

Rappels : Aucun document n'est autorisé. Le barème n'est donné qu'à titre indicatif. Une annexe contenant des rappels de cours est donnée en fin de sujet. **Ce sujet comporte 4 pages.**

Exercice 1 Modélisation : SAT (4pt)

Un restaurant sert deux types de plats, viande ou poisson. Les viandes peuvent être accompagnées de frites, haricots verts ou gratin, et les poissons de carottes ou de gratin. Ce sont les seules possibilités. Chaque client doit choisir un plat entre viande et poisson et doit aussi choisir un accompagnement parmi ceux correspondant à son premier choix. Il ne reste actuellement plus qu'un seul gratin, et un seul client peut donc en commander. Trois clients, Quentin, Roxane et Sylvain arrivent. Roxane veut des frites ou du gratin et Quentin veut du poisson, mais pas de carottes.

Que peuvent commander les trois clients qui correspondent à leurs envies ?

On propose de traduire ce problème en SAT, en utilisant les variables propositionnelles suivantes :

- $a_{x,y}$ est vrai si y est un accompagnement de x , et faux sinon, où $x \in \{v, p\}$ (pour respectivement, (v) viande et (p)oisson) et $y \in \{c, f, g, h\}$ (pour respectivement, (c)arottes, (f)rites, (g)ratin et (h)aricots verts).
- $c_{z,u}$ est vrai si le client z choisit le plat (plat principal ou accompagnement) u et faux sinon, où $z \in \{Q, R, S\}$ (pour respectivement, (Q)uintin, (R)oxane et (S)ylvain) et $u \in \{c, f, g, h, p, v\}$ (comme ci dessus).

Ainsi, $a_{v,f}$ indique si les frites sont un accompagnement de la viande, tandis que $c_{Q,f}$ (resp. $c_{Q,v}$) indique si Quentin commande des frites (resp. de la viande).

On définit les ensembles $Acc = \{c, f, g, h\}$, $PlatPrinc = \{p, v\}$, $Plats = PlatPrinc \cup Acc$ et $Clients = \{Q, R, S\}$.

On peut noter un ensemble de formules à l'aide de quantificateurs et de ces ensembles. Ainsi " $\forall z \in Clients, c_{z,v}$ " signifie que tous les clients mangent de la viande, c'est un raccourci pour " $c_{Q,v} \wedge c_{R,v} \wedge c_{S,v}$ ".

On autorise aussi les contraintes de cardinalité, que l'on note avec des sommes. Par exemple, " $\sum_{y \in Acc} a_{v,y} \geq 1$ " signifie qu'il y a au moins un y tel que $a_{v,y}$, c'est-à-dire que la viande a au moins un accompagnement possible.

Q. 1. Traduire les phrases suivantes :

1. Les viandes peuvent être accompagnées de frites, haricots verts ou gratin, et les poissons de carottes ou de gratin. Ce sont les seules possibilités.

Note : il est possible de traduire tout cela par un ensemble de faits utilisant les variables $a_{x,y}$.

2. Chaque client doit choisir un plat entre viande et poisson.
3. Chaque client doit [...] aussi choisir un accompagnement parmi ceux correspondant à son premier choix.

Note : il peut être plus facile de traduire cela en deux temps en traduisant cette phrase en "Si un client a choisi un plat principal et qu'une garniture donnée n'est pas un accompagnement de ce plat, ce client ne peut choisir cette garniture." et "Chaque client doit choisir exactement un accompagnement".

4. Il ne reste actuellement plus qu'un seul gratin, et un seul client peut donc en commander.
- Note : il suffit de traduire "Il ne peut y avoir plus d'une commande de gratin parmi les clients".

5. Roxane veut des frites ou du gratin et Quentin veut du poisson, mais pas de carottes.

Q. 2. Indiquer comment cet ensemble de formules peut permettre de répondre à la question "Que peuvent commander les trois clients qui correspondent à leurs envies?".

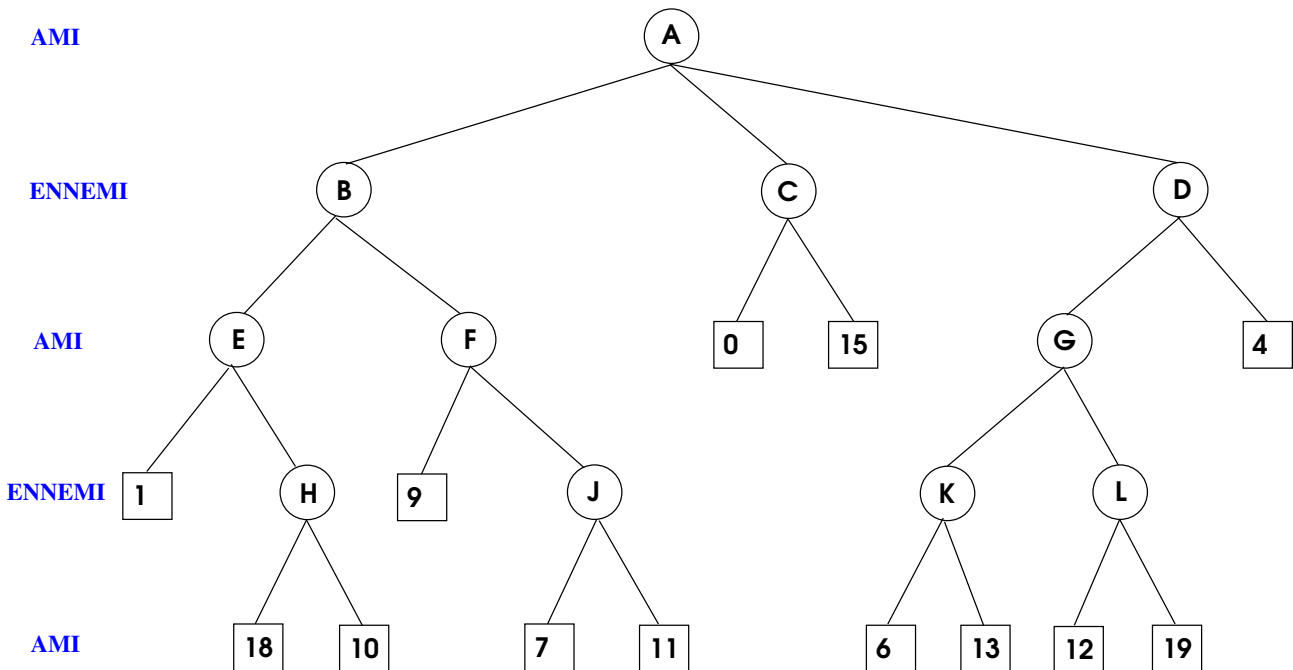
Exercice 2 *Arbre de décision (3pt)*

On considère une base \mathcal{X} définie dans $X_1 \times X_2$ tels que $X_1 = X_2 = [0, 5]$. Chaque exemple de \mathcal{X} est associé à un label de $\mathcal{Y} = \{-, +\}$.

- Q. 1. Donner la représentation graphique (de votre invention) d'une partition de $X_1 \times X_2$ en au moins 5 régions qui pourrait être obtenue par un arbre de décision. Donner aussi un ensemble \mathcal{X} d'au moins 15 exemples correspondant à cette partition. La distribution des labels dans \mathcal{X} doit être équilibrée.
- Q. 2. Donner l'arbre de décision correspondant à la partition précédente.
- Q. 3. Donner, en détaillant votre calcul, la valeur du gain d'information obtenu par l'utilisation de l'attribut racine de votre arbre.
- Q. 4. Expliquer en justifiant par des calculs si le test utilisé par l'attribut racine de votre arbre est le test optimal.

Exercice 3 *Arbre de jeu (5pt)*

On considère l'arbre de recherche suivant. Dans la position racine, notée A , c'est à moi de jouer et je dois trouver un coup.



- Q. 1. Donner une évaluation de la racine de cet arbre de recherche en utilisant l'algorithme minimax.
- Q. 2. Donner une évaluation de la racine de cet arbre en utilisant l'algorithme alpha-béta et en précisant l'ensemble des nœuds de l'arbre visités et, pour chaque branche coupée, le type de la coupe. Une branche est énoncée en donnant le nœud origine et le nœud destination (par exemple : branche (A, B) ou branche $(J, 11)$).
- Q. 3. En permutant les nœuds, il est possible de réorganiser l'arbre de telle sorte que la recherche par l'algorithme alpha-béta soit optimale. Dans une telle réorganisation, il n'est possible que de permuter que les sous-arbres d'un nœud mais sans modifier leur arborescence (chaque nœud doit conserver le même père et les mêmes fils). Par exemple, on peut permuter B et C , ce qui donnerait donc C et son arborescence tout à gauche de A , puis B et son arborescence au centre. Proposer une réorganisation intervenant à chaque niveau de l'arbre pour optimiser la recherche par alpha-béta et donner l'arbre résultat. Donner ensuite une évaluation de cet arbre par l'application de l'algorithme alpha-béta (en précisant l'ensemble des nœuds visités et les coupes mises en œuvre).

Q. 4. Quantifier le gain obtenu en nombre de nœuds visités pour l'arbre originel et l'arbre obtenu après réorganisation.

Exercice 4 *Base de règles (4pt)*

On considère les chances d'un livre d'être adapté comme film cinématographique et d'obtenir alors un Oscar.

R1 : si c'est un livre de science-fiction alors c'est un livre pour geek

R2 : si c'est un livre sur les jeux vidéos alors c'est un livre pour geek

R3 : si c'est un livre de science-fiction alors l'action se passe dans le futur

R4 : si c'est un livre pour geek alors c'est un livre à succès

R5 : si c'est un film réalisé par Spielberg alors c'est un blockbuster ¹

R6 : si l'action se passe dans un monde surpeuplé et que c'est un film réalisé par Spielberg alors il recevra un Oscar

R7 : si c'est un livre sur les jeux vidéos et si l'action se passe dans un monde surpeuplé alors c'est un livre de science-fiction

R8 : si c'est un livre à succès et que le film contient des effets spéciaux alors il recevra un Oscar

R9 : si c'est un livre de science-fiction et qu'il parle de robots alors c'est un film réalisé par Spielberg

R10 : si c'est un blockbuster et que l'action se passe dans le futur alors le film contient des effets spéciaux

Q. 1. Quel ordre de langage peut-on considérer pour représenter ces règles ?

Q. 2. On considère le livre "Player One" ² qui est un livre sur les jeux vidéos, dont l'action se déroule dans un monde surpeuplé, et dont l'adaptation au cinéma est un blockbuster.

En choisissant le mode de chaînage le plus approprié et en justifiant formellement les réponses,

1. prouver que le film associé recevra un Oscar.
2. indiquer combien de preuves peuvent être établies.
3. indiquer le nombre minimal d'inférences à réaliser.

On considère que les règles sont choisies dans l'ordre de la base de règles.

Q. 3. [Question indépendante] Quelle est la différence principale entre les moteurs d'inférence de Prolog et de clips ?

Exercice 5 *Formalisme des "ensembles-réponse" (Answer Sets) (4pt)*

Soit les 6 les propositions suivantes :

(a) Les doctorants sont des étudiants

(b) Les étudiants ne sont généralement pas salariés, autrement dit *si cela n'est pas contradictoire, les étudiants ne sont pas salariés*

(c) Les doctorants sont généralement salariés, autrement dit *si cela n'est pas contradictoire, les doctorants sont salariés*

(d) Les étudiants en formation continue sont salariés.

(e) Jean est doctorant.

(f) Jacques est en formation continue.

Q. 1. Représenter ces propositions en logique des prédicats en faisant appel aux prédicats suivants : *salarié/1, doctorant/1, étudiant/1, formation_continue/1*.

Q. 2. Indiquer le programme correspondant en ASP en indiquant bien le type des variables et le domaine associé.

Q. 3. Donner tous les ensembles réponses de ce programme.

1. c'est-à-dire un film à gros budget.
2. écrit par Ernest Cline.

Annexes

Algorithme de Smodels

L'algorithme utilisé dans `smodels`, un des premiers solveurs ASP performants, est un algorithme de recherche arborescente d'une structure similaire à DPLL (Dédution, Analyse, Choix). L'algorithme `clasp` qui a été utilisé en TME possède un mécanisme de propagation similaire, mais se rapproche plus d'une structure CDCL (avec analyse des conflits et retour arrière non synchrone).

L'algorithme de `smodels` est toutefois plus facile pour rechercher "manuellement" les answer sets d'un programme. Il comporte trois phases.

On note Lit l'ensemble des littéraux du programme P .

On initialise le modèle en cours $M = M_{pos} \cup M_{not} \cup M_{Dpos} \cup M_{Dnot}$ à \emptyset et le programme en cours à $P' = P$, toutes ces variables évoluent au cours de l'algorithme.

1. **Dédution.** On procède en deux temps.

- (a) Littéraux positifs. On ajoute à M_{pos} tous les atomes découlant de faits³, soit $M_{pos} := M_{pos} \cup \{a \in Lit \mid a \in P'\}$. Pour chacun des atomes a ainsi rajouté à M_{pos} , on procède à la *mise à jour par a* du programme courant P' , par suppression ou modification de règles, en traitant chaque règle $r \in P'$ de la façon suivante :
 - Si a est la tête de r , on peut supprimer la règle (elle est satisfaite car sa tête est satisfaite).
 - Si a est dans le corps positif de r , on réécrit la règle r en supprimant a de son corps positif.
 - Si a est dans le corps négatif de r (soit $not\ a$ est dans le corps de r), on supprime la règle (elle est satisfaite car son corps n'est pas satisfait).
- (b) Littéraux négatifs. On ajoute à M_{not} tous les atomes qui ne sont plus présents dans aucune tête de règle de P' , soit $M_{not} := M_{not} \cup \{a \in Lit \mid \forall r \in P' a \notin head(r)\}$. Pour chacun des atomes a ainsi rajouté à M_{not} , on procède à la *mise à jour par not a* du programme courant P' en traitant chaque règle $r \in P'$ de la façon suivante :
 - Si a est dans le corps positif de r , on supprime la règle (elle est satisfaite car son corps n'est pas satisfait).
 - Si a est dans le corps négatif de r (soit $not\ a$ est dans le corps de r), on réécrit la règle r en supprimant a de son corps négatif (i.e. en supprimant $not\ a$ de son corps).

Chacune de ces phases est répétée jusqu'à ce que M atteigne un point fixe.

2. **Analyse.**

- Si M est incohérent ($\perp \in M_{pos}$ ou $(M_{pos} \cup M_{Dpos}) \cap (M_{not} \cup M_{Dnot}) \neq \emptyset$) ou contradictoire ($\exists a \in Lit. \{a, \neg a\} \subseteq (M_{pos} \cup M_{Dpos})$), la branche ne produit aucun answer set. On backtrack (si possible) jusqu'au dernier point de décision (en remettant M et P' dans l'état où ils étaient au moment de la décision).
- Si M est complet ($M_{pos} \cup M_{not} = Lit$), on a trouvé un answer set. On renvoie M_{pos} . Si on cherche plus d'answer set, on backtrack au point de décision précédent.
- Sinon, on doit procéder à une décision.

3. **Décision.** On choisit un littéral a dans $Lit \setminus M$ et on décide de le supposer positif ou négatif en le mettant respectivement dans M_{Dpos} ou M_{Dnot} .

- Si on le rajoute dans M_{Dpos} , on fait une mise à jour de P' par a , mais **sans supprimer les règles qui ont a comme tête.**
- Si on le rajoute dans M_{Dnot} , on fait une mise à jour de P' par $not\ a$.

On reprend alors à l'étape de déduction avec ce nouveau P' .

3. Cette phase est similaire à une propagation unitaire, les faits étant de même nature que des clauses unitaires.