\,\psi$. Soient $s \in \mathsf{Loc}, \lambda \in \mathsf{Exec}(s), A \subseteq \mathsf{Agt}$ et $F \in \Sigma'^{\infty}_A$. On a la suite d'équivalences :
  $\lambda \models_{\tau_A(F)} \delta(\mathbf{X}\,\psi)$ ssi
  $\lambda \models_{\tau_A(F)} \mathbf{X}\,\delta(\psi)$ ssi
  $\lambda[1,\infty] \models_{\tau_A(F)} \delta(\psi)$ ssi, par HI appliquée à l'état $\lambda[1]$, à l'exécution $\lambda[1,\infty]$, à la même coalition A et à la même stratégie F,
  $\lambda[1,\infty] \models_F \psi$ ssi
  $\lambda \models_F \mathbf{X}\,\psi$.
- Si $\varphi = \psi_1\,\mathbf{U}\,\psi_2$, alors un raisonnement similaire donne le résultat.
- Si $\varphi = \lessdot B \gtrdot'_\infty\ \psi$. Soient $s \in \mathsf{Loc}, \lambda \in \mathsf{Exec}(s), A \subseteq \mathsf{Agt}$ et $F \in \Sigma'^{\infty}_A$. On a la suite d'équivalences :
  $\lambda \models_{\tau_A(F)} \delta(\lessdot B \gtrdot'_\infty\ \psi)$ ssi
  $\lambda \models_{\tau_A(F)}\ \lessdot B \gtrdot_\infty\ \delta(\psi)$ ssi
  $s \models_{\tau_A(F)}\ \lessdot B \gtrdot_\infty\ \delta(\psi)$ ssi
  $\exists F' \in \Sigma^{\infty}_B$ tel que $\forall\lambda' \in \mathsf{Out}(s, F' \circ_\infty \tau_A(F)), \lambda' \models_{F'\circ_\infty\tau_A(F)} \delta(\psi)$ ssi, puisque τ_B est une surjection,
  $\exists F' \in \Sigma'^{\infty}_B$ tel que $\forall\lambda' \in \mathsf{Out}(s, \tau_B(F')\circ'_\infty\tau_A(F)), \lambda' \models_{\tau_B(F')\circ_\infty\tau_A(F)} \delta(\psi)$ ssi, puisque $\tau_B(F') \circ_\infty \tau_A(F) = \tau_{A\cup B}(F' \circ'_\infty F)$,
  $\exists F' \in \Sigma'^{\infty}_B$ tel que $\forall\lambda' \in \mathsf{Out}(s, \tau_{A\cup B}(F' \circ'_\infty F)), \lambda' \models_{\tau_{A\cup B}(F'\circ'_\infty F)} \delta(\psi)$ ssi, puisque $\mathsf{Out}(s, \tau_{A\cup B}(F' \circ'_\infty F)) = \mathsf{Out}(s, F' \circ'_\infty F)$,
  $\exists F' \in \Sigma'^{\infty}_B$ tel que $\forall\lambda' \in \mathsf{Out}(s, F' \circ'_\infty F), \lambda' \models_{\tau_{A\cup B}(F'\circ'_\infty F)} \delta(\psi)$ ssi, par HI appliquée à l'état s, à l'exécution λ', à la coalition $A\cup B$ et à la stratégie $F' \circ'_\infty F$,
  $\exists F' \in \Sigma'^{\infty}_B$ tel que $\forall\lambda' \in \mathsf{Out}(s, F' \circ'_\infty F), \lambda' \models_{F'\circ'_\infty F} \psi$ ssi

$s \models_F \lessdot B \gtrdot'_\infty \psi$ ssi
$\lambda \models_F \lessdot B \gtrdot'_\infty \psi$

– Si $\varphi = \gtrdot B \lessdot'_\infty \psi$. Soient $s \in \mathsf{Loc}, \lambda \in \mathsf{Exec}(s), A \subseteq \mathsf{Agt}$ et $F \in \Sigma'^{\infty}_A$. On a la suite d'équivalences :
$\lambda \models_{\tau_A(F)} \delta(\gtrdot B \lessdot'_\infty \psi)$ ssi
$\lambda \models_{\tau_A(F)} \gtrdot B \lessdot_\infty \delta(\psi)$ ssi
$s \models_{\tau_A(F)} \gtrdot B \lessdot_\infty \delta(\psi)$ ssi
$\forall \lambda' \in \mathsf{Out}(s, \tau_A(F)|_{\infty, A\backslash B}), \lambda' \models_{\tau_A(F)|_{\infty,A\backslash B}} \delta(\psi)$ ssi, puisque $\tau_A(F)|_{\infty,A\backslash B} = \tau_{A\backslash B}(F|'_{\infty,A\backslash B})$,
$\forall \lambda' \in \mathsf{Out}(s, \tau_{A\backslash B}(F|'_{\infty,A\backslash B})), \lambda' \models_{\tau_{A\backslash B}(F|'_{\infty,A\backslash B})} \delta(\psi)$ ssi, puisque $\mathsf{Out}(s, \tau_{A\backslash B}(F|'_{\infty,A\backslash B})) = \mathsf{Out}(s, F|'_{\infty,A\backslash B})$,
$\forall \lambda' \in \mathsf{Out}(s, F|'_{\infty,A\backslash B}), \lambda' \models_{\tau_{A\backslash B}(F|'_{\infty,A\backslash B})} \delta(\psi)$ ssi, par HI appliquée à l'état s, à l'exécution λ', à la coalition $A\backslash B$ et à la stratégie $F|'_{\infty,A\backslash B}$,
$\forall \lambda' \in \mathsf{Out}(s, F|'_{\infty,A\backslash B}), \lambda' \models_{F|'_{\infty,A\backslash B}} \psi$ ssi
$s \models_F \gtrdot B \lessdot'_\infty \psi$ ssi
$\lambda \models_F \gtrdot B \lessdot'_\infty \psi$

Ceci, appliqué dans la cas où φ est une formule d'état et F est la stratégie $\emptyset$, établit que

$\forall \varphi \in \mathcal{L}'^{**}_\infty, \forall s \in \mathsf{Loc}, s \models \varphi$ ssi $s \models \delta(\varphi)$, c'est-à-dire que les formules φ et $\delta(\varphi)$ sont équivalentes. □

4.2.2 Model-checking des logiques $\mathcal{L}'^{**}_n$ lorsque n est fini

On va maintenant s'attaquer au problème de model-checking de la logique $\mathcal{L}'^{**}_n$ où n est un entier fini, que l'on notera MC $\mathcal{L}'^{**}_n$. Pour cela, l'idée essentielle est de modéliser la mémoire de chaque joueur par une *CGS* à $n+1$ éléments entièrement controlée par ce joueur, et qui possède toutes les transitions. On va ensuite le faire jouer à la fois sur la *CGS* figurant dans la donnée, et celle modélisant sa mémoire, puis recourir au model-checking de LTL. La méthode pour préserver des contextes imposés par les stratégies est la même que celle qui a été employée dans le model-checking de la logique $\mathcal{L}^{**}_0$.

Définition 32. *Soient* $\mathcal{C}_1 = (\mathsf{Agt}, \mathsf{Loc}_1, \mathsf{AP}_1, \mathsf{Lab}_1, \mathsf{Mov}_1, \mathsf{Edg}_1)$ *et* $\mathcal{C}_2 = (\mathsf{Agt}, \mathsf{Loc}_2, \mathsf{AP}_2, \mathsf{Lab}_2, \mathsf{Mov}_2, \mathsf{Edg}_2)$ *deux* CGS, *avec la notation* $|\mathsf{Agt}| = k$.

On définit $\mathcal{C}_1 \times \mathcal{C}_2 = (\mathsf{Agt}, \mathsf{Loc}_{\mathcal{C}_1\times\mathcal{C}_2}, \mathsf{AP}_{\mathcal{C}_1\times\mathcal{C}_2}, \mathsf{Lab}_{\mathcal{C}_1\times\mathcal{C}_2}, \mathsf{Mov}_{\mathcal{C}_1\times\mathcal{C}_2}, \mathsf{Edg}_{\mathcal{C}_1\times\mathcal{C}_2})$, *le produit des deux structures* $\mathcal{C}_1$ *et* $\mathcal{C}_2$, *de la manière suivante :*

– $\mathsf{Loc}_{\mathcal{C}_1\times\mathcal{C}_2} = \mathsf{Loc}_1 \times \mathsf{Loc}_2$;
– $\mathsf{AP}_{\mathcal{C}_1\times\mathcal{C}_2} = \mathsf{AP}_1 \cup \mathsf{AP}_2$;
– $\mathsf{Lab}_{\mathcal{C}_1\times\mathcal{C}_2} : \left\{ \begin{array}{rcl} \mathsf{Loc}_{\mathcal{C}_1\times\mathcal{C}_2} & \to & 2^{\mathsf{AP}_{\mathcal{C}_1\times\mathcal{C}_2}} \\ (q_1,q_2) & \mapsto & \mathsf{Lab}'(q_1,q_2) = \mathsf{Lab}_1(q_1) \cup \mathsf{Lab}_2(q_2) \end{array} \right.$;
– $\mathsf{Mov}_{\mathcal{C}_1\times\mathcal{C}_2} : \left\{ \begin{array}{rcl} \mathsf{Loc}_{\mathcal{C}_1\times\mathcal{C}_2} \times \mathsf{Agt} & \to & \mathbb{N}_{\geq 1} \\ (q_1,q_2,A_i) & \mapsto & \mathsf{Mov}'(q_1,q_2,A_i) = \mathsf{Mov}_1(q_1) \times \mathsf{Mov}_2(q_2) \end{array} \right.$;
– $\mathsf{Edg}_{\mathcal{C}_1\times\mathcal{C}_2} : \left\{ \begin{array}{rcl} \mathsf{Loc}_{\mathcal{C}_1\times\mathcal{C}_2} \times \mathbb{N}^k & \to & \mathsf{Loc}_{\mathcal{C}_1\times\mathcal{C}_2} \\ ((q_1,q_2),(m_1^{A_1},m_2^{A_1}),\ldots,(m_1^{A_k},m_2^{A_k})) & \mapsto & (\mathsf{Edg}_1(q_1,m_1^{A_1},\ldots,m_1^{A_k}), \mathsf{Edg}_2(q_2,m_2^{A_1},\ldots,m \end{array} \right.$

On généralise trivialement cette définition pour introduire la notion de produit d'une famille finie. Dans le cas d'une famille vide, on prendra le neutre de cette loi produit, qui est la structure à un élément relié à lui-même, sans label.

Remarque :
- Pour pouvoir effectuer le produit de deux structures, il est nécessaire que celles-ci aient les mêmes joueurs. Intuitivement, faire jouer un joueur sur le produit de deux structures, c'est le faire jouer sur les deux structures à la fois.
- Dans un souci de clarté, on notera par la suite, comme on l'a déjà fait dans cette définition, les mouvements de la structure produit, non pas comme des simples nombres, mais comme des couples de nombres associés chacuns à un mouvement des structures d'origine. Par exemple, considérons un état q d'une structure $\mathcal{C}$ depuis lequel le joueur 1 dispose de deux mouvements, ainsi qu'un état q' d'une structure $\mathcal{C}'$ depuis lequel le joueur 1 dispose d'un seul mouvement ; alors depuis l'état (q, q') de la structure produit $\mathcal{C} \times \mathcal{C}'$, les mouvements dont dispose le joueur 1 seront notés $(1, 1)$ et $(2, 1)$ plutôt que simplement 1 et 2.

Définition 33. *Soient* $\mathcal{C}$ *une* CGS, $A_i \in \mathsf{Agt}_{\mathcal{C}}$ *et* $n \in \mathbb{N}$. *On désigne par* $\mathcal{M}_{A_i,n} = (\mathsf{Agt}_{\mathcal{M}_{A_i,n}}, \mathsf{Loc}_{\mathcal{M}_{A_i,n}}, \mathsf{AP}_{\mathcal{M}_{A_i,n}}, \mathsf{Lab}_{\mathcal{M}_{A_i,n}}, \mathsf{Mov}_{\mathcal{M}_{A_i,n}}, \mathsf{Edg}_{\mathcal{M}_{A_i,n}})$ *la* CGS *telle que :*
- $\mathsf{Agt}_{\mathcal{M}_{A_i,n}} = \mathsf{Agt}_{\mathcal{C}}$,
- $\mathsf{Loc}_{\mathcal{M}_{A_i,n}} = \{s_{A_i,0}, ..., s_{A_i,n}\}$,
- $\mathsf{AP}_{\mathcal{M}_{A_i,n}} = \emptyset$,
- $\mathsf{Lab}_{\mathcal{M}_{A_i,n}} = \emptyset$,
- $\mathsf{Mov}_{\mathcal{M}_{A_i,n}} : \left\{ \begin{array}{lcl} \mathsf{Loc} \times \mathsf{Agt} & \rightarrow & \mathbb{N}_{\geq 1} \\ (s_{A_i,l}, P) & \mapsto & \mathsf{Mov}_{\mathcal{M}_{A_i,n}}(s_{A_i,l}, P) = \begin{cases} n+1 & si\ P = A_i \\ 1 & sinon \end{cases} \end{array} \right.$;
- $\mathsf{Edg}_{\mathcal{M}_{A_i,n}} : \left\{ \begin{array}{rcl} \mathsf{Loc} \times \mathbb{N}^k & \rightarrow & \mathsf{Loc} \\ (s_{A_i,l}, m_{A_1}, ..., m_{A_k}) & \mapsto & s_{A_i,m_{A_i}} \end{array} \right.$.

On note $\mathcal{C}_n = \prod_{A_i \in \mathsf{Agt}} \mathcal{M}_{A_i,n} \times \mathcal{C}$.

Remarque :
- Intuitivement, $\mathcal{M}_{A_i,n}$ désigne la structure de Kripke complète à $n + 1$ éléments, contrôlée par le joueur A_i, et modélise la cellule mémoire de ce joueur.
- Si $s_{A_i,l}$ est un état de $\mathcal{M}_{A_i,n}$, on notera $m(s_{A_i,l}) = l$. m est la fonction de lecture de mémoire.
- Si q désigne un état de la structure $\mathcal{C}_n$, on notera respectivement $\pi_{\mathcal{M}_{A_i,n}}(q)$ et $\pi_{\mathcal{C}}(q)$ les états de la structure $\mathcal{M}_{A_i,n}$ et de la structure $\mathcal{C}$ qui lui correspondent.
- De même, si m_{A_i} désigne un mouvement du joueur A_i sur la structure $\mathcal{C}_n$, et A_j un joueur, on notera respectivement $\pi_{\mathcal{M}_{A_j,n}}(m_{A_i})$ et $\pi_{\mathcal{C}}(m_{A_i})$ les mouvements de la structure $\mathcal{M}_{A_j,n}$ et de la structure $\mathcal{C}$ qui lui correspondent. Enfin, si m_A désigne un mouvement de la coalition A sur la struc-

ture $\mathcal{C}_n$, et A_j un joueur, on notera $\pi_{\mathcal{M}_{A_j,n}}(m_A) = (\pi_{\mathcal{M}_{A_j,n}}(m_{A_i}))_{A_i \in A}$ et $\pi_{\mathcal{C}}(m_A) = (\pi_{\mathcal{C}}(m_{A_i}))_{A_i \in A}$.

– Soit $A_j \in A, q \in \mathsf{Loc}_{\mathcal{M}_{A_j,n}}, A \subseteq \mathsf{Agt}$ et $m_A = (m_{A_i})_{A_i \in A} \in \prod_{A_i \in A}\{1, ..., \mathsf{Mov}_{\mathcal{M}_{A_j,n}}(q, A_i)\}$.

On a alors $\mathsf{Next}_{\mathcal{M}_{A_j,n}}(q, A, m_A) = \begin{cases} \{s_{A_j,m_{A_j}}\} & \text{si } A_j \in A \\ \mathsf{Loc}_{\mathcal{M}_{A_j,n}} & \text{sinon} \end{cases}$

Lemme 19. *Pour toute* CGS $\mathcal{C}, \forall n \in \mathbb{N}, \forall q \in \mathsf{Loc}_{\mathcal{C}_n}, \forall A \subseteq \mathsf{Agt}$, $\forall m_A \in \prod_{A_i \in A}\{1, ..., \mathsf{Mov}_{\mathcal{C}_n}(q, A_i)\}$, *on a :*

– $\forall A_j \in A, \pi_{\mathcal{M}_{A_j,n}}(\mathsf{Edg}_{\mathcal{C}_n}(q, m_A)) = \mathsf{Edg}_{\mathcal{M}_{A_j,n}}(\pi_{\mathcal{M}_{A_j,n}}(q), \pi_{\mathcal{M}_{A_j,n}}(m_A))$,

– $\pi_{\mathcal{C}}(\mathsf{Edg}_{\mathcal{C}_n}(q, m_A)) = \mathsf{Edg}_{\mathcal{C}}(\pi_{\mathcal{C}}(q), \pi_{\mathcal{C}}(m_A))$.

Ceci implique que $\forall q, q' \in \mathsf{Loc}_{\mathcal{C}_n}, \forall A \subseteq \mathsf{Agt}, \forall m_A \in \prod_{A_i \in A}\{1, ..., \mathsf{Mov}_{\mathcal{C}_n}(q, A_i)\}$, $q' \in \mathsf{Next}_{\mathcal{C}_n}(q, A, m_A)$ *ssi*

– $\forall A_j \in A, \pi_{\mathcal{M}_{A_j,n}}(q') \in \mathsf{Next}_{\mathcal{M}_{A_j,n}}(\pi_{\mathcal{M}_{A_j,n}}(q), A, \pi_{\mathcal{M}_{A_j,n}}(m_A))$,

– $\pi_{\mathcal{C}}(q') \in \mathsf{Next}_{\mathcal{C}}(\pi_{\mathcal{C}}(q), A, \pi_{\mathcal{C}}(m_A))$.

Et finalement, par la remarque précédente : $\forall q, q' \in \mathsf{Loc}_{\mathcal{C}_n}, \forall A \subseteq \mathsf{Agt}, \forall m_A \in \prod_{A_i \in A}\{1, ..., \mathsf{Mov}_{\mathcal{C}_n}(q, A_i)\}$, $q' \in \mathsf{Next}_{\mathcal{C}_n}(q, A, m_A)$ *ssi*

– $\forall A_j \in A, \pi_{\mathcal{M}_{A_j,n}}(q') = s_{A_j,\pi_{\mathcal{M}_{A_j,n}}(m_{A_j})}$,

– $\pi_{\mathcal{C}}(q') \in \mathsf{Next}_{\mathcal{C}}(\pi_{\mathcal{C}}(q), A, \pi_{\mathcal{C}}(m_A))$.

Définition 34. *Soient* $\mathcal{C}$ *une* CGS, $A_i \subseteq \mathsf{Agt}_{\mathcal{C}}$ *et* $n \in \mathbb{N}$. *On définit*

$$\alpha_{A_i,n} : \begin{cases} \Sigma'^{n}_{A_i,\mathcal{C}} & \to & \Sigma^0_{A_i,\mathcal{C}_n} \\ F & \mapsto & \alpha_{A_i,n}(F) : \begin{cases} \mathsf{Loc}_{\mathcal{C}_n} & \to & \mathbb{N}^{|\mathsf{Agt}|+1} \\ q & \mapsto & m_{A_i}(q) \end{cases} \end{cases},$$

où :

– $\forall A_j \in A, \pi_{\mathcal{M}_{A_j,n}}(m_{A_i}(q)) = \begin{cases} \pi_1(F(m(\pi_{\mathcal{M}_{A_i,n}}(q)), \pi_{\mathcal{C}}(q))) & \textit{si } A_j = A_i \\ 1 & \textit{sinon} \end{cases}$

– $\pi_{\mathcal{C}}(m_{A_i}(q)) = \pi_2(F(m(\pi_{\mathcal{M}_{A_i,n}}(q)), \pi_{\mathcal{C}}(q)))$

Si $A \subseteq \mathsf{Agt}$, *on définit l'application*

$$\alpha_{A,n} : \begin{cases} \Sigma'^{n}_{A,\mathcal{C}} & \to & \Sigma^0_{A,\mathcal{C}_n} \\ F = (F_{A_i})_{A_i \in A} & \mapsto & \alpha_{A,n}(F) = (\alpha_{A_i,n}(F_{A_i}))_{A_i \in A} \end{cases}.$$

Lemme 20. *Pour toute* CGS $\mathcal{C}, \forall n \in \mathbb{N}, \forall A \subseteq \mathsf{Agt}_{\mathcal{C}}, \forall F \in \Sigma'^{n}_{A,\mathcal{C}}, \forall s \in \mathsf{Loc}_{\mathcal{C}}$,
$\mathsf{Out}_{\mathcal{C}}(s, F) = \pi_{\mathcal{C}}(\mathsf{Exec}_{(\mathcal{C}_n,\alpha_{A,n}(F))}(s_{A_1,0}, ..., s_{A_k,0}, s))$.

Preuve : Soient $\mathcal{C}$ une *CGS*, $A \subseteq \mathsf{Agt}_{\mathcal{C}}, F = (F_{A_i})_{A_i \in A} \in \Sigma'^{n}_{A,\mathcal{C}}$ et $s \in \mathsf{Loc}_{\mathcal{C}}$.

On sait par le lemme 26 que $\mathsf{Exec}_{(\mathcal{C}_n,\alpha_{A,n}(F))}(s_{A_1,0}, ..., s_{A_k,0}, s) = \mathsf{Out}_{\mathcal{C}_n}((s_{A_1,0}, ..., s_{A_k,0}, s), \alpha_{A,n}(F))$.

Soit $\lambda \in \mathsf{Out}_{\mathcal{C}}(s, F)$, et $\lambda' \in \mathsf{Out}_{A,\mathcal{C}}(s, F)$ tel que $\pi_{|A|+1}(\lambda) = \lambda'$. Soit maintenant $\lambda'' \in \mathsf{Exec}_{\mathcal{C}_n}(s_{A_1,0}, ..., s_{A_k,0}, s)$ tel que :

– $\forall A_i \in A, \forall k \in \mathbb{N}, \pi_{\mathcal{M}_{A_i,n}}(\lambda''[k]) = s_{A_i,\pi_{A_i}(\lambda'[k])}$,

– $\forall A_i \notin A, \forall k \in \mathbb{N}, \pi_{\mathcal{M}_{A_i,n}}(\lambda''[k]) = s_{A_i,0}$,

– $\forall k \in \mathbb{N}, \pi_{\mathcal{C}}(\lambda''[k]) = \pi_{|A|+1}(\lambda'[k])$.

Il reste à montrer que $\lambda'' \in \mathsf{Out}_{\mathcal{C}_n}((s_{A_1,0},...,s_{A_k,0},s),\alpha_{A,n}(F))$. Il est évident que $\lambda''[0] = (s_{A_1,0},...,s_{A_k,0},s)$. Soit $k \in \mathbb{N}$. On doit montrer que

$\lambda''[k+1] \in \mathsf{Next}_{\mathcal{C}_n}(\lambda''[k], A, (\alpha_{A_i,n}(F_{A_i})(\lambda''[k]))_{A_i\in A})$, ce qui revient par le lemme 50 à montrer que :

- $\forall A_j \in A, \pi_{\mathcal{M}_{A_j,n}}(\lambda''[k+1]) = s_{A_j,\pi_{\mathcal{M}_{A_j,n}}(\alpha_{A_j,n}(F_{A_j})(\lambda''[k]))}$
- $\pi_{|A|+1}(\lambda'[k+1]) \in \mathsf{Next}_{\mathcal{C}}(\pi_{|A|+1}(\lambda'[k]), A, (\pi_{\mathcal{C}}(\alpha_{A_i,n}(F_{A_i})(\lambda''[k])))_{A_i\in A})$.

Premier point : soit $A_j \in A$. On a :

$\pi_{\mathcal{M}_{A_j,n}}(\lambda''[k+1]) = s_{A_j,\pi_{A_j}(\lambda'[k+1])}$, et

$\pi_{\mathcal{M}_{A_j,n}}(\alpha_{A_j,n}(F_{A_j})(\lambda''[k])) =$, par définition de $\alpha_{A_j,n}$

$\pi_1(F_{A_j}(\pi_{A_j}(\lambda'[k]), \pi_{|A|+1}(\lambda'[k]))) =$, puisque $\lambda' \in \mathsf{Out}_{A,\mathcal{C}}(s,F)$,

$\pi_{A_j}(\lambda'[k+1])$.

Deuxième point : on a $\forall A_i \in A, \pi_{\mathcal{C}}(\alpha_{A_i,n}(F_{A_i})(\lambda''[k])) = \pi_2(F_{A_i}(m(\pi_{\mathcal{M}_{A_i,n}}(\lambda''[k])), \pi_{\mathcal{C}}(\lambda''[k])))$. De plus, $m(\pi_{\mathcal{M}_{A_i,n}}(\lambda''[k])) = \pi_{A_i(\lambda'[k])}$ et $\pi_{\mathcal{C}}(\lambda''[k]) = \pi_{|A|+1}(\lambda'[k])$. Le fait que $\lambda' \in \mathsf{Out}_{A,\mathcal{C}}(s,F)$ guarantit donc bien ce que l'on veut.

Soit $\lambda \in \pi_{\mathcal{C}}(\mathsf{Out}_{\mathcal{C}_n}((s_{A_1,0},...,s_{A_k,0},s),\alpha_{A,n}(F)))$, et $\lambda' \in \mathsf{Out}_{\mathcal{C}_n}((s_{A_1,0},...,s_{A_k,0},s),\alpha_{A,n}(F))$ tel que $\pi_{\mathcal{C}}(\lambda') = \lambda$. Soit maintenant λ'' la suite d'éléments de $\mathbb{N}^{|A|} \times \mathsf{Loc}_{\mathcal{C}}$ telle que :

- $\forall A_i \in A, \forall k \in \mathbb{N}, \pi_{A_i}(\lambda''[k]) = m(\pi_{\mathcal{M}_{A_i,n}}(\lambda'[k]))$,
- $\forall k \in \mathbb{N}, \pi_{|A|+1}(\lambda''[k]) = \pi_{\mathcal{C}}(\lambda'[k])$.

Il reste à montrer que $\lambda'' \in \mathsf{Out}_{A,\mathcal{C}}(s,F)$.

Soit $k \in \mathbb{N}$. Puisque $\lambda' \in \mathsf{Out}_{\mathcal{C}_n}((s_{A_1,0},...,s_{A_k,0},s),\alpha_{A,n}(F))$, on a $\lambda'[k+1] \in \mathsf{Next}_{\mathcal{C}_n}(\lambda'[k], A, (\alpha_{A_i,n}(F_{A_i})(\lambda'[k]))_{A_i\in A})$.

Soit $A_i \in A$. On a

$\pi_{A_i}(\lambda''[k+1]) =$, par définition de λ''

$m(\pi_{\mathcal{M}_{A_i,n}}(\lambda'[k+1])) =$, par la première partie du lemme 50

$\pi_{\mathcal{M}_{A_i,n}}(\alpha_{A_i,n}(F_{A_i})(\lambda'[k])) =$, par définition de $\alpha_{A_i,n}$

$\pi_1(F_{A_i}(m(\pi_{\mathcal{M}_{A_i,n}}(\lambda'[k])), \pi_{\mathcal{C}}(\lambda'[k]))) =$, par définition de λ''

$\pi_1(F_{A_i}(\pi_{A_i}(\lambda''[k]), \pi_{|A|+1}(\lambda''[k])))$.

Enfin, $\pi_{|A|+1}(\lambda''[k+1]) = \pi_{\mathcal{C}}(\lambda'[k+1])$ par définition de λ'', donc par la deuxième partie du lemme 50,

$\pi_{|A|+1}(\lambda''[k+1]) \in \mathsf{Next}_{\mathcal{C}}(\pi_{\mathcal{C}}(\lambda'[k]), A, (\pi_{\mathcal{C}}(\alpha_{A_i,n}(F_{A_i})(\lambda'[k])))_{A_i\in A})$

Or $\pi_{\mathcal{C}}(\lambda'[k]) = \pi_{|A|+1}(\lambda''[k])$, et par définition de $\alpha_{A_i,n}$, on a

$\forall A_i \in A, \pi_{\mathcal{C}}(\alpha_{A_i,n}(F_{A_i})(\lambda'[k])) = \pi_2(F_{A_i}(m(\pi_{\mathcal{M}_{A_i,n}}(\lambda'[k])), \pi_{\mathcal{C}}(\lambda'[k])))$.

Comme de plus $m(\pi_{\mathcal{M}_{A_i,n}}(\lambda'[k])) = \pi_{A_i}(\lambda''[k])$ et $\pi_{\mathcal{C}}(\lambda'[k]) = \pi_{|A|+1}(\lambda''[k])$, on a

$\mathsf{Next}_{\mathcal{C}}(\pi_{\mathcal{C}}(\lambda'[k]), A, (\pi_{\mathcal{C}}(\alpha_{A_i,n}(F_{A_i})(\lambda'[k])))_{A_i\in A}) =$
$\mathsf{Next}_{\mathcal{C}}(\pi_{|A|+1}(\lambda''[k]), A, (\pi_2(F_{A_i}(\pi_{A_i}(\lambda''[k]), \pi_{|A|+1}(\lambda''[k]))))_{A_i\in A})$.

□

Soit $n \in \mathbb{N}$ fixé. Voici un algorithme de model-checking pour la logique $\mathcal{L}_n^{'**}$.

On définit par induction sur $\mathcal{L}_{n,p}^{'**}$ l'application σ à valeurs dans LTL (avec toutefois plus de propositions atomiques) telle que :

- Si $\varphi_p = a \in \mathsf{AP}$, alors $\sigma(\varphi_p) = a$,
- Si $\varphi_p = \neg\psi$, alors $\sigma(\varphi_p) = \neg\sigma(\psi)$,
- Si $\varphi_p = \psi_1 \vee \psi_2$, alors $\sigma(\varphi_p) = \sigma(\psi_1) \vee \sigma(\psi_2)$,

Algorithm 2 MC $\mathcal{L}'^{**}_n$

Require: une *CGS* $\mathcal{C}$, une stratégie $F \in \Sigma'^{n}_{A,\mathcal{C}}$, un état $s_0 \in \mathsf{Loc}_{\mathcal{C}}$ et une formule θ_p de $\mathcal{L}'^{**}_{n,p}$

Ensure: OUI ssi $\forall \lambda \in \mathsf{Out}_{\mathcal{C}}(s_0, F), (\mathcal{C}, \lambda) \models_F \theta_p$

$\mathcal{M} := (\mathcal{C}_n, \alpha_{A,n}(F))$

$\Phi_1 := \{\psi \in Sub(\theta_p) \mid \ rq(\psi) = rq(\theta_p)$ et $\exists B \subseteq \mathsf{Agt}, \exists \psi' \in \mathcal{L}'^{**}_n, \psi = \lessdot B \gtrdot'_n \psi'\}$

$\Phi_2 := \{\psi \in Sub(\theta_p) \mid \ rq(\psi) = rq(\theta_p)$ et $\exists B \subseteq \mathsf{Agt}, \exists \psi' \in \mathcal{L}'^{**}_n, \psi = \gtrdot B \lessdot'_n \psi'\}$

for $\psi \in \Phi_1 \cup \Phi_2$ **do**
- $\mathsf{AP}_{\mathcal{M}} := \mathsf{AP}_{\mathcal{M}} \cup \{a_\psi\}$
- **for** $s \in \mathsf{Loc}_{\mathcal{C}}$ **do**
  - **if** $\psi \in \Phi_1$ **then**
    - **for** $F' \in \Sigma'^{n}_{B,\mathcal{C}}$ **do**
      - **if** MC $\mathcal{L}'^{**}_n(\mathcal{C}, F' \circ'_n F, s, \psi')$ **then**
        - $\mathsf{Lab}_{\mathcal{M}}(s_{A_1,0}, ..., s_{A_k,0}, s) := \mathsf{Lab}_{\mathcal{M}}(s_{A_1,0}, ..., s_{A_k,0}, s) \cup \{a_\psi\}$
      - **end if**
    - **end for**
  - **else**
    - **if** MC $\mathcal{L}'^{**}_n(\mathcal{C}, F|'_{n,A\setminus B}, s, \psi')$ **then**
      - $\mathsf{Lab}_{\mathcal{M}}(s_{A_1,0}, ..., s_{A_k,0}, s) := \mathsf{Lab}_{\mathcal{M}}(s_{A_1,0}, ..., s_{A_k,0}, s) \cup \{a_\psi\}$
    - **end if**
  - **end if**
- **end for**

end for

if MC LTL $(\mathcal{M}, (s_{A_1,0}, ..., s_{A_k,0}, s_0), \mathbf{A}\sigma(\theta_p))$ **then**
- retourner OUI

else
- retourner NON

end if

– Si $\varphi_p = \lessdot B \gtrdot'_n \ \psi$, alors $\sigma(\varphi_p) = a_{\varphi_p}$,
– Si $\varphi_p = \gtrdot B \lessdot'_n \ \psi$, alors $\sigma(\varphi_p) = a_{\varphi_p}$,
– Si $\varphi_p = \mathbf{X}\,\psi$, alors $\sigma(\varphi_p) = \mathbf{X}\,\sigma(\psi)$,
– Si $\varphi_p = \psi_1 \,\mathbf{U}\, \psi_2$, alors $\sigma(\varphi_p) = \sigma(\psi_1)\,\mathbf{U}\,\sigma(\psi_2)$.

On va montrer que l'algorithme 2 fait bien ce que l'on affirme. On désigne par $\mathcal{M}_{f,(\mathcal{C},F,\theta_p)}$ la structure contenue dans la variable $\mathcal{M}$ en fin d'exécution de l'algorithme 2 sur l'entrée $(\mathcal{C}, F, s_0, \theta_p)$. Cette notation est justifiée par le fait que la valeur de s_0 n'importe pas.

Il est clair que pour toute entrée $(\mathcal{C}, F, s_0, \theta_p)$ de l'algorithme 2, les structures $(\mathcal{C}_n, \alpha_{A,n}(F))$ et $\mathcal{M}_{f,(\mathcal{C},F,\theta_p)}$ ont mêmes états et mêmes transitions (elles ne diffèrent que par leurs propositions atomiques et leurs labels), donc $\forall s \in \mathsf{Loc}_{\mathcal{C}_n}, \mathsf{Exec}_{(\mathcal{C}_n,\alpha_{A,n}(F))}(s) = \mathsf{Exec}_{\mathcal{M}_{f,(\mathcal{C},F,\theta_p)}}(s)$.

Soit une *CGS* $\mathcal{C}$. On va montrer par induction que pour toute formule $\varphi_p \in \mathcal{L}'^{**}_{n,p}$, on a :

$\forall A \subseteq \mathsf{Agt}, \forall F \in \Sigma'^{n}_{A,\mathcal{C}}, \forall s \in \mathsf{Loc}_{\mathcal{C}}, \forall \lambda \in \mathsf{Exec}_{(\mathcal{C}_n,\alpha_{A,n}(F))}(s_{A_1,0}, ..., s_{A_k,0}, s)$
$(\mathcal{C}, \pi_{\mathcal{C}}(\lambda)) \models_F \varphi_p$ ssi $(\mathcal{M}_{f,(\mathcal{C},F,\varphi_p)}, \lambda) \models \sigma(\varphi_p)$.

– Si $\varphi_p = a \in \mathsf{AP}$. Soient $A \subseteq \mathsf{Agt}, F \in \Sigma'^{n}_{A,\mathcal{C}}, s \in \mathsf{Loc}, \lambda \in \mathsf{Exec}_{(\mathcal{C}_n,\alpha_{A,n}(F))}(s)$.
  On doit montrer que $a \in \mathsf{Lab}_{\mathcal{M}_{f,(\mathcal{C},F,\varphi_p)}}(s_{A_1,0}, ..., s_{A_k,0}, s)$ ssi $a \in \mathsf{Lab}_{\mathcal{C}}(s)$.
  Ceci est vrai, puisque la transformation de $\mathcal{C}_n$ en $\mathcal{M}_{f,(\mathcal{C},F,\varphi_p)}$ ne modifie pas les labels par des propositions atomiques de $\mathcal{C}$,
– Si $\varphi_p = \neg\psi$, alors c'est trivial puisque l'on cherche à montrer une équivalence,
– Si $\varphi_p = \psi_1 \vee \psi_2$, alors c'est trivial aussi,
– Si $\varphi_p = \lessdot B \gtrdot'_n \psi$. Soient $A \subseteq \mathsf{Agt}, F \in \Sigma'^{n}_{A,\mathcal{C}}, s \in \mathsf{Loc}_{\mathcal{C}}, \lambda \in \mathsf{Exec}_{(\mathcal{C}_n,\alpha_{A,n}(F))}(s_{A_1,0}, ..., s_{A_k,0}, s)$.
  On a la suite d'équivalences :
  $(\mathcal{M}_{f,(\mathcal{C},F,\varphi_p)}, \lambda) \models \sigma(\lessdot B \gtrdot'_n \ \psi)$ ssi
  $(\mathcal{M}_{f,(\mathcal{C},F,\varphi_p)}, \lambda) \models a_{\varphi_p}$ ssi
  $a_{\varphi_p} \in \mathsf{Lab}_{\mathcal{M}_{f,(\mathcal{C},F,\varphi_p)}}(s_{A_1,0}, ..., s_{A_k,0}, s)$ ssi
  on a pu trouver une stratégie $F' \in \Sigma'^{n}_{B,\mathcal{C}}$ telle que l'algorithme 2 retourne OUI sur l'entrée $(\mathcal{C}, F' \circ'_n F, s, \psi)$ ssi (par définition de l'algorithme 2)
  $\exists F' \in \Sigma'^{n}_{B,\mathcal{C}}$, MC LTL $(\mathcal{M}_{f,(\mathcal{C},F'\circ'_n F,\psi)}, (s_{A_1,0}, ..., s_{A_k,0}, s), \sigma(\psi))$ ssi
  $\exists F' \in \Sigma'^{n}_{B,\mathcal{C}}, \forall \lambda' \in \mathsf{Exec}_{(\mathcal{C}_n,\alpha_{A,n}(F'\circ'_n F))}(s_{A_1,0}, ..., s_{A_k,0}, s), (\mathcal{M}_{f,(\mathcal{C},F'\circ'_n F,\psi)}, \lambda') \models$
  $\sigma(\psi)$ ssi (par HI appliquée à ψ)
  $\exists F' \in \Sigma'^{n}_{B,\mathcal{C}}, \forall \lambda' \in \mathsf{Exec}_{(\mathcal{C}_n,\alpha_{A,n}(F'\circ'_n F))}(s_{A_1,0}, ..., s_{A_k,0}, s), (\mathcal{C}, \pi_{\mathcal{C}}(\lambda')) \models_{F'\circ'_n F}$
  ψ ssi, par le lemme 52,
  $\exists F' \in \Sigma'^{n}_{B,\mathcal{C}}, \forall \lambda' \in \mathsf{Out}(s, F' \circ'_n F), (\mathcal{C}, \lambda') \models_{F'\circ'_n F} \psi$ ssi
  $(\mathcal{C}, s) \models_F \ \lessdot B \gtrdot'_n \ \psi$ ssi
  $(\mathcal{C}, \pi_{\mathcal{C}}(\lambda)) \models_F \ \lessdot B \gtrdot'_n \ \psi$.
– Si $\varphi_p = \gtrdot B \lessdot'_n \psi$. Soient $A \subseteq \mathsf{Agt}, F \in \Sigma'^{n}_{A,\mathcal{C}}, s \in \mathsf{Loc}_{\mathcal{C}}, \lambda \in \mathsf{Exec}_{(\mathcal{C}_n,\alpha_{A,n}(F))}(s_{A_1,0}, ..., s_{A_k,0}, s)$.
  On a la suite d'équivalences :
  $(\mathcal{M}_{f,(\mathcal{C},F,\varphi_p)}, \lambda) \models \sigma(\gtrdot B \lessdot'_n \ \psi)$ ssi
  $(\mathcal{M}_{f,(\mathcal{C},F,\varphi_p)}, \lambda) \models a_{\varphi_p}$ ssi
  $a_{\varphi_p} \in \mathsf{Lab}_{\mathcal{M}_{f,(\mathcal{C},F,\varphi_p)}}(s_{A_1,0}, ..., s_{A_k,0}, s)$ ssi
  l'algorithme 2 retourne OUI sur l'entrée $(\mathcal{C}, F|'_{n,A\setminus B}, s, \psi)$ ssi (par définition de l'algorithme 2)

MC LTL $(\mathcal{M}_{f,(\mathcal{C},F|'_{n,A\setminus B},\psi)}, (s_{A_1,0},...,s_{A_k,0},s), \sigma(\psi))$ ssi
$\forall \lambda' \in \mathsf{Exec}_{(\mathcal{C},F|'_{n,A\setminus B})}(s_{A_1,0},...,s_{A_k,0},s), (\mathcal{M}_{f,(\mathcal{C},F|'_{n,A\setminus B},\psi)}, \lambda') \models \sigma(\psi)$ ssi
(par HI appliquée à ψ)
$\forall \lambda' \in \mathsf{Exec}_{(\mathcal{C},F|'_{n,A\setminus B})}(s_{A_1,0},...,s_{A_k,0},s), (\mathcal{C}, \pi_{\mathcal{C}}(\lambda')) \models_{F|'_{n,A\setminus B}} \psi$ ssi, par le lemme 52,
$\forall \lambda' \in \mathsf{Out}(s, F|'_{n,A\setminus B}), (\mathcal{C}, \lambda') \models_{F|'_{n,A\setminus B}} \mathbf{A}\psi$ ssi
$(\mathcal{C}, s) \models_F \gg B \lessdot'_n \psi$ ssi
$(\mathcal{C}, \pi_{\mathcal{C}}(\lambda)) \models_F \gg B \lessdot'_n \psi$.

– Si $\varphi_p = \mathbf{X}\,\psi$. Soient $A \subseteq \mathsf{Agt}, F \in \Sigma'^{n}_{A,\mathcal{C}}, s \in \mathsf{Loc}_{\mathcal{C}}, \lambda \in \mathsf{Exec}_{(\mathcal{C}_n,\alpha_{A,n}(F))}(s_{A_1,0},...,s_{A_k,0},s)$.
On a la suite d'équivalences :
$(\mathcal{M}_{f,(\mathcal{C},F,\varphi_p)}, \lambda) \models \sigma(\mathbf{X}\,\psi)$ ssi
$(\mathcal{M}_{f,(\mathcal{C},F,\varphi_p)}, \lambda) \models \mathbf{X}\,\sigma(\psi)$ ssi
$(\mathcal{M}_{f,(\mathcal{C},F,\varphi_p)}, \lambda[1,\infty]) \models \sigma(\psi)$ ssi (puisque l'on voit facilement $\mathcal{M}_{f,(\mathcal{C},F,\varphi_p)} = \mathcal{M}_{f,(\mathcal{C},F,\psi)}$)
$(\mathcal{M}_{f,(\mathcal{C},F,\psi)}, \lambda[1,\infty]) \models \sigma(\psi)$ ssi (par HI)
$((\mathcal{C},F), \pi_{\mathcal{C}}(\lambda[1,\infty])) \models \psi$ ssi
$((\mathcal{C},F), \pi_{\mathcal{C}}(\lambda)) \models \mathbf{X}\,\psi$.
– Si $\varphi_p = \psi_1\,\mathbf{U}\,\psi_2$. Soient $A \subseteq \mathsf{Agt}, F \in \Sigma'^{n}_{A,\mathcal{C}}, s \in \mathsf{Loc}_{\mathcal{C}}, \lambda \in \mathsf{Exec}_{(\mathcal{C}_n,\alpha_{A,n}(F))}(s_{A_1,0},...,s_{A_k,0},s)$.
On a la suite d'équivalences :
$(\mathcal{M}_{f,(\mathcal{C},F,\varphi_p)}, \lambda) \models \sigma(\psi_1\,\mathbf{U}\,\psi_2)$ ssi
$(\mathcal{M}_{f,(\mathcal{C},F,\varphi_p)}, \lambda) \models \sigma(\psi_1)\,\mathbf{U}\,\sigma(\psi_2)$ ssi
$\exists i \geq 0.\ (\mathcal{M}_{f,(\mathcal{C},F,\theta_p)}, \lambda[i,\infty]) \models \sigma(\psi_2)$ et $\forall 0 \leq j < i.\ (\mathcal{M}_{f,(\mathcal{C},F,\theta_p)}, \lambda[j,\infty]) \models \sigma(\psi_1)$ ssi (puisque l'on voit facilement $\mathcal{M}_{f,(\mathcal{C},F,\varphi_p)} = \mathcal{M}_{f,(\mathcal{C},F,\psi_1)} = \mathcal{M}_{f,(\mathcal{C},F,\psi_2)}$)
$\exists i \geq 0.\ (\mathcal{M}_{f,(\mathcal{C},F,\psi_2)}, \lambda[i,\infty]) \models \sigma(\psi_2)$ et $\forall 0 \leq j < i.\ (\mathcal{M}_{f,(\mathcal{C},F,\psi_1)}, \lambda[j,\infty]) \models \sigma(\psi_1)$ ssi (par HI)
$\exists i \geq 0.\ ((\mathcal{C},F), \pi_{\mathcal{C}}(\lambda[i,\infty])) \models \psi_2$ et $\forall 0 \leq j < i.\ ((\mathcal{C},F), \pi_{\mathcal{C}}(\lambda[j,\infty])) \models \psi_1$ ssi
$((\mathcal{C},F), \pi_{\mathcal{C}}(\lambda)) \models \psi_1\,\mathbf{U}\,\psi_2$.

On sait que l'algorithme 2 retourne OUI sur l'entrée $(\mathcal{C}, F, s_0, \theta_p)$ ssi $((\mathcal{M}_{f,(\mathcal{C},F,\theta_p)}), (s_{A_1,0},...,s_{A_k,0},s_0)) \models \mathbf{A}\sigma(\theta_p)$. La propriété que l'on vient d'établir, le lemme 52 et le fait que $(\mathcal{M}_{f,(\mathcal{C},F,\theta_p)})$ et $(\mathcal{C}_n, \alpha_{A,n}(F))$ aient les mêmes exécutions permettent de prouver que l'algorithme 2 retourne OUI sur l'entrée $(\mathcal{C}, F, s_0, \theta_p)$ ssi $\forall \lambda \in \mathsf{Out}_{\mathcal{C}}(s_0, F), (\mathcal{C}, \lambda) \models_F \theta_p$, comme on l'avait annoncé. Cet algorithme est bien un algorithme de model-checking pour $\mathcal{L}'^{**}_n$: étant donnés une *CGS* $\mathcal{C}$, un état s_0 de $\mathcal{C}$ et une formule θ de $\mathcal{L}'^{**}_n$, si l'on veut savoir si $(\mathcal{C}, s_0) \models \theta$, il suffit de lancer l'algorithme 2 sur l'entrée $(\mathcal{C}, \emptyset, s_0, \theta)$. En effet, il retournera OUI ssi $\forall \lambda \in \mathsf{Out}_{\mathcal{C}}(s_0, \emptyset), (\mathcal{C}, \lambda) \models \theta$, c'est-à-dire ssi $(\mathcal{C}, s_0) \models \theta$ car θ est une formule d'état. Cet algorithme est visiblement dans $\mathsf{EXPSPACE}$, toutefois si l'on fixe le nombre de joueur, il devient PSPACE. Il reste à établir la complexité précise de ce problème.

Un point important serait dans l'avenir d'exhiber les relations d'équivalence sur les *CGS* associées aux nouvelles logiques définies dans ce rapport (la relation étant : vérifier les mêmes formules de cette logique). Une autre idée interressante serait certainement de s'écarter à nouveau des principes d'emboitement et d'oubli de stratégies, et de revenir au cadre traditionnel d'ATL^* afin d'y exploiter la définition de stratégie à mémoire bornée. Un objectif pourrait être de classer les formules de chemin en fonction de la mémoire minimale nécéssaire à une stratégie pour les garantir.

Bibliographie

[1] Rajeev Alur, Thomas A. Henzinger, Orna Kupferman, Alternating-time temporal logic, ACM2002

[2] François Laroussinie, Nicolas Markey, Ghassan Oreiby, Expressiveness and complexity of ATL, fossacs 2007

[3] Berard, B., Bidoit, M., Finkel, A., Laroussinie, F., Petit, A., Petrucci, L., Schnoebelen, P., Systems and Software Verification, livre, 1999

[4] Christos Papadimitriou, Computational Complexity, livre, 1994