TD2 : Syntaxe de la logique des propositions

1. Pour chacune des formules suivantes construites sur l'ensemble de symboles propositionnels $S = \{p, q, r, s\}$, les connecteurs $\{\neg, \wedge, \vee, \rightarrow, \leftrightarrow\}$, les constantes $\{\bot, \top\}$ et les parenthèses $\{(,)\}$, vous donnerez :
   - son statut vis-à-vis de la logique des propositions : est-elle ou non une proposition (i.e. une formule bien formée du langage propositionnel) ?
   - l'arborescence associée à la formule (si la formule n'est pas bien formée vous proposerez une correction) ;
   - une expression fonctionnelle de la formule ;
   - une expression infixée débarrassée des parenthèses superflues en considérant les priorités entre connecteurs vues en cours :

   (a) $\neg((q \rightarrow r) \vee p)$
   (b) $((\neg p \rightarrow (\neg r \vee q)) \rightarrow p)$
   (c) $(p \vee\ (\neg q \wedge\ (r \wedge s)))$
   (d) $(((p \wedge q)\neg r) \rightarrow p)$
   (e) $(\neg\neg\ (q \rightarrow r) \vee\ (\neg q \rightarrow \neg r))$
   (f) $(\bot \wedge\ (\neg p \vee \top))$

2. Donner tous les arborescences pouvant correspondre aux formules suivantes sans tenir compte des conventions de priorité entre connecteurs, puis préciser quelle arborescence correspond aux conventions d'écriture infixée données dans le cours.

   (a) $p \wedge q \vee r$
   (b) $p \wedge \neg q$
   (c) $\neg p \wedge q$
   (d) $\neg p \rightarrow \neg q \wedge r$
   (e) $p \rightarrow q \rightarrow r \rightarrow s$

   Remarque : dans la suite on supposera que l'on tient compte des conventions de priorité du cours (y compris celle entre connecteurs de même priorité).

3. Que faudrait-il modifier dans la définition inductive du langage des propositions pour prendre en compte un connecteur binaire *ou exclusif* noté $a \otimes b$: *non (a et b)* ?
   Même question avec un connecteur binaire *non et* noté $\oslash$ et avec un connecteur ternaire *si alors sinon* noté $\dagger\ \ddagger$ (exemple : $(p \dagger q \ddagger r)$: *si p alors q sinon r*).

4. Soit $\mathcal{S}$ l'ensemble des symboles propositionnels et $\mathcal{P}rop(\mathcal{S})$ l'ensemble des fbf construites à partir de $\mathcal{S}$ et soit A et B les fbf suivantes :

   $A =_{def} (\neg p \rightarrow (p \rightarrow (r \vee p)))$

   $B =_{def} (((p \wedge \bot) \vee (\neg r \rightarrow p)) \leftrightarrow r)$

   (a) Dessiner les arborescences associées à A et B
   (b) Calculer les ensembles $\mathcal{SP}(A)$ et $\mathcal{SP}(B)$ où SP est l'application vue en cours qui définit l'ensemble des symboles propositionnels d'une formule.
   (c) Soit l'application $\mathcal{P}rof$ de $\mathcal{P}rop(\mathcal{S})$ dans l'ensemble des entiers naturels qui, à toute fbf P, associe la profondeur de l'arborescence syntaxique associée à P (c'est-à-dire le nombre maximum d'arêtes d'un chemin dans cette arborescence). Donner une définition par induction structurelle de l'application $\mathcal{P}rof$.

(d) Calculer les entiers $\mathcal{P}rof(A)$ et $\mathcal{P}rof(B)$

5. Une sous-formule d'une fbf A est une fbf B qu'il est nécessaire de produire lors de la construction de A par le processus d'induction.

   (a) Dites si les formules suivantes sont des sous-formules de la formule $((\neg p \to q \wedge r) \leftrightarrow ((s \to \neg p \wedge r) \leftrightarrow \neg q \wedge s))$:

   $\neg p$

   $p \to q$

   $p \to q \wedge r$

   $\neg p \to q$

   $\neg p \to q \wedge r$

   $\neg p \wedge r$

   $(s \to \neg p \wedge r) \leftrightarrow \neg q \wedge s$

   (b) Quel est l'ensemble des sous-formules de la formule de la formule $((\neg p \wedge r) \to \neg p)$?

   (c) Donnez une définition par induction de l'application $\mathcal{S}ub$ qui associe à une proposition l'ensemble de ses sous-formules.

6. Soit l'application $\mathcal{S}ub$ précédente et l'application $\mathcal{N}bc$ de $\mathcal{P}rop(\mathcal{S})$ dans $\mathbb{N}$ qui, à toute fbf P, associe le nombre de connecteurs de P, et soit la fbf $A =_{def} ((p \wedge q) \to (\neg r \vee p))$.

   (a) Donner $\mathcal{S}ub(A)$ et $\mathcal{N}bc(A)$, puis vérifier que $|\mathcal{S}ub(A)| \leq 2.\mathcal{N}bc(A) + 1$.

   (b) Donner une définition par induction de l'application $\mathcal{N}bc$.

   (c) Montrer par induction structurelle qu'une fbf P ayant n occurrences de connecteurs a au plus $2n+1$ sous-fbf (autrement dit, que pour toute fbf P, $|\mathcal{S}ub(P)| \leq 2.\mathcal{N}bc(P)+1$).

Préalables aux TP

préalables Scheme

Vous utiliserez l'application **Rackett** qui se lance ...
Dans la bare de menu du haut, vous cliquerez sur Langage→ Sélectionner le Langage et choisirez *Assez gros Scheme.*

La boucle Read Eval Print

Le quote '

L'évaluation récursive

la notation fonctionnelle

l'aide

Dans la bare de menu du haut, vous cliquerez sur Aide→Aide puis sur *Reference : Racket* et vous avez alors dans votre butineur une fenêtre (dans laquelle est écrit ... `search manual`...) dans laquelle vous tappez le nom de la fonction dont vous voulez connaître les spécifications (tappez par exemple `member` puis *entrée*).

Le TDA liste en Scheme

- `NULL` ou alternativement `'()` représente la liste vide
- `(cons x l)`
- `(length l)` : *liste* $\rightarrow \mathbb{N}$ t.q. $(length\ l) = n$ si et seulement si la liste l a n éléments,
- `(car l)`, `(cadr l)`, `(caddr l)` : *liste* $\rightarrow$ *Expression* qui renvoient respectivement (quand ils existent) le premier, le deuxième et le troisième élément de l.

TP1 : partie syntaxique

Représentation des propositions

L'objectif final des TP est d'implanter toutes les notions définies sur les propositions.
Afin de faciliter la manipulation des propositions en Scheme, nous utiliserons une représentation préfixée (fonctionnelle) des propositions. Ainsi les connecteurs seront assimilés à des symboles de fonctions (unaire pour le $\neg$ et binaires pour les autres).

Avec cette représentation, nous noterons les propositions p, $\neg p$ et $p \rightarrow q$ par : `p`, `(non p)`, `(impl p q)`.

Q 1 Représentez les formules suivantes sous forme de listes Scheme (rajoutez éventuellement des parenthèses pour avoir des fbf) :

$$F1 = a \wedge d \leftrightarrow \neg a \vee d$$

$$F2 = \neg(a \wedge \neg b) \vee \neg(a \rightarrow b)$$

$$F3 = \neg(a \rightarrow a \vee b) \wedge \neg\neg(a \wedge (b \vee \neg c))$$

$$F4 = (\neg a \vee b \vee d) \wedge (\neg d \vee c) \wedge (c \vee a) \wedge (\neg c \vee b) \wedge (\neg c \vee \neg b) \wedge (\neg b \vee d)$$

Indication : par exemple, `(define F0 '(impl (ou a (non b)) c))` définit la formule F_0 qui représente $((a \vee \neg b) \rightarrow c)$.

Q 2 Définissez le prédicat [1] `(neg? s)` qui renvoie vrai si et seulement si $s = non$. De même, écrivez les prédicats `ou?, et?, impl?, equi?`.
Indication : par exemple, `(define (pred x) (equal? x 'machin))` définit le prédicat *pred* qui prend en entrée une variable x et qui renvoie vrai si et seulement si $x = machin$, ce que vous pouvez tester en tappant `(pred 'machin)` d'une part et `(pred 'truc)` de l'autre.

Q 3 Définissez le prédicat `(connecteurBinaire? s)` qui renvoie vrai si et seulement si $s \in \{et, ou, impl, equi\}$ (vous pouvez utiliser le prédicat Scheme *member*).

Q 4 Définissez le prédicat `(symboleProp? s)` qui vérifie que s est un symbole Scheme [2] , différent d'un connecteur ;

Q 5 Ecrivez la fonction récursive `(fbf? f)` qui reconnaît si l'expression f est une formule bien formée en vous servant :
- des prédicat que vous venez de définir
- des fonctions Scheme du TDA liste

Q 6 Testez si les 3 formules `'(impl a b), b, '(ou et (non a))` sont des fbf. Vérifiez également vos formules F1, F2, F3 et F4 saisies précédemment.

1. un *prédicat* est une fonction qui renvoie un booléen
2. vous aurez à utiliser le prédicat Scheme *symbol*?.

Manipulation des propositions

Q 7 Définissez la fonction `ssFbf` : $\{1,2\} \times Fbf \rightarrow Fbf$ définie pour des fbf non réduites à un symbole propositionnel t.q. $(ssFbf\ i\ f) = f'$ (avec $i \leq$ arité du connecteur racine de f) ssi f' est la sous formule de f correspondant au sous arbre gauche de la racine si $i = 1$, au sous-arbre droit si $i = 2$.
Il s'agit bien sûr d'un renommage de *cadr* et *caddr*

Q 8 Définissez de même la fonction `connecteur` : $Fbf \rightarrow \{non, et, ou, impl, equiv\}$ définie pour des fbf non réduites à un symbole propositionnel et qui renvoie le connecteur à la racine de l'arborescence représentant la dite formule.

Remarque. On dispose maintenant d'un type abstrait fbf grace à des fonctions de manipulation des objets ce type (*symboleProp*?, *neg*?, *ssFbf*...). Dans la suite seules ces fonctions sont utiles (i.e. on **n'utilisera plus** les fonctions Scheme *length*, *cadr* et *caddr*).

Analyse syntaxique des propositions

Q 9 Ecrivez la fonction `nbc` : $Fbf \rightarrow \mathbb{N}$ qui retourne le nombre de connecteurs d'une proposition (nombre d'occurrences).

Q 10 Ecrivez la fonction `prof` : $Fbf \rightarrow \mathbb{N}$ qui retourne la profondeur de l'arbre associé à la proposition donnée.

Q 11 Ecrivez la fonction `symbProp` : $Fbf \rightarrow 2^{SP}$ qui retourne **l'ensemble** des symboles propositionnels d'une proposition donnée. Vous utiliserez les fonctions et prédicats ensemblistes pour l'appartenance, l'union, l'inclusion... Consultez l'aide, ou réécrivez les fonctions de manipulations d'ensemble (en anglais *set*).
Sachez en particulier, à partir d'un ensemble, reconstruire la liste des éléments de cet ensemble (fabriquez éventuellement la fonction ad-hoc `ensembleVersListe`).
Par exemple `(ensembleVersListe(symbProp '(et p (ou q p))))` devra renvoyer la liste composée (dans un ordre quelconque) des symboles `p` et `q`.

Affichage (infixé) d'une formule (exercice optionnel)

Q 12 Ecrire une fonction `afficher` qui prend en donnée une fbf, ne retourne rien et a comme effet de bord d'afficher la fbf sous forme infixée (affichage classique). Par exemple si
`F:=(impl a (non (et b (non c))))` `Afficher(F)` produira `(a impl non (b et non c))`.

TD3 : Sémantique de la logique des propositions

1. Soient les fbf :
   - $A = p \wedge (\neg q \rightarrow (q \rightarrow p))$
   - $B = (p \vee q) \leftrightarrow (\neg p \vee \neg q)$

   (a) Soit I une interprétation. Déterminer (si c'est possible) $v(A, I)$ et $v(B, I)$ dans chacun des 4 cas suivants :
   - on sait que $I(p)$=*faux* et $I(q)$=*vrai*
   - on sait que $I(p)$=*vrai* et $I(q)$=*faux*
   - on sait que $I(p)$=*faux*
   - on sait que $I(q)$=*vrai*
   - on ne sait rien sur $I(p)$ et $I(q)$

   (b) Les fbf A et B sont-elles satisfiables ? Valides ?

   (c) L'ensemble $\{A, B\}$ est-il consistant ?

2. La sémantique d'une formule est déterminée par celle des symboles propositionnels qui la composent. Combien d'interprétations différentes peut-on donner aux symboles propositionnels de chaque formule suivante. Pour chacune d'elles, donnez la valeur de vérité de la formule.

   (a) $p \vee \neg p$

   (b) $p \wedge p$

   (c) $p \wedge \neg p$

   (d) $p \vee q$

   (e) $p \vee (q \vee \neg q)$

   (f) $(p \rightarrow \neg\neg p) \wedge (\neg\neg p \rightarrow p)$

   (g) $(\neg p \vee q) \leftrightarrow (p \rightarrow q)$

   (h) $\neg(\neg p \vee q) \vee (r \rightarrow (p \leftrightarrow q))$

3. Démontrez pour chaque couple de formules suivantes qu'elles sont "sémantiquement équivalentes" (c'est-à-dire que leur valeur de vérité est identique quelque soit l'interprétation des symboles propositionnels) :

   (a) p et $\neg\neg p$

   (b) $p \rightarrow q$ et $\neg p \vee q$

   (c) $(p \rightarrow q) \wedge (q \rightarrow p)$ et $p \leftrightarrow q$

   (d) $p \wedge \neg p$ et $\bot$

   (e) $(p \wedge q) \vee (\neg p \wedge \neg q) \vee ((p \wedge s) \wedge \neg p)$ et $((p \rightarrow q) \wedge (r \rightarrow s \vee r)) \wedge (\neg p \rightarrow \neg q)$

   (f) $p \vee \neg\bot$ et $\neg\bot$

   (g) $\neg(p \wedge q)$ et $\neg p \vee \neg q$

   (h) $p \wedge (q \vee r)$ et $(p \wedge q) \vee (p \wedge r)$

   (i) $p \wedge (q \vee \neg q)$ et p

4. Le connecteur $\vee$ correspond au "ou inclusif". Nous n'avons pas introduit de connecteur pour exprimer le "ou exclusif". Donnez une table de vérité pour ce connecteur. Quelles formules de la logique des propositions correspondent à ce connecteur c'est à dire ont la même sémantique que "p ou exclusif q" ? Mêmes questions avec le connecteur ternaire "si p alors q sinon r" dont la sémantique est celle de l'instruction correspondante des langages de programmations.

5. Dites parmi les formules suivantes lesquelles sont équivalentes :
   (a) $(A \wedge B) \rightarrow C$
   (b) $(A \rightarrow C) \wedge (B \rightarrow C)$
   (c) $(A \rightarrow C) \vee (B \rightarrow C)$
   (d) $(A \vee B) \rightarrow C$
   (e) $A \rightarrow (C \wedge D)$
   (f) $A \rightarrow (C \vee D)$
   (g) $(A \rightarrow C) \wedge (A \rightarrow D)$
   (h) $(A \rightarrow C) \vee (A \rightarrow D)$
6. Dire si les formules suivantes sont valides, insatisfiables ou contingentes
   (a) $((p \rightarrow q) \rightarrow p)$
   (b) $(p \rightarrow (q \rightarrow p))$
   (c) $((p \wedge q) \leftrightarrow (p \rightarrow \neg q))$
   (d) $(p \vee q) \wedge (\neg p \vee q) \wedge (\neg q \vee p) \wedge (\neg p \vee \neg q \vee r) \wedge (\neg r \vee s)$
7. Que pensez-vous des affirmations suivantes :
   (a) si une formule est contingente, sa négation l'est également
   (b) si G et H sont 2 formules contingentes, alors $G \vee H$ et $G \wedge H$ sont 2 formules contingentes
   (c) si $G \vee H$ est insatisfiable alors G et H sont 2 formules insatisfiables
   (d) si $G \vee H$ est valide alors G et H sont 2 formules valides
8. Montrez que pour toute formule H, il existe une formule H' équivalente à H et n'ayant comme connecteurs logiques que la négation et l'implication.
   Appliquer à la formule$((p \vee q) \leftrightarrow (r \wedge s))$
9. Soit le connecteur *nand* défini par p *nand* $q =_{def} \neg(p \wedge q)$. Montrez que toute formule est équivalente à une formule ayant *nand* comme seul connecteur. Appliquez à la formule $p \rightarrow q$

TP2 : interprétation

Définition d'une structure d'interprétation

On représente l'interprétation d'un symbole propositionnel en Scheme par une liste à deux éléments (p, v) où p est un symbole propositionnel et $v \in \{0, 1\}$.
La **représentation d'une interprétation** I est alors une liste de tels couples, telle que chaque symbole propositionnel n'a qu'une seule valeur associée (sa *valeur d'interprétation* pour I).
Une représentation d'une interprétation est donc la représentation d'une fonction.
Par exemple après `(define I0 '((a 0)(b 1)))`, `I0` sera la représentation de l'interprétation I_0 telle que $I_0(a) = faux$ et que $I_0(b) = vrai$.

Q 13 Définissez en Scheme les 3 interprétations $I1$, $I2$, $I3$ suivantes : $I1(a) = I1(c) = vrai$ et $I1(b) = faux$, $I2(a) = I2(b) = I2(c) = faux$, $I3(a) = I3(b) = I3(c) = vrai$.

Q 14 Ecrivez la fonction `symbInt` : $Interpretation \rightarrow 2^{SP}$ qui retourne l'ensemble des symboles propositionnels d'une interprétation.
Par exemple, *symbInt(I1)* renverra en l'ensemble des trois éléments a, b et c.

Sémantique d'une proposition

Q 15 Ecrivez le prédicat `intComp?` : $Interpretation \times Fbf \rightarrow Booleen$ qui retourne vrai si l'interprétation donnée permet d'associer une valeur à tous les symboles propositionnels de la formule donnée (l'interprétation est alors *complète* pour la formule).

Q 16 Ecrivez la fonction `intSP` : $SP \times Interpretation \rightarrow \{0, 1\}$ qui retourne la valeur d'interprétation d'un symbole propositionnel donné dans une interprétation donnée (on supposera que ce symbole propositionnel apparaît dans la structure d'interprétation).

Q 17 Ecrivez les fonctions qui sont les interprétations fixes (telles qu'elles on été vues en cours) des connecteurs logiques : `intNon` : $\{0, 1\} \rightarrow \{0, 1\}$, `intEt` : $\{0, 1\} \times \{0, 1\} \rightarrow \{0, 1\}$, `intOu, intImpl, intEqui`.
Par exemple `(define (intNand x y) (*x y))` fournirait la fonction interprétation du connecteur logique `nand`.

Q 18 Finalement, écrivez la fonction `valV` : $Fbf \times Interpretation \rightarrow \{0, 1\}$ qui calcule la valeur de vérité d'une formule f pour une interprétation i complète pour f.

Satisfiabilité et validité d'une proposition

Pour étudier les propriétés sémantiques des propositions, on utilisera des listes d'interprétations.
Pour tester la satisfiabilité d'une proposition, il faut calculer la liste de ses interprétations qui ne dépend que de la liste des symboles propositionnels apparaissant dans la proposition.

Remarque. Pour ce qui suit les fonctionnelles Scheme de type `map` peuvent être utiles [3]

3. E étant une liste et f une fonction unaire définie sur E : `(map f E)` renvoie $E' = \{(f\ e) | e \in E\}$ (i.e. construit une liste composé des résultats de l'application de f à chaque élément de E). Ex. : (*map* (*lambda* (*x*) (*sqrt x*)) (*list* 16 9 4 81)) renvoie (4 3 2 9) ;

Q 19 Liste des interprétations d'une liste de SP. Ecrire une fonction `lisInt` qui prend en donnée une liste de symboles propositionnels (`lisSP`) et retourne la liste de toutes les interprétations (`lisIntp`) de ces symboles propositionnels. Lorsqu'il n'y a qu'un symbole propositionnel, il n'y a que 2 interprétations possibles. Si il y en a plus d'une, il est judicieux de calculer récursivement la liste $\mathcal{I}$ des interprétations de tous les symboles sauf le premier, puis de prendre en compte le premier symbole en ajoutant à chaque interprétation de $\mathcal{I}$ l'interprétation du premier symbole (une fois à 0 et une fois à 1).

Q 20 Écrire un prédicat `satisfiable?` qui retourne true si et seulement si une fbf donnée est satisfiable. Tester votre prédicat sur les propositions $a, \neg a, (a \wedge b), ((a \wedge b) \wedge \neg a), F_1, F_2, F_3, F_4$.

Q 21 Écrire un prédicat `valide?` qui retourne true si et seulement si une fbf donnée est valide. Tester votre prédicat sur les propositions $a, \neg a, (a \vee b), ((a \vee b) \vee \neg a), F_1, F_2, F_3, F_4$.

TD 4 Modélisation

1. Le lieutenant Colombo enquête sur le crime ayant eu lieu dans la nuit du 1 au 2 octobre. Il dispose des informations suivantes :
   (a) Jacques ou Martin est coupable
   (b) si Martin est coupable alors le crime a eu lieu avant minuit
   (c) si le crime a eu lieu à partir de minuit alors Jacques est coupable
   (d) le crime a eu lieu avant minuit
      i. Que peut-il en déduire sur l'identité du (ou des) coupable(s) ?
      ii. Même question s'il dispose de l'information supplémentaire
   (e) si Jacques est coupable alors le crime a eu lieu à partir de minuit.
2. On considère les énoncés suivants, où $p,\ q,\ r,\ s$ et t représentent eux-mêmes des énoncés. Les écrire sous forme de fbf.
   (a) si p alors q
   (b) pour que p, il suffit que q
   (c) pour que p, il faut que q
   (d) p est une CNS pour que q
   (e) soit p, soit q, mais pas les deux
   (f) si p alors q sinon r
   (g) si p alors q à moins que r
   (h) si p alors q sauf si r
   (i) si p alors si q alors r sinon s
   (j) si p alors si q alors r sinon s sinon t
3. Modélisez les énoncés suivants en logique des propositions
   (a) Il suffit à Eric d'assister aux cours et aux TD pour qu'il ait la moyenne.
   (b) R est une relation d'équivalence si et seulement si R est réflexive, symétrique et transitive.
   (c) Si Rose n'est pas vaccinée, il suffit d'une coupure pour qu'elle attrape le tétanos.
   (d) Si Pierre est chez lui, il lit ou il écoute de la musique.
   (e) Le Sida ne sera pas éradiqué à moins qu'un nouveau vaccin ne soit découvert.
   (f) Il est nécessaire d'avoir du courage et de l'habileté pour escalader cette paroi.
4. – Modélisez en logique des propositions les phrases suivantes :
   (a) Jean est à son bureau, à moins qu'il soit en train de déjeuner.
   (b) Jean a un fils ou deux filles.
   (c) Jean n'a ni fils, ni filles
   (d) Ni Jean ni Bernard ne possède à la fois un lecteur MP3 et un graveur.
   – En vous aidant de cette modélisation, proposez pour chaque phrase, une phrase exprimant la négation de la phrase d'origine.

5. *Enigme* Vous vous trouvez sur une île un peu étrange : l'île de Puro-Pira.
   Vous savez qu'à part vous, on y trouve deux catégories de gens : les Purs, qui ne disent que des choses vraies, et les Pires, qui ne disent que des choses fausses.
   Alice et Bernard sont deux habitants de l'île de Puro-Pira. Il se peut qu'ils soient deux Purs, deux Pires, une Pure et un Pire,... Tout est possible !
   Déterminez dans chacune des situations indépendantes suivantes si Alice et Bernard sont des Purs ou des Pires.
   Vous justifierez votre réponse en modélisant et résolvant formellement ce problème.
   – 1 : Vous rencontrez Bernard qui vous dit : *Alice et moi sommes tous les deux des Pires*
   – 2 : Vous rencontrez Alice qui vous dit : *Si je suis une Pure alors Bernard est un Pire*
   – 3 : Vous rencontrez Alice et Bernard : Alice dit : *Je suis une Pure ou Bernard est un Pur* et Bernard dit : *Nous ne sommes pas du même type.*
   – Trouvez une phrase que ni un Pur ni un Pire ne peut dire.
   – Trouvez une phrase qui peut-être dite à la fois par un Pur et un Pire.
6. On cherche à deviner la position d'un certain nombre de bateaux sur une grille de bataille navale possédant 2 lignes, appelées a et b, et 3 colonnes, appelées 1, 2, 3. Pour cela on dispose des informations suivantes :
   – Il y a au moins un bateau sur la ligne b
   – Il y a au moins un bateau sur la ligne a
   – Il n'y a pas 2 bateaux sur une même colonne
   – Il n'y a pas de bateau en (b,1)
   – Si il y a un bateau sur la ligne a alors il n'y a pas de bateau en (b,3).
   En notant x_i (pour $x \in \{a, b\}$ et $i \in \{1, 2, 3\}$) l'affirmation : *Il y a un bateau à la position* (x, i) , modélisez par une formule de la logique des propositions les 5 affirmations ci-dessus. Puis traduisez sous forme logique le problème posé.
7. On dispose de 3 cases alignées notées 1, 2 et 3 (la case 1 est à gauche, la 2 au centre et la 3 à droite) et de pions de différentes formes : triangle, rond ou carré. Les pions peuvent être placés dans les cases. On notera c_1 l'affirmation : *L'emplacement 1 contient un pion carré* et on fera de même pour les autres formes et les autres numéros de case.
   Modélisez les deux phrases : *Il y a un pion rond immédiatement à droite d'un pion carré.* et *Chaque emplacement contient soit un pion rond soit un pion carré soit un pion triangulaire.*
   Donnez les modèles communs à ces 2 formules. Que peut- on en conclure quant au pion situé dans la case 2 ?
8. Soient les énoncés suivants :
   (a) Si Robert a mal au genou, il ne peut pas jouer à la fois les matchs 1 et 2.
   (b) L'équipe de Bouzols perd le match 1 si et seulement si Robert ne joue pas ce match.
   (c) Si Robert ne participe pas au second match alors il est nécessaire qu'il fasse chaud pour que Bouzols gagne le match 2.
   (d) Pour être qualifiée, l'équipe de Bouzols doit nécessairement gagner les matchs 1 et 2.
   (e) Si l'équipe de Bouzols est qualifiée et qu'il ne fait pas chaud alors Robert n'a pas mal au genou.

   Modélisez chacune de ces phrases en logique des propositions. Vous modéliserez chaque phrase telle qu'elle est énoncée. Le sport pratiqué par Robert et les habitants de Bouzols n'admet pas de résultat nul. En conséquence dire qu'un match est perdu est équivalent à dire qu'il n'est pas gagné.

TD 5 : Formes normales et clausales

1. Donnez des formules correspondant à la table de vérité suivante, en particulier les FNC et FND :

p	q	r	
vrai	*vrai*	*vrai*	*faux*
vrai	*vrai*	*faux*	*faux*
vrai	*faux*	*vrai*	*vrai*
vrai	*faux*	*faux*	*faux*
faux	*vrai*	*vrai*	*vrai*
faux	*vrai*	*faux*	*vrai*
faux	*faux*	*vrai*	*vrai*
faux	*faux*	*faux*	*faux*

2. Soit la fbf : $F =_{def} ((r \to p) \to (\neg(q \vee r) \to p))$
dessinez l'arborescence syntaxique de F puis mettez F sous forme conjonctive. Finalement donnez sa forme clausale.
3. Mettez sous forme clausale les fbf suivantes :
   (a) $(p \leftrightarrow (q \vee \neg(r \wedge p)))$
   (b) $\neg((b \to a) \to \neg c \wedge \neg(d \to e \wedge f))$
4. *Subsomption* : Soit deux clauses C_1 et C_2, on dit que C_1 subsume C_2 (i.e. C_2 est une clause subsumée par C_1) si et seulement si $C_1 \subseteq C_2$. Montrez que si C_1 subsume C_2 et que C_1 est satisfiable alors C_2 est satisfiable. En déduire qu'une forme clausale F est insatisfiable si et seulement si F débarrassée des clauses subsumées est insatisfiable.
5. *Tautologie* : Montrez qu'une forme clausale F est insatisfiable si et seulement si F débarrassée des clauses tautologiques (valides) est insatisfiable.

   Notation : Soit F une forme clausale et l un littéral, on note $F[l]$ la forme clausale obtenue à partir de F en supprimant les clauses contenant le littéral l et en supprimant le littéral opposé à l des autres clauses.

6. Un littéral pur d'une forme clausale F est un littéral qui n'apparaît que sous une seule forme dans F, i.e. F contient des occurrences de l mais aucune occurrence de $\neg l$. Soit F une forme clausale et l un littéral pur de F, montrez que $F[l]$ est insatisfiable si et seulement si F est insatisfiable.
7. Une clause unitaire est une clause ne contenant qu'un seul littéral. Montrez que si $\{l\}$ est une clause unitaire d'une forme clausale alors F est insatisfiable si et seulement si $F[l]$ est insatisfiable.
8. Soit F une forme clausale et l un littéral de F (et soit l' le littéral opposé), montrez que F est insatisfiable si et seulement si $F[l]$ et $F[l']$ sont insatisfiables.

TP4 : formes conjonctives et formes clausales

Mise sous forme conjonctive

Les 4 premières questions visent à fournir les transformations de fbf permettant un passage à la forme conjonctive. Pour ces 4 fonctions, il faut raisonner sur l'arbre syntaxique associé à la formule. La 5e vise à fournir cette fonction. Attention, la 4e fonction (`distOu`) est particulièrement délicate. Ex. : `distOu(et(et(a,non(b)),ou(c,ou(non(d),et(e,f)))))` doit retourner `et(et(a,non(b)),et(ou(c,ou(non(d),e)),ou(c,ou(non(d),f))))`.

Q 22 Ecrire une fonction récursive `oteEqui` qui prend en paramètre une fbf et retourne une fbf logiquement équivalente qui ne contient pas de connecteur $\leftrightarrow$.
Rappel : $(A \leftrightarrow B) \equiv ((A \rightarrow B) \wedge (B \rightarrow A))$

Q 23 Ecrire une fonction récursive `oteImpl` qui prend en paramètre une fbf et retourne une fbf logiquement équivalente qui ne contient pas de connecteur $\rightarrow$.
Rappel : $(A \rightarrow B) \equiv (\neg A \vee B)$

Q 24 Ecrire une fonction récursive `redNeg` qui prend en paramètre une fbf ne contenant pas de connecteur $\leftrightarrow$ et $\rightarrow$ et retourne une fbf logiquement équivalente dont la négation ne porte que sur les symboles propositionnels.
Rappel : $\neg\neg A \equiv A$, $\neg(A \wedge B) \equiv (\neg A \vee \neg B)$, $\neg(A \vee B) \equiv (\neg A \wedge \neg B)$

Q 25 Ecrire une fonction récursive `distOu` qui prend en paramètre une fbf composée de littéraux connectés par des $\wedge$ et $\vee$ et retourne une fbf logiquement équivalente sous forme conjonctive (i.e. conjonction de disjonctions de littéraux). Rappel : $(A \vee (B \wedge C)) \equiv ((A \vee B) \wedge (A \vee C))$

Q 26 Ecrire alors la fonction `formeConj` qui prend en paramètre une fbf quelconque et retourne une fbf logiquement équivalente sous forme conjonctive.

Forme clausale

Q 27 On veut représenter une forme clausale comme une liste de listes de littéraux. Ecrire un prédicat `litteral?` qui quelque soit son (ses) paramètre(s) retourne vrai si et seulement si son unique paramètre est un littéral (positif ou négatif).

Q 28 Définir alors les prédicats `clause?` et `lisClauses?`.

Q 29 Ecrire une fonction récursive `transfClause` qui prend en paramètre une fbf disjonction de littéraux et retourne la clause correspondante à cette fbf.

Q 30 Ecrire une fonction récursive `transfLisClauses` qui prend en paramètre une fbf sous forme conjonctive et retourne la liste de clauses (`lisClauses`) correspondant à cette fbf.

Q 31 Finalement, écrire une fonction `formeClausale` qui prend en paramètre une fbf quelconque et retourne la liste de clauses (`lisClauses`) correspondant à sa forme clausale.

TD 6 : méthodes de preuve

1. Démontrez en utilisant exclusivement la méthode de résolution que :
   $\{a \to b, (c \wedge d) \to a, e \to c, d \wedge e\} \models b$
2. Dites en utilisant la méthode des tableaux si les fbfs suivantes sont satisfiables :
   (a) $((\neg(b \wedge a) \to (a \leftrightarrow b)) \wedge \neg(\neg a \vee b))$
   (b) $((\neg p \wedge (\neg q \vee r)) \vee (p \to (q \wedge \neg r))) \wedge (p \leftrightarrow \neg q)$
3. Montrez que les formules suivantes sont valides à l'aide de la méthode des tableaux :
   (a) $((A \vee B) \to (A \vee C)) \to (A \vee (B \to C))$
   (b) $((A \wedge (B \to C)) \to D) \to ((\neg A \vee B \vee D) \wedge (\neg A \vee \neg C \vee D))$
   (c) $((A \vee B) \wedge (\neg A \vee B) \wedge (A \vee \neg B)) \to \neg(\neg A \vee \neg B)$
4. Montrez à l'aide de la méthode de résolution puis à l'aide de la méthode des tableaux que le raisonnement suivant n'est pas valide :
   $\{(p \to (q \vee r)), (q \to (r \to s)), \neg s\} \models \neg p$
5. Soit les formules bien formées suivantes de la logique des propositions :
   – $A =_{def} \neg(q \wedge \neg r) \wedge (p \to q \vee (r \wedge \neg p))$
   – $B =_{def} \neg(r \to s) \vee \neg p \vee (r \wedge s)$
   (a) Calculez les valeurs de vérité de A et B pour l'interprétation I suivante :
   $I(p) =$ *vrai*, $I(q) =$ *faux*, $I(r) =$ *vraiet*$I(s) =$ *vrai*.
   Que peut-on en conclure sur A et B ?
   (b) Mettre A sous forme clausale.
   (c) Montrez par la méthode de résolution que B se déduit logiquement de A ($A \models B$).
   (d) Montrez cette déduction par la méthode des tableaux.
6. Le résultat de la méthode de résolution est : *une forme clausale FC est insatisfiable si et seulement si il existe une résolution de FC terminant par la clause vide (une telle résolution est appelée réfutation)* . Mais comment obtenir une réfutation ? Un moyen simple consiste à calculer toutes les résolvantes possibles. L'ensemble des résolvantes possibles étant fini cette stratégie garantit de ne pas rater la clause vide (on dit que la stratégie est complète). Mais elle est coûteuse. Nous proposons ici d'autres stratégies
   – Une première stratégie est la unit-résolution qui ne calcule les résolvantes qu'entre une clause unitaire et une clause quelconque.
   (a) Appliquez l'unit-résolution aux clauses $\{a \vee c, \neg a \vee b, \neg c, \neg b\}$
   (b) L'unit-résolution est-elle complète ?
   – Une seconde stratégie possible construit à partir d'une forme clausale FC, la séquence de formes clausales $S_0, S_1, \ldots S_n$ où :
   – $S_0 =_{def} FC$,
   – Pour tout $i \in [0 \ldots n-1]$, S_{i+1} est l'ensemble des résolvantes de clauses de S_i.
   On ne calcule donc les résolvantes qu'entre clauses produites à l'étape précédente.
   (a) Calculez la séquence produite par les clauses précédentes ;
   (b) Cette stratégie est-elle complète ?
7. Soit la forme clausale suivante (correspondant à la description de la bataille navale du TD4) :
   $D =_{def} \{\{a_1, a_2, a_3\}, \{b_1, b_2, b_3\}, \{\neg a_1, \neg b_1\}, \{\neg a_2, \neg b_2\}, \{\neg a_3, \neg b_3\}, \{\neg b_1\}\{\neg a_1, \neg b_3\}\{\neg a_2, \neg b_3\}\}$
   Prouver par la méthodede l'exercice précédent que l'on peut déduire qu'il y a un bateau en b_2 et pas de bateau en a_2, en montrant que $D \cup \{\{\neg b_2, a_2\}\}$ est insatisfiable :

TP5 : Méthodes de preuve

Il s'agit d'implanter l'une des méthodes vues en cours. Il est préférable d'en implanter une correctement et complètement plutôt que d'essayer de toutes mal les faire ! On s'attachera en particulier à se doter d'un jeu d'essais permettant de tester la méthode.

Q 32 Mettre en œuvre la méthode de résolution.

Q 33 Mettre en œuvre la méthode des Tableaux.

Application

Q 34 Modéliser un problème en logique des propositions et résolvez-le à l'aide d'une des 3 méthodes précédentes.

TD 7 : Logique des prédicats

1. Modélisez en logique des prédicats du premier ordre les énoncés ci-dessous :
   - Toute femme a au moins une fille.
   - Il y a au moins une femme qui a au moins une fille.
   - Aucune brouette n' est confortable. Tout véhicule inconfortable n' a aucun succès. Certains véhicules ont du succès.
2. Soit la formule $\forall x\ (\exists y\ p(x,y) \rightarrow q(x))$
   - Dessinez l'arborescence syntaxique associée à cette formule
   - Donnez un modèle et un contre-modèle
3. Parmi les formules suivantes, lesquelles sont valides. Justifiez votre réponse en donnant un contre-modèle pour les non-valide et une preuve pour les valides.
   - $A =_{def} ((\forall x\ p(x) \wedge (\exists y\ q(y) \vee \forall y p(y))) \rightarrow \exists x\ p(x))$
   - $B =_{def} ((\forall x\ p(x) \wedge (\exists y\ q(y) \vee \forall y p(y))) \rightarrow \forall x\ p(x))$
   - $C =_{def} ((\forall x\ p(x) \rightarrow \forall x\ q(x)) \rightarrow \forall x\ (p(x) \rightarrow q(x)))$
   - $D =_{def} (\forall x\ \exists y\ p(x,y) \rightarrow \exists y\ \forall x\ p(x,y))$
4. Trouver des formules de la logique des prédicats pouvant être interprétées par les phrases suivantes. Donner pour chaque formule l'interprétation des constantes et des symboles de prédicats :
   - Tout ce qui est en or brille.
   - Tout ce qui brille n'est pas de l'or.
   - Tous les oiseaux volent sauf les autruches qui sont des oiseaux mais ne volent pas.
   - Si Eric est un étudiant sérieux alors tous les étudiants sont sérieux
   - Toto a lu un livre qui a été lu par tous ses amis sauf par son ami Juju.

   Pour chaque formule montrez qu'elle est contingente en choisissant un domaine non vide D et une interprétation I une fois modèle, une fois contre-modèle de la formule.
5. Soient les formules :
   - $A =_{def} \forall x\ (p(x) \rightarrow p(a))$
   - $B =_{def} (\forall x\ p(x) \rightarrow p(a))$
   - $C =_{def} (p(a) \rightarrow \forall x\ p(x))$
   - $E =_{def} \exists x\ (p(x) \rightarrow p(a))$
   - $F =_{def} (\exists x\ p(x) \rightarrow p(a))$
   - $G =_{def} (p(a) \rightarrow \exists x\ p(x))$

   et l'interprétation I dont le domaine D est l'ensemble des salles de cours de l'UM2 et telle que $I(a)$ est la salle de cours numéro (SC.1.01) et $I(p)$ est l'application de D dans $\{vrai, faux\}$ telle que pour tout élément d de D , $I(p)(d) = vrai$ si et seulement si la salle d est ouverte
   - Calculer $v(A,I),\ v(B,I),\ldots$ et $v(F,I)$ sachant que la salle de cours 1 est ouverte et que toutes les autres sont fermées.
   - Calculer $v(A,I),\ v(B,I),\ldots$ et $v(F,I)$ sachant que la salle de cours 1 est fermée et que toutes les autres sont ouvertes.
   - Lesquelles des formules $A,\ B,\ C,\ E,\ F$ et G sont elles valides ?
6. Soit $\mathcal{L}$ un langage de la logique du premier ordre contenant deux symboles de prédicats unaires p et q. On considère les formules suivantes :
   - $A =_{def} \forall x\ (p(x) \rightarrow q(x))$
   - $B =_{def} \exists x\ (p(x) \rightarrow q(x))$

   On se propose d'étudier les interprétations pour lesquelles ces formules sont vraies.
   Soit D un ensemble non vide et I une interprétation de $\mathcal{L}$ de domaine D.
   On note $D_p =_{def} \{d \in D \ \mid\ I(p)(d) = 1\}$ et $D_q =_{def} \{d \in D \ \mid\ I(q)(d) = 1\}$

(a) Dans cette question, $D = \{1, 2, 3\}$ et $D_p = \{1, 2\}$. Quelle condition doit vérifier D_q pour que A soit vraie pour I ? Pour que B soit vraie pour I ?

(b) On suppose que D est un ensemble non vide quelconque. Quelles conditions doivent vérifier D_p et D_q pour que A soit vraie pour I ? Pour que B soit vraie pour I ?

7. Soit une relation binaire R dans un ensemble E et un graphe orienté $G = (X, U)$. Trouver des formules de la logique des prédicats pouvant être interprétées par les phrases suivantes. Donner pour chaque formule le domaine d'interprétation et l'interprétation des constantes et des symboles de prédicats.

   (a) R est une relation d'équivalence (c'est-à-dire R est réflexive, symétrique et transitive)

   (b) R n'est pas une relation d'équivalence

   (c) G a au moins une source et au moins un puits

   (d) G n'a ni source ni puits

   On suppose que $E = X = \{1, 2\}$ et $R = U = \{(1, 1), (1, 2)\}$. Préciser pour chaque formule l'interprétation correspondante et calculer la valeur de vérité de la formule pour cette interprétation.

8. Parmi les équivalences suivantes où F et G sont des fbf quelconques contenant au moins une occurrence libre de x , lesquelles sont correctes ?
   - $(\exists x\ F \vee \exists x\ G) \equiv \exists x\ (F \vee G)$
   - $(\exists x\ F \wedge \exists x\ G) \equiv \exists x\ (F \wedge G)$
   - $(\forall x\ F \vee \forall x\ G) \equiv \forall x\ (F \vee G)$
   - $(\forall x\ F \wedge \forall x\ G) \equiv \forall x\ (F \wedge G)$