Une logique épistémique modeste basée sur les observations d'ordre supérieur

Andreas Herzig **Emiliano Lorini** **Faustine Maffre**

University of Toulouse, IRIT
118, Route de Narbonne
F-31062 Toulouse, France

{Andreas.Herzig, Emiliano.Lorini, Faustine.Maffre}@irit.fr

Résumé

Nous présentons une logique épistémique dynamique basée sur ce qu'un agent peut observer, y compris les observations jointes et l'observation de ce que les autres observent, généralisant ainsi les logiques ECL-PC(PO) et LRC de van der Hoek, Wooldridge et al. où les variables propositionnelles que chaque agent observe sont connaissance commune. Dans notre logique, les propositions ainsi que la visibilité de ces propositions peuvent être modifiées par des programmes d'affectation. Nous montrons comment les opérateurs épistémiques peuvent alors être interprétés. Nous proposons une axiomatisation adéquate et complète et prouvons que le problème de satisfiabilité est PSpace-complet. Enfin, nous montrons comment les annonces publiques et privées peuvent être exprimées dans notre logique.

1 Introduction

Dans son livre pionnier de 1962 [11], Hintikka proposa d'analyser les phrases de la forme "l'agent i sait φ dans un monde possible w" par "φ est vrai dans tous les mondes que i ne peut pas distinguer de w". De telles logiques épistémiques ont recours à la sémantique des mondes possibles de Kripke. Elles sont été popularisées en informatique par Fagin et al. dans les années 90 [4].

Au cours des dernières années, plusieurs auteurs ont étudié comment la logique épistémique pourrait plutôt se baser sur la notion de visibilité (ou observabilité) de variables propositionnelles. Les approches les plus remarquables sont Epistemic Coalition Logic of Propositional Control with Partial Observability ECL-PC(PO) [17] et Logic of Revelation and Concealment LRC [16]. Ces logiques trouvent leur origine dans les langages utilisés dans des vérificateurs de modèle tels que MOCHA pour la description des systèmes distribués [2]. L'idée de base est que la phrase "l'agent i sait que le fait atomique p est vrai" peut être réduit à (1) p et vrai et (2) l'agent i observe la valeur de vérité de p. De la même façon, "l'agent i sait que p est faux" peut être réduit à (1) p et faux et (2) l'agent i observe la valeur de vérité de p. On suppose donc que chaque agent a un ensemble de faits atomiques (alias variables propositionnelles) qu'il peut observer, dans le sens où il connaît leurs valeurs de vérité. A l'inverse, toute combinaison des valeurs de vérité des variables non observées est possible pour l'agent. Ces informations d'observabilité permettent de reconstruire la sémantique de Hintikka : une relation d'accessibilité kripkéenne $\sim_i$ entre les mondes possibles w et w' peut être associée à l'agent i par la définition suivante : $w \sim_i w'$ si et seulement si chaque variable observée par i dans w a la même valeur de vérité dans w et dans w'. Cela permet de donner la sémantique de phrases de la forme "i sait que φ" où φ est une formule complexe arbitraire (et pas seulement une variable propositionnelle ou sa négation).

À la fois ECL-PC(PO) et LRC, et plus généralement toutes les logiques épistémiques existantes basées sur la visibilité dans la littérature, supposent que ce que chaque agent peut voir est connaissance commune. C'est une hypothèse forte que nous allons relâcher dans ce papier : nous allons permettre des situations où l'agent i voit la variable p, mais l'agent j ne voit pas si i voit p. Ceci permet de modéliser de façon naturelle des scénarios tels que le problème de bavardage

où six amis échangent leurs secrets via des appels téléphoniques ou le problème des deux généraux où deux agents doivent se coordonner via une communication asynchrone non fiable. Dans notre approche, nous modélisons les informations de visibilité non seulement en termes de variables associées à des agents (comme cela est fait dans ECL-PC(PO) and LRC), mais plus généralement en termes de variables associées à des *séquences* d'agents. Syntaxiquement, nous représentons ceci au moyen de formules atomiques que nous appelons des *atomes de visibilité*. Ils sont de la forme $S_{i_1} S_{i_2} ... S_{i_n} p$, où p est une variable propositionnelle, et $i_1, i_2, ..., i_n$ sont des agents. Si n=0, alors on a seulement une variable propositionnelle. Si n=1, l'atome $S_{i_1} p$ se lit "l'agent i_1 voit la valeur de la variable p", et si n=2, l'observation du second ordre $S_{i_1} S_{i_2} p$ se lit "l'agent i_1 voit si i_2 voit la valeur de p"; etc.

Notre sémantique est basée sur des *évaluations*, qui sont simplement des ensembles d'atomes de visibilité. Afin de garantir l'introspection positive et négative, nous devons veiller à ce que l'agent soit toujours conscient des variables propositionnelles qu'il voit : pour chaque agent i et chaque variable propositionnelle p, nous exigeons que $S_i\, S_i\, p$ soit dans chaque évaluation. Plus généralement, on considère que l'évaluation V est introspective si elle contient chaque atome de visibilité ayant deux S_i consécutifs. Ainsi, dans une évaluation introspective, l'agent est conscient de ce qu'il voit, et tout agent voit cela, et tout agent voit que tout agent voit cela, etc. Les informations de visibilité permettent d'interpréter les opérateurs épistémiques : pour les variables propositionnelles p, la formule $K_i p$ est vraie dans une évaluation V si V contient à la fois p et $S_i\, p$. Plus généralement, la condition de vérité pour $K_i \varphi$ est basée sur une relation entre les évaluations qui peut être construite grâce à nos atomes de visibilité : $V \sim_i V'$ si chaque atome que i voit dans V a la même valeur de vérité dans V et dans V'. Alors que les relations $\sim_i$ sont toujours réflexives, elles sont symétriques et transitives — et sont donc des relations d'équivalence — sur l'ensemble des évaluations introspectives seulement.

Une autre nouveauté de notre approche par rapport aux logiques épistémiques existantes fondées sur la visibilité est que notre langage comprend une formule atomique spéciale pour l'attention conjointe, de la forme JSp, qui signifie "tous les agents voient conjointement la valeur de p". Métaphoriquement, l'attention conjointe peut être comprise comme un contact visuel entre les agents lors de l'observation de quelque chose. Tout comme la visibilité individuelle, nous permettons les observations jointes d'ordre supérieur, et ajoutons une contrainte sur les évaluations qui garantit l'introspection de la connaissance commune. Nous exigeons en outre que la visibilité conjointe implique la visibilité individuelle en imposant que $S_i\, p \in V$ lorsque $JS\, p \in V$. Ceci permet d'interpréter un opérateur de connaissance commune CK modal de la même façon que celui de la connaissance individuelle.

Comme plusieurs propositions existantes, nous nous inspirons des logiques épistémiques dynamiques DEL [18] et ajoutons un côté dynamique à notre logique épistémique basée sur l'observation. Plus précisément, nous adaptons la logique LRC de van der Hoek et al. qui dispose de deux opérations de mise à jour modifiant la visibilité : révéler et cacher la valeur d'une variable à un agent. Ces deux primitives ne peuvent toutefois pas être utilisées en l'état car la mise à jour naïve d'une évaluation peut la rendre non introspective. Nous évitons ceci par une définition appropriée de la mise à jour d'une évaluation. Nous utilisons les mêmes programmes d'affectation que ceux de Dynamic Logic of Propositional Assignments DL-PA [10] [3], qui est un dialecte de Propositional Dynamic Logic PDL [6] où les programmes atomiques abstraits de PDL sont instanciés par des affectations de valeurs de vérité à des formules atomiques. L'avantage de ceci est d'obtenir une borne supérieure de la complexité PSPACE, à la fois pour le problème de satisfiabilité et celui de vérification de modèle. Nous montrons aussi comment les mises à jour de visibilité peuvent capturer les annonces publiques et privées des atomes et de leur négation.

Nous appelons notre logique DEL-PAO : Dynamic Epistemic Logic of Propositional Assignment and Observation.

Le papier est organisé comme suit : les sections 2 et 3 introduisent le langage et la sémantique de DEL-PAO. Les sections 4 et 5 contiennent un axiomatisation et le résultat de complexité. La section 6 illustre notre logique par deux applications : l'encodage des annonces et une modélisation du problème des généraux. La section 7 traite des travaux connexes et la section 8 conclut. [1]

2 Langage

Soit $Prop$ un ensemble non vide dénombrable de variables propositionnelles et soit Agt un ensemble fini non vide d'agents. Les formules atomiques de notre langage sont des séquences d'opérateurs de visibilité suivies de variables propositionnelles. La définition formelle est la suivante.

L'ensemble des *opérateurs d'observabilité* est

$$OBS = \{S_i : i \in Agt\} \cup \{JS\},$$

où S_i représente de visibilité individuelle de l'agent i et JS représente la visibilité conjointe de tous les agents.

1. Une version complète de ce papier incluant les preuves (en anglais) est disponible à `http://tinyurl.com/o3q86ay`.

L'ensemble de toutes les séquences d'opérateurs de visibilité est noté OBS^* et l'ensemble de toutes les séquences non vides est noté OBS^+. Nous utilisons $\sigma, \sigma', \ldots$ pour les éléments de OBS^*. Enfin, l'ensemble des formules atomiques est

$$ATM = \{\sigma\, p : \sigma \in OBS^*, p \in Prop\}.$$

Les éléments de cet ensemble sont également appelés *atomes de visibilité*, ou simplement atomes. Par exemple, $JS\, S_2\, q$ signifie "tous les agents voient conjointement si l'agent 2 voit la valeur de q"; en d'autres termes, il est attention conjointe dans le groupe de tous les agents concernant la visibilité sur q de 2. Nous utilisons $\alpha, \alpha', \ldots, \beta, \beta', \ldots$ pour les éléments de ATM.

Le langage de DEL-PAO est alors défini par la grammaire suivante :

$$\pi ::= +\alpha \mid -\alpha \mid \pi;\pi \mid \pi \sqcup \pi \mid \varphi?$$
$$\varphi ::= \alpha \mid \neg\varphi \mid \varphi \wedge \varphi \mid K_i\varphi \mid CK\varphi \mid [\pi]\varphi$$

où α prend ses valeurs dans ATM et i dans Agt.

Nos programmes atomiques sont des affectations de valeurs de vérité à des atomes de ATM : $+\alpha$ rends α vrai et $-\alpha$ le rend faux. Les programmes complexes sont construits avec les opérateurs de PDL : $\pi;\pi'$ est la composition séquentielle, $\pi \sqcup \pi'$ est le choix non déterministe, et $\varphi?$ est le test. Comme dans PDL, la formule $[\pi]\varphi$ signifie "après chaque exécution de π, φ est vrai". La formule $K_i\varphi$ signifie "i sait que φ est vrai sur la base de ce qu'il observe", et $CK\varphi$ signifie "tous les agents savent conjointement que φ est vrai sur la base de ce qu'ils observent conjointement". Nos opérateurs épistémiques captent des formes de connaissances individuelles et communes qui sont respectivement obtenus par observation individuelle et l'observation conjointe des faits. Cela diffère donc conceptuellement des opérateurs classiques de la connaissance individuelle et commune comme étudié dans le domaine de la logique épistémique [4]. Nous reviendrons sur ce point dans la section 3.4.

Les autres opérateurs booléens $\top$, $\bot$, $\vee$, $\rightarrow$ and $\leftrightarrow$ sont définis comme d'habitude, et $\neg K_i \neg\varphi$ est abrégé par $\widehat{K}_i\varphi$. Le programme $\top?$ est abrégé par *skip* ("ne rien faire") et $\bot?$ par *fail*.

Nous notons $ATM(\varphi)$ (respectivement $ATM(\pi)$) l'ensemble des atomes apparaissant dans la formule φ (respectivement le programme π). Par exemple, pour $\varphi = [\pi]S_1\, JS\, p \rightarrow q$ avec $\pi = q?;+S_2\, p$, on a $ATM(\varphi) = \{q, S_2\, p, S_1\, JS\, p\}$ et $ATM(\pi) = \{q, S_2\, p\}$. $JS\, p$ n'est donc pas un atome de φ. La longueur des formules φ et des programmes π, notée $length(\varphi)$ et $length(\pi)$, est le nombre de symboles utilisés pour les écrire, où nous ne comptons pas [,] et les parenthèses et considérons que la longueur de JS, CK, les noms des agents et des variables propositionnelles est de 1. Ainsi, les symboles S_i et K_i ont une longueur de 2. Par exemple, on a $length(S_2\, S_2\, p) = 5$ et $length([+S_2\, p]JS\, p \rightarrow q) = 8$. On note que $ATM(\varphi) \leq length(\varphi)$.

3 Sémantique

Nous définissons les évaluations et précisons les contraintes qui sont motivées par le fait que les informations de visibilité devrait être introspectives et que la visibilité commune devrait impliquer la visibilité individuelle. Nous définissons ensuite les relations d'indistinguabilité entre évaluations ainsi que l'interprétation des formules et des programmes.

3.1 Évaluations introspectives

Definition 1. *Une* évaluation *est un sous-ensemble de l'ensemble des atomes ATM. Une évaluation $V \in 2^{ATM}$ est* introspective *si et seulement si elle respecte les propriétés suivantes pour tout $\alpha \in ATM$ et $i \in Agt$:*

$$S_i\, S_i\, \alpha \in V \quad \text{(C1)}$$
$$JS\, JS\, \alpha \in V \quad \text{(C2)}$$
$$JS\, S_i\, S_i\, \alpha \in V \quad \text{(C3)}$$
$$\text{si } JS\, \alpha \in V, \text{ alors } S_i\, \alpha \in V \quad \text{(C4)}$$
$$\text{si } JS\, \alpha \in V, \text{ alors } JS\, S_i\, \alpha \in V \quad \text{(C5)}$$

L'ensemble des évaluations introspectives est noté $INTR$.

(C1) concerne l'introspection de la vue individuelle : un agent voit toujours s'il voit la valeur de n'importe quel atome. (C2) requiert la même propriété pour la vue commune; en effet, si $JS\, \alpha$ est vrai, alors $JS\, JS\, \alpha$ doit être vrai par introspection, et si $JS\, \alpha$ est faux alors tous les agents voient conjointement qu'au moins un d'entre eux a perdu le contact visuel. (C3) oblige la première propriété à être connaissance commune. (C4) garantit que la visibilité conjointe implique la visibilité individuelle. (C5) garantit que $JS\, \alpha \in V$ implique $JS\, \sigma\, \alpha \in V$ pour $\sigma \in OBS^*$.

Ensemble, les deux dernières contraintes assurent que lorsque $JS\, \alpha \in V$, alors $\sigma\, \alpha \in V$ pour $\sigma \in OBS^+$. Ceci motive la relation de *conséquence introspective* entre les atomes comme suit :

$$\alpha \rightsquigarrow \beta \text{ ssi soit } \alpha = \beta,$$
$$\text{soit } \alpha = JS\, \alpha' \text{ et } \beta = \sigma\, \alpha' \text{ pour } \sigma \in OBS^+$$

Ainsi, un atome de la forme $JS\, p$ a un nombre infini de conséquences : $JS\, p$ lui-même; $S_i\, p$, $JS\, S_i\, p$ et $S_i\, JS\, p$ pour tout agent i; $S_i\, S_j\, p$ pour tout i et j, et ainsi de suite. Les atomes qui ne sont pas de la forme $JS\, \alpha$ ont seulement comme conséquence eux-mêmes.

Nous pouvons caractériser évaluations introspectives comme les évaluations qui sont fermées par conséquence introspective.

Proposition 1. *Une évaluation $V \subseteq ATM$ est introspective si et seulement si, pour tout $\alpha \in ATM$ et $i \in Agt$:*

$$\sigma\, S_i\, S_i\, \alpha \in V \text{ pour tout } \sigma \in OBS^* \quad (1)$$

$$\sigma\, JS\, \alpha \in V \text{ pour tout } \sigma \in OBS^+ \quad (2)$$

$$\text{si } \alpha \in V \text{ et } \alpha \rightsquigarrow \beta \text{ alors } \beta \in V \quad (3)$$

Ainsi, tout atome contenant une séquence $S_i\, S_i$ est dans une évaluation introspective, ainsi que tout atome contenant un JS (en supposant qu'il n'est pas le premier élément). Enfin, toute conséquence est également présente. Ces trois éléments sont équivalents aux cinq contraintes (voir la preuve dans la version complète du papier), mais sont techniquement plus faciles à manipuler.

On dit qu'un atome $\alpha \in ATM$ est *valide dans INTR* si et seulement si α appartient à toute évaluation de $INTR$. Par la proposition 1, α est valide dans $INTR$ si et seulement s'il est de la forme soit $\sigma\, S_i\, S_i\, \alpha$ avec $\sigma \in OBS^*$ ou $\sigma\, JS\, \alpha$ avec $\sigma \in OBS^+$.

On observe que nous n'imposons pas la contrainte "si $\sigma\, \alpha \in V$ pour chaque $\sigma \in OBS^*$, alors $JS\, \alpha \in V$", qui correspond à la définition du plus grand point fixe de l'opérateur de connaissance commune. Nous commenterons ceci dans la section 3.4.

3.2 Relations d'indistinguabilité

Nous définissons les relations $\sim_i$ entre évaluations, une par agent i. Deux évaluations sont liées par $\sim_i$ si chaque α que i voit a la même valeur dans l'évaluation liée. De la même façon, nous définissons une relation $\sim_{Agt}$.

On note $V(\alpha) = V'(\alpha)$ lorsque soit $\alpha \in V$ et $\alpha \in V'$, ou $\alpha \notin V$ et $\alpha \notin V'$. On définit :

$$V \sim_i V' \text{ ssi } S_i\, \alpha \in V \text{ implique } V(\alpha) = V'(\alpha)$$

$$V \sim_{Agt} V' \text{ ssi } JS\, \alpha \in V \text{ implique } V(\alpha) = V'(\alpha)$$

Par exemple, supposons que l'ensemble des variables propositionnelles $Prop$ est le singleton $\{p\}$. Alors une évaluation contenant $S_i\, p$ et p est liée à toute évaluation contenant p, alors que l'évaluation avec $S_i\, p$ mais sans p est liée à toute évaluation ne contenant pas p.

Les relations $\sim_i$ et $\sim_{Agt}$ sont réflexives, mais ni transitives ni symétriques. Cependant, ces deux propriétés sont vérifiées sur les évaluations satisfaisant les contraintes (C1) et (C2) de la définition 1.

Proposition 2. *La relation $\sim_{Agt}$ et chaque relation $\sim_i$ sont des relations d'équivalence sur INTR.*

Lemma 1. *Soit $V \in INTR$, $V \sim_i V_1$ et $V \sim_{Agt} V_2$. Alors $V_1 \in INTR$ et $V_2 \in INTR$.*

Ce dernier lemme établit que, si nous sommes dans une évaluation introspective, alors une relation épistémique ne permet jamais d'accéder à une évaluation non introspective.

3.3 Conditions de vérité et validité

Étant donné une évaluation introspective V, nos opérations de mise à jour ajoutent ou enlèvent des atomes de V. Cela demande une certaine prudence : nous voulons que l'évaluation résultante soit introspective. Par exemple, quand V ne contient pas $S_i\, p$, alors $V \cup \{JS\, p\}$ viole (C4). Donc, lors de l'ajout d'un atome à V, nous devons aussi ajouter toutes ses *conséquences introspectives positives*. Symétriquement, lors du retrait d'un atome, il faut aussi de supprimer ses *conséquences introspectives négatives*. Nous définissons les éléments suivants :

$$\mathit{Eff}^+(\alpha) = \{\beta \in ATM : \alpha \rightsquigarrow \beta\}$$

$$\mathit{Eff}^-(\alpha) = \{\beta \in ATM : \beta \rightsquigarrow \alpha\}$$

Par exemple, $\mathit{Eff}^-(S_i\, S_j\, p) = \{JS\, p, JS\, S_j\, p\}$ car ces deux atomes impliquent $S_i\, S_j\, p$. Les variables propositionnelles n'ont pas de conséquence positive ou négative, à part eux-mêmes. De toute évidence, quand V est introspective alors à la fois $V \cup \mathit{Eff}^+(\alpha)$ et $V \setminus \mathit{Eff}^-(\alpha)$ le sont aussi, (sauf si α est valide).

Les conditions de vérité sont les suivantes :

$V \models \alpha$ ssi $\alpha \in V$

$V \models \neg\varphi$ ssi $V \not\models \varphi$

$V \models \varphi \wedge \psi$ ssi $V \models \varphi$ et $V \models \psi$

$V \models K_i\varphi$ ssi $V' \models \varphi$ pour tout V' tel que $V \sim_i V'$

$V \models CK\varphi$ ssi $V' \models \varphi$ pour tout V' tel que $V \sim_{Agt} V'$

$V \models [\pi]\varphi$ ssi $V' \models \varphi$ pour tout V' tel que $V R_\pi V'$

où R_π est une relation binaire sur les évaluations définie (par récurrence mutuelle avec la définition de $\models$) par :

$V R_{+\alpha} V'$ ssi $V' = V \cup \mathit{Eff}^+(\alpha)$

$V R_{-\alpha} V'$ ssi $V' = V \setminus \mathit{Eff}^-(\alpha)$ and α n'est pas valide dans $INTR$

$V R_{\pi_1;\pi_2} V'$ ssi il existe U tel que $V R_{\pi_1} U$ et $U R_{\pi_2} V'$

$V R_{\pi_1 \sqcup \pi_2} V'$ ssi $V R_{\pi_1} V'$ ou $V R_{\pi_2} V'$

$V R_{\varphi?} V'$ ssi $V = V'$ et $V \models \varphi$

La relation R_π est définie comme dans PDL pour la opérateurs ;, $\sqcup$ et ?. L'interprétation des affectations est conçu de telle manière que nous restons dans $INTR$: lors de l'exécution $+\alpha$, on ajoute toutes les

conséquences positives de α ; pour $-\alpha$, le programme devrait échouer si α est valide dans *INTR* et sinon, enlever toutes les conséquences négatives de α. Par exemple, nous n'avons jamais $VR_{-S_1\,S_1\,p}V'$, i.e., le programme $-S_1\,S_1\,p$ échoue toujours. En revanche, le programme $-S_1\,S_2\,p$ réussit toujours, et on a

$$VR_{-S_1\,S_2\,p}\,(V \setminus \{S_1\,S_2\,p, JS\,S_2\,p, JS\,p\}).$$

car $JS\,S_2\,p \rightsquigarrow S_1\,S_2\,p$ et $JS\,p \rightsquigarrow S_1\,S_2\,p$. Donc V $V \not\models [-S_1\,S_2\,p]JS\,p$ quel que soit V.

Lemma 2. *Si $V \in INTR$ et $VR_\pi V'$ alors $V' \in INTR$.*

Proposition 3. *Pour tout $V \in INTR$, $i \in Agt$ et π, V est seulement liée à des évaluations de INTR par $\sim_i$, $\sim_{Agt}$ et R_π.*

Lorsqu'on a $V \models \varphi$, on dit que V est un *modèle* de φ. L'ensemble des modèles (pas nécessairement introspectifs) de φ est noté $\|\varphi\|$.

Une formule φ est *satisfiable dans INTR* si φ a un modèle introspectif, à savoir, si $\|\varphi\| \cap INTR \neq \emptyset$. Par exemple, $JS\,p \wedge \neg S_i\,p$ a un modèle, mais n'a pas de modèle introspectif et est donc insatisfiable dans *INTR*.

Une formule φ est *valide dans INTR* si $INTR \subseteq \|\varphi\|$. On dit aussi que φ est DEL-PAO-valide. Par exemple, $\neg[-S_1\,S_2\,p]JS\,p$ est valide dans *INTR*. On note que $\neg\beta \rightarrow [+\alpha]\neg\beta$ est valide dans *INTR* si et seulement si $\alpha \not\rightsquigarrow \beta$.

La proposition suivante exprime le fait que, quand φ ne contient aucun opérateur épistémique, alors la valeur de vérité de φ ne dépend que des atomes présents dans φ.

Proposition 4. *Soit φ sans opérateurs épistémiques. Soit $V, V' \in 2^{ATM}$ telles que que $V(\alpha) = V'(\alpha)$ pour chaque atome $\alpha \in ATM(\varphi)$. Alors $V \models \varphi$ si et seulement si $V' \models \varphi$.*

Cette proposition sera déterminante pour le reste du papier. On remarque qu'elle n'est pas nécessairement correcte quand la formule contient des opérateurs épistémiques. Par exemple, considérons $K_i p$: sa valeur de vérité dépend de la valeur de vérité de l'atome de visibilité $S_i\,p$, qui ne fait pas partie $ATM(K_i p)$.

3.4 Discussion

Les opérateurs de la connaissance individuelle de DEL-PAO satisfont tous les principes de S5. Ils sont plus forts que ceux de la logique épistémique standard S5. On peut le voir avec la formule invalide de S5 $K_i(p \vee q) \rightarrow (K_i p \vee K_i q)$ qui est valide dans *INTR* (voir la proposition 5). Pour donner un exemple, cela signifie que si l'on sait que le maître d'hôtel ou le jardinier est le meurtrier alors on sait lequel d'entre eux l'est ; elle implique aussi que l'on ne peut pas connaître une disjonction exclusive de deux variables propositionnelles p et q. C'est un principe fort, qui est toutefois également validé par toutes les autres logiques épistémiques basées sur la visibilité.

Notre opérateur de connaissance commune satisfait l'axiome de point fixe de la connaissance commune

$$CKp \rightarrow p \wedge \Big(\bigwedge_{i \in Agt} K_i CKp\Big).$$

Cela correspond au fait que par des contraintes (C2) et (C4), $JS\,p \wedge p \rightarrow \Big(\bigwedge_{i \in Agt} S_i\,JS\,p\Big)$ est valide dans *INTR*. Notre notion de connaissance commune est toutefois plus faible que la connaissance commune standard car l'axiome d'induction

$$\Big(\varphi \wedge CK\Big(\varphi \rightarrow \bigwedge_{i \in Agt} K_i\varphi\Big)\Big) \rightarrow CK\varphi$$

n'est pas valide dans *INTR*. Au-delà de la justification technique de ce choix (une telle contrainte ne peut pas être capturée par une formule finie constituée d'atomes de visibilité) nous nous alignons sur [14, 9] et supposons que ce principe est trop fort pour une logique de la connaissance commune.

4 Axiomatization

Nous présentons maintenant une axiomatisation adéquate et complète de DEL-PAO.

4.1 Axiomes de réduction pour les opérateurs modaux

La proposition suivante sera utilisée pour éliminer les opérateurs épistémiques d'une formule qui ne contient pas d'opérateurs dynamiques.

Proposition 5. *Les formules suivantes sont* DEL-PAO*-valides, avec A^+ et A^- des ensembles d'atomes tels que $A^+ \cap A^- = \emptyset$:*

$$K_i\alpha \leftrightarrow S_i\,\alpha \wedge \alpha \qquad (Red_{K,\alpha})$$

$$CK\alpha \leftrightarrow JS\,\alpha \wedge \alpha \qquad (Red_{CK,\alpha})$$

$$K_i\neg\alpha \leftrightarrow S_i\,\alpha \wedge \neg\alpha \qquad (Red_{K,\neg})$$

$$CK\neg\alpha \leftrightarrow JS\,\alpha \wedge \neg\alpha \qquad (Red_{CK,\neg})$$

$$K_i(\varphi \wedge \varphi') \leftrightarrow K_i\varphi \wedge K_i\varphi' \qquad (Red_{K,\wedge})$$

$$CK(\varphi \wedge \varphi') \leftrightarrow CK\varphi \wedge CK\varphi' \qquad (Red_{CK,\wedge})$$

$$K_i\left(\bigvee_{\alpha\in A^+}\alpha \vee \bigvee_{\alpha\in A^-}\neg\alpha\right) \leftrightarrow \left(\bigvee_{\alpha\in A^+}K_i\alpha\right) \vee \left(\bigvee_{\alpha\in A^-}K_i\neg\alpha\right) \qquad (Red_{K,\vee})$$

$$CK\left(\bigvee_{\alpha\in A^+}\alpha \vee \bigvee_{\alpha\in A^-}\neg\alpha\right) \leftrightarrow \left(\bigvee_{\alpha\in A^+}CK\alpha\right) \vee \left(\bigvee_{\alpha\in A^-}CK\neg\alpha\right) \qquad (Red_{CK,\vee})$$

La proposition suivante permet d'éliminer les opérateurs dynamiques de formules ne contenant pas d'opérateurs épistémiques.

Proposition 6. *Les formules suivantes sont* DEL-PAO*-valides.*

$$[\pi;\pi']\varphi \leftrightarrow [\pi][\pi']\varphi \qquad (Red_;)$$
$$[\pi \sqcup \pi']\varphi \leftrightarrow [\pi]\varphi \wedge [\pi']\varphi \qquad (Red_\sqcup)$$
$$[\varphi?]\varphi' \leftrightarrow \varphi \to \varphi' \qquad (Red_?)$$
$$[+\alpha]\neg\varphi \leftrightarrow \neg[+\alpha]\varphi \qquad (Red_{+\alpha,\neg})$$
$$[-\alpha]\neg\varphi \leftrightarrow \begin{cases} \top & si\ \alpha\ est\ valide\ dans\ INTR \\ \neg[-\alpha]\varphi & sinon \end{cases} \qquad (Red_{-\alpha,\neg})$$
$$[+\alpha](\varphi \wedge \varphi') \leftrightarrow [+\alpha]\varphi \wedge [+\alpha]\varphi' \qquad (Red_{+\alpha,\wedge})$$
$$[-\alpha](\varphi \wedge \varphi') \leftrightarrow [-\alpha]\varphi \wedge [-\alpha]\varphi' \qquad (Red_{-\alpha,\wedge})$$
$$[+\alpha]\beta \leftrightarrow \begin{cases} \top & si\ \alpha \rightsquigarrow \beta \\ \beta & sinon \end{cases} \qquad (Red_{+\alpha})$$
$$[-\alpha]\beta \leftrightarrow \begin{cases} \top & si\ \alpha\ est\ valide\ dans\ INTR \\ \bot & si\ \alpha\ n'est\ pas\ valide\ dans\ INTR \\ & and\ \beta \rightsquigarrow \alpha \\ \beta & sinon \end{cases} \qquad (Red_{-\alpha})$$

Theorem 1. *Pour toute formule φ, il existe une formule sans opérateurs modaux φ' telle que $\varphi \leftrightarrow \varphi'$ est* DEL-PAO*-valide.*

4.2 Axiomes d'introspection

Proposition 7. *Les formules suivantes sont* DEL-PAO*-valides, pour tout $i \in Agt$ et $\alpha \in ATM$:*

$$S_i\, S_i\, \alpha \qquad (Vis_{C1})$$
$$JS\ JS\ \alpha \qquad (Vis_{C2})$$
$$JS\, S_i\, S_i\, \alpha \qquad (Vis_{C3})$$
$$JS\ \alpha \to S_i\, \alpha \qquad (Vis_{C4})$$
$$JS\ \alpha \to JS\, S_i\, \alpha \qquad (Vis_{C5})$$

On appelle $\mathcal{T}_{vis}$ la collection des formules ci-dessus, i.e.,

$$\begin{aligned} \mathcal{T}_{vis} = &\{S_i\, S_i\, \alpha : i \in Agt, \alpha \in ATM\} \\ &\cup \{JS\ JS\ \alpha : \alpha \in ATM\} \\ &\cup \{JS\, S_i\, S_i\, \alpha : i \in Agt, \alpha \in ATM\} \\ &\cup \{JS\ \alpha \to S_i\, \alpha : i \in Agt, \alpha \in ATM\} \\ &\cup \{JS\ \alpha \to JS\, S_i\, \alpha : i \in Agt, \alpha \in ATM\} \end{aligned}$$

Proposition 8. *Pour φ sans opérateurs modaux, φ est* DEL-PAO*-valide si et seulement si $\mathcal{T}_{vis} \models_{\mathsf{CPL}} \varphi$, où $\models_{\mathsf{CPL}}$ est la conséquence logique en logique propositionnelle classique.*

4.3 Adéquation et complétude

L'axiomatisation de DEL-PAO est donnée par :

- les axiomes de CPL (logique propositionnelle classique) ;
- les axiomes de réduction de la proposition 5 ;
- les axiomes de réduction de la proposition 6 ;
- les axiomes d'introspection de la proposition 7 ;
- la règle du Modus Ponens et les règles d'équivalence pour K_i, CK et $[\pi]$:

$$\frac{\varphi \leftrightarrow \varphi'}{K_i\varphi \leftrightarrow K_i\varphi'} \qquad \frac{\varphi \leftrightarrow \varphi'}{CK\varphi \leftrightarrow CK\varphi'} \qquad \frac{\varphi \leftrightarrow \varphi'}{[\pi]\varphi \leftrightarrow [\pi]\varphi'}$$

Proposition 9. *Pour les formules φ sans opérateurs modaux, $\vdash_{\mathsf{DEL\text{-}PAO}} \varphi$ si et seulement si $\mathcal{T}_{vis} \vdash_{\mathsf{CPL}} \varphi$.*

Theorem 2. *L'axiomatisation de* DEL-PAO *est adéquate et complète.*

5 Complexité

Theorem 3. *Les problèmes de satisfiabilité et de vérification de modèle de* DEL-PAO *sont* PSPACE*-complets.*

Nous consacrons le reste de la section à la preuve de ce résultat. Nous commençons par observer que les problèmes de satisfiabilité et de vérification de modèle ont la même complexité.

Proposition 10. *Soit φ une formule de* DEL-PAO *sans opérateurs épistémiques telle que $ATM(\varphi) = \{\alpha_1, \ldots, \alpha_n\}$. Alors φ est satisfiable dans INTR si et seulement si la formule*

$$\langle (+\alpha_1 \sqcup -\alpha_1); \ldots; (+\alpha_n \sqcup -\alpha_n) \rangle \varphi$$

est valide dans INTR.

On voit que $(+\alpha_1 \sqcup -\alpha_1); \ldots; (+\alpha_n \sqcup -\alpha_n)$ a une longueur linéaire dans $length(\varphi)$ car $ATM(\varphi) \leq length(\varphi)$.

Dans le reste de la section, nous établissons les bornes inférieures et supérieures du problème de vérification de modèle.

5.1 Borne inférieure

La borne inférieure du problème de vérification de modèle est établie au moyen d'un encodage de formules booléennes quantifiées (QBF), dont le problème de satisfiabilité est PSPACE-complet. Plus de détails sont disponibles dans la version complète du papier.

Il est un peu plus difficile d'établir la borne supérieure. Nous allons encoder notre logique dans le fragment sans étoile (*) de Dynamic Logic of Propositional Assignments DL-PA [10, 3], dont le problème de satisfiabilité est PSPACE-complet. D'abord, nous rappelons brièvement que cette logique.

5.2 Dynamic Logic of Propositional Assignments

Tout comme le langage de DEL-PAO, le langage de DL-PA est constitué de formules et de programmes. Ils sont définis par la grammaire suivante :

$$\pi ::= +\alpha \mid -\alpha \mid \pi;\pi \mid \pi \sqcup \pi \mid \varphi?$$
$$\varphi ::= \alpha \mid \neg\varphi \mid \varphi \wedge \varphi \mid [\pi]\varphi$$

où α prend ses valeurs dans ATM et i dans Agt. Ce langage ne contient pas d'opérateurs épistémiques, et est constitué des mêmes atomes que DEL-PAO.

Les formules sont interprétées dans des évaluations $V \in 2^{ATM}$ exactement de la même manière que dans DEL-PAO, sauf que l'interprétation des programmes atomiques ne prend pas en compte les conséquences introspectives. Donc :

$$VR_{+\alpha}V' \text{ ssi } V' = V \cup \{p\}$$
$$VR_{-\alpha}V' \text{ ssi } V' = V \setminus \{p\}$$

On a un équivalent de la proposition 4 dans DL-PA.

Proposition 11 ([3], Proposition 1)**.** *Soit $V, V' \in 2^{ATM}$ telles que $V(\alpha) = V'(\alpha)$ pour tout atome $\alpha \in ATM(\varphi)$. Alors $V \models_{\text{DL-PA}} \varphi$ si et seulement si $V' \models_{\text{DL-PA}} \varphi$.*

5.3 Borne supérieure

Nous transformons d'abord nos opérateurs épistémiques en programmes équivalents ; il suffira alors de donner une borne supérieure pour le problème de vérification de modèle pour les formules sans opérateurs épistémiques. Nous établissons ensuite la complexité en traduisant nos formules et programmes en formules et programmes DL-PA. Ce faisant, nous prenons en compte les contraintes d'introspection de DEL-PAO sur les évaluations.

Tout d'abord, nous définissons les programmes :

$$\pi_{i,\alpha} = S_i\,\alpha? \sqcup (\neg S_i\,\alpha?; (+\alpha \sqcup -\alpha))$$
$$\pi_{Agt,\alpha} = JS\,\alpha? \sqcup (\neg JS\,\alpha?; (+\alpha \sqcup -\alpha))$$

Le premier programme vérifie si i voit α, et si ce n'est pas le cas, fait varier la valeur de vérité de α ; le second fait la même chose mais pour la visibilité conjointe. Étant donné un ensemble d'atomes $A = \{\alpha_1, ..., \alpha_n\}$, on définit :

$$\pi_{i,A} = \pi_{i,\alpha_1}; ...; \pi_{i,\alpha_n}$$
$$\pi_{Agt,A} = \pi_{Agt,\alpha_1}; ...; \pi_{Agt,\alpha_n}$$

où nous supposons que les atomes sont ordonnés de telle sorte que $k \leq l$ implique que $length(\alpha_k) \leq length(\alpha_l)$. De plus, nous supposons que le programme correspond à *skip* si l'ensemble A est vide. Par exemple,

$$\pi_{i,\{S_j\,p,p\}} = \big(S_i\,p? \sqcup (\neg S_i\,p?; (+p \sqcup -p))\big);$$
$$\big(S_i\,S_j\,p? \sqcup (\neg S_i\,S_j\,p?; (+S_j\,p \sqcup -S_j\,p))\big)$$

car $length(p) \leq length(S_j\,p)$.

Proposition 12. *Soit φ une formule de* DEL-PAO *sans opérateurs épistémiques. Alors les équivalences*

$$K_i\varphi \leftrightarrow [\pi_{i,ATM(\varphi)}]\varphi$$
$$CK\varphi \leftrightarrow [\pi_{Agt,ATM(\varphi)}]\varphi$$

sont DEL-PAO*-valides.*

La proposition 12 peut être transformée en une procédure éliminant opérateurs épistémiques : il suffit d'appliquer plusieurs fois les équivalences, à commencer par les opérateurs les plus intérieurs.

Procedure 1. *Tant qu'il y a un opérateur épistémique dans φ :*

1. *s'il existe une sous-formule $K_i\varphi'$ telle que φ' ne contient pas les opérateurs épistémiques, remplacer φ par $[\pi_{i,ATM(\varphi)}]\varphi'$;*
2. *s'il existe une sous-formule $CK\varphi'$ telle que φ' ne contient pas les opérateurs épistémiques, remplacer φ par $[\pi_{Agt,ATM(\varphi)}]\varphi'$.*

Proposition 13. *Pour chaque formule* DEL-PAO *φ, il existe une formule* DEL-PAO *φ' sans opérateurs épistémiques telle que $\varphi \leftrightarrow \varphi'$ est* DEL-PAO*-valide. La longueur de φ' est polynomiale dans $length(\varphi)$.*

Il suffit donc de considérer le problème de vérification de modèle pour les formules sans opérateurs épistémiques. La prochaine étape est de traduire en temps polynomial les formules et les programmes de DEL-PAO en formules et programmes DL-PA. Nous allons en même temps capter les contraintes d'introspection par des affectations DL-PA.

Étant donné d'un atome α et un ensemble d'atomes pertinents $A \subseteq ATM$, soit $Eff^+(\alpha) \cap A = \{\beta_1, ..., \beta_n\}$ et $Eff^-(\alpha) \cap A = \{\beta'_1, ..., \beta'_m\}$. On traduit les affectations de α comme suit :

$$tr(+\alpha, A) = +\beta_1; ...; +\beta_n$$
$$tr(-\alpha, A) = \begin{cases} fail & \text{si } \alpha \text{ est valide dans } INTR \\ -\beta'_1; ...; -\beta'_m & \text{sinon} \end{cases}$$

Encore une fois, nous supposons que les atomes sont ordonnés de telle sorte que $k \leq l$ implique que $length(\beta_k) \leq length(\beta_l)$ et que le programme devient *skip* si l'ensemble $\{\beta_1, ..., \beta_n\}$ est vide.

Nous étendons tr de façon homomorphique aux programmes et formules complexes, avec $tr(\alpha) = \alpha$ et

$$tr([\pi]\varphi) = [tr(\pi, ATM(\varphi))]tr(\varphi).$$

On note que $ATM(tr(\pi, A)) \subseteq A$ et $ATM(tr(\varphi)) \subseteq ATM(\varphi)$.

Proposition 14. *Soit φ une formule* DEL-PAO *sans opérateurs épistémiques. Alors $V \models_{\text{DEL-PAO}} \varphi$ si et seulement si $V \models_{\text{DL-PA}} tr(\varphi)$.*

La grande finale résulte de propositions 10, 13 et 14 et de l'observation que la longueur de $tr(\varphi)$ est polynomiale dans $length(\varphi)$.

Corollary 1. *Dans* DEL-PAO*, les problèmes de satisfiabilité et de vérification de modèle sont* PSPACE*-complet.*

6 Applications

Nous montrons maintenant comment exprimer les annonces publiques et privées des atomes et des formules épistémiques. Cela nous permet de formaliser le problème de bavardage et des deux généraux.

6.1 Annonces publiques et privées

Public Announcement Logic PAL [15] est une logique de la famille DEL étendant la logique épistémique standard avec un opérateur $[\psi!]$, de sorte que $[\psi!]\varphi$ signifie "après que le fait réel ψ ait été annoncé publiquement, φ est vrai". Ses validités sont axiomatisées au moyen des axiomes de réduction suivants :

$$[\psi!]p \leftrightarrow \psi \rightarrow p$$
$$[\psi!]\neg\varphi \leftrightarrow \psi \rightarrow \neg[\psi!]\varphi$$
$$[\psi!](\varphi \wedge \varphi') \leftrightarrow [\psi!]\varphi \wedge [\psi!]\varphi'$$
$$[\psi!]K_i\varphi \leftrightarrow \psi \rightarrow K_i[\psi!]\varphi$$

Nous pouvons exprimer les annonces publiques de littéraux dans DEL-PAO comme suit :

$$p! = p?; +JS\, p$$
$$\neg p! = \neg p?; +JS\, p$$

ainsi que celle de la connaissance des atomes :

$$K_i p! = K_i p?; +JS\, p$$

Il peut être vérifié qu'avec ces définitions, tous les axiomes de réduction de PAL sont valides dans notre logique lorsque ψ est soit un littéral soit la connaissance d'un atome (voir la version longue).

Un supplément intéressant est le fait que nous pouvons également modéliser facilement les annonces *privées* du même type de formules. Avec $j : \psi!$ signifiant "ψ est annoncé en privé à l'agent j", on a :

$$j : p! = p?; +S_j\, p$$
$$j : \neg p! = \neg p?; +S_j\, p$$
$$j : K_i p! = K_i p?; +S_j\, p; +S_j\, S_i\, p$$

6.2 Le problème de bavardage

Six amis ont chacun un secret. Quand ils s'appellent les uns des autres, ils échangent tous les secrets qu'ils connaissent. Le problème est de trouver combien d'appels sont nécessaires pour diffuser tous les secrets parmi tous les amis. Il a été prouvé [12] que le nombre minimal d'appels est 8 ; par exemple, si nous écrivons ij le fait que i appelle j (ou j appelle i), la séquence suivante propage tous les secrets : 12, 34, 56, 13, 45, 16, 24, 35. Une étude détaillée de ce problème peut être trouvée dans [18]. Le problème est également abordé dans [19] avec une logique spécifique aux réseaux de communication.

Nous modélisons ce problème avec des annonces privées. Avec $Agt = \{i : 1 \leq i \leq 6\}$ et $Prop = \{s_i : i \in Agt\}$ (s_i signifiant que i a le secret s_i), on définit le programme $Call_{ij}$, pour $i, j \in Agt$, par :

$$Call_{ij} =$$
$$(S_i\, s_1?; j : s_1! \sqcup \neg S_i\, s_1?); ...; (S_i\, s_6?; j : s_6! \sqcup \neg S_i\, s_6?);$$
$$(S_j\, s_1?; i : s_1! \sqcup \neg S_j\, s_1?); ...; (S_j\, s_6?; i : s_6! \sqcup \neg S_j\, s_6?)$$

En exécutant notre programme, i raconte tout ce qu'il sait à j, et inversement. Par conséquent, on a :

$$\{\alpha : \alpha \text{ est valide dans } INTR\} \cup Prop \cup \{S_i\, s_i : i \in Agt\} \models$$
$$[Call_{12}; Call_{34}; Call_{56}; Call_{13};$$
$$Call_{45}; Call_{16}; Call_{24}; Call_{35}] \bigwedge_{i \in Agt} K_i \left(\bigwedge_{j \in Agt} s_j \right)$$

Notre évaluation contient tous les secrets, et le fait que chaque agent connaît son secret. Elle est introspective, puisque nous avons tous les atomes valides et nous n'ajoutons pas d'autre atome de la forme $JS\, \alpha$.

Intuitivement, chaque appel va ajouter la visibilité de chaque atome connu par l'un des deux agents aux deux ; une annonce de la forme $j : s_i!$ est égale, dans DEL-PAO, à $s_i?; +S_j\, s_i$, et puisque s_i est vrai dans notre évaluation, cela va se réduire à $+S_j\, s_i$. En fin de compte, $S_j\, s_i$ sera mis à vrai seulement si $S_i\, s_i$ est vrai. Donc $Call_{12}$ est réduit à $+S_2\, s_1; +S_1\, s_2$, et après son exécution, $S_1\, s_1$, $S_2\, s_2$, $S_1\, s_2$ et $S_2\, s_1$ sont tous vrais, et donc $K_1(s_1 \wedge s_2) \wedge K_2(s_1 \wedge s_2)$ est également vrai, ce qui est le résultat attendu. En continuant ainsi, et puisque cette séquence d'appels conduit à une solution dans le problème initial, la formule sera vraie à la fin du programme.

6.3 Le problème des deux généraux

Nous avons vu que dans le problème de bavardage, chaque ami finira par connaître chaque secret. Cela ne signifie pas toutefois qu'il y a connaissance commune de tous les secrets. La difficulté d'obtenir la connaissance commune par communication asynchrone non fiable est mis en évidence par le 'problème des deux généraux' [1] [7]. Sa généralisation est le célèbre problème de généraux byzantins [13].

En bref, deux généraux ont besoin de coordonner une attaque ; ils ne peuvent gagner que s'ils attaquent en même temps. Pour communiquer, ils doivent en-

voyer un messager qui peut être capturé. Ils doivent décider du moment de l'attaque, se mettre d'accord sur ce moment, et chaque général doit savoir que l'autre est d'accord. Pour ce faire, un général envoie un message avec l'heure de l'attaque, mais comme il ne peut pas être sûr que le message ait été reçu, l'autre général doit envoyer un accusé de réception. Bien sûr, tout comme le message d'origine, l'accusé de réception peut être perdu, le premier général doit donc envoyer un accusé de réception, et ainsi de suite. Pour être parfaitement coordonnés, à savoir, pour garantir qu'il y a connaissance commune de l'heure de l'attaque, les généraux doivent envoyer un nombre infini de messages.

Formellement, avec $Agt = \{1, 2\}$ et $Prop = \{ta\}$, nous définissons l'atome α_n par :

— $\alpha_n = \underbrace{S_1\, S_2\, S_1\, ...S_2\, ta}_{n \text{ alternations}}$ si n est pair ;

— $\alpha_n = \underbrace{S_2\, S_1\, S_2\, ...S_2\, ta}_{n \text{ alternations}}$ si n est impair.

Ensuite, nous définissons le programme

$$\pi_n = +ta; +S_1\, ta; +S_2\, ta; +\alpha_2; +\alpha_3; ...; +\alpha_n,$$

qui formalise le fait que le moment de l'attaque a été défini, que chaque agent le sait, que 1 sait que 2 le sait, et ainsi de suite jusqu'à n.

L'illustration du problème de deux généraux est alors le suivant. Pour chaque $n > 0$,

$$\{\alpha : \alpha \text{ est valide dans } INTR\} \models [\pi_n] \neg CK\, ta.$$

Intuitivement, pour avoir $V \models CK\, ta$, il faut avoir $ta \in V$ et $JS\, ta \in V$. Comme aucun d'entre eux est présent dans notre évaluation, les deux devraient être ajoutés par le programme π_n. En outre, et parce que $JS\, ta$ est une conséquence introspective de lui-même seulement, il devrait être ajouté explicitement par π_n. Cependant, alors que π_n ajoute ta quelle que soit la valeur de n>0, l'atome $JS\, ta$ n'est jamais ajouté. Ainsi, $CK\, ta$ n'est jamais vrai quel que soit $n > 0$.

7 Travaux connexes

Ce travail est lié à plusieurs autres logiques développées au cours des dernières années. La logique *Epistemic Coalition Logic of Propositional Control with Partial Observability* ECL-PC(PO) [17] utilise le langage de la logique épistémique standard ; mais les formules sont interprétées sur des évaluations. La principale différence avec DEL-PAO est que la visibilité est représentée par un ensemble d'atomes pour chaque agent, contenant les variables dont l'agent peut percevoir la valeur. Par conséquent, ils n'est pas possible d'exprimer des perceptions d'ordre supérieur telles que "l'agent i voit si l'agent j perçoit p". Au lieu de cela, et comme cela a déjà été mentionné, la perception de chaque agent devient connaissance commune parmi tous les agents. Par conséquent, nous conjecturons que ECL-PC(PO) peut être encodé dans DEL-PAO.

Une autre logique proche de la nôtre est la *Logic of Revelation and Concealment* LRC [16]. Cette logique permet d'exprimer, par des programmes, qu'une variable est révélée ou cachée à un agent. Sémantiquement, les formules sont interprétées dans des modèles pointés avec un ensemble de visibilité pour chaque agent. Révéler une variable à un agent l'ajoute à son ensemble, alors que la lui cacher la supprime de son ensemble. Cependant, comme dans [17], les ensembles de visibilité sont connaissance commune parmi tous les agents.

Nous pouvons également citer la logique 'knowing whether' [5] qui ajoute un opérateur signifiant "i sait si φ" au langage de la logique épistémique standard, interprété comme "φ a la même valeur dans tous les mondes indistinguables pour i". Cela peut être comparé à nos atomes de visibilité S_i qui expriment la même idée sur les atomes.

8 Conclusion

Nous avons introduit une logique épistémique dynamique avec affectations et observations DEL-PAO qui permet d'exprimer des observations d'ordre supérieur et conjointes ainsi que leurs mises à jour. Elle évite l'hypothèse forte des autres logiques épistémiques basées sur les observations, où qui voit quoi qui est connaissance commune. On remarque que l'ajout de l'ordre supérieur des observations et en particulier des observations communes se fait sans coût supplémentaire : la satisfiabilité et la vérification de modèle restent PSPACE-complet. Cela contraste avec les logiques standards de connaissance commune où la satisfiabilité est EXPTIME-difficile [8].

Une extension simple de notre logique serait de généraliser l'opérateur de connaissance commune CK à des sous-ensembles de Agt. Il faudrait ajouter des atomes de visibilité $JS_J\alpha$, un par groupe d'agents J. Une autre généralisation intéressante serait de considérer la croyance à la place de la connaissance. Un moyen d'y parvenir pourrait être de remplacer S_i par deux opérateurs O_i et C_i, signifiant respectivement que i a une opinion sur quelque chose et que l'opinion de i est correcte sur quelque chose. Cela nécessiterait d'autres contraintes sur les évaluations qui doivent correspondre aux propriétés de la croyance. D'autres extensions possibles concernent la partie dynamique : comme [10], on peut ajouter des atomes représentant le contrôle de i sur une variable propositionnelle p, dans le sens où i peut changer la valeur de vérité de p à volonté. On peut alors associer à chaque affectation un auteur, qui est l'agent exécutant l'affectation. Comme montré dans [10], cela permet d'intégrer la Coalition

Logic of Propositional Control [17]. Il reste à savoir comment ceci se combine avec les observations d'ordre supérieur.

Remerciements

Nous sommes reconnaissants aux relecteurs pour leur lecture approfondie et leurs commentaires.

Références

[1] E. A. Akkoyunlu, K. Ekanadham, and R. V. Hubert. Some constraints and tradeoffs in the design of network communications. In *Proceedings of the 5th ACM Symposium on Operating Systems Principles*, pages 67–74. ACM Press, 1975.

[2] Rajeev Alur, Thomas A. Henzinger, Freddy Y. C. Mang, Shaz Qadeer, Sriram K. Rajamani, and Serdar Tasiran. MOCHA : modularity in model checking. In Alan J. Hu and Moshe Y. Vardi, editors, *Computer Aided Verification, 10th International Conference, CAV '98, Vancouver, BC, Canada, June 28 - July 2, 1998, Proceedings*, volume 1427 of *Lecture Notes in Computer Science*, pages 521–525. Springer, 1998.

[3] Philippe Balbiani, Andreas Herzig, and Nicolas Troquard. Dynamic logic of propositional assignments : a well-behaved variant of PDL. In Orna Kupferman, editor, *Proceedings of the 28th Annual IEEE/ACM Symposium on Logic in Computer Science*, pages 143–152, 2013.

[4] Ronald Fagin, Joseph Y. Halpern, Yoram Moses, and Moshe Y. Vardi. *Reasoning about Knowledge*. MIT Press, 1995.

[5] Jie Fan, Yanjing Wang, and Hans van Ditmarsch. Knowing whether. *CoRR*, abs/1312.0, 2013.

[6] Michael J. Fischer and Richard E. Ladner. Propositional dynamic logic of regular programs. *Journal of Computer and System Sciences*, 18(2) :194–211, 1979.

[7] Jim Gray. Notes on data base operating systems. In *Operating Systems, An Advanced Course*, pages 393–481. Springer-Verlag, 1978.

[8] Joseph Y. Halpern and Yoram Moses. A guide to completeness and complexity for modal logics of knowledge and belief. *Artificial Intelligence*, 54(3) :319–379, 1992.

[9] Andreas Herzig. Logics of knowledge and action : critical analysis and challenges. *Journal of Autonomous Agents and Multi-Agent Systems*, pages 1–35, 2014. Online July 2, 2014, doi : 10.1007/s10458-014-9267-z.

[10] Andreas Herzig, Emiliano Lorini, Nicolas Troquard, and Frédéric Moisan. A dynamic logic of normative systems. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, pages 228–233, 2011.

[11] Jaakko Hintikka. *Knowledge and belief*. Cornell University Press, Ithaca, N.Y., 1962.

[12] Cor A. J. Hurkens. Spreading gossip efficiently. *Nieuw Archief voor Wiskunde*, 5/1(2) :208–210, 2000.

[13] Leslie Lamport, Robert Shostak, and Marshall Pease. The Byzantine Generals Problem. *ACM Transactions on Programming Languages and Systems*, 4(3) :382–401, 1982.

[14] Emiliano Lorini and Andreas Herzig. Direct and indirect common belief. In *Institutions, Emotions, and Group Agents*, pages 355–372. Springer Netherlands, 2014.

[15] Jan Plaza. Logics of public communications. In M. L. Emrich, M. S. Pfeifer, M. Hadzikadic, and Z.W. Ras, editors, *Proceedings of the 4th International Symposium on Methodologies for Intelligent Systems*, pages 201–216. Oak Ridge National Laboratory, ORNL/DSRD- 24, 1989.

[16] Wiebe van der Hoek, Petar Iliev, and Michael Wooldridge. A logic of revelation and concealment. In Wiebe van der Hoek, Lin Padgham, Vincent Conitzer, and Michael Winikoff, editors, *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems*, pages 1115–1122. IFAAMAS, 2012.

[17] Wiebe van der Hoek, Nicolas Troquard, and Michael Wooldridge. Knowledge and control. In Liz Sonenberg, Peter Stone, Kagan Tumer, and Pinar Yolum, editors, *Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems*, pages 719–726. IFAAMAS, 2011.

[18] Hans van Ditmarsch, Wiebe van der Hoek, and Barteld Kooi. *Dynamic Epistemic Logic*. Springer Publishing Company, Incorporated, 1st edition, 2007.

[19] Yanjing Wang, Floor Sietsma, and Jan van Eijck. Logic of information flow on communication channels. In Andrea Omicini, Sebastian Sardiña, and Wamberto Weber Vasconcelos, editors, *DALT*, pages 130–147. Springer, 2010.

Proofs

1 Semantics

1.1 Proposition 1

Proposition 1. *A valuation $V \subseteq ATM$ is introspective if and only if, for every $\alpha \in ATM$ and $i \in Agt$:*

$$\sigma\, S_i\, S_i\, \alpha \in V \text{ pour every } \sigma \in OBS^* \quad (1)$$

$$\sigma\, JS\, \alpha \in V \text{ pour every } \sigma \in OBS^+ \quad (2)$$

$$\text{if } \alpha \in V \text{ and } \alpha \rightsquigarrow \beta \text{ then } \beta \in V \quad (3)$$

Démonstration. First, we prove that constraints of Definition 1 implies these validities.

Suppose V is introspective.

We begin with the third validity. The interesting case is when $\alpha = JS\,\alpha'$. Suppose $JS\,\alpha' \in V$. By constraint (C5), we know that $JS\,S_{i_1}\,\alpha' \in V$, and also that $JS\,S_{i_2}\,S_{i_1}\,\alpha' \in V$ for any i_1 and i_2, and so on. By repeating the application of (C5), we can generate any $JS\,S_{i_n}...S_{i_1}\,\alpha'$ and then, by (C4), we can obtain $S_i\,S_{i_n}...S_{i_1}\,\alpha'$ for any agents $i_1,...i_n$ and i. Moreover, we have $JS\,JS\,\alpha' \in V$ by (C2), and in the same way, we can generate $S_i\,S_{i_n}...S_{i_1}\,JS\,\alpha'$ for any agents $i_1,...i_n$ and i. By choosing α' appropriately, we can therefore obtain any sequence containing any S_i and JS, that is, all $\sigma\,\alpha'$ for any $\sigma \in OBS^+$.

Using the same technique, we obtain $\sigma\,S_i\,S_i\,\alpha$ and $\sigma\,JS\,\alpha$, the first with (C1) (for σ empty) and (C3), and the second with (C2).

The other way round, constraints (C1), (C2) and (C3) are respectively instances of (1), (2) and (1) ; and constraints (C4) and (C5) are both guaranteed by (3). □

1.2 Proposition 2

Proposition 2. *The relation $\sim_{Agt}$ and every relation $\sim_i$ are equivalence relations on INTR.*

1.2.1 Relation $\sim_i$

Démonstration. We prove reflexivity, symmetry and transitivity of the relation $\sim_i$.

Reflexivity is guaranteed by the definition of $\sim_i$: for any α such that $S_i\,\alpha \in V$, the value stays the same when staying in the same valuation.

To prove symmetry, suppose $V \sim_i V'$. As V satisfies (C1), $S_i\,S_i\,\alpha \in V$ for every $\alpha \in ATM$, and thus $V(S_i\,\alpha) = V'(S_i\,\alpha)$. Therefore, for every α such that $S_i\,\alpha \in V$, $V(\alpha) = V'(\alpha)$ and $S_i\,\alpha \in V'$, and for every α such that $S_i\,\alpha \notin V$, $S_i\,\alpha \notin V'$. Therefore $V' \sim_i V$.

For transitivity, suppose $V \sim_i V'$ and $V' \sim_i V''$. As observed above, we have $V(S_i\,\alpha) = V'(S_i\,\alpha)$, and $V'(S_i\,\alpha) = V''(S_i\,\alpha)$, thus $V(S_i\,\alpha) = V''(S_i\,\alpha)$ for every $\alpha \in ATM$. Moreover, for every α such that $S_i\,\alpha \in V$, $V(\alpha) = V'(\alpha)$ and $S_i\,\alpha \in V'$, and thus $V'(\alpha) = V''(\alpha)$, so $V(\alpha) = V''(\alpha)$. Therefore $V \sim_i V'$. □

1.2.2 Relation $\sim_{Agt}$

Démonstration. Identical to the previous proof but with $\sim_{Agt}$, $JS\,\alpha$ and constraint $JS\,JS\,\alpha$ instead of $\sim_i$, $S_i\,\alpha$ and constraint $S_i\,S_i\,\alpha$. □

1.3 Lemma 1

Lemma 1. *Let $V \in INTR$, $V \sim_i V_1$ and $V \sim_{Agt} V_2$. Then $V_1 \in INTR$ and $V_2 \in INTR$.*

1.3.1 Relation $\sim_j$

Démonstration. Suppose $V \sim_j V_1$ and $V \in INTR$. We prove that V_1 satisfies the constraints (C1)-(C5) of Definition 1 thanks to validities of Proposition 1.

Consider an arbitrary agent i.

For (C1), we have $S_j\,S_i\,S_i\,\alpha \in V$ and $S_i\,S_i\,\alpha \in V$ due to validity (1). So $S_i\,S_i\,\alpha \in V_1$.

For (C2), we have $S_j\,JS\,JS\,\alpha \in V$ and $JS\,JS\,\alpha \in V$ due to validity (2). So $JS\,JS\,\alpha \in V_1$.

For (C3), we have $S_j\,JS\,S_i\,S_i\,\alpha \in V$ and $JS\,S_i\,S_i\,\alpha \in V$ due to validity (1). So $JS\,S_i\,S_i\,\alpha \in V_1$.

For (C4), suppose $JS\,\alpha \in V_1$. Since $S_j\,JS\,\alpha \in V$ (due to (2)), we must have $JS\,\alpha \in V$ (because $V \sim_j V_1$). Thus $S_j\,S_i\,\alpha \in V$ and $S_i\,\alpha \in V$ (validity (3)). So $S_i\,\alpha \in V_1$.

For (C5), suppose $JS\,\alpha \in V_1$. Since $S_j\,JS\,\alpha \in V$ (due to (2)), we must have $JS\,\alpha \in V$ (because $V \sim_j V_1$). Thus $JS\,S_i\,\alpha \in V$ (by (3)), and $S_j\,JS\,S_i\,\alpha \in V$ (by (2)). So $JS\,S_i\,\alpha \in V_1$.

Therefore $V_1 \in INTR$. □

1.3.2 Relation $\sim_{Agt}$

Démonstration. Suppose $V \sim_{Agt} V_2$. We prove that V_2 satisfies the constraints (C1)-(C5) of Definition 1 thanks to validities of Proposition 1.

Consider an arbitrary agent i.

For (C1), we have $JS\,S_i\,S_i\,\alpha \in V$ and $S_i\,S_i\,\alpha \in V$ due to validity (1). So $S_i\,S_i\,\alpha \in V_2$.

For (C2), we have $JS\,JS\,JS\,\alpha \in V$ and $JS\,JS\,\alpha \in V$ due to validity (2). So $JS\,JS\,\alpha \in V_2$.

For (C3), we have $JS\,JS\,S_i\,S_i\,\alpha \in V$ and $JS\,S_i\,S_i\,\alpha \in V$ due to validity (1). So $JS\,S_i\,S_i\,\alpha \in V_2$.

For (C4), suppose $JS\,\alpha \in V_2$. Since $JS\,JS\,\alpha \in V$ (due to (2)), we must have $JS\,\alpha \in V$ (because $V \sim_{Agt} V_2$). Thus $JS\,S_i\,\alpha \in V$ and $S_i\,\alpha \in V$ (validity (3)). So $S_i\,\alpha \in V_2$.

For (C5), suppose $JS\,\alpha \in V_2$. Since $JS\,JS\,\alpha \in V$ (due to (2)), we must have $JS\,\alpha \in V$ (because $V \sim_{Agt} V_2$). Thus $JS\,S_i\,\alpha \in V$ (by (3)), and $JS\,JS\,S_i\,\alpha \in V$ (by (2)). So $JS\,S_i\,\alpha \in V_2$.

Therefore $V_2 \in INTR$. □

1.4 Lemma 2

Lemma 2. *If $V \in INTR$ and $VR_\pi V'$ then $V' \in INTR$.*

Démonstration. Suppose π is of the form $+\alpha$.

— If α is of the form $JS\,\alpha'$, then every atom β such that $\alpha \rightsquigarrow \beta$ is added to V to obtain V'. Thus validities (1) and (2) hold since we did not subtract anything from V, and (3) holds because it was already satisfied in V and every consequence of α was added to V'.

— If α is not of the form $JS\,\alpha'$, then V' is simply equal to $V \cup \{\alpha\}$. Thus validities (1) and (2) are true because corresponding atoms were already present in V, and (3) holds because it was already satisfied in V and an atom without any consequences but itself was added.

Suppose π is of the form $-\alpha$.

— If α is valid in $INTR$ then there does not exist any valuation V' such that $VR_{-\alpha}V'$.

— If α is not valid in $INTR$, then all atoms having α as a consequence are removed, that is, any $JS\,\alpha'$ such that $\alpha = \sigma\,\alpha'$ and $\sigma \in OBS^+$, does not belong to V'. Thus, validities (1) and (2) are true in V' since atoms of these forms were not removed, and (3) is also true in V' because every $JS\,\alpha'$ that could imply $\sigma\,\alpha'$ was removed.

When π is of the form $\pi_1;\pi_2$ or $\pi_1 \sqcup \pi_2$, as by induction hypothesis R_{π_1} and R_{π_2} only relate valuations from $INTR$, their composition and union will only relate valuations from $INTR$. Finally, tests relate the current valuation to itself and thus stay in $INTR$.

Therefore $V' \in INTR$. □

1.5 Proposition 3

Proposition 3. *For every $V \in INTR$, $i \in Agt$ and π, V is only related to valuations in $INTR$ by $\sim_i$, $\sim_{Agt}$ and R_π.*

Démonstration. Direct consequence of lemmas 1 and 2. □

1.6 Proposition 4

Proposition 4. *Let φ be without epistemic operators. Let $V, V' \in 2^{ATM}$ such that $V(\alpha) = V'(\alpha)$ for every atom $\alpha \in ATM(\varphi)$. Then $V \models \varphi$ if and only if $V' \models \varphi$.*

Démonstration. We prove by mutual induction on formulas and programs that for any formula φ and program π :

— if $V(\alpha) = V'(\alpha)$ for every $\alpha \in ATM(\varphi)$ then $V \models \varphi$ if and only if $V' \models \varphi$, and

— if $V(\alpha) = V'(\alpha)$ for every $\alpha \in ATM(\pi) \cup A$ for every $A \in 2^{ATM}$, then for all $U \in 2^{ATM}$ such that $VR_\pi U$, there exists $U' \in 2^{ATM}$ such that $V'R_\pi U'$ and $U(\alpha) = U'(\alpha)$ for every $\alpha \in ATM(\pi) \cup A$.

□

1.6.1 Hypothesis for formulas

Démonstration. The base case is obvious : suppose $\varphi = \alpha$. Then $V(\alpha) = V'(\alpha)$ means $V \models \alpha$ if and only if $V' \models \alpha$.

First, suppose $\varphi = \neg\varphi'$ and $V(\alpha) = V'(\alpha)$ for every $\alpha \in ATM(\neg\varphi') = ATM(\varphi')$. Our induction hypothesis implies $V \models \varphi'$ if and only if $V' \models \varphi'$, thus $V \not\models \varphi'$ if and only if $V' \not\models \varphi'$, which is equivalent to $V \models \neg\varphi'$ if and only if $V' \models \neg\varphi'$.

Now suppose $\varphi = \varphi_1 \wedge \varphi_2$ and $V(\alpha) = V'(\alpha)$ for every $\alpha \in ATM(\varphi_1 \wedge \varphi_2) = ATM(\varphi_1) \cup ATM(\varphi_2)$. Our induction hypothesis implies both $V \models \varphi_1$ if and only if $V' \models \varphi_1$ and $V \models \varphi_2$ if and only if $V' \models \varphi_2$. Therefore $V \models \varphi_1$ and $V \models \varphi_2$ if and only if $V' \models \varphi_1$ and $V' \models \varphi_2$, which is equivalent to $V \models \varphi_1 \wedge \varphi_2$ if and only if $V' \models \varphi_1 \wedge \varphi_2$.

Finally, suppose $\varphi = [\pi]\varphi'$.

Suppose $V \not\models [\pi]\varphi'$ and $V' \models [\pi]\varphi'$.

The first is equivalent to : there exists a valuation $U \in 2^{ATM}$ such that $VR_\pi U$ and $U \not\models \varphi'$. Our induction hypothesis for programs implies that there exists $U' \in 2^{ATM}$ such that $V'R_\pi U'$ and $U(\alpha) = U'(\alpha)$ for every $\alpha \in ATM(\pi) \cup A$, for every $A \in 2^{ATM}$, and in particular, for every $\alpha \in ATM(\varphi') \cup ATM(\pi)$. Then, our induction hypothesis on formulas implies that $U \models \varphi'$ if and only if $U' \models \varphi'$.

Since we know that $U \not\models \varphi'$, we have : there exists a valuation $U' \in 2^{ATM}$ such that $V'R_\pi U'$ and $U' \not\models \varphi'$. This is equivalent to $V' \not\models [\pi]\varphi'$, which gives a contradiction with our hypothesis. Therefore $V \models [\pi]\varphi'$ if and only if $V' \models [\pi]\varphi'$. □

1.6.2 Hypothesis for programs

Démonstration. First, take $\pi = +\alpha$ with $V(\alpha') = V'(\alpha')$ for every $\alpha' \in ATM(+\alpha) \cup A = \{\alpha\} \cup A$, for every $A \in 2^{ATM}$. Thus $U = V \cup \mathit{Eff}^+(\alpha)$.

Let us take $U' = V' \cup \mathit{Eff}^+(\alpha)$. Then we have $V'R_{+\alpha}U'$ (by definition of U') and $U(\alpha') = U'(\alpha')$

for every $\alpha' \in \{\alpha\} \cup A = ATM(+\alpha) \cup A$ for every $A \in 2^{ATM}$ since $V(\alpha') = V'(\alpha')$ for every $\alpha' \in \{\alpha\} \cup A$.

Now suppose $\pi = -\alpha$ with $V(\alpha') = V'(\alpha')$ for every $\alpha' \in ATM(-\alpha) \cup A = \{\alpha\} \cup A$, for every $A \in 2^{ATM}$.

If α is valid in $INTR$, then V is not related to any valuation and the hypothesis is satisfied.

If α is not valid, then the proof is analogous to the previous one : we have $U = V \setminus \mathit{Eff}^-(\alpha)$. Let us take $U' = V' \setminus \mathit{Eff}^-(\alpha)$. Then $V' R_{-\alpha} U'$ (by definition of U') and $U(\alpha') = U'(\alpha')$ for every $\alpha' \in \{\alpha\} \cup A = ATM(-\alpha) \cup A$ for every $A \in 2^{ATM}$ since $V(\alpha') = V'(\alpha')$ for every $\alpha' \in \{\alpha\} \cup A$.

Suppose $\pi = \pi_1; \pi_2$ with $V(\alpha) = V'(\alpha)$ for every $\alpha \in ATM(\pi_1; \pi_2) \cup A = ATM(\pi_1) \cup ATM(\pi_2) \cup A$, for every $A \in 2^{ATM}$. $V R_{\pi_1;\pi_2} U$ is equivalent to : there exists a valuation $V_1 \in 2^{ATM}$ such that $V R_{\pi_1} V_1$ and $V_1 R_{\pi_2} U$.

Since $V(\alpha) = V'(\alpha)$ for every $\alpha \in ATM(\pi_1) \cup A$ and $V R_{\pi_1} V_1$, then by induction hypothesis, there exists a valuation $V_1' \in 2^{ATM}$ such that $V' R_{\pi_1} V_1'$ and $V_1(\alpha) = V_1'(\alpha)$ for every $\alpha \in ATM(\pi_1) \cup A$, for every $A \in 2^{ATM}$. Thus, this property holds for every $\alpha \in ATM(\pi_1) \cup ATM(\pi_2) \cup A$ and therefore for every $\alpha \in ATM(\pi_2) \cup A$.

Since $V_1 R_{\pi_2} U$, there exists $U' \in 2^{ATM}$ such that $V_1' R_{\pi_2} U'$ and $U(\alpha) = U'(\alpha)$ for every $\alpha \in ATM(\pi_2) \cup A$, for every $A \in 2^{ATM}$, that is for every $\alpha \in ATM(\pi_1) \cup ATM(\pi_2) \cup A = ATM(\pi_1; \pi_2) \cup A$. Therefore, $V' R_{\pi_1;\pi_2} U'$ and $U(\alpha) = U'(\alpha)$ for every $\alpha \in ATM(\pi_1; \pi_2) \cup A$.

Suppose $\pi = \pi_1 \sqcup \pi_2$ with $V(\alpha) = V'(\alpha)$ for every $\alpha \in ATM(\pi_1 \sqcup \pi_2) \cup A = ATM(\pi_1) \cup ATM(\pi_2) \cup A$, for every $A \in 2^{ATM}$. $V R_{\pi_1 \sqcup \pi_2} U$ is equivalent to $V R_{\pi_1} U$ or $V R_{\pi_2} U$.

Since $V(\alpha) = V'(\alpha)$ for every $\alpha \in ATM(\pi_1) \cup A$ and $V R_{\pi_1} U$, then by induction hypothesis, there exists a valuation $U' \in 2^{ATM}$ such that $V' R_{\pi_1} U'$ and $U(\alpha) = U'(\alpha)$ for every $\alpha \in ATM(\pi_1) \cup A$, for every $A \in 2^{ATM}$, and thus, for every $\alpha \in ATM(\pi_1) \cup ATM(\pi_2) \cup A$.

On the other hand, since $V(\alpha) = V'(\alpha)$ for every $\alpha \in ATM(\pi_2) \cup A$ and $V R_{\pi_2} U$, then by induction hypothesis, there exists a valuation $U' \in 2^{ATM}$ such that $V' R_{\pi_2} U'$ and $U(\alpha) = U'(\alpha)$ for every $\alpha \in ATM(\pi_2) \cup A$, for every $A \in 2^{ATM}$, and thus, for every $\alpha \in ATM(\pi_1) ATM(\pi_2) \cup A$.

Thus, we have that there exists a valuation $U' \in 2^{ATM}$ such that $V' R_{\pi_1} U'$ or $V' R_{\pi_2} U'$ and $U(\alpha) = U'(\alpha)$ for every $\alpha \in ATM(\pi_1) \cup ATM(\pi_2) \cup A = ATM(\pi_1 \sqcup \pi_2) \cup A$, and therefore, $V' R_{\pi_1 \sqcup \pi_2} U'$ and $U(\alpha) = U'(\alpha)$ for every $\alpha \in ATM(\pi_1 \sqcup \pi_2) \cup A$.

Finally, suppose $\pi = \varphi?$ with $V(\alpha) = V'(\alpha)$ for every $\alpha \in ATM(\varphi?) \cup A = ATM(\varphi) \cup A$, for every $A \in 2^{ATM}$. First, suppose $V \not\models \varphi$, then V is not related to any valuation and the hypothesis is satisfied. If $V \models \varphi$, then $V R_{\varphi?} U$ is equivalent to $V = U$. By induction hypothesis on formulas, we know that since $V \models \varphi$ and $V(\alpha) = V'(\alpha)$ for every $\alpha \in ATM(\varphi)$, then $V' \models \varphi'$.

Let us take $U' = V'$, then $V' R_{\varphi?} U'$ and $U(\alpha) = U'(\alpha)$ for every $\alpha \in ATM(\varphi) \cup A = ATM(\varphi?) \cup A$ (because $V = U$). □

2 Axiomatization

2.1 Proposition 5

Proposition 5. *The following formulas are* DEL-PAO*-valid, where α is atomic and A^+ and A^- are sets of atoms such that $A^+ \cap A^- = \emptyset$:*

$$K_i \alpha \leftrightarrow S_i\, \alpha \wedge \alpha \qquad (Red_{K,\alpha})$$

$$CK\, \alpha \leftrightarrow JS\, \alpha \wedge \alpha \qquad (Red_{CK,\alpha})$$

$$K_i \neg\alpha \leftrightarrow S_i\, \alpha \wedge \neg\alpha \qquad (Red_{K,\neg})$$

$$CK \neg\alpha \leftrightarrow JS\, \alpha \wedge \neg\alpha \qquad (Red_{CK,\neg})$$

$$K_i(\varphi \wedge \varphi') \leftrightarrow K_i\varphi \wedge K_i\varphi' \qquad (Red_{K,\wedge})$$

$$CK(\varphi \wedge \varphi') \leftrightarrow CK\varphi \wedge CK\varphi' \qquad (Red_{CK,\wedge})$$

$$K_i \left(\bigvee_{\alpha \in A^+} \alpha \vee \bigvee_{\alpha \in A^-} \neg\alpha \right) \leftrightarrow \left(\bigvee_{\alpha \in A^+} K_i \alpha \right) \vee \left(\bigvee_{\alpha \in A^-} K_i \neg\alpha \right) \qquad (Red_{K,\vee})$$

$$CK \left(\bigvee_{\alpha \in A^+} \alpha \vee \bigvee_{\alpha \in A^-} \neg\alpha \right) \leftrightarrow \left(\bigvee_{\alpha \in A^+} CK \alpha \right) \vee \left(\bigvee_{\alpha \in A^-} CK \neg\alpha \right) \qquad (Red_{CK,\vee})$$

Démonstration. We prove every equivalence.

$Red_{K,\alpha}$: If $V \models S_i\, \alpha \wedge \alpha$, that is, $S_i\, \alpha \in V$ and $V \models \alpha$, then $V' \models \alpha$ for every valuation $V \sim_i V'$, by the definition of $\sim_i$, which is equivalent to $V \models K_i \alpha$.

The other way around, suppose $V' \models \alpha$ for every valuation $V \sim_i V'$ and $V \not\models S_i\, \alpha \wedge \alpha$ (thus, either $V \not\models S_i\, \alpha$ or $V \not\models \alpha$).

- Suppose $S_i\, \alpha \notin V$, then there exists a valuation V' related to V such that $V' \not\models \alpha$ (contradiction).
- Now suppose $\alpha \notin V$ and $S_i\, \alpha \in V$, then for every valuation V' related to V, $V' \not\models \alpha$ (contradiction).

($Red_{CK,\alpha}$) : Identical to the previous proof but with $JS\, \alpha$, CK and $\sim_{Agt}$ instead of $S_i\, \alpha$, K_i and $\sim_i$.

($Red_{K,\neg}$) : Analogous to the proof of $Red_{K,\alpha}$ but with the opposite value of α.

($Red_{CK,\neg}$) : Identical to the previous proof but with $JS\, \alpha$, CK and $\sim_{Agt}$ instead of $S_i\, \alpha$, K_i and $\sim_i$.

($Red_{K,\wedge}$) : The proof is standard.

($Red_{CK,\wedge}$) : Identical to the previous proof but with CK and $\sim_{Agt}$ instead of K_i and $\sim_i$.

($Red_{K,\vee}$) : The '$\leftarrow$' direction is obvious : it is already valid in the basic modal logic K. For the '$\rightarrow$' direction, take an arbitrary valuation $V \in INTR$ and suppose

$$V \models K_i \left(\bigvee_{\alpha \in A^+} \alpha \vee \bigvee_{\alpha \in A^-} \neg\alpha \right)$$

and

$$V \models \bigwedge_{\alpha \in A^+} \widehat{K}_i \neg\alpha \wedge \bigwedge_{\alpha \in A^-} \widehat{K}_i \alpha.$$

Let us define the four following sets of atoms :

$$A^+_S = \{\alpha : \alpha \in A^+ \text{ and } S_i\,\alpha \in V\}$$
$$A^+_{\neg S} = \{\alpha : \alpha \in A^+ \text{ and } S_i\,\alpha \notin V\}$$
$$A^-_S = \{\alpha : \alpha \in A^- \text{ and } S_i\,\alpha \in V\}$$
$$A^-_{\neg S} = \{\alpha : \alpha \in A^- \text{ and } S_i\,\alpha \notin V\}$$

These sets are mutually disjoint because A^+ and A^- are. Then we can reformulate our second hypothesis by splitting each conjunction into two parts :

$$V \models \bigwedge_{\alpha \in A^+_S} \widehat{K}_i \neg\alpha \wedge \bigwedge_{\alpha \in A^-_S} \widehat{K}_i \alpha \wedge \bigwedge_{\alpha \in A^+_{\neg S}} \widehat{K}_i \neg\alpha \wedge \bigwedge_{\alpha \in A^-_{\neg S}} \widehat{K}_i \alpha.$$

For every $\alpha \in A^+_S \cup A^-_S$, $S_i\,\alpha \in V$ and therefore, since $V \models \widehat{K}_i \neg\alpha$ and $V \models \widehat{K}_i \alpha$, we have for every V' related to V : $V' \models \neg\alpha$ for all $\alpha \in A^+_S$; and $V' \models \alpha$ for all $\alpha \in A^-_S$. In words, the values of atoms from $A^+_S \cup A^-_S$ are identical in every valuation related to V.

Since $S_i\,\alpha \notin V$ for every $\alpha \in A^+_{\neg S} \cup A^-_{\neg S}$, all these α can vary : the current valuation is related to other valuations where they are true or false. In particular, there exists a valuation V'' such that $V'' = (V \setminus A^+_{\neg S}) \cup A^-_{\neg S}$ and $V \sim_i V''$. Therefore $V'' \models \neg\alpha$ for every $\alpha \in A^+ = A^+_S \cup A^+_{\neg S}$, and $V'' \models \alpha$ for every $\alpha \in A^- = A^-_S \cup A^-_{\neg S}$ which contradicts our first hypothesis.

($Red_{CK,\vee}$) : Identical to the previous proof but with CK and $\sim_{Agt}$ instead of K_i and $\sim_i$. □

2.2 Proposition 6

Proposition 6. *The following formulas are* DEL-PAO*-valid.*

$$[\pi;\pi']\varphi \leftrightarrow [\pi][\pi']\varphi \qquad (Red_{;})$$
$$[\pi \sqcup \pi']\varphi \leftrightarrow [\pi]\varphi \wedge [\pi']\varphi \qquad (Red_{\sqcup})$$
$$[\varphi?]\varphi' \leftrightarrow \varphi \rightarrow \varphi' \qquad (Red_{?})$$
$$[+\alpha]\neg\varphi \leftrightarrow \neg[+\alpha]\varphi \qquad (Red_{+\alpha,\neg})$$
$$[-\alpha]\neg\varphi \leftrightarrow \begin{cases} \top & \text{if } \alpha \text{ is valid in } INTR \\ \neg[-\alpha]\varphi & \text{otherwise} \end{cases} \qquad (Red_{-\alpha,\neg})$$
$$[+\alpha](\varphi \wedge \varphi') \leftrightarrow [+\alpha]\varphi \wedge [+\alpha]\varphi' \qquad (Red_{+\alpha,\wedge})$$
$$[-\alpha](\varphi \wedge \varphi') \leftrightarrow [-\alpha]\varphi \wedge [-\alpha]\varphi' \qquad (Red_{-\alpha,\wedge})$$
$$[+\alpha]\beta \leftrightarrow \begin{cases} \top & \text{if } \alpha \rightsquigarrow \beta \\ \beta & \text{otherwise} \end{cases} \qquad (Red_{+\alpha})$$
$$[-\alpha]\beta \leftrightarrow \begin{cases} \top & \text{if } \alpha \text{ is valid in } INTR \\ \bot & \text{if } \alpha \text{ is not valid in } INTR \\ & \text{and } \beta \rightsquigarrow \alpha \\ \beta & \text{otherwise} \end{cases} \qquad (Red_{-\alpha})$$

Démonstration. We prove equivalences which are specific to DEL-PAO.

($Red_{;}$) : PDL validity.

($Red_{\sqcup}$) : PDL validity.

($Red_{?}$) : PDL validity.

($Red_{+\alpha,\neg}$) : This equivalence is valid because $R_{+\alpha}$ is a function : it relates every valuation to exactly one valuation. Indeed, $V \models [+\alpha]\neg\varphi$ is equivalent to $V \cup \mathit{Eff}^+(\alpha) \models \neg\varphi$, that is, $V \cup \mathit{Eff}^+(\alpha) \not\models \varphi$ which is equivalent to $V \models \neg[+\alpha]\varphi$.

($Red_{-\alpha,\neg}$) : Unlike $+\alpha$, the program $-\alpha$ brings about two cases : if α is valid, then no valuation is related to V and the formula is equivalent to $\top$. Otherwise, as for $+\alpha$, $V \models [-\alpha]\neg\varphi$ is equivalent to $V \setminus \mathit{Eff}^-(\alpha) \models \neg\varphi$, that is, $V \setminus \mathit{Eff}^-(\alpha) \not\models \varphi$ which is equivalent to $V \models \neg[-\alpha]\varphi$.

($Red_{+\alpha,\wedge}$) : PDL validity.

($Red_{-\alpha,\wedge}$) : PDL validity.

($Red_{+\alpha}$) : Suppose $V \models [+\alpha]\beta$, which is equivalent to $V \cup \mathit{Eff}^+(\alpha) \models \beta$. If $\alpha \rightsquigarrow \beta$, which means by definition that $\beta \in \mathit{Eff}^+(\alpha)$, then the formula is satisfied because β was already in V or added. Otherwise, since β was not added to V, then it is satisfied in $V \cup \mathit{Eff}^+(\alpha)$ if and only if it was satisfied in V, so the formula is equivalent to β.

($Red_{-\alpha}$) : Suppose $V \models [-\alpha]\beta$. If α is valid in $INTR$, then there is no valuation related to V, which means that every related valuation satisfies β, so the formula is satisfied. Now suppose β is not valid in $INTR$, thus our hypothesis is equivalent to $V \setminus \mathit{Eff}^-(\alpha) \models \beta$. As for the previous validity, if $\beta \rightsquigarrow \alpha$, which means by definition that $\beta \in \mathit{Eff}^-(\alpha)$, then the formula cannot be satisfied because β was already not in V or removed. Otherwise, since β was not removed from V, then it is satisfied in $V \cup \mathit{Eff}^+(\alpha)$ if and only if it was satisfied in V, so the formula is equivalent to β. □

2.3 Theorem 1

Theorem 1. *For every φ there exists a formula without modal operators φ' such that $\varphi \leftrightarrow \varphi'$ is valid in $INTR$.*

Démonstration. While there is a modal operator in φ, take the innermost one ($K_i\varphi'$, $CK\varphi'$ or $[\pi]\varphi'$ such that φ' does not contain modal operators) :

— in the case of $K_i\varphi'$, put φ' in conjunctive normal form and eliminate the epistemic operator by applying equivalences $(Red_{K,\wedge})$, $(Red_{K,\vee})$, $(Red_{K,\alpha})$ and $(Red_{K,\neg})$ of Proposition 5 ;

— in the case of $CK\varphi'$, put φ' in conjunctive normal form and eliminate the epistemic operator by applying equivalences $(Red_{CK,\wedge})$, $(Red_{CK,\vee})$, $(Red_{CK,\alpha})$ and $(Red_{CK,\neg})$ of Proposition 5 ;

— in the case of $[\pi]\varphi'$, eliminate the dynamic operator by applying equivalences of Proposition 6.

So by iterating elimination of innermost modal operators we obtain a formula without modal operators. □

2.4 Proposition 7

Proposition 7. *The following formulas are* DEL-PAO*-valid, for every $i \in Agt$ and $\alpha \in ATM$:*

$$S_i\, S_i\, \alpha \qquad (Vis_{C1})$$
$$JS\, JS\, \alpha \qquad (Vis_{C2})$$
$$JS\, S_i\, S_i\, \alpha \qquad (Vis_{C3})$$
$$JS\, \alpha \rightarrow S_i\, \alpha \qquad (Vis_{C4})$$
$$JS\, \alpha \rightarrow JS\, S_i\, \alpha \qquad (Vis_{C5})$$

Démonstration. These five formulas are the syntactical counterparts of the five constraints (C1)-(C5) of Definition 1. □

2.5 Proposition 8

Proposition 8. *For φ without modal operators, φ is valid in INTR if and only if $\mathcal{T}_{vis} \models_{\mathsf{CPL}} \varphi$, where $\models_{\mathsf{CPL}}$ is logical consequence in classical propositional logic.*

Démonstration. The theory $\mathcal{T}_{vis}$ corresponds exactly to the constraints defining the set of introspective valuations *INTR*. □

2.6 Proposition 9

Proposition 9. *For formulas φ without modal operators, $\vdash_{\mathsf{DEL\text{-}PAO}} \varphi$ if and only if $\mathcal{T}_{vis} \vdash_{\mathsf{CPL}} \varphi$.*

Démonstration. For the left-to-right direction, first, if φ is DEL-PAO-provable then φ is DEL-PAO-valid, since our axioms are sound and rules of inference preserve validity.

Moreover, if φ is DEL-PAO-valid then φ is CPL-valid in the $\mathcal{T}_{vis}$ models. This is trivial because the models are isomorphic. (It is actually an equivalence.)

Then, if φ is CPL-valid in the $\mathcal{T}_{vis}$ models then φ is derivable from $\mathcal{T}_{vis}$ in CPL, by completeness of classical propositional logic. (It is again actually an equivalence.)

Thus, if φ is DEL-PAO-provable then φ is derivable from $\mathcal{T}_{vis}$ in CPL.

For the right-to-left direction, if φ is derivable from $\mathcal{T}_{vis}$ in CPL then φ is DEL-PAO-provable is obviously true because the axiomatics of DEL-PAO contains the elements of $\mathcal{T}_{vis}$ as schemas. □

2.7 Theorem 2

Theorem 2. *The axiomatization of* DEL-PAO *is sound and complete.*

Démonstration. Let φ be a DEL-PAO formula. Let φ' be its reduction to a boolean formula.

To prove soundness, suppose φ is a theorem of DEL-PAO. By the Reduction Theorem 1, φ' is a theorem of DEL-PAO, too. By Proposition 9 we have $\mathcal{T}_{vis} \vdash_{\mathsf{CPL}} \varphi'$. Then $\mathcal{T}_{vis} \models_{\mathsf{CPL}} \varphi'$ by soundness of classical propositional logic. Hence φ' is valid in *INTR* by Proposition 8. Finally, φ is valid in *INTR* because all the equivalences used in the Reduction Theorem 1 are valid in *INTR*.

The completeness proof follows the lines of the soundness proof in reverse order, resorting to completeness of classical logic instead of soundness. □

3 Complexity

3.1 Theorem 3

Theorem 3. *The* DEL-PAO *satisfiability and* DEL-PAO *model checking problems are both* PSPACE*-complete.*

Démonstration. Proven by the following results. □

3.2 Proposition 10

Proposition 10. *Let φ be a* DEL-PAO *formula without epistemic operators such that $ATM(\varphi) = \{\alpha_1, \ldots, \alpha_n\}$. Then φ is satisfiable in INTR if and only if the formula*

$$\langle (+\alpha_1 \sqcup -\alpha_1); \ldots; (+\alpha_n \sqcup -\alpha_n) \rangle \varphi$$

is valid in INTR.

Démonstration. Suppose there exists a valuation $V \in INTR$ such that $V \models \varphi$. We take an arbitrary valuation U and we construct the valuation $V' = (U \cup \{\alpha : \alpha \in ATM(\varphi) \text{ and } \alpha \in V\}) \setminus \{\alpha : \alpha \in ATM(\varphi) \text{ and } \alpha \notin V\}$. By construction, we have :

— $U R_{(+\alpha_1 \sqcup -\alpha_1);\ldots;(+\alpha_n \sqcup -\alpha_n)} V'$ and

— $V(\alpha) = V'(\alpha)$ for every $\alpha \in ATM(\varphi)$.

The second item is equivalent, by Proposition 4, to $V \models \varphi$ if and only if $V' \models \varphi$, thus $V' \models \varphi$. Therefore, for every valuation $U \in INTR$, there exists a valuation V' such that $UR_{(+\alpha_1 \sqcup -\alpha_1);\ldots;(+\alpha_n \sqcup -\alpha_n)}V'$ and $V' \models \varphi$, which is the truth condition of $U \models \langle (+\alpha_1 \sqcup -\alpha_1); \ldots; (+\alpha_n \sqcup -\alpha_n)\rangle\varphi$. □

3.3 Lower bound

Proposition. *Let φ be a quantified Boolean formula built over the set of propositional variables Prop. We have :*

$$V \models_{\mathsf{QBF}} \varphi \Longleftrightarrow V \models_{\mathsf{DEL\text{-}PAO}} tr_\pi(\varphi)$$

with $tr_\pi(p) = p$ and $tr_\pi(\forall p.\varphi) = [+p \sqcup -p]tr_\pi(\varphi)$; the Boolean connectives are translated homomorphically.

Démonstration. The case of propositional variables and boolean operators is obvious since they are translated homomorphically.

$V \models_{\mathsf{QBF}} \forall p.\varphi$ is verified if and only if φ is true when p is replaced by $\top$ and when p is replaced by $\bot$, which means : $V \models_{\mathsf{QBF}} \forall p.\varphi$ if and only if

$$V \cup \{p\} \models_{\mathsf{QBF}} \varphi \text{ and } V \setminus \{p\} \models_{\mathsf{QBF}} \varphi.$$

By induction hypothesis, this is equivalent to :
$V \cup \{p\} \models_{\mathsf{DEL\text{-}PAO}} tr_\pi(\varphi)$ and $V \setminus \{p\} \models_{\mathsf{DEL\text{-}PAO}} tr_\pi(\varphi)$, which means :

$$V \models_{\mathsf{DEL\text{-}PAO}} [+p]tr_\pi(\varphi) \wedge [-p]tr_\pi(\varphi),$$

that is :

$$V \models_{\mathsf{DEL\text{-}PAO}} [+p \sqcup -p]tr_\pi(\varphi).$$

□

This proves that the fragment without epistemic operators has a PSPACE-hard satisfiability problem.

We know prove the same for the fragment without dynamic operators.

Proposition. *Let φ be a quantified Boolean formula built over the set of propositional variables Prop. Let the set of agents be defined as $Agt = \{i_p : p \in Prop\}$. With $Diff = \{S_{i_p} p' : p, p' \in Prop, p \neq p'\}$ a set of atoms and $V \subseteq Prop$ a valuation :*

$$V \models_{\mathsf{QBF}} \varphi \Longleftrightarrow V' \models_{\mathsf{DEL\text{-}PAO}} tr_K(\varphi)$$

where $V' = V \cup \{\alpha : \alpha \text{ is valid in } INTR\} \cup Diff$, and for all $p \in Prop$, $tr_K(p) = p$ and $tr_K(\forall p.\varphi) = K_{i_p} tr_K(\varphi)$; the Boolean connectives are translated homomorphically.

Démonstration. The case of propositional variables and boolean operators is obvious since they are translated homomorphically.

Now suppose $V \models \varphi \Longleftrightarrow V' \models tr_K(\varphi)$ where φ does not contain any quantifier. We can remark that $V' \in INTR$, since it contains all valid atoms and no $JS\,\alpha$ (thus no consequence is needed). We prove that $V \models \forall p.\varphi$, with $p \in Prop$ an arbitrary propositional variable, if and only if $V' \models K_{i_p} tr_K(\varphi)$.

We saw in the previous proof that the left part, $V \models \forall p.\varphi$, is true if and only if $V \cup \{p\} \models \varphi$ and $V \setminus \{p\} \models \varphi$.

The right part, $K_{i_p} tr_K(\varphi)$, is equivalent to : for all V' such that $V \sim_{i_p} V'$, $V' \models tr_K(\varphi)$. However, since we have $V' \not\models S_{i_p} p$ (because $V \subseteq Prop$ and $S_{i_p} p$ was not added between V and V') and, for each $p' \neq p$, $V' \models S_{i_p} p'$, we know that each $p' \neq p$ must have the same value in V and in any V' such that $V \sim_{i_p} V'$, and that p can have any value. Thus, this is equivalent to $V \cup \{p\} \models tr_K(\varphi)$ and $V \setminus \{p\} \models tr_K(\varphi)$. Therefore $V \models \forall p.\varphi \Longleftrightarrow V' \models K_{i_p} tr_K(\varphi)$. □

3.4 Proposition 11

Proposition 11 ([3], Proposition 1)**.** *Let $V, V' \in 2^{ATM}$ such that $V(\alpha) = V'(\alpha)$ for every atom $\alpha \in ATM(\varphi)$. Then $V \models_{\mathsf{DL\text{-}PA}} \varphi$ if and only if $V' \models_{\mathsf{DL\text{-}PA}} \varphi$.*

Démonstration. From DL-PA. □

3.5 Proposition 12

Lemma 3. *$VR_{\pi_{i,A}}V'$ if and only if $V(\alpha) = V'(\alpha)$ for every $\alpha \in ATM$ such that $\alpha \notin A$ or $S_i\,\alpha \in V$.*

Démonstration. By definition, the program $\pi_{i,A}$ only considers atoms α from A and changes the ones such that $\neg S_i\,\alpha?$ succeeds, that is, atoms contained in A and for which $S_i\,\alpha$ is false. Other atoms remain the same. □

Proposition 12. *Let φ be a* DEL-PAO *formula without epistemic operators. Then the equivalences*

$$K_i\varphi \leftrightarrow [\pi_{i,ATM(\varphi)}]\varphi$$
$$CK\varphi \leftrightarrow [\pi_{Agt,ATM(\varphi)}]\varphi$$

are valid in INTR.

Démonstration. Suppose $V \models K_i\varphi$ with $V \in INTR$. This is equivalent to, for every $V' \in INTR$ such that $V \sim_i V'$, $V' \models \varphi$, that is :

for every $V' \in INTR$ such that for every $\alpha \in ATM$,
if $S_i\,\alpha \in V$ then $V(\alpha) = V'(\alpha)$, $V' \models \varphi$.

For the other side of the equivalence, we want $V \models [\pi_{i,ATM(\varphi)}]\varphi$, which is equivalent to for every $V' \in INTR$ such that $VR_{\pi_{i,ATM(\varphi)}}V'$, $V' \models \varphi$, that is, by Lemma 3 :

for every $V' \in INTR$ such that for every $\alpha \in ATM$,
if $\alpha \notin ATM(\varphi)$ or $S_i\,\alpha \in V$, then $V(\alpha) = V'(\alpha)$, $V' \models \varphi$,

or, in other words :

for every $V' \in INTR$ such that for every $\alpha \in ATM$,
if $\alpha \notin ATM(\varphi)$ then $V(\alpha) = V'(\alpha)$
and if $S_i\,\alpha \in V$ then $V(\alpha) = V'(\alpha)$,
$V' \models \varphi$.

Since atoms are ordered by their length, assignments of $\pi_{i,\alpha}$ cannot interfere with tests of $\pi_{i,\alpha'}$ if α is "smaller" than α'.

Therefore, we want to prove that, for an arbitrary $V \in INTR$, the first statement is equivalent to the latter.

The left-to-right direction is obvious.

For the opposite direction, suppose there exists a valuation $U \in INTR$ such that for every $\alpha \in ATM$, if $S_i\,\alpha \in V$ then $V(\alpha) = U(\alpha)$ and $U \not\models \varphi$.

Let us define $U' = (U \cup \{\alpha \notin ATM(\varphi) : \alpha \in V\}) \setminus \{\alpha \notin ATM(\varphi) : \alpha \notin V\}$. By definition, $V(\alpha) = U'(\alpha)$ for every $\alpha \notin ATM(\varphi)$ and moreover, $V(\alpha) = U'(\alpha)$ whenever $S_i\,\alpha \in V$. Therefore, by hypothesis, $U' \models \varphi$.

However, by Proposition 4, and since atoms from $ATM(\varphi)$ are not modified between U and U', $U \models \varphi$ if and only if $U' \models \varphi$, which produces a contradiction since we supposed $U \not\models \varphi$.

Therefore the equivalence holds.

The proof for the second program is identical but with CK, JS and $\sim_{Agt}$ instead of K_i, S_i and $\sim_i$. □

3.6 Proposition 13

Proposition 13. *For every* DEL-PAO *formula* φ*, there exists a* DEL-PAO *formula* φ' *without epistemic operators such that* $\varphi \leftrightarrow \varphi'$ *is valid in INTR. The length of* φ' *is polynomial in* $length(\varphi)$*.*

Démonstration. This formula is computed by Procedure 1, which terminates and is correct by Proposition 12. The length of φ' is quadratic in $length(\varphi)$. □

3.7 Proposition 14

Proposition 14. *Let* φ *be a* DEL-PAO *formula without epistemic operators. Then* $V \models_{\text{DEL-PAO}} \varphi$ *if and only if* $V \models_{\text{DL-PA}} tr(\varphi)$*.*

Démonstration. We prove by mutual induction on formulas and programs that for any formula φ without epistemic operators and for any program π :

— $V \models_{\text{DEL-PAO}} \varphi$ if and only if $V \models_{\text{DL-PA}} tr(\varphi)$, and

— $V R^{\text{DEL-PAO}}_{\pi} V'$ if and only if for every $A \subseteq ATM$ such that $ATM(\varphi) \subseteq A$, $(V \cap A) R^{\text{DL-PA}}_{tr(\pi,A)} (V' \cap A)$.

□

3.7.1 Hypothesis for formulas

Démonstration. The case of propositional variables and boolean operators is obvious since they are translated homomorphically.

For dynamic operators, we have to show :

$$V \models_{\text{DEL-PAO}} [\pi]\varphi \text{ if and only if } V \models_{\text{DL-PA}} [tr(\pi, ATM(\varphi))]tr(\varphi).$$

In virtue of the truth conditions, the left hand side means :

$V' \models_{\text{DEL-PAO}} \varphi$ for every V' such that $V R^{\text{DEL-PAO}}_{\pi} V'$.

By induction hypothesis, the latter is equivalent to :

$$V' \models_{\text{DL-PA}} tr(\varphi) \text{ for every } V' \text{ and } ATM(\varphi) \subseteq A \text{ such that } (V \cap A) R^{\text{DL-PA}}_{tr(\pi,A)} (V' \cap A).$$

By Proposition 11, this is equivalent to

$$V' \models_{\text{DL-PA}} tr(\varphi) \text{ for every } V' \text{ such that } (V \cap ATM(\varphi)) R^{\text{DL-PA}}_{tr(\pi, ATM(\varphi))} (V' \cap ATM(\varphi))$$

which is equivalent, again by Proposition 11, to

$V' \models_{\text{DL-PA}} tr(\varphi)$ for every V' such that $V R^{\text{DL-PA}}_{tr(\pi, ATM(\varphi))} V'$

and therefore to $V \models_{\text{DL-PA}} [tr(\pi, ATM(\varphi))]tr(\varphi)$. □

3.7.2 Hypothesis for programs

Démonstration. First, when π is of the form $+\alpha$ then $V R^{\text{DEL-PAO}}_{+\alpha} V'$ if and only if $V' = V \cup \mathit{Eff}^{+}(\alpha)$. The latter is the case if and only if for all $ATM(\varphi) \subseteq A$, $V' \cap A = (V \cup \mathit{Eff}^{+}(\alpha)) \cap A$, i.e., if and only if $V' \cap A = (V \cap A) \cup (\mathit{Eff}^{+}(\alpha) \cap A)$, which means $(V \cap A) R^{\text{DL-PA}}_{tr(+\alpha,A)} (V' \cap A)$ for all $ATM(\varphi) \subseteq A$.

Second, when π is of the form $-\alpha$ then we have to distinguish two subcases. If α is valid in $INTR$ then $tr(-\alpha, A) = \mathit{fail}$ and $V R^{\text{DEL-PAO}}_{-\alpha} V'$ is never the case, which matches that for all $ATM(\varphi) \subseteq A$, $(V \cap A) R^{\text{DL-PA}}_{\mathit{fail}} (V' \cap A)$ is never the case.

If α is not valid in $INTR$ then $V R_{-\alpha} V'$ if and only if $V' = V \setminus \mathit{Eff}^{-}(\alpha)$ and we reason similarly to that for $+\alpha$: this is equivalent to $V' \cap A = (V \setminus \mathit{Eff}^{-}(\alpha)) \cap A$ for all $ATM(\varphi) \subseteq A$, that is, $V' \cap A = (V \cap A) \setminus (\mathit{Eff}^{-}(\alpha) \cap A)$ and thus $(V \cap A) R^{\text{DL-PA}}_{tr(-\alpha,A)} (V' \cap A)$ for all $ATM(\varphi) \subseteq A$.

Suppose $\pi = \pi_1; \pi_2$. Then $V R^{\text{DEL-PAO}}_{\pi_1;\pi_2} V'$ if and only if there exists a valuation V_1 such that $V R^{\text{DEL-PAO}}_{\pi_1} V_1$ and $V_1 R^{\text{DEL-PAO}}_{\pi_2} V'$. By induction hypothesis, this is equivalent to : there exists a valuation V_1 such that $(V \cap A) R^{\text{DL-PA}}_{tr(\pi_1,A)} (V_1 \cap A)$ and $(V_1 \cap A) R^{\text{DL-PA}}_{tr(\pi_2,A)} (V' \cap A)$, for all $ATM(\varphi) \subseteq A$, which means $(V \cap A) R^{\text{DL-PA}}_{tr(\pi_1,A);tr(\pi_2,A)} (V' \cap A)$. By definition of tr, we have $(V \cap A) R^{\text{DL-PA}}_{tr(\pi_1;\pi_2,A)} (V' \cap A)$ for all $ATM(\varphi) \subseteq A$.

Now suppose $\pi = \pi_1 \sqcup \pi_2$. We have $V R^{\text{DEL-PAO}}_{\pi_1 \sqcup \pi_2} V'$ if and only if $V R^{\text{DEL-PAO}}_{\pi_1} V'$ or $V R^{\text{DEL-PAO}}_{\pi_2} V'$. By induction hypothesis, this is equivalent to $(V \cap A) R^{\text{DL-PA}}_{tr(\pi_1,A)} (V' \cap A)$ or $(V \cap A) R^{\text{DL-PA}}_{tr(\pi_2,A)} (V' \cap A)$, for all $ATM(\varphi) \subseteq A$, which means $(V \cap A) R^{\text{DL-PA}}_{tr(\pi_1,A) \sqcup tr(\pi_2,A)} (V' \cap A)$. By definition of tr, we have $(V \cap A) R^{\text{DL-PA}}_{tr(\pi_1 \sqcup \pi_2,A)} (V' \cap A)$ for all $ATM(\varphi) \subseteq A$.

Finally, suppose $\pi = \varphi?$. Then $V R^{\mathsf{DEL\text{-}PAO}}_{\varphi?} V'$ if and only if $V = V'$ and $V \models_{\mathsf{DEL\text{-}PAO}} \varphi$. The first is equivalent to $V \cap A = V' \cap A$ for all $ATM(\varphi) \subseteq A$, while the second means, by induction hypothesis on formulas, $V \models_{\mathsf{DL\text{-}PA}} tr(\varphi)$. By Proposition 11, the latter is equivalent to $V \cap ATM(\varphi) \models_{\mathsf{DL\text{-}PA}} tr(\varphi)$ which means, again by Proposition 11, $V \cap A \models_{\mathsf{DL\text{-}PA}} tr(\varphi)$ for all $ATM(\varphi) \subseteq A$. Therefore we have $(V \cap A) R^{\mathsf{DL\text{-}PA}}_{tr(\varphi)?}(V' \cap A)$ and thus $(V \cap A) R^{\mathsf{DL\text{-}PA}}_{tr(\varphi?, A)}(V' \cap A)$ for all $ATM(\varphi) \subseteq A$. □

4 Applications

4.1 PAL axioms

Before moving to the proof of PAL axioms, we show that the relation of introspective consequence is transitive and we add some new useful equivalences.

Proposition 15. *For three atoms $\alpha, \beta, \gamma \in ATM$, if $\alpha \rightsquigarrow \gamma$ and $\gamma \rightsquigarrow \beta$, then $\alpha \rightsquigarrow \beta$.*

Démonstration. First, if $\gamma = \alpha$ or $\gamma = \beta$, then $\alpha \rightsquigarrow \beta$.

Now, if $\gamma \neq \alpha$ and $\gamma \neq \beta$, then $\alpha = JS\,\alpha'$ and $\gamma = \sigma\,\alpha'$ with $\sigma \in OBS^+$.

We know that γ cannot begin with a S_i since it would have no consequence but itself (and thus $\gamma = \beta$). Therefore γ is of the form $JS\,\sigma'\alpha'$, with $\sigma' \in OBS^*$. Thus its consequences, including β, are of the form $\sigma''\sigma'\alpha'$, with $\sigma'' \in OBS^+$, that is, $\beta = \sigma'''\alpha'$ with $\sigma''' = \sigma''\sigma' \in OBS^+$. Therefore $\alpha \rightsquigarrow \beta$. □

Proposition 16. *The following equivalences are* DEL-PAO*-valid.*

$$[+\alpha][+\beta]\varphi \leftrightarrow [+\beta][+\alpha]\varphi \tag{1}$$

$$[+\alpha][-\beta]\varphi \leftrightarrow [-\beta][+\alpha]\varphi \quad \text{if } \alpha \not\rightsquigarrow \beta \tag{2}$$

$$[\varphi?][\psi?]\varphi \leftrightarrow [\psi?][\varphi?]\varphi \tag{3}$$

$$[+\alpha][\beta?]\varphi \leftrightarrow \begin{cases} [+\alpha]\varphi & \text{if } \alpha \rightsquigarrow \beta \\ [\beta?][+\alpha]\varphi & \text{otherwise} \end{cases} \tag{4}$$

$$[-\alpha][\neg\beta?]\varphi \leftrightarrow \begin{cases} [-\alpha]\varphi & \text{if } \beta \rightsquigarrow \alpha \\ [\neg\beta?][-\alpha]\varphi & \text{otherwise} \end{cases} \tag{5}$$

$$[+\alpha][\neg\beta?]\varphi \leftrightarrow \begin{cases} \top & \text{if } \alpha \rightsquigarrow \beta \\ [\neg\beta?][+\alpha]\varphi & \text{otherwise} \end{cases} \tag{6}$$

$$[-\alpha][\beta?]\varphi \leftrightarrow \begin{cases} \top & \text{if } \beta \rightsquigarrow \alpha \\ [\beta?][-\alpha]\varphi & \text{otherwise} \end{cases} \tag{7}$$

$$[+\alpha][+JS\,p]\varphi \leftrightarrow [+JS\,p]\varphi \quad \text{if } JS\,p \rightsquigarrow \alpha \tag{8}$$

$$[-\alpha][+JS\,p]\varphi \leftrightarrow [+JS\,p]\varphi \quad \text{if } JS\,p \rightsquigarrow \alpha \tag{9}$$

Démonstration. We prove every equivalence for an arbitrary valuation V.

(1) : Due to the associativity and commutativity of union : $(V \cup \mathit{Eff}^+(\alpha)) \cup \mathit{Eff}^+(\beta) = (V \cup \mathit{Eff}^+(\beta)) \cup \mathit{Eff}^+(\alpha)$.

(2) : Suppose $\alpha \not\rightsquigarrow \beta$. We want to prove that $(V \cup \mathit{Eff}^+(\alpha)) \setminus \mathit{Eff}^-(\beta) = (V \setminus \mathit{Eff}^-(\beta)) \cup \mathit{Eff}^+(\alpha)$. We know that $(V \setminus \mathit{Eff}^-(\beta)) \cup \mathit{Eff}^+(\alpha) = (V \cup \mathit{Eff}^+(\alpha)) \setminus (\mathit{Eff}^-(\beta) \setminus \mathit{Eff}^+(\alpha))$, thus we want to prove $\mathit{Eff}^-(\beta) \setminus \mathit{Eff}^+(\alpha) = \mathit{Eff}^-(\beta)$, that is, $\mathit{Eff}^+(\alpha) \cap \mathit{Eff}^-(\beta) = \emptyset$, or $\{\gamma : \alpha \rightsquigarrow \gamma\} \cap \{\gamma : \gamma \rightsquigarrow \beta\} = \emptyset$.

Suppose there exists a γ such that $\alpha \rightsquigarrow \gamma$ and $\gamma \rightsquigarrow \beta$. By the Proposition 15, this implies that $\alpha \rightsquigarrow \beta$ (contradiction). Therefore $\{\gamma : \alpha \rightsquigarrow \gamma\} \cap \{\gamma : \gamma \rightsquigarrow \beta\} = \emptyset$.

(3) : PDL validity.

(4) : Suppose $\alpha \rightsquigarrow \beta$, then $\beta \in \mathit{Eff}^+(\alpha)$ and thus $\beta \in V \cup \mathit{Eff}^+(\alpha)$. Therefore $V \models [+\alpha][\beta?]\varphi$ is equivalent to $V \cup \mathit{Eff}^+(\alpha) \models \beta \rightarrow \varphi$, which means $V \cup \mathit{Eff}^+(\alpha) \models \varphi$, that is $V \models [+\alpha]\varphi$.

Now suppose $\alpha \not\rightsquigarrow \beta$, then $\beta \notin \mathit{Eff}^+(\alpha)$ and thus $\beta \in V \Leftrightarrow \beta \in V \cup \mathit{Eff}^+(\alpha)$. Therefore $V \models [+\alpha][\beta?]\varphi$ is equivalent to $V \cup \mathit{Eff}^+(\alpha) \models \beta \Rightarrow V \cup \mathit{Eff}^+(\alpha) \models \varphi$, which means $V \models \beta \Rightarrow V \cup \mathit{Eff}^+(\alpha) \models \varphi$, that is $V \models \beta \rightarrow [+\alpha]\varphi$, same as $V \models [\beta?][+\alpha]\varphi$.

(5) : The reasoning is similar to the previous proof.

Suppose $\beta \rightsquigarrow \alpha$, then $\beta \in \mathit{Eff}^-(\alpha)$ and thus $\beta \notin V \setminus \mathit{Eff}^-(\alpha)$. Therefore $V \models [-\alpha][\neg\beta?]\varphi$ is equivalent to $V \setminus \mathit{Eff}^-(\alpha) \models \neg\beta \rightarrow \varphi$, which means $V \setminus \mathit{Eff}^-(\alpha) \models \varphi$, that is $V \models [-\alpha]\varphi$.

Now suppose $\beta \not\rightsquigarrow \alpha$, then $\beta \notin \mathit{Eff}^-(\alpha)$ and thus $\beta \notin V \Leftrightarrow \beta \notin V \setminus \mathit{Eff}^-(\alpha)$. Therefore $V \models [-\alpha][\neg\beta?]\varphi$ is equivalent to $V \setminus \mathit{Eff}^-(\alpha) \models \neg\beta \Rightarrow V \setminus \mathit{Eff}^-(\alpha) \models \varphi$, which means $V \models \neg\beta \Rightarrow V \setminus \mathit{Eff}^-(\alpha) \models \varphi$, that is $V \models \neg\beta \rightarrow [-\alpha]\varphi$, same as $V \models [\neg\beta?][-\alpha]\varphi$.

(6) : Suppose $\alpha \rightsquigarrow \beta$, then $\beta \in \mathit{Eff}^+(\alpha)$ and thus $\beta \in V \cup \mathit{Eff}^+(\alpha)$. Therefore $V \cup \mathit{Eff}^+(\alpha) \models [\neg\beta?]\varphi$ is equivalent to $V \cup \mathit{Eff}^+(\alpha) \models [\mathit{fail}]\varphi$, that is, $\top$.

Now suppose $\alpha \not\rightsquigarrow \beta$, then $\beta \notin V \Leftrightarrow \beta \notin V \cup \mathit{Eff}^+(\alpha)$. Thus $V \models [+\alpha][\neg\beta?]\varphi$ is equivalent to $V \cup \mathit{Eff}^+(\alpha) \models \neg\beta \Rightarrow V \cup \mathit{Eff}^+(\alpha) \models \varphi$, which means $V \models \neg\beta \Rightarrow V \cup \mathit{Eff}^+(\alpha) \models \varphi$, that is $V \models [\neg\beta?][+\alpha]\varphi$.

(7) : The reasoning is similar to the previous proof.

Suppose $\beta \rightsquigarrow \alpha$, then $\beta \in \mathit{Eff}^-(\alpha)$ and thus $\beta \notin V \setminus \mathit{Eff}^-(\alpha)$. Therefore $V \setminus \mathit{Eff}^-(\alpha) \models [\beta?]\varphi$ is equivalent to $V \setminus \mathit{Eff}^-(\alpha) \models [\mathit{fail}]\varphi$, that is, $\top$.

Now suppose $\beta \not\rightsquigarrow \alpha$, then $\beta \in V \Leftrightarrow \beta \in V \setminus \mathit{Eff}^-(\alpha)$. Thus $V \models [-\alpha][\beta?]\varphi$ is equivalent to $V \setminus \mathit{Eff}^-(\alpha) \models \beta \Rightarrow V \setminus \mathit{Eff}^-(\alpha) \models \varphi$, which means $V \models \beta \Rightarrow V \setminus \mathit{Eff}^-(\alpha) \models \varphi$, that is $V \models [\beta?][-\alpha]\varphi$.

(8) : Since $JS\,p \rightsquigarrow \alpha$, then for every atom γ such that $\alpha \rightsquigarrow \gamma$, we have $JS\,p \rightsquigarrow \gamma$. Thus $\mathit{Eff}^+(\alpha) \subseteq \mathit{Eff}^+(JS\,p)$. Therefore $V \models [+\alpha][+JS\,p]\varphi$ is equivalent to $V \cup \mathit{Eff}^+(\alpha) \cup \mathit{Eff}^+(JS\,p) \models \varphi$, that is $V \cup \mathit{Eff}^+(JS\,p) \models \varphi$, which means $V \models [+JS\,p]\varphi$.

(9) : Since $JS\,p \rightsquigarrow \alpha$, then $\alpha = \sigma\,p$ with $\sigma \in OBS^+$, and thus $\mathit{Eff}^-(\alpha) = \{\gamma : \gamma \rightsquigarrow \sigma\,p \text{ with } \sigma \in OBS^+\} = \{JS\,\sigma'p : \sigma' \in OBS^*\}$ which is included in $\mathit{Eff}^+(JS\,p) = \{\sigma\,p : \sigma \in OBS^+\}$. Therefore $V \models [-\alpha][+JS\,p]\varphi$ is equivalent to $(V \setminus \mathit{Eff}^-(\alpha)) \cup \mathit{Eff}^+(JS\,p) \models \varphi$, that is $V \cup \mathit{Eff}^+(JS\,p) \models \varphi$, which means $V \models [+JS\,p]\varphi$. □

4.1.1 Literals

We define $Pr(\psi)$, with ψ a literal, the propositional variable associated to this literal. More formally, $Pr(p) = Pr(\neg p) = p$.

Proposition. *With ψ a literal and $\psi! = \psi?; +JS\ Pr(\psi)$, the following equivalences are* DEL-PAO*-valid.*

$$[\psi!]p \leftrightarrow \psi \rightarrow p \quad (1)$$

$$[\psi!]\neg\varphi \leftrightarrow \psi \rightarrow \neg[\psi!]\varphi \quad (2)$$

$$[\psi!](\varphi \wedge \varphi') \leftrightarrow [\psi!]\varphi \wedge [\psi!]\varphi' \quad (3)$$

$$[\psi!]K_i\varphi \leftrightarrow \psi \rightarrow K_i[\psi!]\varphi \quad (4)$$

Démonstration.

(1) : $[\psi!]p \equiv [\psi?; +JS\ Pr(\psi)]p$
$\equiv [\psi?][+JS\ Pr(\psi)]p$
$\equiv \psi \rightarrow [+JS\ Pr(\psi)]p$
$\equiv \psi \rightarrow p$ (p is not consequence of $+JS\ Pr(\psi)$).

(2) : $[\psi!]\neg\varphi \equiv [\psi?; +JS\ Pr(\psi)]\neg\varphi$
$\equiv [\psi?; \psi?; +JS\ Pr(\psi)]\neg\varphi$
$\equiv [\psi?][\psi?; +JS\ Pr(\psi)]\neg\varphi$
$\equiv \psi \rightarrow [\psi?; +JS\ Pr(\psi)]\neg\varphi$
$\equiv \psi \rightarrow [\psi?][+JS\ Pr(\psi)]\neg\varphi$
$\equiv \psi \rightarrow [\psi?]\neg[+JS\ Pr(\psi)]\varphi$
$\equiv \psi \rightarrow \neg[\psi?][+JS\ Pr(\psi)]\varphi$
$\equiv \psi \rightarrow \neg[\psi?; +JS\ Pr(\psi)]\varphi$
$\equiv \psi \rightarrow \neg[\psi!]\varphi$
because $\psi \rightarrow [\psi?]\neg\varphi \equiv \psi \rightarrow \neg[\psi?]\varphi$ (PDL validity).

(3) : PDL validity.

(4) : This proof uses equivalences from Proposition 16.

We will use the program associated to $K_i\varphi$ to prove this property. However, to do so, φ is required to not contain epistemic operators. To ensure this, we define φ', the Boolean formula equivalent to φ, obtained thanks to the procedure described in the proof of Theorem 1, and thus prove $[\psi!]K_i\varphi' \leftrightarrow \psi \rightarrow K_i[\psi!]\varphi'$.

With $ATM(\varphi') = \{\alpha_1, \alpha_2, \ldots, \alpha_n\}$ we have, first :
$[\psi!]K_i\varphi'$
$\equiv [\psi?; +JS\ Pr(\psi)]K_i\varphi'$
$\equiv [\psi?; \psi?; +JS\ Pr(\psi)]K_i\varphi'$
$\equiv [\psi?][\psi?; +JS\ Pr(\psi)]K_i\varphi'$
$\equiv \psi \rightarrow [\psi?; +JS\ Pr(\psi)]K_i\varphi'$
$\equiv \psi \rightarrow [\psi?; +JS\ Pr(\psi)][\pi_{i,ATM(\varphi')}]\varphi'$
$\equiv \psi \rightarrow [\psi?; +JS\ Pr(\psi)][\pi_{i,\alpha_1}; \pi_{i,\alpha_2}; \ldots; \pi_{i,\alpha_n}]\varphi'$
$\equiv \psi \rightarrow [\psi?; +JS\ Pr(\psi)][\pi_{i,\alpha_1}][\pi_{i,\alpha_2}] \ldots [\pi_{i,\alpha_n}]\varphi'$.
On the other hand :
$\psi \rightarrow K_i[\psi!]\varphi'$
$\equiv \psi \rightarrow K_i[\psi?; +JS\ Pr(\psi)]\varphi'$
$\equiv \psi \rightarrow [\pi_{i,ATM(\varphi')}][\psi?; +JS\ Pr(\psi)]\varphi'$
$\equiv \psi \rightarrow [\pi_{i,\alpha_1}; \pi_{i,\alpha_2}; \ldots; \pi_{i,\alpha_n}][\psi?; +JS\ Pr(\psi)]\varphi'$
$\equiv \psi \rightarrow [\pi_{i,\alpha_1}][\pi_{i,\alpha_2}] \ldots [\pi_{i,\alpha_n}][\psi?; +JS\ Pr(\psi)]\varphi'$.

One can argue that some programs may be lacking in the second part, since we are replacing $K_i[\psi?; +JS\ Pr(\psi)]\varphi'$ and only considering atoms from φ', thus $\pi_{i,Pr(\psi)}$ and $\pi_{i,JS\ Pr(\psi)}$ should also be added. However, for each of these atoms, we have two cases :

- if it belongs to $ATM(\varphi')$, then it is already added along with other atoms of φ' ;
- if it does not belong to $ATM(\varphi')$, and since φ' does not contain any epistemic operator, then by Proposition 4, its value does not influence the value of φ'. Therefore even if the program updates its value (in the case of agent i not seeing it), the value of φ' will not be modified.

In both cases, it is safe to not consider programs associated to these atoms.

Therefore it suffices to prove :

$$\psi \rightarrow [\psi?; +JS\ Pr(\psi)][\pi_{i,\alpha}]\varphi' \equiv \psi \rightarrow [\pi_{i,\alpha}][\psi?; +JS\ Pr(\psi)]\varphi'$$

Recall that $\pi_{i,\alpha} = S_i\,\alpha? \sqcup (\neg S_i\,\alpha?; (+\alpha \sqcup -\alpha))$.
Thus the left part is equivalent to :
$\psi \rightarrow [\psi?; +JS\ Pr(\psi)][S_i\,\alpha? \sqcup (\neg S_i\,\alpha?; (+\alpha \sqcup -\alpha))]\varphi'$
$\equiv \psi \rightarrow [\psi?][+JS\ Pr(\psi)][S_i\,\alpha?]\varphi'$
$\wedge\ [\psi?][+JS\ Pr(\psi)][\neg S_i\,\alpha?; (+\alpha \sqcup -\alpha)]\varphi'$
$\equiv \psi \rightarrow [\psi?][+JS\ Pr(\psi)][S_i\,\alpha?]\varphi'$
$\wedge\ [\psi?][+JS\ Pr(\psi)][\neg S_i\,\alpha?][+\alpha]\varphi'$
$\wedge\ [\psi?][+JS\ Pr(\psi)][\neg S_i\,\alpha?][-\alpha]\varphi'$,
while the right part is equivalent to :
$\psi \rightarrow [S_i\,\alpha? \sqcup (\neg S_i\,\alpha?; (+\alpha \sqcup -\alpha))][\psi?; +JS\ Pr(\psi)]\varphi'$
$\equiv \psi \rightarrow [S_i\,\alpha?][\psi?][+JS\ Pr(\psi)]\varphi'$
$\wedge\ [\neg S_i\,\alpha?; (+\alpha \sqcup -\alpha)][\psi?][+JS\ Pr(\psi)]\varphi'$
$\equiv \psi \rightarrow [S_i\,\alpha?][\psi?][+JS\ Pr(\psi)]\varphi'$
$\wedge\ [\neg S_i\,\alpha?][+\alpha][\psi?][+JS\ Pr(\psi)]\varphi'$
$\wedge\ [\neg S_i\,\alpha?][-\alpha][\psi?][+JS\ Pr(\psi)]\varphi'$.

Therefore we prove :

$\psi \rightarrow [\psi?][+JS\ Pr(\psi)][S_i\,\alpha?]\varphi'$
$\quad \wedge [\psi?][+JS\ Pr(\psi)][\neg S_i\,\alpha?][+\alpha]\varphi'$
$\quad \wedge [\psi?][+JS\ Pr(\psi)][\neg S_i\,\alpha?][-\alpha]\varphi'$
$\equiv \psi \rightarrow [S_i\,\alpha?][\psi?][+JS\ Pr(\psi)]\varphi'$
$\quad \wedge [\neg S_i\,\alpha?][+\alpha][\psi?][+JS\ Pr(\psi)]\varphi'$
$\quad \wedge [\neg S_i\,\alpha?][-\alpha][\psi?][+JS\ Pr(\psi)]\varphi'.$

We make two main cases.

First, suppose $JS\ Pr(\psi) \rightsquigarrow S_i\,\alpha$.

Then the first part is equivalent to :

$\psi \rightarrow [\psi?][+JS\ Pr(\psi)][skip]\varphi'$
$\quad \wedge [\psi?][+JS\ Pr(\psi)][fail][+\alpha]\varphi'$
$\quad \wedge [\psi?][+JS\ Pr(\psi)][fail][-\alpha]\varphi'$
$\equiv \psi \rightarrow [\psi?][+JS\ Pr(\psi)]\varphi'$
$\equiv \psi \rightarrow [+JS\ Pr(\psi)]\varphi'.$

For the second part, we split the formula in two subparts, one for each truth value of $S_i\,\alpha$. We have :

$\psi \rightarrow (S_i\,\alpha \rightarrow [S_i\,\alpha?][\psi?][+JS\ Pr(\psi)]\varphi'$
$\quad \wedge [\neg S_i\,\alpha?][+\alpha][\psi?][+JS\ Pr(\psi)]\varphi'$
$\quad \wedge [\neg S_i\,\alpha?][-\alpha][\psi?][+JS\ Pr(\psi)]\varphi')$
$\wedge (\neg S_i\,\alpha \rightarrow [S_i\,\alpha?][\psi?][+JS\ Pr(\psi)]\varphi'$
$\quad \wedge [\neg S_i\,\alpha?][+\alpha][\psi?][+JS\ Pr(\psi)]\varphi'$
$\quad \wedge [\neg S_i\,\alpha?][-\alpha][\psi?][+JS\ Pr(\psi)]\varphi')$
$\equiv \psi \rightarrow (S_i\,\alpha \rightarrow [skip][\psi?][+JS\ Pr(\psi)]\varphi'$
$\quad \wedge [fail][+\alpha][\psi?][+JS\ Pr(\psi)]\varphi'$
$\quad \wedge [fail][-\alpha][\psi?][+JS\ Pr(\psi)]\varphi')$
$\wedge (\neg S_i\,\alpha \rightarrow [fail][\psi?][+JS\ Pr(\psi)]\varphi'$
$\quad \wedge [skip][+\alpha][\psi?][+JS\ Pr(\psi)]\varphi'$
$\quad \wedge [skip][-\alpha][\psi?][+JS\ Pr(\psi)]\varphi')$
$\equiv \psi \rightarrow (S_i\,\alpha \rightarrow [\psi?][+JS\ Pr(\psi)]\varphi')$
$\wedge (\neg S_i\,\alpha \rightarrow [+\alpha][\psi?][+JS\ Pr(\psi)]\varphi'$
$\quad \wedge [-\alpha][\psi?][+JS\ Pr(\psi)]\varphi')$
$\equiv \psi \rightarrow (S_i\,\alpha \rightarrow [+JS\ Pr(\psi)]\varphi')$
$\wedge (\neg S_i\,\alpha \rightarrow [+\alpha][+JS\ Pr(\psi)]\varphi')$
$\equiv \psi \rightarrow (S_i\,\alpha \rightarrow [+JS\ Pr(\psi)]\varphi')$
$\wedge (\neg S_i\,\alpha \rightarrow [+JS\ Pr(\psi)]\varphi')$ (since $\alpha = \psi$ is already true)
$\equiv \psi \rightarrow [+JS\ Pr(\psi)]\varphi'$, which is exactly what we wanted.

Now suppose $\alpha = Pr(\psi)$ and $\psi = \neg p$. Then we have :

$\psi \rightarrow (S_i\,\alpha \rightarrow [+JS\ Pr(\psi)]\varphi')$
$\wedge (\neg S_i\,\alpha \rightarrow [+\alpha][fail][+JS\ Pr(\psi)]\varphi'$
$\quad \wedge [-\alpha][skip][+JS\ Pr(\psi)]\varphi')$
$\equiv \psi \rightarrow (S_i\,\alpha \rightarrow [+JS\ Pr(\psi)]\varphi')$
$\wedge (\neg S_i\,\alpha \rightarrow [-\alpha][+JS\ Pr(\psi)]\varphi')$
$\equiv \psi \rightarrow (S_i\,\alpha \rightarrow [+JS\ Pr(\psi)]\varphi')$
$\wedge (\neg S_i\,\alpha \rightarrow [+JS\ Pr(\psi)]\varphi')$ (since $\neg\alpha = \psi$ is already true)
$\equiv \psi \rightarrow [+JS\ Pr(\psi)]\varphi'$, which is also what we wanted.

Finally, suppose $\alpha \neq Pr(\psi)$. Thus $\alpha \not\rightsquigarrow Pr(\psi)$ and $Pr(\psi) \not\rightsquigarrow \alpha$. Moreover, since $JS\ Pr(\psi) \rightsquigarrow S_i\,\alpha$, then $\alpha \in \{\sigma\ Pr(\psi) : \sigma \in OBS^*\}$, but since $\alpha \neq Pr(\psi)$, we have $\alpha \in \{\sigma\ Pr(\psi) : \sigma \in OBS^+\}$, that is, $JS\ Pr(\psi) \rightsquigarrow \alpha$.

Therefore we have, by equivalences (4) and (7) or (6) and (5) of Proposition 16 (depending on ψ) :

$\psi \rightarrow (S_i\,\alpha \rightarrow [+JS\ Pr(\psi)]\varphi')$
$\wedge (\neg S_i\,\alpha \rightarrow [\psi?][+\alpha][+JS\ Pr(\psi)]\varphi'$
$\quad \wedge [\psi?][-\alpha][+JS\ Pr(\psi)]\varphi')$
$\equiv \psi \rightarrow (S_i\,\alpha \rightarrow [+JS\ Pr(\psi)]\varphi')$
$\wedge (\neg S_i\,\alpha \rightarrow [+\alpha][+JS\ Pr(\psi)]\varphi'$
$\quad \wedge [-\alpha][+JS\ Pr(\psi)]\varphi')$
$\equiv \psi \rightarrow (S_i\,\alpha \rightarrow [+JS\ Pr(\psi)]\varphi')$
$\wedge (\neg S_i\,\alpha \rightarrow [+JS\ Pr(\psi)]\varphi'$
$\quad \wedge [+JS\ Pr(\psi)]\varphi')$ (by equivalences (8) and (9))
$\equiv \psi \rightarrow [+JS\ Pr(\psi)]\varphi'$, which is what we wanted.

Now, suppose $JS\ Pr(\psi) \not\rightsquigarrow S_i\,\alpha$. This implies that $\alpha \notin \{\sigma\ Pr(\psi) : \sigma \in OBS^*\}$, thus, $\alpha \notin \{\sigma\ Pr(\psi) : \sigma \in OBS^+\}$ and thus $JS\ Pr(\psi) \not\rightsquigarrow \alpha$. Moreover, we also have $\alpha \neq Pr(\psi)$ (and thus $\alpha \not\rightsquigarrow Pr(\psi)$ and $Pr(\psi) \not\rightsquigarrow \alpha$).

The left part of the equivalence is therefore equivalent to :

$\psi \rightarrow [\psi?][S_i\,\alpha?][+JS\ Pr(\psi)]\varphi'$ (eq. (4))
$\quad \wedge [\psi?][\neg S_i\,\alpha?][+JS\ Pr(\psi)][+\alpha]\varphi'$ (eq. (6))
$\quad \wedge [\psi?][\neg S_i\,\alpha?][+JS\ Pr(\psi)][-\alpha]\varphi'$ (eq. (6))
$\equiv \psi \rightarrow [S_i\,\alpha?][\psi?][+JS\ Pr(\psi)]\varphi'$ (eq. (3))
$\quad \wedge [\neg S_i\,\alpha?][\psi?][+JS\ Pr(\psi)][+\alpha]\varphi'$ (eq. (3))
$\quad \wedge [\neg S_i\,\alpha?][\psi?][+JS\ Pr(\psi)][-\alpha]\varphi'$ (eq. (3))
$\equiv \psi \rightarrow [S_i\,\alpha?][\psi?][+JS\ Pr(\psi)]\varphi'$
$\quad \wedge [\neg S_i\,\alpha?][\psi?][+\alpha][+JS\ Pr(\psi)]\varphi'$ (eq. (1))
$\quad \wedge [\neg S_i\,\alpha?][\psi?][-\alpha][+JS\ Pr(\psi)]\varphi'$ (eq. (2)).

Suppose $\psi = p$, then the latter is equivalent to :

$\psi \rightarrow [S_i\,\alpha?][\psi?][+JS\ Pr(\psi)]\varphi'$
$\quad \wedge [\neg S_i\,\alpha?][+\alpha][\psi?][+JS\ Pr(\psi)]\varphi'$ (eq. (4))
$\quad \wedge [\neg S_i\,\alpha?][-\alpha][\psi?][+JS\ Pr(\psi)]\varphi'$ (eq. (7)).

Suppose $\psi = p$, the reasoning is the same but use different equivalences :

$\psi \rightarrow [S_i\,\alpha?][\psi?][+JS\ Pr(\psi)]\varphi'$
$\quad \wedge [\neg S_i\,\alpha?][+\alpha][\psi?][+JS\ Pr(\psi)]\varphi'$ (eq. (6))
$\quad \wedge [\neg S_i\,\alpha?][-\alpha][\psi?][+JS\ Pr(\psi)]\varphi'$ (eq. (5)).

Both are exactly equal to the right part.

□

4.1.2 Positive knowledge of atoms

Proposition. *With $K_j q! = S_j\, q?; q?; +JS\, q$, the following equivalences are* DEL-PAO*-valid.*

$$[K_j q!]p \leftrightarrow K_j q \to p \quad (1)$$

$$[K_j q!]\neg\varphi \leftrightarrow K_j q \to \neg[K_j q!]\varphi \quad (2)$$

$$[K_j q!](\varphi \wedge \varphi') \leftrightarrow [K_j q!]\varphi \wedge [K_j q!]\varphi' \quad (3)$$

$$[K_j q!]K_i\varphi \leftrightarrow K_j q \to K_i[K_j q!]\varphi \quad (4)$$

Démonstration. Proofs are similar to the ones for literals.

(1) : $[K_j q!]p \equiv [S_j\, q?; q?; +JS\, q]p$
$\equiv S_j\, q \to (q \to [+JS\, q]p)$
$\equiv (S_j\, q \wedge q) \to [+JS\, q]p$
$\equiv K_j q \to [+JS\, q]p$
$\equiv K_j q \to p$ (p is not consequence of $+JS\, q$).

(2) : $[K_j q!]\neg\varphi \equiv [S_j\, q?; q?; +JS\, q]\neg\varphi$
$\equiv [S_j\, q?; q?; S_j\, q?; q?; +JS\, q]\neg\varphi$
$\equiv [S_j\, q?; q?][S_j\, q?; q?; +JS\, q]\neg\varphi$
$\equiv K_j q \to [S_j\, q?; q?][+JS\, q]\neg\varphi$
$\equiv K_j q \to [S_j\, q?; q?]\neg[+JS\, q]\varphi$
$\equiv K_j q \to \neg[S_j\, q?; q?][+JS\, q]\varphi$
$\equiv K_j q \to \neg[S_j\, q?; q?; +JS\, q]\varphi$
$\equiv K_j q \to \neg[K_j q!]\varphi$

(3) : PDL validity.

(4) : This proof uses equivalences from Proposition 16.

We will use the program associated to $K_i\varphi$ to prove this property. However, to do so, φ is required to not contain epistemic operators. To ensure this, we define φ', the Boolean formula equivalent to φ, obtained thanks to the procedure described in the proof of Theorem 1, and thus prove $[K_j q!]K_i\varphi' \leftrightarrow K_j q \to K_i[K_j q!]\varphi'$.

With $ATM(\varphi') = \{\alpha_1, \alpha_2, \ldots, \alpha_n\}$, we have, first :
$[K_j q!]K_i\varphi'$
$\equiv [S_j\, q?; q?; +JS\, q]K_i\varphi'$
$\equiv [S_j\, q?; q?; S_j\, q?; q?; +JS\, q]K_i\varphi'$
$\equiv [S_j\, q?; q?][S_j\, q?; q?; +JS\, q]K_i\varphi'$
$\equiv K_j q \to [S_j\, q?; q?; +JS\, q]K_i\varphi'$
$\equiv K_j q \to [S_j\, q?; q?; +JS\, q][\pi_{i,ATM(\varphi')}]\varphi'$
$\equiv K_j q \to [S_j\, q?; q?; +JS\, q][\pi_{i,\alpha_1}; \pi_{i,\alpha_2}; \ldots; \pi_{i,\alpha_n}]\varphi'$
$\equiv K_j q \to [S_j\, q?; q?; +JS\, q][\pi_{i,\alpha_1}][\pi_{i,\alpha_2}] \ldots [\pi_{i,\alpha_n}]\varphi'$.

On the other hand :
$K_j q \to K_i[K_j q!]\varphi'$
$\equiv K_j q \to K_i[S_j\, q?; q?; +JS\, q]\varphi'$
$\equiv K_j q \to [\pi_{i,ATM(\varphi')}][S_j\, q?; q?; +JS\, q]\varphi'$
$\equiv K_j q \to [\pi_{i,\alpha_1}; \pi_{i,\alpha_2}; \ldots; \pi_{i,\alpha_n}][S_j\, q?; q?; +JS\, q]\varphi'$
$\equiv K_j q \to [\pi_{i,\alpha_1}][\pi_{i,\alpha_2}] \ldots [\pi_{i,\alpha_n}][S_j\, q?; q?; +JS\, q]\varphi'$.

One can argue that some programs may be lacking in the second part, since we are replacing $K_i[S_j\, q?; q?; +JS\, q]\varphi'$ and only considering atoms from φ', thus $\pi_{i,S_j\, q}$, $\pi_{i,q}$ and $\pi_{i,JS\, q}$ should also be added. However, for each of these atoms, we have two cases :

- if it belongs to $ATM(\varphi')$, then it is already added along with other atoms of φ' ;
- if it does not belong to $ATM(\varphi')$, and since φ' does not contain any epistemic operator, then by Proposition 4, its value does not influence the value of φ'. Therefore even if the program updates its value (in the case of agent i not seeing it), the value of φ' will not be modified.

In both cases, it is safe to not consider programs associated to these atoms.

Therefore it suffices to prove :
$K_j q \to [S_j\, q?; q?\,; +JS\, q][\pi_{i,\alpha}]\varphi'$
$\equiv K_j q \to [\pi_{i,\alpha}][S_j\, q?; q?; +JS\, q]\varphi'$

Recall that $\pi_{i,\alpha} = S_i\, \alpha? \sqcup (\neg S_i\, \alpha?; (+\alpha \sqcup -\alpha))$.
Thus the left part is equivalent to :
$K_j q \to [S_j\, q?; q?; +JS\, q]$
$[S_i\, \alpha? \sqcup (\neg S_i\, \alpha?; (+\alpha \sqcup -\alpha))]\varphi'$
$\equiv K_j q \to [S_j\, q?; q?][+JS\, q][S_i\, \alpha?]\varphi'$
$\wedge\, [S_j\, q?; q?][+JS\, q][\neg S_i\, \alpha?; (+\alpha \sqcup -\alpha)]\varphi'$
$\equiv K_j q \to [S_j\, q?; q?][+JS\, q][S_i\, \alpha?]\varphi'$
$\wedge\, [S_j\, q?; q?][+JS\, q][\neg S_i\, \alpha?][+\alpha]\varphi'$
$\wedge\, [S_j\, q?; q?][+JS\, q][\neg S_i\, \alpha?][-\alpha]\varphi'$,
while the right part is equivalent to :
$K_j q \to [S_i\, \alpha? \sqcup (\neg S_i\, \alpha?; (+\alpha \sqcup -\alpha))]$
$[S_j\, q?; q?; +JS\, q]\varphi'$
$\equiv K_j q \to [S_i\, \alpha?][S_j\, q?; q?][+JS\, q]\varphi'$
$\wedge\, [\neg S_i\, \alpha?; (+\alpha \sqcup -\alpha)][S_j\, q?; q?][+JS\, q]\varphi'$
$\equiv K_j q \to [S_i\, \alpha?][S_j\, q?; q?][+JS\, q]\varphi'$
$\wedge\, [\neg S_i\, \alpha?][+\alpha][S_j\, q?; q?][+JS\, q]\varphi'$
$\wedge\, [\neg S_i\, \alpha?][-\alpha][S_j\, q?; q?][+JS\, q]\varphi'$.

Therefore we prove :
$K_j q \to [S_j\, q?; q?][+JS\, q][S_i\, \alpha?]\varphi'$
$\wedge\, [S_j\, q?; q?][+JS\, q][\neg S_i\, \alpha?][+\alpha]\varphi'$
$\wedge\, [S_j\, q?; q?][+JS\, q][\neg S_i\, \alpha?][-\alpha]\varphi'$
$\equiv K_j q \to [S_i\, \alpha?][S_j\, q?; q?][+JS\, q]\varphi'$
$\wedge\, [\neg S_i\, \alpha?][+\alpha][S_j\, q?; q?][+JS\, q]\varphi'$
$\wedge\, [\neg S_i\, \alpha?][-\alpha][S_j\, q?; q?][+JS\, q]\varphi'$.

We make two main cases.
First, suppose $JS\, q \rightsquigarrow S_i\, \alpha$.
Then the first part is equivalent to :
$K_j q \to [S_j\, q?; q?][+JS\, q][skip]\varphi'$
$\wedge\, [S_j\, q?; q?][+JS\, q][fail][+\alpha]\varphi'$

$\wedge\, [S_j\, q?; q?][+JS\, q][fail][-\alpha]\varphi'$

$\equiv K_j q \to [S_j\, q?; q?][+JS\, q]\varphi'$

$\equiv K_j q \to [+JS\, q]\varphi'$.

For the second part, we split the formula in two subparts, one for each truth value of $S_i\, \alpha$. We have :

$K_j q \to (S_i\, \alpha \to [S_i\, \alpha?][S_j\, q?; q?][+JS\, q]\varphi'$
$\quad \wedge [\neg S_i\, \alpha?][+\alpha][S_j\, q?; q?][+JS\, q]\varphi'$
$\quad \wedge [\neg S_i\, \alpha?][-\alpha][S_j\, q?; q?][+JS\, q]\varphi')$
$\wedge\, (\neg S_i\, \alpha \to [S_i\, \alpha?][S_j\, q?; q?][+JS\, q]\varphi'$
$\quad \wedge [\neg S_i\, \alpha?][+\alpha][S_j\, q?; q?][+JS\, q]\varphi'$
$\quad \wedge [\neg S_i\, \alpha?][-\alpha][S_j\, q?; q?][+JS\, q]\varphi')$

$\equiv K_j q \to (S_i\, \alpha \to [skip][S_j\, q?; q?][+JS\, q]\varphi'$
$\quad \wedge [fail][+\alpha][S_j\, q?; q?][+JS\, q]\varphi'$
$\quad \wedge [fail][-\alpha][S_j\, q?; q?][+JS\, q]\varphi')$
$\wedge\, (\neg S_i\, \alpha \to [fail][S_j\, q?; q?][+JS\, q]\varphi'$
$\quad \wedge [skip][+\alpha][S_j\, q?; q?][+JS\, q]\varphi'$
$\quad \wedge [skip][-\alpha][S_j\, q?; q?][+JS\, q]\varphi')$

$\equiv K_j q \to (S_i\, \alpha \to [S_j\, q?; q?][+JS\, q]\varphi')$
$\wedge\, (\neg S_i\, \alpha \to [+\alpha][S_j\, q?; q?][+JS\, q]\varphi'$
$\quad \wedge [-\alpha][S_j\, q?; q?][+JS\, q]\varphi')$

$\equiv K_j q \to (S_i\, \alpha \to [+JS\, q]\varphi')$
$\wedge\, (\neg S_i\, \alpha \to [+\alpha][+JS\, q]\varphi'$
$\quad \wedge [-\alpha][S_j\, q?; q?][+JS\, q]\varphi')$ (since $K_j q$ implies both $S_i\, q$ and q and that they will not be removed after adding α).

Suppose $\alpha = S_j\, q$ or $\alpha = q$. Then the latter is equivalent to :

$K_j q \to (S_i\, \alpha \to [+JS\, q]\varphi')$
$\wedge\, (\neg S_i\, \alpha \to [+\alpha][+JS\, q]\varphi'$
$\quad \wedge [-\alpha][fail][+JS\, q]\varphi')$

$\equiv K_j q \to (S_i\, \alpha \to [+JS\, q]\varphi')$
$\wedge\, (\neg S_i\, \alpha \to [+\alpha][+JS\, q]\varphi')$

$\equiv K_j q \to (S_i\, \alpha \to [+JS\, q]\varphi')$
$\wedge\, (\neg S_i\, \alpha \to [+JS\, q]\varphi')$ (since $\alpha = S_j\, q$ or $\alpha = q$ is already true since $K_j q$ is true)

$\equiv K_j q \to [+JS\, q]\varphi'$, which is exactly what we wanted.

Now suppose $\alpha \neq S_j\, q$ and $\alpha \neq q$. Thus $S_j\, q \not\rightsquigarrow \alpha$ and $q \not\rightsquigarrow \alpha$, since these atoms have no consequence but themselves. Moreover, since $JS\, q \rightsquigarrow S_i\, \alpha$, then $\alpha \in \{\sigma\, q : \sigma \in OBS^*\}$, but since $\alpha \neq q$, we have $\alpha \in \{\sigma\, q : \sigma \in OBS^+\}$, that is, $JS\, q \rightsquigarrow \alpha$.

Therefore we have, by equivalence (7) or (5) of Proposition 16 (depending on ψ) :

$K_j q \to (S_i\, \alpha \to [+JS\, q]\varphi')$
$\wedge\, (\neg S_i\, \alpha \to [+\alpha][+JS\, q]\varphi'$
$\quad \wedge [S_j\, q?; q?][-\alpha][+JS\, q]\varphi')$

$\equiv K_j q \to (S_i\, \alpha \to [+JS\, q]\varphi')$
$\wedge\, (\neg S_i\, \alpha \to [+\alpha][+JS\, q]\varphi'$
$\quad \wedge [-\alpha][+JS\, q]\varphi')$

$\equiv K_j q \to (S_i\, \alpha \to [+JS\, q]\varphi')$
$\wedge\, (\neg S_i\, \alpha \to [+JS\, q]\varphi'$
$\quad \wedge [+JS\, q]\varphi')$ (by equivalences (8) and (9))

$\equiv K_j q \to [+JS\, q]\varphi'$, which is what we wanted.

Now, suppose $JS\, q \not\rightsquigarrow S_i\, \alpha$. This implies that $\alpha \notin \{\sigma\, q : \sigma \in OBS^*\}$, thus, $\alpha \notin \{\sigma\, q : \sigma \in OBS^+\}$ and thus $JS\, q \not\rightsquigarrow \alpha$. Moreover, we have $\alpha \neq S_j\, q$ and $\alpha \neq q$ (and thus $S_j\, q \not\rightsquigarrow \alpha$, $\alpha \not\rightsquigarrow q$ and $q \not\rightsquigarrow \alpha$) and also $\alpha \not\rightsquigarrow S_j\, q$, because $\alpha \neq JS\, q$.

The left part of the equivalence is therefore equivalent to :

$K_j q \to [S_j\, q?; q?][S_i\, \alpha?][+JS\, q]\varphi'$ (eq. (4))
$\quad \wedge [S_j\, q?; q?][\neg S_i\, \alpha?][+JS\, q][+\alpha]\varphi'$ (eq. (6))
$\quad \wedge [S_j\, q?; q?][\neg S_i\, \alpha?][+JS\, q][-\alpha]\varphi'$ (eq. (6))

$\equiv K_j q \to [S_i\, \alpha?][S_j\, q?; q?][+JS\, q]\varphi'$ (eq. (3))
$\quad \wedge [\neg S_i\, \alpha?][S_j\, q?; q?][+JS\, q][+\alpha]\varphi'$ (eq. (3))
$\quad \wedge [\neg S_i\, \alpha?][S_j\, q?; q?][+JS\, q][-\alpha]\varphi'$ (eq. (3))

$\equiv K_j q \to [S_i\, \alpha?][S_j\, q?; q?][+JS\, q]\varphi'$
$\quad \wedge [\neg S_i\, \alpha?][S_j\, q?; q?][+\alpha][+JS\, q]\varphi'$ (eq. (1))
$\quad \wedge [\neg S_i\, \alpha?][S_j\, q?; q?][-\alpha][+JS\, q]\varphi'$ (eq. (2))

$\equiv K_j q \to [S_i\, \alpha?][S_j\, q?; q?][+JS\, q]\varphi'$
$\quad \wedge [\neg S_i\, \alpha?][+\alpha][S_j\, q?; q?][+JS\, q]\varphi'$ (eq. (4))
$\quad \wedge [\neg S_i\, \alpha?][-\alpha][S_j\, q?; q?][+JS\, q]\varphi'$ (eq. (7)).

This is exactly equal to the right part. □

4.2 Gossip problem

Proposition. *With*

$$Call_{ij} = S_i\, s_1?; j : s_1!; ...; S_i\, s_6?; j : s_6!;$$
$$S_j\, s_1?; i : s_1!; ...; S_j\, s_6?; i : s_6!$$

we have :

$$\{\alpha : \alpha \text{ is valid in } INTR\} \cup Prop \cup \{S_i\, s_i : i \in Agt\} \models$$
$$[Call_{12}; Call_{34}; Call_{56}; Call_{13};$$
$$Call_{45}; Call_{16}; Call_{24}; Call_{35}] \bigwedge_{i \in Agt} K_i \left(\bigwedge_{j \in Agt} s_j \right)$$

Démonstration. First, we can remark that our valuation $\{\alpha : \alpha \text{ is valid in } INTR\} \cup Prop \cup \{S_i\, s_i : i \in Agt\}$ is introspective, since it contains all valid atoms and no more atoms of the form $JS\, \alpha$.

Each announcement of the form $j : s_k!$ is equal, in our framework, to $s_k?; +S_j\, s_k$, and since every s_k is true in our valuation and will never be removed, this will always reduce to $+S_j\, s_k$. Thus, for every call $Call_{ij}$, and for every $1 \leq k \leq 6$, $S_j\, s_k$ will be set to true only if $S_i\, s_k$ is true and $S_i\, s_k$ will be set to true only if $S_i\, s_k$ is true.

The first call, $Call_{12}$, thus reduces to $+S_2\, s_1; +S_1\, s_2$, and after its execution, $S_1\, s_1$, $S_2\, s_2$, $S_1\, s_2$ and $S_2\, s_1$ are present in the valuation, along with s_1 and s_2. Thus : $V \models (S_1\, s_1 \wedge s_1) \wedge (S_1\, s_2 \wedge s_2)\, \wedge (S_2\, s_1 \wedge s_1) \wedge (S_2\, s_2 \wedge s_2)$, that is, $V \models K_1 s_1 \wedge K_1 s_2 \wedge K_2 s_1 \wedge K_2 s_2$, which means :

$$V \models K_1(s_1 \wedge s_2) \wedge K_2(s_1 \wedge s_2).$$

In the same way, the two following calls produce $V \models K_3(s_3 \wedge s_4) \wedge K_4(s_3 \wedge s_4)$ and $V \models K_5(s_5 \wedge s_6) \wedge K_6(s_5 \wedge s_6)$.

Then $Call_{13}$ adds $S_3\, s_1$, $S_3\, s_2$, $S_1\, s_3$ and $S_1\, s_4$, and thus we have

$V \models K_1(s_1 \wedge s_2 \wedge s_3 \wedge s_4) \wedge K_3(s_1 \wedge s_2 \wedge s_3 \wedge s_4)$.

If we keep going, and since we know that the sequence of calls solves the problem, the formula will be true after the execution of the whole program. $\square$