Faculté des Sciences d'Orsay — Université Paris-Saclay
Licence Informatique, LDD Informatique, Mathématiques, semestre 5 — 2024-25
Eléments de logique pour l'informatique (Info 315) — 6 janvier 2025

Examen - 18 décembre 2024

L'examen dure 2 heures. L'énoncé est composé de 9 pages, le barème est indicatif.
Toutes les réponses (sauf QCM) devront être clairement justifiées.
Le seul document autorisé est le polycopié de cours (document orange) non annoté. Le tableau résumant les règles du système G est donné à la fin du sujet.

L'exercice 4 et les 4 premières questions du QCM sont à faire uniquement pour les licences mention informatique (parcours informatique et MIAGE). L'exercice 5 est à faire pour les LDD Informatique, Mathématiques (dont magistères).

Recopiez le numéro d'anonymat sur les intercalaires et sur le QCM.

Exercice 1 *QCM (LDD 4 pts et Licence 5 pts)* Le **Nom Prénom ou numéro d'anonymat de la copie principale** doit être reporté sur l'énoncé du QCM. Utiliser un stylo bleu ou noir pour cocher les cases (pas de crayon papier). En cas d'erreur, indiquer en toute lettre la réponse souhaitée (vrai, faux ou pas de réponse).

Sauf indication contraire, une bonne réponse rapporte $\frac{1}{4}$, une mauvaise réponse retire $\frac{1}{8}$ point, l'absence de réponse vaut 0 point.

N'oubliez pas de rendre le QCM avec vos copies.

Correction : *Voir correction individuelle des QCM.*

Exercice 2 *Système G (3 points)*
Soit p, q, r trois variables propositionnelles.

1. A l'aide du système G, faire les arbres de dérivation des deux séquents suivants :
   (a) $(((p \Rightarrow q) \wedge (r \Rightarrow q)) \Rightarrow r) \vdash p \vee r$
   (b) $(p \Rightarrow q) \Rightarrow q \vdash \neg p \vee q$

   Correction :

   (a) séquent $(((p \Rightarrow q) \wedge (r \Rightarrow q)) \Rightarrow r) \vdash p \vee r$

$$\vee d\ \dfrac{\Rightarrow g\ \dfrac{\text{hyp}\ \dfrac{}{r \vdash p, r} \qquad \wedge d\ \dfrac{\Rightarrow d\ \dfrac{\text{hyp}\ \dfrac{}{p \vdash p, r, q}}{\vdash p, r, p \Rightarrow q} \qquad \Rightarrow d\ \dfrac{\text{hyp}\ \dfrac{}{r \vdash p, r, q}}{\vdash p, r, r \Rightarrow q}}{\Gamma \vdash p, r, (p \Rightarrow q) \wedge (r \Rightarrow q)}}{((p \Rightarrow q) \wedge (r \Rightarrow q)) \Rightarrow r \vdash p, r}}{((p \Rightarrow q) \wedge (r \Rightarrow q)) \Rightarrow r \vdash p \vee r}$$

   (b) séquent $(p \Rightarrow q) \Rightarrow q \vdash \neg p \vee q$

$$\vee d\ \dfrac{\neg d\ \dfrac{\Rightarrow g\ \dfrac{\text{hyp}\ \dfrac{}{q, p \vdash q} \qquad \Rightarrow d\ \dfrac{p \vdash q}{\vdash q, p \Rightarrow q}}{(p \Rightarrow q) \Rightarrow q, p \vdash q}}{(p \Rightarrow q) \Rightarrow q \vdash \neg p, q}}{(p \Rightarrow q) \Rightarrow q \vdash \neg p \vee q}$$

2. En déduire si les séquents sont ou non valides. Si un séquent est non valide, on donnera une interprétation qui le rend faux et on indiquera s'il est satisfiable ou non.

   Correction : *Le premier séquent admet un arbre de dérivation complet, il est donc valide. Pour le second séquent, on a une feuille $p \vdash q$ qui ne correspond pas à une règle hypothèse. Le séquent est donc faux dans une interprétation avec p vrai et q faux. Il est par contre trivialement vrai dans un interprétation où p est faux ou bien q est vrai. Il est donc non valide mais satisfiable.*

Exercice 3 *Modélisation, résolution (8 points)*
Cet exercice propose de modéliser quelques règles du jeu "Hunt the Wumpus" un jeu historique écrit en Basic dans les années 70 par Gregory Yob.

Le jeu se passe dans un espace composé de salles reliées entre elles. Dans l'une des salles se trouve le "Wumpus", un monstre qu'un chasseur doit découvrir sans se faire prendre. Les salles reliées à la salle du monstre ont une mauvaise odeur, ce qui permet de détecter le monstre sans se faire dévorer. Certaines salles contiennent des puits dont on ne peut pas sortir, d'autres cases contiennent des chauve-souris qui vont attraper le chasseur et le déposer dans une autre salle de leur choix.

Pour modéliser le jeu avec de la logique, on se place dans un univers où les objets sont les salles et on se donne les symboles suivants :

- `départ` : une constante qui correspond à la salle dont part le chasseur de Wumpus.
- `wumpus`, `puits`, `odeur`, trois prédicats unaires :
  - `wumpus` est vrai pour la salle dans laquelle se cache le wumpus.
  - `puits` est vrai pour une salle si elle contient un puits sans fond.
  - `odeur` est vrai pour une salle qui a une mauvaise odeur (on pourra dire que la salle sent mavais).
- `lie`, `trajet`, `depl` des prédicats binaires :
  - $\texttt{lie}(s, s')$ est vrai si les salles s et s' sont reliées entre elles par un couloir.
  - $\texttt{trajet}(s, s')$ est vrai s'il y a une chauve-souris dans la salle s et que cette chauve-souris peut transporter le chasseur de la salle s à la salle s' (on suppose que la chauve-souris revient à sa salle de départ après avoir déposé le chasseur).
  - $\texttt{depl}(s, s')$ est vrai lorsque le chasseur a un déplacement possible de la salle s à la salle s' en une étape du jeu. En général ce sera entre deux salles reliées, mais à cause des puits et des chauve-souris, une étape de déplacement ne coïncide pas toujours avec deux salles liées.
- un prédicat binaire d'égalité noté de manière infixe $s = s'$

1. Traduire en langue naturelle (en français ou en anglais correct) les formules suivantes :
   (a) $\exists s, \texttt{wumpus}(s) \land \neg s = \texttt{départ} \land \forall s', \texttt{wumpus}(s') \Rightarrow s = s'$
   (b) $\forall s, \exists s', (\neg \texttt{wumpus}(s) \land \neg \texttt{puit}(s)) \Rightarrow \texttt{depl}(s, s')$
   (c) $\exists s', \forall s, (\neg \texttt{wumpus}(s) \land \neg \texttt{puit}(s)) \Rightarrow \texttt{depl}(s, s')$
   (d) $\forall x\, y, \texttt{odeur}(x) \land \texttt{odeur}(y) \Rightarrow \exists s, (\texttt{lie}(x, s) \land \texttt{lie}(y, s) \land \texttt{wumpus}(s))$

   Correction :

   (a) Il existe une unique salle dans laquelle il y a un wumpus, cette salle n'est pas la salle de départ.

   (b) Pour toute salle qui ne comporte ni puit ni wumpus, il existe une autre salle avec un déplacement possible pour le chasseur entre ces deux salles.

   (c) Il existe une salle que le chasseur peut atteindre en une étape à partir de n'importe quelle salle qui ne comporte ni puit ni wumpus.

(d) *Si deux salles ont une mauvaise odeur alors elles sont toutes les deux liées à une salle qui contient le wumpus.*

2. Mettre la négation de la formule 1d en forme normale de négation.

   Correction :
   Le parenthésage de la formule 1d est :
   $\forall x\, y, ((\mathtt{odeur}(x) \wedge \mathtt{odeur}(y)) \Rightarrow \exists s, (\mathtt{lie}(x,s) \wedge \mathtt{lie}(y,s) \wedge \mathtt{wumpus}(s)))$.
   La négation de cette formule est donc équivalente à la formule suivante en forme normale de négation :
   $\exists x\, y, \mathtt{odeur}(x) \wedge \mathtt{odeur}(y) \wedge \forall s, (\neg \mathtt{lie}(x,s) \vee \neg \mathtt{lie}(y,s) \vee \neg \mathtt{wumpus}(s))$

3. Donner des formules logiques du premier ordre qui n'utilisent que les symboles de la signature donnée pour représenter les propositions suivantes :

   (a) Il y a au plus deux puits.

   (b) Les salles qui ont une mauvaise odeur sont exactement les salles qui sont reliées à une salle où se trouve le wumpus.

   (c) Un déplacement d'une salle à une autre salle est possible lorsque les deux salles sont liées et que la première salle ne contient ni un puits, ni le wumpus, ni une chauve-souris.

   (d) Si un chasseur arrive par un couloir dans une salle qui comporte une chauve-souris alors il se déplace directement (en un coup) de sa salle d'origine jusqu'à une salle destination du trajet de la chauve-souris.

   Correction :

   (a) $\forall x\, y\, z, \mathtt{puits}(x) \wedge \mathtt{puits}(y) \wedge \mathtt{puits}(z) \Rightarrow x = y \vee y = z \vee x = z$

   (b) $\forall x, \mathtt{odeur}(x) \Leftrightarrow (\exists y, \mathtt{wumpus}(y) \wedge \mathtt{lie}(x,y))$

   (c) $\forall x\, y, \mathtt{lie}(x,y) \wedge \neg \mathtt{puits}(x) \wedge \neg \mathtt{wumpus}(x) \wedge \neg(\exists z, \mathtt{trajet}(x,z)) \Rightarrow \mathtt{depl}(x,y)$

   (d) $\forall x\, y\, z, \mathtt{lie}(x,y) \wedge \mathtt{trajet}(y,z) \Rightarrow \mathtt{depl}(x,z)$

4. On se donne les formules suivantes
   — $A_1 \stackrel{\text{def}}{=} \forall s, (\mathtt{odeur}(s) \Rightarrow \exists s', \mathtt{lie}(s,s') \wedge \mathtt{wumpus}(s'))$
   — $A_2 \stackrel{\text{def}}{=} \forall s\, w, (\mathtt{odeur}(s) \Rightarrow \mathtt{wumpus}(w) \Rightarrow \mathtt{lie}(s,w))$
   — $A_3 \stackrel{\text{def}}{=} \forall x\, y\, t\, u, (x \neq y \wedge \mathtt{lie}(x,t) \wedge \mathtt{lie}(y,t) \wedge \mathtt{lie}(x,u) \wedge \mathtt{lie}(y,u)) \Rightarrow (\mathtt{wumpus}(t) \Rightarrow \mathtt{wumpus}(u))$
   — $D \stackrel{\text{def}}{=} \forall s, (\exists x\, y, x \neq y \wedge \mathtt{odeur}(x) \wedge \mathtt{odeur}(y) \wedge \mathtt{lie}(x,s) \wedge \mathtt{lie}(y,s)) \Rightarrow \mathtt{wumpus}(s)$

   En utilisant la méthode de résolution dont on détaillera les étapes, montrer que la propriété D est conséquence logique des propriétés A_1, A_2, A_3.

   Correction :
   — *On met les formules* $A_1, A_2, A_3, \neg D$ *en forme clausale et on cherche une dérivation de la clause vide.*
   — *Mise en forme clausale de* A_1
     — *Forme normale de négation :* $A_1 \equiv \forall s, (\neg \mathtt{odeur}(s) \vee \exists s', \mathtt{lie}(s,s') \wedge \mathtt{wumpus}(s'))$
     — *Skolemisation, on introduit un nouveau symbole* f *d'arité 1 :* $\forall s, (\neg \mathtt{odeur}(s) \vee (\mathtt{lie}(s,f(s)) \wedge \mathtt{wumpus}(f(s))))$
     — *On développe pour avoir la forme normale conjonctive et on obtient deux clauses* $C_1 \stackrel{\text{def}}{=} \neg \mathtt{odeur}(s) \vee \mathtt{lie}(s,f(s))$ $C_2 \stackrel{\text{def}}{=} \neg \mathtt{odeur}(s) \vee \mathtt{wumpus}(f(s))$
   — *Mise en forme clausale de* A_2

    - *Mise en forme normale de négation* $\forall s\, w, (\neg \mathtt{odeur}(s) \vee \neg \mathtt{wumpus}(w) \vee \mathtt{lie}(s,w))$
      La formule ne contient pas de quantificateur eistentiel et est déjà en FNC, on a une clause
      $C_3 \stackrel{\text{def}}{=} \neg \mathtt{odeur}(s) \vee \neg \mathtt{wumpus}(w) \vee \mathtt{lie}(s,w)$
- *Mise en forme clausale de* A_3 *: après mise en FNN, on obtient une seule clause* $C_4 \stackrel{\text{def}}{=} x = y \vee \neg \mathtt{lie}(x,t) \vee \neg \mathtt{lie}(y,t) \vee \neg \mathtt{lie}(x,u) \vee \neg \mathtt{lie}(y,u) \vee \neg \mathtt{wumpus}(t) \vee \mathtt{wumpus}(u)$
- *Mise en forme clausale de* $\neg D$
    - *Mise en forme normale de négation* $\neg D \equiv \exists s, (\exists x\, y, x \neq y \wedge \mathtt{odeur}(x) \wedge \mathtt{odeur}(y) \wedge \mathtt{lie}(x,s) \wedge \mathtt{lie}(y,s)) \wedge \neg \mathtt{wumpus}(s)$
    - *On skolémise en commençant par* s *puis* x *et* y *ce qui fait introduire trois constantes* a *(pour* s*),*b *(pour* x*) et* c *(pour* y*). La propriété devient* $b \neq c \wedge \mathtt{odeur}(b) \wedge \mathtt{odeur}(c) \wedge \mathtt{lie}(b,a) \wedge \mathtt{lie}(c,a)) \wedge \neg \mathtt{wumpus}(a)$
    - *La formule est en FNC et donne 6 clauses*
      $C_5 \stackrel{\text{def}}{=} b \neq c \quad C_6 \stackrel{\text{def}}{=} \mathtt{odeur}(b) \quad C_7 \stackrel{\text{def}}{=} \mathtt{odeur}(c)$
      $C_8 \stackrel{\text{def}}{=} \mathtt{lie}(b,a) \quad C_9 \stackrel{\text{def}}{=} \mathtt{lie}(c,a) \quad C_{10} \stackrel{\text{def}}{=} \neg \mathtt{wumpus}(a)$
- *On a donc un ensemble de 10 clauses à partir desquels il faut dériver la clause vide par les règles de résolution*

$$
\begin{array}{lcl}
C_1 & \stackrel{\text{def}}{=} & \neg \mathtt{odeur}(s) \vee \mathtt{lie}(s, f(s)) \\
C_2 & \stackrel{\text{def}}{=} & \neg \mathtt{odeur}(s) \vee \mathtt{wumpus}(f(s)) \\
C_3 & \stackrel{\text{def}}{=} & \neg \mathtt{odeur}(s) \vee \neg \mathtt{wumpus}(w) \vee \mathtt{lie}(s,w) \\
C_4 & \stackrel{\text{def}}{=} & x = y \vee \neg \mathtt{lie}(x,t) \vee \neg \mathtt{lie}(y,t) \vee \neg \mathtt{lie}(x,u) \vee \neg \mathtt{lie}(y,u) \vee \neg \mathtt{wumpus}(t) \vee \mathtt{wumpus}(u) \\
C_5 & \stackrel{\text{def}}{=} & b \neq c \\
C_6 & \stackrel{\text{def}}{=} & \mathtt{odeur}(b) \\
C_7 & \stackrel{\text{def}}{=} & \mathtt{odeur}(c) \\
C_8 & \stackrel{\text{def}}{=} & \mathtt{lie}(b,a) \\
C_9 & \stackrel{\text{def}}{=} & \mathtt{lie}(c,a) \\
C_{10} & \stackrel{\text{def}}{=} & \neg \mathtt{wumpus}(a)
\end{array}
$$

On réalise les étapes suivantes

- $C_1 + C_6$ *avec la substitution* $\{s \leftarrow b\}$ *donne* $C_{11} \stackrel{\text{def}}{=} \mathtt{lie}(b, f(b))$
- $C_2 + C_6$ *avec la substitution* $\{s \leftarrow b\}$ *donne* $C_{12} \stackrel{\text{def}}{=} \mathtt{wumpus}(f(b))$
- $C_3 + C_7$ *avec la substitution* $\{s \leftarrow c\}$ *donne* $C_{13} \stackrel{\text{def}}{=} \neg \mathtt{wumpus}(w) \vee \mathtt{lie}(c,w)$
- $C_{13} + C_{12}$ *avec la substitution* $\{w \leftarrow f(b)\}$ *donne* $C_{14} \stackrel{\text{def}}{=} \mathtt{lie}(c, f(b))$
- *On fait une suite d'étapes de résolution à partir de la clause* C_4 *:*

$$
\dfrac{\dfrac{\dfrac{\{x \leftarrow b, y \leftarrow c\}\ \dfrac{C_4 \qquad C_5}{\neg \mathtt{lie}(b,t) \vee \neg \mathtt{lie}(c,t) \vee \neg \mathtt{lie}(b,u) \vee \neg \mathtt{lie}(c,u) \vee \neg \mathtt{wumpus}(t) \vee \mathtt{wumpus}(u)} \qquad C_{11}}{\{t \leftarrow f(b)\}\ \ \neg \mathtt{lie}(c,f(b)) \vee \neg \mathtt{lie}(b,u) \vee \neg \mathtt{lie}(c,u) \vee \neg \mathtt{wumpus}(f(b)) \vee \mathtt{wumpus}(u)} \qquad C_{14}}{\neg \mathtt{lie}(b,u) \vee \neg \mathtt{lie}(c,u) \vee \neg \mathtt{wumpus}(f(b)) \vee \mathtt{wumpus}(u)} \qquad C_{12}}{\neg \mathtt{lie}(b,u) \vee \neg \mathtt{lie}(c,u) \vee \mathtt{wumpus}(u)}
$$

Soit $C_{15} \stackrel{\text{def}}{=} \neg \mathtt{lie}(b,u) \vee \neg \mathtt{lie}(c,u) \vee \mathtt{wumpus}(u)$ *la clause ainsi obtenue*

— *On conclut avec les étapes suivantes*

$$\dfrac{\dfrac{\{u \leftarrow a\}\ \dfrac{C_{15} \qquad C_8}{\neg\texttt{lie}(c,a) \vee \texttt{wumpus}(a)} \qquad C_9}{\texttt{wumpus}(a)} \qquad C_{10}}{\bot}$$

5. Dans la question précédente, la formule A_3 est une propriété de la manière dont les salles sont reliées entre elles. Dans le jeu d'origine, les salles sont positionnées sur les sommets d'un dodécahèdre dont les faces sont des pentagones et elles sont reliées par les arêtes. Si on a deux points différents x et y reliés tous deux à une salle s alors cette salle est unique. Comme il n'y a qu'un seul wumpus, on en déduit que la formule A_3 est vérifiée. Donner une interprétation avec un domaine contenant 4 salles $\{a, b, c, d\}$ dans lequel les formules A_1 et A_2 sont vraies et les formules A_3 et D sont fausses. On précisera donc les interprétations des symboles de prédicat qui apparaissent dans les formules à savoir `lie`, `odeur` et `wumpus`. Le prédicat d'égalité est interprété de manière standart par l'égalité du domaine, les 4 éléments sont donc deux à deux différents.

   Correction : *On positionne les salles en carré, on a donc pour interprétation de* $\texttt{lie}$ *l'ensemble de coupes* $\{(a, b),(b, a),(b, c),(c, b),(c, d),(d, c),(d, a),(a, d)\}$. *On peut prendre comme interprétation de* $\texttt{wumpus}$ *l'ensemble réduit à un élément* $\{a\}$ *et pour interprétation de* $\texttt{odeur}$ *les cases adjacentes à savoir* $\{b, d\}$

   Dans cette interprétation :

   — A_1 *est vrai puisque les deux salles avec odeurs sont* b *et* d *toutes les deux reliées à la salle* a *dans laquelle se trouve bien* $\texttt{wumpus}$

   — A_2 *est vrai puisque les conditions contraignent* s *à être* b *ou* d *et* w *à être* a *et que dans chacun des cas on a bien* $\texttt{lie}(s, w)$ *qui est vrai.*

   — A_3 *est faux, en effet en prenant pour* x *la salle* b *pour* y *la salle* c *pour* t *la salle* a *et pour* u *la salle* d *on a bien que* $(x \neq y \wedge \texttt{lie}(x, t) \wedge \texttt{lie}(y, t) \wedge \texttt{lie}(x, u) \wedge \texttt{lie}(y, u))$ *est vrai ainsi que* $\texttt{wumpus}(t)$ *par contre* $\texttt{wumpus}(u)$ *est faux et donc la formule* A_3 *est fausse.*

   — D *est également faux, il suffit de prendre pour* s *la salle* d. *Avec* $x = b$ *et* $y = c$ *on justifie que* $\exists x\, y, x \neq y \wedge \texttt{odeur}(x) \wedge \texttt{odeur}(y) \wedge \texttt{lie}(x, s) \wedge \texttt{lie}(y, s)$ *est vrai mais* $\texttt{wumpus}(s)$ *est faux et donc la formule* D *est également fausse.*

Exercice 4 *Transformation de formules (Licence, 4 points)*

Exercice à faire uniquement par les étudiants de Licence informatique (dont parcours Miage) les autres étudiants (LDD Informatique, Mathématiques et Magistère) traiteront à la place l'exercice 5.

On dira qu'une formule est en forme **minimale** si elle n'utilise que le connecteur $\Rightarrow$, et comme formule atomique $\bot$ et les variables propositionnelles. Le but de l'exercice est de montrer que toute formule propositionnelle admet une forme minimale équivalente. Dans la suite p et q représentent des variables propositionnelles.

1. Donner une formule en forme minimale équivalente à $\neg p$.
2. Sachant que les formules $p \Rightarrow q$ et $\neg p \vee q$ sont logiquement équivalentes, donner une formule équivalente à $p \vee q$ en forme minimale, justifier.

3. (a) Donner la table de vérité de $(p \Rightarrow (q \Rightarrow \bot)) \Rightarrow \bot$.
   (b) En déduire une formule équivalente à $\neg p \wedge q$ qui soit en forme minimale.
4. Déduire des questions précédentes une fonction `min` qui transforme toute formule propositionnelle A (sans oublier $\top$) en une formule équivalente en forme minimale.

Correction :

1. *.* $\neg p \equiv p \Rightarrow \bot$.
2. *. Une forme minimale pour* $p \vee q$ *est* $(p \Rightarrow \bot) \Rightarrow q$ *(qui est équivalent à* $\neg p \Rightarrow q$*)*
3. *(a)*

p	q	$p \Rightarrow (q \Rightarrow \bot)$	$(p \Rightarrow (q \Rightarrow \bot)) \Rightarrow \bot$
F	_	V	F
V	F	V	F
V	V	F	V

On reconnait la table de $p \wedge q$

(b) $((p \Rightarrow \bot) \Rightarrow q \Rightarrow \bot) \Rightarrow \bot$, *une autre forme possible est* $(q \Rightarrow p) \Rightarrow \bot$

4.

$$\begin{array}{llll}
\mathtt{min}(\bot) & = \bot & \mathtt{min}(P \vee Q) & = (\mathtt{min}(P) \Rightarrow \bot) \Rightarrow \mathtt{min}(Q) \\
\mathtt{min}(\top) & = \bot \Rightarrow \bot & \mathtt{min}(P \wedge Q) & = (\mathtt{min}(P) \Rightarrow (\mathtt{min}(Q) \Rightarrow \bot)) \Rightarrow \bot \\
\mathtt{min}(p) & = p \quad \mathtt{si}\ p\ \mathtt{var.\ prop.} & \mathtt{min}(P \Rightarrow Q) & = \mathtt{min}(P) \Rightarrow \mathtt{min}(Q) \\
\mathtt{min}(\neg P) & = \mathtt{min}(P) \Rightarrow \bot & &
\end{array}$$

Exercice 5 ***uniquement pour les parcours LDD/Magistère*** *(5 points)*

L'objet de l'exercice est de montrer que la stratégie qui consiste à utiliser dans la résolution une clause qui n'a que des littéraux négatifs permet toujours de dériver la clause vide à partir d'un ensemble insatisfiable de clauses.

1. Montrer que tout ensemble insatisfiable de clauses propositionnelles qui ne contient pas la clause vide a forcément au moins une clause dite négative qui n'a que des littéraux négatifs, c'est-à-dire de la forme $\neg p_1 \vee \ldots \vee \neg p_n$, et une clause dite positive, qui n'a que des littéraux positifs, c'est-à-dire de la forme $p_1 \vee \ldots \vee p_n$ avec p_i des variables propositionnelles.

   Correction : *Si un ensemble de clauses propositionnelles qui ne contient pas la clause vide n'a pas de clauses négative alors toutes les clauses ont au moins un littéral positif, l'interprétation dans laquelle toutes les variables propositionnelles sont vraies rend toutes les clauses vraies donc l'ensemble est satisfiable.*

   De même s'il n'y a pas de clause positive alors toutes les clauses ont un littéral négatif et donc l'interprétation qui rend toutes les variables propositionnelles fausses rend vraies toutes les clauses et donc l'ensemble est satisfiable.

   Certains ont invoqué le fait que si l'ensemble est insatisfiable alors on devrait avoir "une clause et son contraire". Tout d'abord la négation d'une clause n'est pas en général une clause mais une conjonction de littéraux. L'exemple $\{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q\}$ *qui est insatisfiable montre qu'on n'a pas forcément de clauses réduites à un littéral. La seule présence de* $p \vee q$ *et de* $\neg p \vee \neg q$ *ne rend pas l'ensemble insatisfiable.*

   Un argument possible est de regarder effectivement l'arbre des résolutions qui mène à la clause vide. Tout d'abord la résolution de deux clauses $C = x \vee D$ *et* $C' = \neg x \vee D'$ *produit une clause* $D \vee D'$*. Si la clause résultante est négative alors* C' *est négative et si elle est* *positive alors C est positive. Donc on peut remonter d'une clause négative obtenue à une étape de résolution jusqu'à une clause négative de l'ensemble initial (et de même pour une clause positive). La dernière étape de la résolution est une résolution sur les clauses p et $\neg p$ respectivement positive et négative, en remontant dans l'arbre jusqu'à une feuille qui correspond à une clause de l'ensemble initial, on trouve une clause positive et une clause négative.*

De manière alternative on peut regarder l'arbre de décision de l'ensemble des clauses et remarquer que la clause la plus à gauche dans l'arbre (qui est fausse avec toutes les variables à vrai) est négative et celle la plus à droite (fausse avec toutes les variables fausses) est positive.

2. Soit l'ensemble de clauses $\{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q\}$. Construire par résolution une dérivation de la clause vide qui utilise à chaque étape une clause négative.

   Correction :

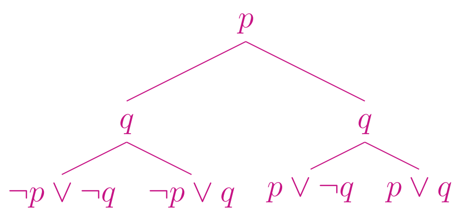
   — $\neg p \vee \neg q$ *et* $\neg p \vee q$ *donne la clause* $\neg p$ *qui est négative*
   — $\neg p \vee \neg q$ *et* $p \vee \neg q$ *donne la clause* $\neg q$ *qui est négative*
   — $\neg p$ *et* $p \vee q$ *donne la clause* q
   — $\neg q$ *et* q *donne la clause vide*

   A chaque étape de résolution, on a bien utilisé une clause négative.

3. On se donne un ensemble de variables propositionnelles qu'on ordonne de manière arbitraire et un ensemble $\mathcal{E}$ de clauses. On suppose $\mathcal{E}$ insatisfiable.

   La méthode de base pour prouver la complétude consiste à construire un arbre binaire de décision (on dit aussi arbre sémantique) dont les nœuds internes sont étiquetés par les variables propositionnelles, ces variables apparaissent en ordre croissant de la racine vers les feuilles. Le branchement est binaire, le sous-arbre de gauche du nœud étiqueté par la variable x correspond au cas où x est vrai et le sous-arbre de droite correspond au cas où x est faux. Ainsi chaque branche correspond à une interprétation partielle des variables (toutes ne sont pas définies). Les feuilles sont des clauses de $\mathcal{E}$ qui sont fausses dans l'interprétation associée à la branche. On demande de plus que la variable du nœud père de la feuille apparaisse effectivement dans la clause (sinon la clause est fausse quelque soit la valeur de la variable et donc la clause était déjà fausse dans la branche qui menait au père et pouvait donc être remontée).

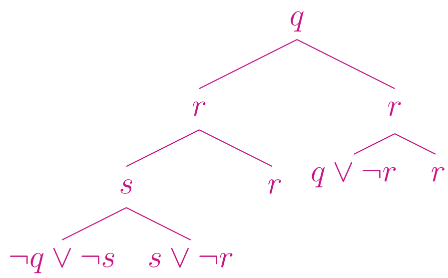
   Par exemple pour l'ensemble de clauses $\{p \vee q, \neg p \vee q, p \vee \neg q, \neg p \vee \neg q\}$ avec comme ordre pour les variables $p < q$, on obtient l'arbre



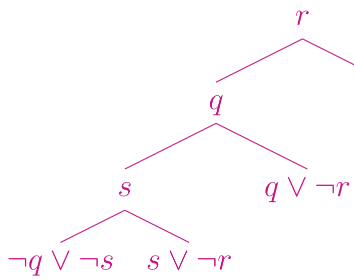
   (a) Construire deux arbres de décision pour l'ensemble de clauses ci-dessous, le premier en prenant comme ordre sur les variables $q < r < s$ et le second en prenant $r < q < s$

$$\{r, q \vee \neg r, s \vee \neg r, \neg q \vee \neg s\}$$

Correction : *Avec l'ordre* $q < r < s$ *:*



Avec l'ordre $r < q < s$ *:*



(b) Pour la complétude de la résolution générale, on prend un nœud de profondeur maximale, associé à la variable x. Il a deux feuilles qui contiennent des clauses, l'une avec x, l'autre avec $\neg x$. On peut donc faire une étape de résolution qui donne une clause C. Dans l'arbre sémantique, on remplace le nœud x par la clause C, qui ne contient plus x. On obtient ainsi un arbre sémantique strictement plus petit.

Cette approche ne s'applique pas à la stratégie de résolution avec une clause négative, en effet cette clause n'est pas forcément à profondeur maximale dans l'arbre.

On suppose qu'on a construit un arbre de décision pour un ensemble $\mathcal{E}$ de clauses supposé insatisfiable. On a au moins une clause négative (sinon l'ensemble serait satisfiable) que l'on note C. Soit x la variable associée au nœud père de C. Puisque x est négatif, on a que C est à gauche du nœud x.

On s'intéresse au sous-arbre droit de ce nœud x que l'on note d.

i. Montrer que si aucune clause du sous-arbre d ne contient la variable x alors on peut simplifier l'arbre sémantique en remplaçant le nœud x par le sous-arbre d.

ii. Soit une clause D du sous-arbre d qui contient la variable x et qui se trouve sur une branche correspondant à une interprétation I. On a donc que D est fausse dans l'interprétation I. Montrer que la clause obtenue par résolution de la clause négative C et de D est également fausse dans l'interprétation I.

iii. En conclure qu'il est possible en n'utilisant que des résolutions avec une clause négative de construire un arbre sémantique strictement plus petit.

Correction :

i. Soit J *l'interprétation partielle obtenue jusqu'au nœud* x*. Soit* D *une clause de* d*, elle se trouve sur un chemin correspondant à une interprétation partielle* I *de la forme* $J, x \mapsto F, J'$ *avec* J' *l'interprétation des variables strictement plus grandes que* x*. La clause* D *est fausse dans l'interprétation* I *mais comme* x *n'apparaît pas dans la formule, la clause est également fausse dans l'interprétation* J, J'*. Ce* *résultat est vrai pour toutes les clauses de* d*, et donc l'arbre obtenu en remplaçant le nœud* x *par l'arbre* d *est encore un arbre sémantique qui justifie l'insatisfiabilité de* $\mathcal{E}$*.*

ii. Soit D *une clause de* d *qui contient* x*, elle se trouve sur un chemin correspondant à une interprétation partielle* I *de la forme* $J, x \mapsto F, J'$ *avec* J' *l'interprétation des variables strictement plus grandes que* x*.*

On décompose $C = C' \vee \neg x$ *et* $D = D' \vee x$*, on a* $J, \models \neg C'$ *et* $J, x \mapsto F, J' \models \neg D'$*. La résolution entre* C *et* D *donne la clause* $C' \vee D'$*. On a bien* $J, x \mapsto F, J' \models \neg(C' \vee D')$*.*

iii. Il suffit de faire la résolution entre la clause négative et toutes les clauses du sous-arbre droit qui contiennent la variable x*. La variable* x *n'apparaît alors plus dans le sous-arbre droit, et on peut donc remplacer le nœud* x *par ce sous-arbre droit, ce qui donne un arbre sémantique strictement plus petit.*

Remarque : *il est possible que dans le processus, la condition que la clause contienne la variable du nœud père ne soit pas satisfaite, auquel cas, il faut remonter la clause un cran plus haut, mais cela ne fait que diminuer encore plus la taille de l'arbre sémantique.*

Rappel des règles du système G (propositionnel)

hypothèse	$(\textsc{Hyp})\dfrac{}{A, \Gamma \vdash \Delta, A}$	
	gauche	droite
$\bot$	$\dfrac{}{\bot, \Gamma \vdash \Delta}$	$\dfrac{\Gamma \vdash \Delta}{\Gamma \vdash \Delta, \bot}$
$\top$	$\dfrac{\Gamma \vdash \Delta}{\top, \Gamma \vdash \Delta}$	$\dfrac{}{\Gamma \vdash \Delta, \top}$
$\neg$	$\dfrac{\Gamma \vdash \Delta, A}{\neg A, \Gamma \vdash \Delta}$	$\dfrac{A, \Gamma \vdash \Delta}{\Gamma \vdash \Delta, \neg A}$
$\wedge$	$\dfrac{A, B, \Gamma \vdash \Delta}{A \wedge B, \Gamma \vdash \Delta}$	$\dfrac{\Gamma \vdash \Delta, A \quad \Gamma \vdash \Delta, B}{\Gamma \vdash \Delta, A \wedge B}$
$\vee$	$\dfrac{A, \Gamma \vdash \Delta \quad B, \Gamma \vdash \Delta}{A \vee B, \Gamma \vdash \Delta}$	$\dfrac{\Gamma \vdash \Delta, A, B}{\Gamma \vdash \Delta, A \vee B}$
$\Rightarrow$	$\dfrac{\Gamma \vdash \Delta, A \quad B, \Gamma \vdash \Delta}{A \Rightarrow B, \Gamma \vdash \Delta}$	$\dfrac{A, \Gamma \vdash \Delta, B}{\Gamma \vdash \Delta, A \Rightarrow B}$