UFR Sciences et Techniques

Licence S&T – 1ère année

Logique pour l'informatique

TRAVAUX DIRIGÉS

Enseignant **Jean-Yves ANTOINE** (Jean-Yves.Antoine AT univ-tours.fr)

Introduction à la logique

QUESTIONS DE COURS

Exercice 1 — Les profs de maths de lycée sont-ils menteurs ? (contrôle 2001 – Objectif 1.1.2.)

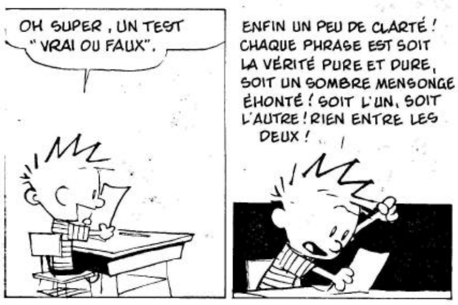
Tout au long de votre scolarité, on vous a parlé en mathématiques de théorèmes. Mais en utilisant cette notion, ne vous aurait-on pas menti par omission ? Que pensez-vous en particulier des assertions suivantes :

1 — *Pour être un théorème il suffit qu'un énoncé mathématique soit vrai*

☐ vrai ☐ faux

2 — *Un théorème peut-être faux*

☐ vrai ☐ faux



Exercice 2 — La vérité est toujours relative (contrôle 2007-2008 – Objectif 1.1.6.)

On considère un système logique qui est sain (consistant) mais pas complet**.** Que pensez-vous des deux affirmations suivantes, prises globalement :

a) *Dans ce système, tout théorème est vrai*

b) *Dans ce système, tout ce qui est vrai n'est pas forcément un théorème*

En utilisant les atomes `Est_Theoreme` et `Est_Vrai`, donnez la représentation **la plus simple** possible de cet ensemble de deux affirmations en logique des propositions.

Logique des propositions (LP)

QUESTIONS DE COURS

Exercice 3 — Syntaxe LP (objectif : 2.1.1)

On considère les propositions suivantes :

a) $\neg B \Rightarrow (B \vee R)$ b) $\neg B \Rightarrow (B \Rightarrow P)$ c) $\neg(B \vee R)$

d) $(B \wedge R) \Rightarrow C$ e) $((B \vee R) \wedge \neg C) \Rightarrow \neg P$ f) $P \Rightarrow (B \Leftrightarrow C)$

Quelles sont les propositions dans lesquelles les parenthèses sont inutiles ?

Exercice 4 — Météorologie logique (objectif : 2.2.1)

En utilisant les atomes S (le soleil brille), P (il pleut), B (il bruine), A (il y a un arc en ciel), O (il a un vent d'Ouest), E (il y a du vent d'Est) Traduire dans la logique des propositions les énoncés suivants :

1) *S'il pleut et que le soleil brille en même temps, alors il y a un arc en ciel*

2) *Si le vent d'ouest amène la pluie, on n'a jamais vu qu'un vent d'est soit porteur de pluie.*

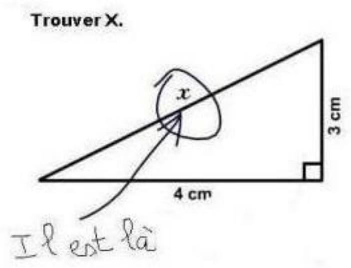
3) *La bruine est une forme de pluie.*

TRADUCTION EN LP : EXERCICES

Exercice 5 — Petit précis de géométrie euclidienne (objectif 2.2.1)

Représenter dans le formalisme de la logique des propositions les théorèmes de géométrie suivants :

a — *Si un triangle est équilatéral alors il est isocèle.*
b — *Un triangle rectangle n'est jamais équilatéral.*
c — *Un carré est à la fois un parallélogramme et un rectangle.*
d — *Un losange n'est ni un quadrilatère rectangle ni un triangle.*
e — *Deux droites coplanaires sont soit sécantes, soit parallèles.*
f — *Deux droites ne peuvent être à la fois sécantes et parallèles.*



Exercice 6 — Rencontres logiques (10' ; objectif 2.2.1.)

Traduire chacune des phrases suivantes en une formule du langage des propositions :

a — *Jamais il ne me rencontre sans me dire bonjour.*
b — *Lorsqu'il ne me dit pas bonjour, c'est qu'il est fâché.*
c — *Il s'arrange pour ne jamais me rencontrer lorsqu'il est fâché..*
d — *Ou bien on va se dire bonjour ou bien il est fâché..*

Exercice 7 — Un peu de gastronomie... (objectif 2.2.1)

On considère l'ensemble des connaissances suivantes, extraite d'un improbable cours d'école hôtelière.

Si le client a commandé un plat du jour, on accompagnera le repas d'une carafe d'eau..
Si le client a demandé un menu plus conséquent, ou a commandé à la carte, on servira obligatoirement du vin.
Trois sortes de vins peuvent être servis : du blanc, du rosé ou du rouge.
Avec de la viande, on servira du vin rouge et du rosé ou du blanc avec le poisson.

Représentez ces informations sous la forme d'un ensemble de formules de la logique des propositions.

Exercice 8 — Prix de l'essence et prévention routière (objectif 2.2.1)

Les forces de police ont observé qu'en France, la vitesse moyenne des automobilistes s'est réduite autour des années 2010. Quel est la cause de cette nouvelle sagesse des français, d'ordinaire plus prompts à acheter un détecteur de radar qu'à penser aux milliers de morts annuels sur les routes de l'Hexagone ? Pour y voir plus clair, donnez la traduction en logique des propositions des énoncés suivants (en ne traduire pas les parties entre crochets :

a. *La baisse des réserves pétrolières et la hausse de la consommation chinoise expliquent la cherté du pétrole*
b. *Ce n'est pas parce qu'il y a des radars [*à certains endroits*] que les gens réduisent leur vitesse [*ailleurs*].*
c. *C'est parce que le pétrole est cher que les gens roulent moins vite.*
d. *Avec l'assouplissement des règles de rattrapage de points de permis* [sous le gouvernement de François Fillon] *être flashé par un radar coûte moins cher.*
e. *Pour que les français roulent moins vite, il faut augmenter le prix du pétrole et non pas augmenter le nombre de radars*

Exercice 9 — Le diplômé motivé (objectif 2.2.1)

A peine sorti de licence informatique, Alphonse Oboulo se rue sur les annonces d'emploi, persuadé que la qualité de la formation reçue dans cette université sera immédiatement reconnue par les entreprises les plus prestigieuses. En réponse à l'une de ses 5 678 candidatures spontanée, le DRH de la World Company Inc., le convoque pour un entretien décisif. Remonté à bloc, Alphonse Oboulo, se précipite dans la première gare venue pour réserver un billet de train pour la ville où il est convoqué. Dans la file d'attente, il fait le raisonnement suivant (on ne considérera pour cela que les parties en caractères italique) :

Grèves SNCF : montée du mécontentement en province

(1) *En cas de grève, le train sera en retard ou annulé.*
(2) *Ne pas réussir l'audition et manquer le rendez-vous sont les 2 causes les plus sures de ne pas être embauché.*
(3) Compte tenu de la pré-sélection effectuée, *réussir à l'entretien est synonyme d'embauche.*
(4) *Qui dit train en retard dit rendez-vous manqué.*
(5) *Une chose est sûre, je ne raterai pas l'audition.*

(C) *Donc, seule une grève peut empêcher mon embauche*

Angoissé à l'idée d'avoir son avenir suspendu au climat social au sein de la SNCF, Alphonse Oboulo tente d'écarter ses idées noires en traduisant en logique des propositions son raisonnement. Saurez-vous l'aider ?

Exercice 10 — Après le 11 septembre : logique et géopolitique (environ 5' ; objectif 2.2.1💣)

1 — Complétez la traduction dans la logique des propositions de l'énoncé suivant : *Palestiniens et Israéliens ne sont pas tous nécessairement opposés à la paix*.

Palestinien Israélien Pacifiste

2 — Même question avec l'énoncé suivant : *Il n'y a pas nécessairement que des islamistes extrémistes parmi les Afghans et les Pakistanais.*

Afghan Pakistanais Islamiste_extrémiste

Exercice 11 — To tweet or not to tweet ? (objectif 2.2.1. 💣)

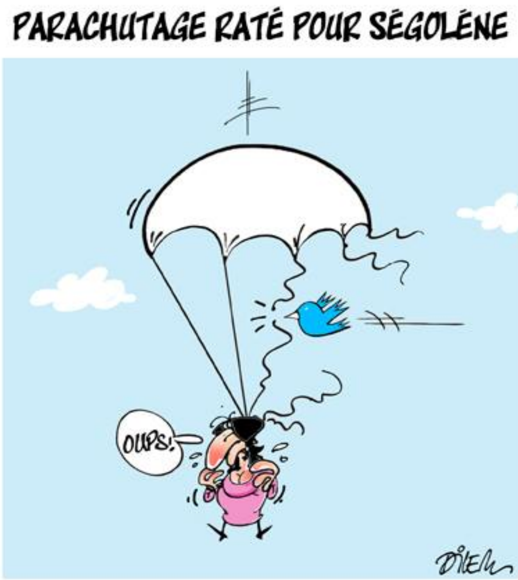
En envoyant un tweet de soutien à l'adversaire de Ségolène Royale aux dernières élections législatives, Valérie Trierweiler est parvenue à un exploit rare : générer des milliers d'articles journalistiques (une des quotidiens, mais aussi des hebdomadaires politiques) et commentaires sur les forum à partir d'un petit message de moins de 140 caractères... De quoi s'interroger sur les relations des français avec une politique de plus en plus people.

De quoi également donner des idées à votre enseignant pour son cours de logique : il vous est en effet demandé de traduire en logique des propositions les énoncés ci-dessous :

a — *Parler du tweet de Valerie Treilerweiler n'est pas la seule solution pour vendre un magazine.*

b — *Parler de Valerie Treiweiler et parler de Ségolène Royal, voilà deux façons d'avoir une vente assurée.*

c — *Ce n'est pas parce que François Hollande a changé de compagne qu'il faut parler de cette dernière.*



Exercice 12 — Histoires de connecteurs (environ 5'; objectifs 2.2.1 & 2.1.3)

1 — Complétez la traduction dans la logique des propositions de l'énoncé suivant : *La gloire pas plus que la richesse n'apportent le bonheur*.

Gloire Richesse Bonheur

2 — Donnez la traduction dans la logique des propositions de l'énoncé suivant : *Au resto U c'est toujours fromage ou dessert.*

Exercice 13 — La télé rend fou (10'; objectifs 2.2.1 & 2.1.3)

1. Quelle(s) fbf(s) représente(nt) l'énoncé suivant : *Pour gagner 2 heures de temps libre quotidien ou encore voir son Quotient Intellectuel augmenter, il est nécessaire (mais pas nécessairement suffisant, bien sûr !) de jeter sa télévision*

- ☐ Temps_libre_en_plus $\wedge$ QI_en_hausse $\Rightarrow$ Jeter_Television
- ☐ Temps_libre_en_plus $\vee$ QI_en_hausse $\Rightarrow$ Jeter_Television
- ☐ Temps_libre_en_plus $\wedge$ QI_en_hausse $\Leftrightarrow$ Jeter_Television
- ☐ Jeter_Television $\Leftrightarrow$ Temps_libre_en_plus $\vee$ QI_en_hausse
- ☐ Jeter_Television $\Rightarrow$ Temps_libre_en_plus $\wedge$ QI_en_hausse
- ☐ Jeter_Television $\Rightarrow$ Temps_libre_en_plus $\vee$ QI_en_hausse

2. De même, quelle(s) fbf(s) représente(nt) l'énoncé suivant : *Ne pas perdre ce n'est pas obligatoirement obtenir une victoire, alors qu'obtenir une victoire c'est toujours ne pas perdre*

- ☐ $Victoire \Rightarrow \neg Perdre$
- ☐ $\neg Perdre \Rightarrow Victoire$
- ☐ $\neg Perdre \Rightarrow \neg Victoire$
- ☐ $Victoire \Leftrightarrow \neg Perdre$
- ☐ $(Victoire \Rightarrow \neg Perdre) \wedge (\neg Perdre \Rightarrow Victoire)$
- ☐ $\neg(\neg Perdre \Rightarrow Victoire) \wedge (Victoire \Rightarrow \neg Perdre)$
- ☐ $(\neg Perdre \Rightarrow \neg Victoire) \wedge (Victoire \Rightarrow \neg Perdre)$

TRADUCTION EN LP : POUR ALLER PLUS LOIN

Exercice 14 — Logique, langage et littérature (objectif 2.2.1)

De tous temps, les logiciens ont cherché à représenter le langage, manifestation par excellence de la pensée humaine, sous forme logique. A premier abord, ce projet peut paraître raisonnable. Par exemple, représentez dans le formalisme de la logique des propositions les citations suivantes :

a — *Je pense, donc je suis* (Descartes, Discours de la méthode)
b — *Je me révolte, donc je suis* (Camus, L'Eté)
c — *Cet animal est triste, et la crainte le ronge.* (La Fontaine, Fables: Le lièvre et les grenouilles)
d — *Il n'y a pas d'amour de vivre sans désespoir de vivre* (Albert Camus, L'Envers et l'endroit)
e — *Qu'on cède à la peur du mal, on ressent déjà le mal de la peur* (Beaumarchais, Le barbier de Séville)
f — *Ventre affamé n'a point d'oreilles* (La Fontaine, ib. : Le milan et le rossignol)
g — *Je meurs si je vous perds, mais je meurs si j'attends* (Racine, Andromaque)

Très vite, on réalise cependant que la tâche est complexe : les représentations logiques ont du mal à rendre compte de certaines subtilités de compréhension traduites par des verbes de modalité (*croire que, penser que, savoir que..*), par de petits mots outils (*déjà*, *seulement*, *toujours*, *mais*...) voire par certaines tournures. Des logiques complexes telles que la logique modale ou encore la logique floue ont été inventées pour atteindre un niveau de finesse plus important. Bien des détails de la langue échappent cependant toujours à la représentation logique. Tentez tout de même de représenter les citations suivantes en logique des propositions, tout en relevant les insuffisances de cette représentation :

a — *Aller jusqu'au bout, ce n'est pas seulement résister, c'est aussi se laisser aller* (Camus, Carnets)
b — *Prouver que j'ai raison serait accorder que je puis avoir tort* (Beaumarchais, Le mariage de Figaro)
c — *Je le haïrais davantage* [Voltaire] *si je le méprisais moins* (Rousseau, Correspondance)
d — [*Va,*] *je ne te hais point* (Corneille, Le Cid)

Calcul des propositions

QUESTIONS DE COURS

Exercice 15 — Tables de vérité (objectif 2.2.2)

Donner la table de vérité des propositions suivantes :

a) $\neg P \wedge Q$ b) $\neg(P \vee Q)$
c) $\neg P \Rightarrow (Q \wedge P)$ d) $(\neg P \vee Q) \Leftrightarrow (P \Rightarrow \neg Q)$

Exercice 16 — Intuition logique (objectif 2.2.3)

Donnez un modèle de la fbf suivante : $(\neg A \vee B \vee \neg C) \Rightarrow C$

TABLES DE VERITE : EXERCICES CALCULATOIRES

Exercice 17 — Question de rapidité : table de vérité ou calcul mental (objectif 2.2.3 ;)

Donnez les interprétations des atomes P, Q et R (par exemple P = Vrai ; Q = Faux ; R = Faux) qui rendent fausses les formules suivantes :

1) $R \wedge \neg P \Rightarrow (Q \vee (R \Rightarrow P))$
2) $[\, Q \Rightarrow (R \Rightarrow P)\,] \vee \neg R \vee P$

Exercice 18 — Tautologies, contradictions (objectif 2.2.2.)

1 — Précisez, en utilisant la méthode des tables de vérité, si les formules suivantes sont des tautologies, des contradictions, ou des propositions simplement satisfaisables. On cherchera à limiter au maximum les calculs.

a) $(P \wedge Q) \wedge (\neg P \vee Q)$ b) $P \vee \neg(P \wedge Q)$
c) $\neg P \Rightarrow (P \wedge Q)$
d) $(P \Rightarrow Q) \Rightarrow (Q \Rightarrow P)$
e) $(P \Rightarrow Q) \wedge (Q \Rightarrow R) \Rightarrow (P \Rightarrow R)$

Réponse — tautologies : b, e ; formules satisfaisables : a, c, d

2 — Quelle remarque peut on faire au sujet du résultat d) ?

Exercice 19 — Tautologies (objectif 2.2.2) **d'après A. Aho & J. Ullman,** *Concepts fondamentaux de l'informatique*, **Dunod**

Précisez, en utilisant la méthode des tables de vérité, quelles sont les expressions, parmi les formules suivants, qui sont des tautologies. On cherchera à limiter au maximum les calculs.

1 — $(P \Rightarrow Q) \Rightarrow P$
2 — $P \wedge Q \wedge R \Rightarrow P \vee Q$
3 — $(P \Leftrightarrow (Q \vee R)) \Rightarrow (\neg Q \Rightarrow P \vee R)$

Réponse — tautologie : b

Exercice 20 — Tables de vérité et ou exclusif (objectif 2.2.2)

On considère la fbf suivante : $\neg(A \oplus B) \wedge (C \Leftrightarrow B) \wedge (\neg A \vee \neg B)$.

1) Donnez le nombre d'interprétations pour lesquelles la formule est vraie :
2) Donnez le nombre d'interprétations pour lesquelles la formule est fausse :
3) Que peut-on dire de la formule :

☐ Tautologique uniquement ☐ Tautologique et satisfaisable
☐ Satisfaisable (sans plus) ☐ Contradictoire

Exercice 21 — Tables de vérité et ou exclusif (10' ; objectif 2.2.2 ; 💣)

Précisez, en utilisant la méthode des tables de vérité, la nature (tautologique, satisfaisable, contradictoire) des expressions suivantes. On cherchera à limiter au maximum les calculs.

a) $(P \vee Q) \wedge ((R \wedge Q) \oplus (\neg P \Leftrightarrow \neg Q))$ b) $(P \wedge Q) \vee ((R \vee Q) \Leftrightarrow (\neg P \oplus \neg Q))$

Exercice 22 — Equivalence logique (objectif 2.2.2 ; 💣)

Retrouvez, à l'aide de la méthode des tables de vérité, les règles d'équivalence suivantes :

a) $P \vee (Q \wedge R) \equiv (P \vee Q) \wedge (P \vee R)$ (distributivité de la disjonction sur la conjonction)

b) $\neg(P \wedge Q) \equiv \neg P \vee \neg Q$ (1ère loi de De Morgan)

c) $P \equiv P \wedge (P \vee Q) \equiv P \vee (P \wedge Q)$ (règle d'absorption)

MODELES D'UN ENSEMBLE DE FORMULES : EXERCICES AVEC TRADUCTIONS

Exercice 23 — Coût de la vie (objectif 2.2.1 et 2.2.2)

Monsieur Truisme nous fait les deux affirmations suivantes :

(1) *Si cet objet est rare, alors il est cher*
(2) *De plus, si cet objet est cher, alors il n'est pas rare*

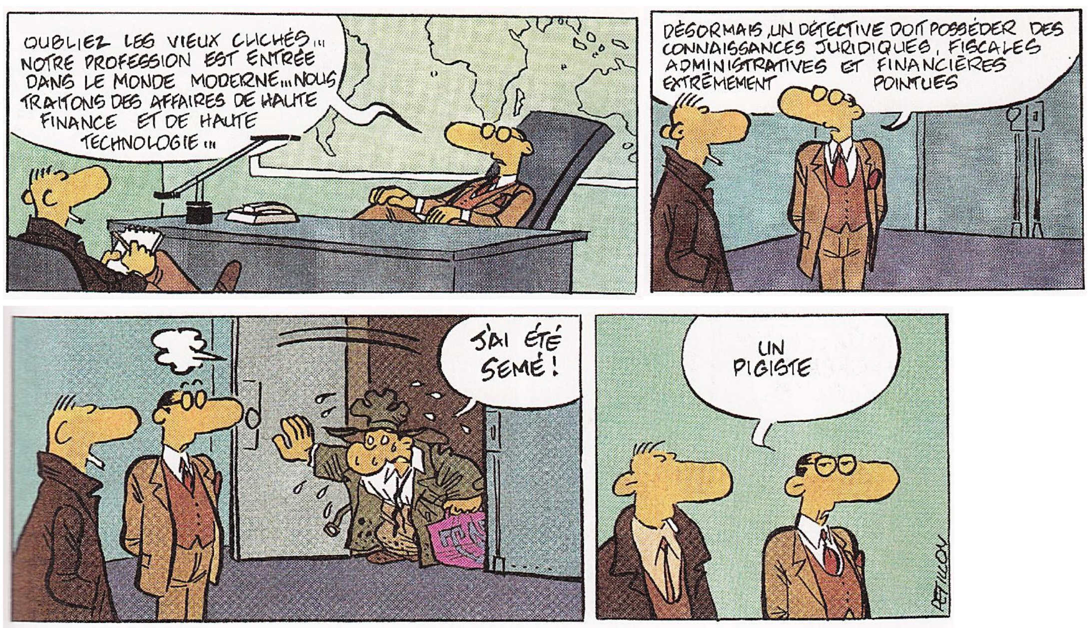
A première vue ces deux affirmations semblent contradictoires. Montrez, à l'aide de la méthode des tables de vérité, que cet ensemble de propositions est satisfaisable. C'est à dire que Monsieur Truisme n'a pas forcément toujours tort...

Exercice 24 — Détective malgré lui (objectifs 2.2.1. et 2.2.3)

Un crime horrible a été commis au département informatique : les PC de la salle d'informatique qui faisaient l'an dernier encore de l'Université un musée, ont été remplacés par des ordinateurs flambants neufs ! Le détective Jack Palmer, chargé de cette difficile enquête, est arrivé à isoler trois suspects. Voici le résumé de leur dépositions :

(TA) André : *Bernard est coupable. Claude n'a rien à voir là dedans..*
(TB) Bernard : *Si André a fait le coup alors Claude est innocent.*
(TC) Claude : *Je suis innocent, mais l'un des deux autres au moins est coupable.*

La logique va aider ce pauvre détective, qui se perd comme à l'accoutumée en conjonctures.



1. Donner les formules correspondant aux trois témoignages TA, TB et TC.
2. Cet ensemble de formules est-il satisfaisable ? Pour répondre à cette question, étudiez les différents cas de culpabilité à l'aide d'une table de vérité.

Réponse indicative — l'analyse de la table de vérité nous renseigne à coup sûr sur la culpabilité ou l'innocence de deux personnes, mais ne nous permet pas de décider pour la troisième.

Exercice 25 — Déduction à l'aide de la méthode des tables de vérités (objectifs 2.2.1. et 2.2.3)

Deux amis, qui ne se mentent jamais, sont en conversation :

Bertrand : *Alors, t'es tu vraiment décidé entre Juliette et Isabelle ?*

Paul : *Je vais te dire deux choses. D'abord, j'aime Juliette ou Isabelle, et ensuite, si j'aime Juliette alors c'est plus fort que moi, j'aime aussi Isabelle.*

Bertrand *: Peux-tu me jurer que si tu aimes Juliette alors tu aimes aussi Isabelle ?*

Paul *: Tout ce que je peux t'affirmer, c'est que si c'est vrai alors j'aime Juliette et que si c'est faux alors je ne l'aime pas*

De qui Paul est-il réellement amoureux ? Pour le savoir, on représentera chaque proposition par une formule logique, et on regardera par la méthode des tables de vérité quelle interprétation peut satisfaire l'ensemble des formules.

Réponse indicative — *Abondance de biens ne nuit pas* ou encore *quand il y en a pour une il y en a pour deux...*

CP : Mise sous forme normale

QUESTIONS DE COURS

Exercice 26 — Equivalence logique et formules de transformations (objectif 2.2.5)

Montrez les équivalences suivantes en appliquant les formules de transformation de la logique des propositions :

a) $\neg P \vee Q \quad \equiv \quad (P \wedge Q) \vee \neg P$
b) $P \wedge Q \quad \equiv \quad P \wedge (\neg P \vee Q)$

Exercice 27 — Vérification de la validité des formules par transformations (objectif 2.2.2)

En utilisant les règles de simplification par équivalence logique vues en cours, montrez que :

a) $(P \wedge (P \Rightarrow Q))$ et $P \wedge Q$ sont logiquement équivalentes.
b) $((P \Rightarrow Q) \Rightarrow P) \wedge \neg P$ est contradictoire.

Exercice 28 — Equivalence logique (objectifs 2.2.2 et 2.2.5)

Dans chacun des couples de formules suivantes, les deux formules sont-elles logiquement équivalentes?

(1) $A \wedge B \Rightarrow C \qquad (A \Rightarrow C) \vee (B \Rightarrow C)$
(2) $A \vee B \Rightarrow C \qquad (A \Rightarrow C) \vee (B \Rightarrow C)$
(3) $A \qquad (A \Rightarrow B) \Rightarrow A$

Réponse — équivalence pour les cas 1 et 3 ; vous pouvez trouver un contre-exemple pour le cas 2)

FORME NORMALE : EXERCICES CALCULATOIRES

Exercice 29 — Formes normales conjonctives et disjonctives (objectif 2.2.5 ; 💣)

1 — Mettre sous forme normale conjonctive, puis disjonctive la formule suivante : $Fa \equiv (P \wedge Q) \Leftrightarrow (\neg P \Rightarrow Q)$

2 — Montrez, en les mettant au choix sous forme conjonctive ou disjonctive, que la formule suivante est valide :
$((P \Rightarrow Q) \wedge (P \Rightarrow R)) \Rightarrow ((P \wedge Q) \Rightarrow (P \wedge R))$

Exercice 30 — Tautologies (objectif 2.2.5) **d'après A. Aho & J. Ullman,** *Concepts fondamentaux de l'informatique*, **Dunod**

En cherchant à mettre les formules suivants sous forme normale, préciser si ces expressions sont des tautologies, des contradictions ou simplement des formules satisfaisables.

1 — $P \wedge Q \wedge R \Rightarrow P \vee Q$
2 — $((P \Rightarrow Q) \wedge (Q \Rightarrow R)) \Rightarrow (P \Rightarrow R)$
3 — $(P \Rightarrow Q) \Rightarrow P$
4 — $(P \Leftrightarrow (Q \vee R)) \Rightarrow (\neg Q \Rightarrow P \vee R)$

Réponse — Tautologies : 1, 2.

Exercice 31 — Validité : choix de méthodes (objectifs 2.2.2 et 2.2.5)

Répondez, en utilisant la méthode de votre choix (tables de vérité ou formules d'équivalence), aux question suivantes :

1 — Que peut-on dire de la formule $(A \Rightarrow C) \Leftrightarrow (A \vee C \Leftrightarrow B)$

Réponse : ☐ tautologique ☑ satisfaisable ☐ contradictoire

2 — Les formules suivantes sont-elles logiquement équivalentes :

$A \Rightarrow (B \Leftrightarrow C \vee D)$ et $A \wedge B \Leftrightarrow D \wedge C \wedge A$

Réponse : ☐ équivalentes ☑ non équivalentes

FORME NORMALE : PROBLÈMES

Exercice 32 — Système complet de connecteurs (15' ; objectif 2.3.1)

$\downarrow$ est le connecteur logique représentant le NON_OU (NOR en anglais) : $A \downarrow B \equiv \neg(A \vee B)$. En partant du système complet de connecteurs étudié en cours de votre choix, trouvez et donnez les équivalences qui prouvent que $\{ \downarrow \}$ constitue également un système complet.

Exercice 33 — Système complet de connecteurs (15' ; objectif 2.3.1)

Le connecteur booléen ou exclusif est défini par $A \oplus B \equiv (A \wedge \neg B) \vee (\neg A \wedge B)$. En utilisant la méthode de votre choix, montrez que :

(1) cet opérateur est associatif et commutatif.

(2) $A \oplus \mathcal{F} \equiv A$ et $A \oplus A \equiv \mathcal{F}$

(3) Toute fonction booléenne f peut-être représentée dans le système de connecteurs $\{1, \wedge, \oplus \}$.

Exercice 34 — Complétude et connecteurs logiques (objectif 2.3.1)

En logique, le terme "complétude" peut revêtir diverses acceptations. Il peut ainsi caractériser la capacité d'un système formel à démontrer toutes les conséquences logiques du langage sur lequel il est défini (théorème d'incomplétude de Gödel, par exemple). Il peut également servir à caractériser la difficulté algorithmique des problème de décidabilité. Ainsi, savoir si une expression logique est une tautologie constitue un problème **NP-complet**, c'est à dire faisant partie des problèmes que l'on ne sait résoudre qu'avec un temps exponentiel à la dimension du problème. Dans cet exercice, nous nous intéresserons à la troisième acceptation de ce terme, qui concerne les ensembles de connecteurs logiques.

Un ensemble de connecteurs est en effet dit complet si et seulement si toute expression logique peut s'exprimer uniquement à l'aide de ces opérateurs. Ainsi, nous avons vu en cours que l'ensemble $\{ \wedge, \vee, \neg, \Rightarrow, \Leftrightarrow \}$ constituait un système complet de connecteurs pour la logique des propositions. Pour montrer qu'un ensemble de connecteurs logiques est complet, il faut et il suffit de montrer qu'on peut exprimer les connecteurs d'un autre système complet dans ce nouvel ensemble. Partant de $\{ \wedge, \vee, \neg, \Rightarrow, \Leftrightarrow \}$, on montre ainsi que $\{ \wedge, \vee, \neg \}$ constitue un système complet en vertu des règles d'équivalences suivantes :

$$P \Rightarrow Q \quad \equiv \quad \neg P \vee Q$$

$$P \Leftrightarrow Q \quad \equiv \quad (\neg P \vee Q) \wedge (P \vee \neg Q)$$

1 — On considère ici l'opérateur NAND (NON-ET) noté $\uparrow$ défini comme la négation de la conjonction :

$$P \uparrow Q \quad \equiv \quad \neg(P \wedge Q)$$

Montrez que l'ensemble $\{ \uparrow \}$ réduit à ce seul opérateur constitue un système complet. Cet opérateur est de première importance en informatique et en électronique, puisqu'il permet à lui seul de représenter l'ensemble des fonctions logiques nécessaire à la mise en œuvre des circuits des ordinateurs.

Indication — Montrer qu'un système de connecteur est un système complet revient à montrer que tout connecteur quelconque peut se réécrire sous la forme d'une combinaison de connecteurs du système ... et non l'inverse !

2 — On considère maintenant l'opérateur NOR (NON-OU) noté $\downarrow$ et défini comme la négation de la disjonction. Montrez que l'ensemble $\{ \downarrow \}$ constitue lui aussi un système complet de connecteurs.

Déduction en LP

QUESTIONS DE COURS

Exercice 35 — Tables de vérité (objectif 2.2.7)

Montrez, par la méthode des tables de vérité, que :

1 — $P \Leftrightarrow Q$ implique logiquement $P \Rightarrow Q$
2 — $P \Leftrightarrow \neg Q$ n'implique pas logiquement $P \Rightarrow Q$

Exercice 36 — Tables de vérité et résolution (objectif 2.2.7)

Vérifiez, en utilisant les tables de vérité puis la méthode de résolution, la validité des raisonnements suivants :

a) $P \Rightarrow Q \quad {}^{?}\!\models \quad Q \Rightarrow P$
b) $(A \vee C) \Rightarrow B \quad {}^{?}\!\models \quad A \vee B$
c) $A \Rightarrow \neg B,\ \neg C \Rightarrow A \quad {}^{?}\!\models \quad B \Rightarrow C$

Exercice 37 — Un peu de recul sur le cours… (objectif 2.1.9)

1) Que peut-on dire de l'énoncé ci-dessous ? (objectifs 2.2.7, 2.1.4 , 2.1.9 et indirectement 4.1.2)

« En logique des propositions, on peut toujours démontrer si une fbf est contradictoire ou pas »

☐ vrai ☐ faux ☐ cela dépend de la fbf

2) On considère une formule mise sous forme clausale sur laquelle on applique la méthode de résolution. Que peut-on dire sur la formule si on arrive :

a) à en tirer une résolvante tautologique ($\mathcal{V}$)
b) à la clause vide $\varnothing$.

☐ la formule est contradictoire ☐ la formule est valide
☐ la formule est satisafaisable ☐ on ne peut rien dire

Exercice 38 — Fausses idées sur la résolution (objectif 2.1.9)

1 — Lorsqu'on applique la méthode de résolution, que peut-on dire si on arrive à tirer d'un ensemble de clauses la clause toujours vraie ?

2 — On cherche à montrer qu'un raisonnement est valide à l'aide de la méthode de résolution. Après avoir mis la formule de réfutation sous forme normale, on applique la résolution sur toutes les clauses obtenues et on arrive à obtenir la clause vide. Néanmoins, on remarque qu'une des clauses initiales n'a pas été utilisée pour arriver à ce résultat. Que peut-on en conclure : la raisonnement est-il valide ou non ?

Exercice 39 — Résolution (objectif 2.1.9)

On veut démontrer la validité du raisonnement :$A \Leftrightarrow \neg B,\ C \Rightarrow B \vee A,\ C \Rightarrow B \wedge A \ {}^{?}\!\models B \wedge C$

1. Donnez les clauses correspondant à la mise sous forme clausale de la formule de réfutation
2. Le raisonnement est-il valide? Si non, donnez toutes les résolvantes obtenues.

DEDUCTION : EXERCICES CALCULATOIRES

Exercice 40 — Quelques déductions plus complexes (objectifs 2.2.7 et 2.2.8)

Vérifiez, *en utilisant la méthode de votre choix*, la validité des raisonnements suivants :

1) $A \Rightarrow B,\ A \Rightarrow C,\ \neg(B \vee C) \quad \overset{?}{\models} \quad D$

2) $\neg P \Rightarrow \neg Q \vee R \quad \overset{?}{\models} \quad (\neg P \Rightarrow Q) \Rightarrow (P \vee R)$

Exercice 41 — Raisonnements avec un grand nombre de propositions atomiques (objectif 2.2.8)

1) En employant la méthode de résolution, vérifiez la validité des raisonnements suivants :

a) $A \Rightarrow \neg B \vee C,\ B \Rightarrow A \wedge \neg C \quad \overset{?}{\models} \quad C \Rightarrow B$

b) $\neg A \Rightarrow B,\ C \Rightarrow \neg A \quad \overset{?}{\models} \quad \neg A \Rightarrow (\neg C \Rightarrow B)$

c) $P \Rightarrow R \wedge T,\ T \vee S \Rightarrow \neg Q \quad \overset{?}{\models} \quad \neg(P \wedge Q)$

d) $(S \Rightarrow R) \wedge P,\ Q \quad \overset{?}{\models} \quad \neg R \wedge \neg S \wedge P$

2) Le résultat concernant le raisonnement d) était-il prévisible ?

Réponse — raisonnements valides : b, c.

Exercice 42 — Une déduction avec « ou exclusif » (10' ; objectif 2.2.8)

En utilisant la méthode de résolution, discuter de la validité du raisonnement suivant.

$A \oplus B,\ C \Rightarrow B \vee A \quad \overset{?}{\models} B \wedge C$

DEDUCTION : PROBLEMES AVEC TRADUCTION

Exercice 43 — Suicide alimentaire (objectifs 2.2.1 et 2.2.8)

Voici Janvier qui arrive, et bonjour les examens de fin de semestre ! Comme à l'accoutumée, Pierre a passé ses vacances de Noël à surfer sur les pistes enneigées. Il avait bien emmené quelques polys histoire de se donner bonne conscience, mais l'appel de l'or blanc a une fois de plus été le plus fort. Et une fois de plus, il va falloir trouver une solution pour s'en sortir, du genre "*le tout pour le tout*". Pierre fait alors le raisonnement suivant :

(a) *Un élève malade ne peut passer son examen.*

(b) *Si je mange au R.U. ce midi, je risque une intoxication alimentaire et serai très certainement malade.*

(c) *Je vais aller manger au R.U. ce midi.*

(d) *Donc, je ne passerai pas cet examen, c'est une certitude absolue !*

Impressionné par son courage, mais tout de même inquiet à l'idée de se restaurer dans ce temple de la haute gastronomie française, Pierre s'interroge sur la validité de son raisonnement : ce serait tout de même stupide de risquer sa vie pour rien... Malheureusement, il a précisément fait une impasse totale sur son cours de logique. Sauriez-vous conseiller notre infortuné skieur en utilisant la méthode des tables de vérité et la méthode de résolution ?

Réponse — le raisonnement est valide.

Exercice 44 — Comment préparer ses examens (20' ; objectifs 2.2.1 et 2.2.8)

Jean-Antoine se prépare à réviser ses examens de septembre. Seul dans sa chambre, il réfléchit sur la stratégie à adopter et se tient le raisonnement suivant :

(H1) *S'il fait beau demain, pour travailler il faut que je me lève tôt.*
(H2) *Il va faire beau demain.*
(H3) *C'est décidé ; je vais travailler ou bien me lever tôt.*

(C) Donc, c'est sûr, si je me lève tôt, je vais travailler.

1 — Traduisez les trois hypothèses et la conclusion sous forme de formules logiques.
2 — Le raisonnement de Jean-Antoine est-il valide? Justifiez la réponse par trois méthodes différentes.

***Réponse** — le raisonnement est non valide.*

Exercice 45 — Jack Palmer sur l'île de Beauté (objectifs 2.2.1 et 2.2.8)

Cette semaine, notre ami Jack Palmer est sur une affaire de vol de saucisson de cochon sauvage près de Rossignoli, typique village Corse. Après avoir auditionné l'ensemble des témoins, le détective fait le raisonnement suivant :

Si Figoli n'a pas rencontré Paoli dans le maquis la nuit du vol, c'est que Paoli est le voleur ou que Figoli est un menteur. Si Paoli n'est pas le voleur, alors Figoli ne l'a pas rencontré cette nuit là et le forfait a été commis après minuit. Si le crime a eu lieu après minuit, alors Paoli est le voleur ou Figoli a menti. Donc c'est sûr: Paoli est le voleur.

Convaincu de la culpabilité de Paoli, il s'apprête à accuser ce dernier. Jack Palmer aurait-il enfin réussi à résoudre une affaire ? Utilisez la méthode de résolution pour vous faire une opinion sur la validité de son raisonnement. Donner un contre-exemple en cas de raisonnement non valide. [**Réponse** — le raisonnement est non valide].

Exercice 46 — Adieu veaux, vaches, cochons, couvées (objectifs 2.2.1 et 2.2.8)

La PAC (Politique Agricole Commune de l'Union Européenne) est souvent au centre des discussions concernant la construction européenne. Sujet souvent embarrassant pour des gouvernements soucieux à la fois de ménager un électorat de première importance et de réduire le coût budgétaire induit par le soutien artificiel des prix agricoles. Fraîchement nommé et déjà en proie à de terribles migraines à la veille du Salon de l'agriculture, Marcel Pichedru, jeune Ministre en charge du sujet, fait le raisonnement suivant :

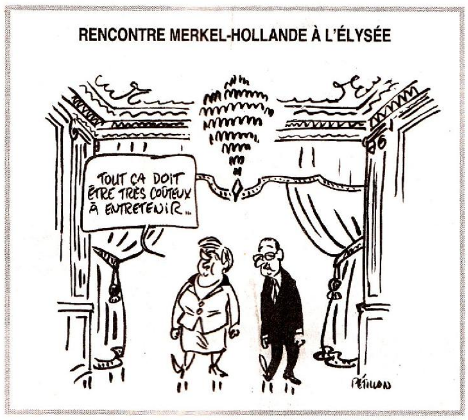
1) *Nous perdrons les voix des agriculteurs si nous ne poursuivons pas notre politique de soutiens des prix.*
2) *A moins d'entreprendre des réformes structurelles, il y aura suproduction si nous conservons cette politique,*
3) *Nous avons besoin des voix des agriculteurs pour être réelus.*
4) *A l'avenir, il faut éviter à tout prix toute surproduction.*

C) *Donc si nous sommes réélus, il nous faudra mettre en place des réformes structurelles.*

Ce raisonnement est-il valide ? Utilisez le principe de résolution pour répondre à cette question.

Réponse — le raisonnement est valide.

Exercice 47 — L'hôte encombrant (objectifs 2.2.1 et 2.2.8)



Le chef du protocole de la République Française doit organiser prochainement un bal dans les superbes salons de l'Élysée en l'honneur de la chancelière allemande. Fraîchement nommé suite à l'élection récente du nouveau président de la République, il est très ennuyé, car il ne connaît pas les goûts musicaux de cette dernière, et encore moins ses qualités de danseuses. Il craint de ternir les relations entre les deux pays (déjà passablement ébréchées par le débat sur la limitation des dépenses publiques) en proposant un programme qui lui déplaise. Un rien facétieux, son homologue allemand lui fournit des indications sous la forme de la devinette suivante :

1) *Si elle déteste le jazz, alors elle aime le tango ou la valse.*
2) *Elle a les mêmes préférences pour le tango et la rumba.*
3) *Si elle déteste la valse alors elle aime le tango.*
4) *Elle déteste le jazz ou la valse, ou les deux d'ailleurs.*
5) *Si elle aime la valse et pas le jazz, alors elle aime la rumba.*

Après longue réflexion, il décide d'inviter Stacey Kent à donner un récital jazzy, car, selon lui

C) *Elle déteste la rumba, la valse et le tango.*

Donnez une représentation logique de ce raisonnement et cherchez si sa conclusion est une conséquence valide des affirmations qui lui ont été énoncées.

PARADOXES LOGIQUES

Exercice 48 — Le paradoxe de la carte de Jourdain (objectifs 2.2.1 et 2.2.8)

Les ouvrages de logique amusante fourmillent de paradoxe propres à donner la migraine au lecteur ordinaire, alors qu'une bonne connaissance de la logique permet de surmonter ce type de contradictions sans problème. nous allons étudier ici un paradoxe proposé pour la première fois par le mathématicien anglais P.E.B Jourdain en 1913.

Dans un premier temps, intéressons nous à un paradoxe équivalent à celui de Jourdain, et que l'on doit au mathématicien contemporain R. Smullyan (tiré de : *Quel est le titre de ce livre*, Dunod). On se trouve sur une île peuplée de deux types d'individus : les purs, qui ne mentent jamais, et les pires, qui mentent toujours. Vous rencontrez soudain deux autochtones, Annie Ouininon (que l'on notera A) et Robin Chsepa (noté B). Ceux-ci vous déclarent respectivement :

(A) *Robin est un pire*
(B) *Annie est une pure*

1 — Cette rencontre a-t-elle vraiment pu avoir lieu, ou est-elle paradoxale ? Pour répondre à cette interrogation, cherchez si ces deux énoncés peuvent être satisfaisables simultanément en appliquant successivement les méthodes des tables de vérité et de résolution.

Revenons maintenant au paradoxe de la carte de Jourdain. Sur un côté d'une carte est écrit :

(TA) *La phrase écrite de l'autre côté de cette carte est vraie.*

Quand on la retourne, on découvre une autre affirmation :

(TB) *La phrase écrite de l'autre côté de cette carte est fausse.*

2 — Montrez que ce paradoxe est équivalent au précédent en en donnant une représentation logique.

Exercice 49 — Panique aux JO (objectifs 2.2.1 et 2.2.8) Adapté de E. Busser & G. Cohen, *La Recherche*, 302(1997)

En patinage artistique, les décisions des jurys classant les candidats relèvent d'une subjectivité qu'il est souvent de bon ton de dénigrer, a fortiori lorsque les sportifs d'un pays n'obtiennent pas les résultats escomptés par leurs compatriotes... Lassés de ces critiques perpétuelles, un trio de juges facétieux a le don d'exaspérer les commentateurs et les sportifs par des affirmations qui sont tantôt vraies, tantôt fausses. Laetitia, Patricia et Surya voudraient bien savoir si elles sont qualifiées dans le dernier groupe de la finale des Jeux Olympique d'Hiver. Ayant plus révisé leurs triples axels que leur logique des propositions, elles font appel à vos compétences pour démêler cet écheveaux. Sauriez vous les aider en utilisant la méthode de votre choix ?

1 — Laetitia recueille les affirmations suivantes auprès du jury :

Nikolaï : *L'un de nous au moins va vous mentir.*
Omar : *Vous êtes qualifiée, toute mes félicitations !*
Peter : *Nikolaï a dit la vérité.*

Après mure réflexion, Laetitia saute de joie en s'écriant "Je suis qualifiée !". Une cruelle désillusion l'attend-elle ?

Réponse — oui, le raisonnement de Laetitia n'est pas valide.

2 — C'est au tour de Patricia de se présenter maintenant face aux juges.

Nikolaï : *Désolé, vous n'êtes pas qualifiée.*
Omar : *Ne l'écoutez pas, Nikolaï ne dit jamais la vérité.*
Peter : *Je ne sais pas si vous devez croire Omar ! En effet, deux d'entre nous, au moins, vous mentent !*

Croyant Nikolaï sur parole, Patricia repart la tête basse en marmonnant "*Ma triple boucle piquée m'a perdue*". Un petit calcul logique saura-t-elle la réconforter ?

Réponse — oui, le raisonnement de Patricia est valide.

3 — Enfin, Surya affronte ce trio diabolique.

Nikolaï : *Je n'ai qu'une chose à dire : Peter vous dira la vérité..*
Omar : *Si vous êtes qualifiée, c'est que Nikolaï vous a menti !*
Peter : *Omar dit la vérité. En tous cas, je vous félicité, vous êtes qualifiée.*

Se perdant en conjoncture, Surya reste prostrée face au jury. Sauriez-vous trouver la conclusion que l'on peut tirer de ces trois affirmations ?

PROBLÈMES

Exercice 50 — Lewis Caroll et les paradoxes de l'implication (objectifs 2.1.3, 2.2.1, 2.2.7 et 2.2.8)

Si la définition des opérateurs logiques tels que la conjonction, la disjonction ou l'équivalence ne pose aucun problème à "l'honnête homme", il n'en va pas de même pour l'implication. L'auteur d'Alice au Pays des Merveilles, qui se piquait de logique, s'est penché sur ce problème dans un traité de logique qu'il avait rédigé (*Symbolic Logic*, 1896). De même, dans ses *Principes des Mathématiques* (1980, ed. Blanchard), Louis Couturat note ce qu'il appelle "les paradoxes de l'implication matérielle" :

*Toutes les propositions vraies sont équivalentes. Toutes les propositions fausses sont équivalentes. Chaque proposition fausse implique toutes les propositions (vraies ou fausses); chaque proposition vraie est impliquée par toutes les propositions (fausses ou vraies). Ces paradoxes inévitables (car ce sont des conséquences nécessaires au calcul, et cela dans n'importe quel système de Logique) s'expliquent par le fait que l'implication ici considérée est l'**implication matérielle**, et non pas l'**implication formelle** [...] à laquelle tout le monde pense quand on parle d'implication [dans la vie courante]. L'implication matérielle ($P \Rightarrow Q$) ne signifie rien de plus que ceci : "Ou P est fausse, ou Q est vraie". Peu importe que les propositions P et Q aient entre elles un rapport logique ou empirique quelconque: l'implication est vérifiée dès que P est fausse (quelle que soit Q) ou dès que Q est vraie (quelle que soit P). Voilà pourquoi on arrive à ce résultat paradoxal, que le faux implique le vrai.*

Ces vérités paradoxales servent d'ailleurs à résoudre correctement certains paralogismes ou certains paradoxes où le bon sens vulgaire risquerait de s'embarasser. Tel est par exemple le problème de Lewis Caroll : [supposons que] *"Q implique R ; mais* [aussi que] *P implique que Q implique non-R; Que faut-il en déduire ?* [...] *Lewis Caroll* [adoptant le point de vue du sens commun] *raisonne ainsi : si Q implique R, il est impossible que Q implique non-R; donc P implique l'impossible, et par suite est faux.*

1 — En utilisant la méthode de votre choix (résolution ou tables de vérités), montrez que la conclusion, et donc le raisonnement, de Lewis Caroll sont erronés.

2 — Quel est le "maillon" de la démonstration de Caroll qui pose problème. Pourquoi ?

3 — Finalement, quelle conclusion peut-on tirer des deux hypothèses initiales ? Montrer que cette conclusion est bien une conséquence logique de ces prémisses.

Logique des prédicats du premier ordre (LP1)

QUESTIONS DE COURS

Exercice 51 — Un peu de syntaxe (objectif 3.1.3)

Dans la logique des prédicats du premier ordre, on considère la formule bien formée suivante : $\forall x\ (\forall y\ (\ (\exists z\ \ R(f\ (x),\ g(y,z))) \Rightarrow (S(x,y) \vee T(z))))$

Reconnaître, dans cette formule les symboles fonctionnels et les symboles de prédicat. Préciser leur arité.

Exercice 52 — Un peu de syntaxe, suite et fin (objectifs 3.1.2 et 3.1.4)

On se place dans la logique des prédicats du premier ordre et on considère :

R	prédicat d'arité 1	S et T	prédicats d'arité 2
f et g	fonctions d'arité 1	h	fonction d'arité 2

On rappelle que "=" est un raccourci d'écriture pour le prédicat d'arité 2 Egal. On considère les formules suivantes :

a) $\exists x\ (\ (\ \forall y\ (\ \exists z\ (\ R(x)\ )\ )\ ) \vee (\ \exists y\ (\ \neg(\ \forall z\ (\ S\ (\ h\ (x,z),\ x))))))$
b) $(\ \forall x\ (\ T\ (\ f(x),\ y\ ))) \Rightarrow (\ \neg(\ \exists x\ (\ f(x,y)\ )\ )\ )$
c) $(\ \forall z\ (\ T\ (\ x,\ y\ ))) \Rightarrow (\ \ \exists y\ ((\ \forall x\ (\ \neg(\ f(x) = y\ ))) \vee T(y,z)\ ))$

1 — Quelles sont, parmi ces formules, celles qui sont bien formées ?

2 — Simplifiez-les formules bien formées en enlevant les parenthèses inutiles au regard des règles de priorités.

3 — Déterminez les occurrences liées et les occurrences libres dans les formules bien formées.

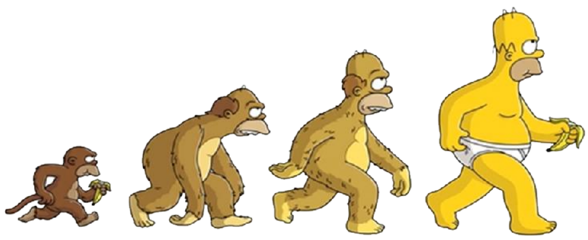
4 — Standardiser l'écriture de ces formules si nécessaire (mise sous forme propre).

Exercice 53 — Un traduction primaire (objectif 3.1.1 et 3.2.1)

Traduire en logique des prédicats du 1er ordre l'ensemble des assertions de ce raisonnement.

Tout homme est un singe supérieur
Tout singe supérieur est un primate
Les dauphins ne sont pas des primates
Il y a des dauphins qui sont intelligents

Donc on peut ne pas être un homme et être intelligent



Exercice 54 — Savez-vous choisir vos connecteurs ? (objectifs 3.1.2 et 3.2.1)

Indiquez quels est (sont) la (ou les) traduction(s) correcte(s) des énoncés suivants.

1. *Tous les enfants ne sont pas des anges*

- ☐ $\forall x\ (\ enfant(x) \Rightarrow \neg anges(x)\ )$
- ☐ $\exists x\ (\ enfant(x) \wedge \neg anges(x)\ )$
- ☐ $\exists x\ (\ enfant(x) \Rightarrow \neg anges(x)\ )$
- ☐ $\forall x\ (\ anges(x) \Rightarrow \neg enfant(x)\ )$
- ☐ $\neg\ \forall x\ (\ enfant(x) \Rightarrow anges(x)\ )$
- ☐ $\neg\ \forall x\ (\ anges(x) \Leftrightarrow enfant(x)\ )$
- ☐ $enfant \Rightarrow \neg anges$
- ☐ $\neg(\ enfant \Leftrightarrow\ \ anges)$

2. *Ce n'est pas parce qu'on n'est pas une femme qu'on ne doit pas participer au ménage dans le couple.*

- ☐ $\forall x\ (\ menage(x) \Rightarrow \neg femme(x)\ )$
- ☐ $\neg\ \forall x\ (\ menage(x) \Leftrightarrow \neg femme(x)\ )$
- ☐ $\exists x\ (\ \neg\ femme(x)\ \wedge menage(x)\ )$
- ☐ $\exists x\ (\ \neg\ femme(x)\ \wedge\ \neg menage(x)\ )$
- ☐ $\neg\ \forall x\ (\ menage(x) \Rightarrow \neg femme(x)\ )$
- ☐ $\neg\ \forall x\ (\ \neg femme(x) \Rightarrow menage(x)\ )$
- ☐ $\neg\ \exists x\ (\ \neg femme(x) \wedge menage(x)\ )$

3. *Les carrés sont des parallépipèdes rectangles*

- ☐ $\forall x\ (\ carré(x) \wedge parallépidède(x) \wedge rectangle(x)\ )$
- ☐ $\exists x\ (\ carré(x) \wedge parallépidède(x) \wedge rectangle(x)\ )$
- ☐ $\forall x\ (\ carré(x) \Rightarrow parallépidède(x) \wedge rectangle(x)\ )$
- ☐ $\exists x\ (\ carré(x) \Rightarrow\ parallépidède(x) \wedge rectangle(x)\ )$
- ☐ $\forall x\ (\ (\neg parallépidède(x) \wedge\ \neg rectangle(x)\ ) \Rightarrow\ \neg carré(x)\ )$
- ☐ $\forall x\ (\ (\neg parallépidède(x) \vee\ \neg rectangle(x)\ ) \Rightarrow\ \neg carré(x)\ )$
- ☐ $\forall x\ (\ carré(x) \Leftrightarrow parallépidède(x) \wedge rectangle(x)\ )$

4. *Chaque entreprise a un PDG*

- ☐ $\forall x\ \exists y\ (\ entreprise(x) \wedge personne(y) \wedge PDG(x,y)\ )$
- ☐ $\exists x \forall y\ (\ entreprise(x) \wedge personne(y) \wedge PDG(x,y)\ )$
- ☐ $\forall x\ \forall y\ (\ entreprise(x) \wedge personne(y) \wedge PDG(x,y)\ )$
- ☐ $\exists x \exists\ y\ (\ entreprise(x) \wedge personne(y) \wedge PDG(x,y)\ )$
- ☐ $\forall x\ (\ entreprise(x) \Rightarrow \exists\ y\ \ (personne(y) \wedge PDG(x,y)\ ))$
- ☐ $\exists x\ (\ entreprise(x)\ \ \Rightarrow \forall y\ (personne(y) \wedge PDG(x,y)\ )\ )$
- ☐ $\forall x(\ entreprise(x) \Rightarrow \forall y\ (personne(y) \wedge PDG(x,y)\ )\ )$
- ☐ $\exists x\ (\ entreprise(x) \Rightarrow \exists\ y\ \ (personne(y) \wedge PDG(x,y)\ )\ )$

TRADUCTIONS EN LP1 : EXERCICES

Exercice 55 — Mourir de faim ou de la vache folle ? (10' ; objectif 3.2.1)

Vache folle, tremblante du mouton, fièvre aphteuse, le consommateur français perd d'année en année sa légendaire attirance pour les plaisirs de la table. Totalement désorienté par les annonces alarmistes de médias plus avides de sensationnel que de réflexion, certains parents s'empressent ainsi de retirer leurs chérubins d'une cantine scolaire proposant de la nourriture a priori douteuse, afin de pouvoir les gaver de gallettes de tofu bio directement importées des hauts plateaux du Népal. Une seule réponse à ces comportements passionnels : un recours à la raison, donc à la logique. Mais pour cela, assurons nous au préalable que nous disposons du bagage nécessaire en logique des prédicats du 1er ordre pour mener à bien ce programme salutaire. Commençons donc par quelques travaux pratiques.

On demande de donner la traduction dans la logique des prédicats du 1er ordre des énoncés ci-dessous. On utilisera pour cela les prédicats suivants : Creutzfeld/1, Manger/2, Boeuf/1, Farines/1, Francais/1.

1) *Aucun boeuf n'est nourrit qu'avec des farines animales.*

2) *Si un boeuf a consommé des farines animales c'est qu'il n'est pas français.*

3) *Personne ne peut contracter la maladie de Creutzfeld-Jacob sans avoir consommé de boeuf.*

Exercice 56 — Jeu de construction logique (objectif 3.2.1)

Monsieur Patate, Madame Patate et Junior Patate sont bien embêtés : après une activité que la morale réprouve certainement, ils ont mélangé tous les éléments de leur anatomie, et ont quelque mal à bien remettre tout en place. La logique va tenter de les aider. Pour cela, il vous est demandé de représenter les énoncés ci-dessous en utilisant les prédicats suivants de la logique des prédicats du 1er ordre :

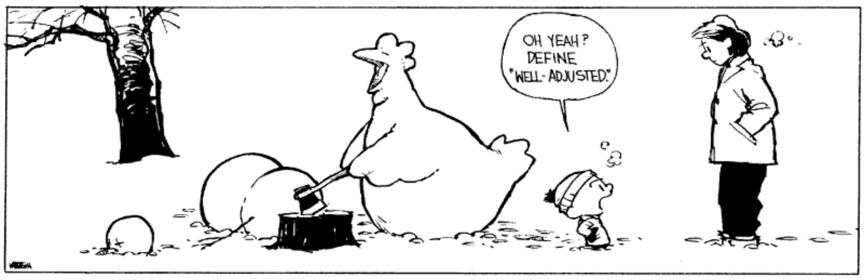
- `Lunette(X)` est vrai si X est une paire de lunette.
- `Oreille(X)` est vrai si X est une paire d'oreilles.
- `Chapeau(X)` est vrai si X est un chapeau.
- `Moustache(X)` est vrai si X est un chapeau
- `Sur(X,Y)` est vrai si X est sur Y (ou peut aller sur Y)
- `Patate(X)` est vrai si X est un membre de la famille patate
- Homme(X) est vrai si X est un homme (enfin, une patate du genre masculin !)

Donnez les traductions des énoncés dans les cadres prévus à cet effet. Ah, au fait, Mr Patate s'appelle Robert… :

1. *Tous les membres de la famille Patate ont un chapeau sur eux.*
2. *Toutes les paires d'oreille peuvent aller sur tous les membres de la famille Patate*
3. *Il n'existe pas de paire de lunettes qui peut aller sur tous les membres de la famille Patate*
4. *Si un élément est sur Robert, il ne sera pas sur une femme de la famille*
5. *Seuls les hommes de la famille Patate ont une moustache*

Exercice 57 — **La télé rend fou** (objectif 3.2.1)

A chaque révélation d'un crime perpétré par un adolescent, la télévision est accusée d'amplifier les tendances psychopathes des jeunes du par la diffusion de séries policières trop sanglantes. Ce qui est clair, c'est que les téléspectateurs y trouvent ce qu'ils cherchent comme va nous le montrer un petit raisonnement.



Donnez la traduction, en logique des prédicats du 1er ordre, du raisonnement ci-dessous. On utilisera pour cela les prédicats `personne/1, tele/1, regarde/2, psycho/1, criminel/1, nevrose/1, montre/2.`

Tout téléspectateur qui regarde la télévision devient psychotique à la longue
Certains psychotiques ne regardent jamais télévision
Si un criminel n'est pas psychotique, alors il est névrosé
TF1 ne montre à longueur d'antenne de des criminels

Donc les gens que l'on voit à la télé sont … des téléspectateurs

Exercice 58 — Le panda est un loup pour le bambou (15' environ ; objectif 3.2.1)

Symbole de la Chine éternelle, le panda est un animal dont le régime alimentaire est un des plus spécialisés, ce qui n’est pas fait pour freiner sa dramatique extinction. Il se nourrit en effet quasiment exclusivement de feuilles du bambou. Autant dire que pour le bambou, le panda n’apparaît pas comme un bon gros nounours tout mignon, mais un terrible criminel contre l’humanité, sanguinaire et sans pitié. Mais qui se soucie des angoisses du pauvre bambou...

Traduire en LP1 les énoncés ci-après en utilisant les 5 prédicats suivants :

`Mange(x,y)` est vrai ssi x mange y.
`Herbivore(x)` est vrai ssi x est un animal herbivore.
`Vegetal(x)` est vrai ssi x est un végétal.
`Bambou(x)` est vrai ssi x est un bambou.
`Panda(x)` est vrai ssi x est un panda.

1. *Les herbivores ne mangent que des végétaux.*
2. *Aucun herbivore ne mange tout type de végétal.*
3. *Il y a des végétaux que ne mange aucun herbivore.*
4. *Les pandas sont des herbivores qui ne consomment que des bambous*

Exercice 59 — Casse-tête au Département Informatique : emplois du temps (10’; objectif 3.2.1;)

Chaque année à la rentrée, la consommation d’aspirine augmente dangereusement dans les rangs des directeurs de filières du département informatique. Tous sont en effet taraudés par la même question : comment arriver à caser un emploi du temps sachant que le nombre de salles disponibles est inférieur au nombre de groupes d’étudiants… Les difficultés vont d’année en année croissants. En effet, bien des enseignants ne se contentent plus d’une salle avec craie et tableau noir : bien souvent, il faut également tenir compte des besoins en tableau blanc interactif (TBI) ou en vidéoprojecteurs. Puisque chaque année vos serviteurs s’arrachent les cheveux à vous concocter un emploi du temps aux petits oignons, à vous maintenant d’attraper la migraine sur les traductions suivantes…

Traduire les énoncés suivants dans la logique des prédicats du 1er ordre. On utilisera pour cela les prédicats `tbi/1` , `vidéoproj/1`, `panne/1`, `amphi/1`, `salle_td/1`, `est_dans/2`.

a) *On trouve toujours un TBI dans une salle de TD.*
b) *Il n’y a pas de vidéoprojecteur en salle « B301 ».*
c) *Tous les vidéoprojecteurs sont dans des amphis.*
d) *S’il y a un vidéoprojecteur quelque part, c’est qu’il s’agit d’un amphi.*
e) *Il n’y a que des TBI dans les salles de TD.*

Exercice 60 — Un peu de rangement (objectif 3.2.1)

Deux heures du matin. Sous perfusion de caféine depuis le début de l'après-midi, Bertrand Jémoitou vient enfin de finir la révision de son cours de logique. Après 10 heures de combats homériques contre la multitude de polycopiés que le prof distribue chaque semaine, sa chambre est un vrai capharnaüm. Dire qu'il va falloir tout ranger avant d'aller se coucher ! Procédant avec méthode, Bertrand Jémoitou énonce :

(1) *Tous les tiroirs contiennent des feuilles.*
(2) *Aucun des classeurs ne contient de polycopiés.*
(3) *Dans l'un des classeurs, il n'y a que des polycopiés.*
(4) *Si les polys ne sont pas dans les classeurs, elles sont dans des tiroirs.*
(5) *S'il y a des feuilles dans un tiroir, on est certain de ne pas y trouver de polys.*

Traduire les énoncés suivants en langage des prédicats du premier ordre. On utilisera 4 prédicats unaires destinés à l'identification des objets décrits dans ces énoncés (par exemple, `Feuille(x)` a pour signification : x est une feuille), et le prédicat `Dans(x,y)` vrai si et seulement si x est dans y.

Exercice 61 — Harry Potter et la formule logique (objectif 3.2.1)

En utilisant les prédicats `Sorcier/1`, `Parent/2`, `Malefique/1`, `Ensorcele/1`, `Ancetre/2`, `Grifondor/1` donnez la traduction en LP1 des énoncés suivants :

1 — *Les parents des sorciers ne sont pas tous des sorciers*

2 — *Tout sorcier a au moins un Moldu parmi ses ancêtres*

3 — *Les sorciers ne sont pas tous maléfiques*

4 — *Un ancien de Grifondor est soit non maléfique, soit ensorcelé.*

5 — *Il n'y a aucun ancien étudiant de Grifondor qui soit devenu maléfique*

6 — *Il faut nécessairement être sorcier pour pouvoir des pouvoirs maléfiques.*

Exercice 62 — La réussite au bout du chagrin (objectif 3.2.1)

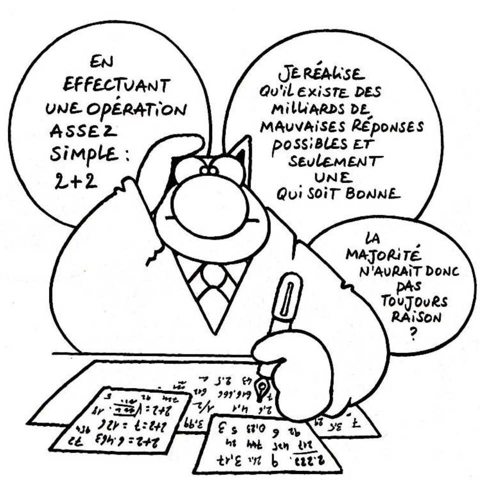
« *Si l'école m'apprend pas ça alors je dis halte à tout* », chantait Renaud, qui avait apparemment du mal à envoyer sa fille chaque matin « *au chagrin* ». Il est vrai que le système éducatif apparaît souvent (à tort ou à raison ?) à ses usagers comme une course d'obstacle dont la finalité est avant tout d'obtenir des diplômes. En témoigne malheureusement l'inspiration de l'auteur de cet exercice...

Utiliser le prédicat binaire R(x,y) dont l'interprétation est x réussit y ainsi que des variables et des constantes laissées à votre choix pour traduire en LP1 les énoncés suivants :

1. *Jean ne réussira pas son licence sans que Yves le réussisse*.
2. *Nul ne peut obtenir sa licence s'il a échoué à son bac*.
3. *Il y a qui réussissent leur bac tout en échouant à leur permis de conduire*.
4. *Si on a un master ou un doctorat, c'est qu'on a déjà obtenu son licence*.

APPLICATION AUX MATHEMATIQUES

Exercice 63 — Encore des mathématiques... (objectif 3.3.1)



Pour traduire leurs propositions dans leurs démonstrations, les mathématiciens utilisent un langage formel qui ressemble à de la logique des LP1 mais n'en est pas vraiment. Afin de ne pas faire de confusion entre syntaxe de la LP1 et syntaxe mathématique, nous allons traduire en logique des prédicats un certain nombre de propositions mathématiques.

1. Traduire dans la logique des prédicats les affirmations suivantes en les symboles mathématiques =, + et *, le prédicat unaire *E(n)* pour exprimer que la variable *n* est un entier ainsi que le prédicat binaire ≥.

1 — *Tout entier est le carré d'un entier.*
2 — *Tout entier a pour carré la somme des carrés de deux entiers.*
3 — *Certains entiers ont pour carré la somme des carrés de deux autres entiers.*
4 — *Aucun entier n'est plus grand que tous les autres.*

2. Traduisez ensuite la négation des ces formules, puis simplifiez les fbf obtenues pour enfin en donner une traduction la plus simple possible en français.

TRADUCTIONS EN LP1 : POUR ALLER PLUS LOIN

Exercice 64 — Instruction civique... (objectif 3.2.1)

Donnez **une** représentation des énoncés suivant en logique des prédicats du premier ordre :

a — *Nul n'est sensé ignorer la loi*

b — *Les hommes naissent et demeurent libres et égaux en droits*
(Déclaration des Droits de l'Homme et du Citoyen du 26 Août 1789, article premier).

c — *Tout homme étant présumé innocent jusqu'à ce qu'il ait été déclaré coupable*
(Déclaration des Droits de l'Homme et du Citoyen du 26 Août 1789, article IX).

d — *Nul ne doit être inquiété pour ses opinions,* [même religieuses] *pourvu que leur manifestation ne trouble pas l'ordre public établi par la loi*
(Déclaration des Droits de l'Homme et du Citoyen du 26 Août 1789, article X).

Calcul des prédicats du premier ordre

QUESTIONS DE COURS

Exercice 65 — Tout était si simple en LP (objectif 3.1.7 et 2.1.4)

Expliquez pourquoi la méthode des tables de vérité n'est pas applicable en logique des prédicats du 1er ordre

Exercice 66 — Transitivité et antisymétrie de la relation inférieure ou égale sur N (objectif 3.2.2)

On considère les deux fbf closes suivantes :

$\phi 1 = \forall x \forall y\ (\exists z\ (\ P(x,z) \wedge P(z,y)) \Rightarrow P(x,y)\ )$
$\phi 2 = \forall x \forall y\ (\ P(x,y) \wedge P(y,x) \Leftrightarrow (x = y)\ )$

Déterminez la valeur de validité de ces formules pour l'interprétation

$I = \{D = \mathbb{n}$; Ip : P(x) vrai ssi $x \leq y$; = : égalité }

Exercice 67 — Interprétation en LP1 (objectif 3.2.2)

On considère les deux formules suivantes de la logique des prédicats du premier ordre :

(F1) $\exists x\ \forall y\ \exists z\ ((P(x) \Rightarrow S(x,y)) \wedge P(y) \wedge \neg S(y,z)\ )$
(F2) $\exists x\ \ \exists z\ ((S(z,x) \Rightarrow S(x,z)) \Rightarrow\ \forall y\ S(x,y))$

On donne les deux interprétations A et B suivantes :

(Ia) le domaine est N ; S est la relation d'ordre usuelle et P(n) si et seulement si n est pair.
(Ib) le domaine est R ; S(x,y) si et seulement si $y=x^2$ et P(x) si et seulement si x est un nombre rationnel.

De quelles formules F1 ou F2 les interprétations A et B sont-elles des modèles ? (**Réponse** — F2).

Exercice 68 — Question de quantificateur (objectif 3.1.3 et 3.1.7)

Lorsque vous êtes amenés à faire des traductions en LP1, le choix des bonnes combinaisons entre quantificateurs et connecteurs logiques vous pose souvent problème. Considérons par exemple les deux énoncés suivants :

(1) $\forall x\ (\ P(x) \Rightarrow Q(x)\ )$ (2) $\forall x\ (\ P(x) \wedge Q(x)\ )$

Afin de saisir la distinction entre ces propositions, trouver un modèle de (1) qui rend faux le second.

INTERPRETATIONS : EXERCICES

Exercice 69 — Validité et domaine d'interprétation (objectif 3.2.2)

On se place dans la logique des prédicats du premier ordre. On considère les prédicats d'arité 2 *Egal* et *Diff* dont les raccourcis d'écriture sont respectivement = et ≠, c'est à dire que x = y est vrai ssi $I_v(x)$ et $I_v(y)$ sont égaux et $x \neq y$ est vrai ssi $I_v(x)$ et $I_v(y)$ sont différents. Soient les formules suivantes :

a — $\forall x \forall y \forall z \forall t\ (\ x \neq y \wedge y \neq z \wedge z \neq x \wedge (\ t = x \vee t = y \vee t = z))$

b — $\exists t \forall x \forall y \forall z\ (\ x \neq y \wedge y \neq z \wedge z \neq x \wedge (\ t = x \vee t = y \vee t = z))$

c — $\exists x\ \exists y\ \exists z \exists t\ (\ x \neq y \wedge y \neq z \wedge z \neq x \wedge (\ t = x \vee t = y \vee t = z))$

d — $\exists x\ \exists y\ \exists z \forall t\ (\ x \neq y \wedge y \neq z \wedge z \neq x \wedge (\ t = x \vee t = y \vee t = z))$

Déterminez la validité de ces formules (validité, satisfaisabilité, inconsistance) dans les cinq interprétations suivantes, variant par leur domaine d'interprétation respectifs :

D1 = { 0 } D2 = { 0, 1} D3 = {0, 1, 2} D3 = {0, 1, 2, 3} D4 = N

Exercice 70 — Encore des mathématiques (objectif 3.2.2)

R et *S* sont deux prédicats d'arité 2 du langage des prédicats du 1er ordre. On considère les deux formules suivantes :

(F1) $\exists x \, \forall y \, (R(x,y) \Rightarrow S(x,y))$
(F2) $\forall x \, \exists y \, (R(x,y) \Rightarrow S(x,y))$

Le domaine d'interprétation est l'ensemble des entiers naturels supérieurs ou égaux à 2. L'interprétation de R est la relation d'inégalité $\leq$, celle de S est la relation divise au sens de la division euclidienne définie sur les entiers.

1. Donner, dans cette interprétation, les valeurs de vérité respectives des formules F1 et F2.

Réponse — I(F1) = $\mathscr{F}$; I(F2) = $\mathscr{V}$.

2. En gardant les mêmes interprétations pour R et S, modifier le domaine d'interprétation pour que les deux formules soient vraies.

RECHERCHE DE MODELES : EXERCICES

Exercice 71 — On est toujours le modèle de quelqu'un… (10' ; objectifs 3.1.2 et 3.2.2)

On considère les deux formules bien formées de la LP1 suivantes :

(1) $\forall x \, \exists y \, (P(x) \wedge Q(x,y)$
(2) $\exists x \, \forall y \, (P(x) \wedge Q(x,y)$

a) Donnez un modèle de la formule (1) pour lequel la formule (2) est fausse.

b) Inversement, donnez un modèle de la formule (2) pour lequel la formule (1) est fausse.

Exercice 72 — Gare au quanti-fi-eur ! (10' ; objectifs 3.1.2 et 3.2.2)

On considère l'ensemble de formules de la logique des prédicats du 1er ordre suivant :

1) $\forall x \; [P(x) \Rightarrow \exists y \, Q(x,y)]$
2) $\exists x \; [P(x) \Rightarrow \forall y \, Q(x,y)]$

Donnez un modèle de la formule (2) pour lequel la formule (1) est fausse.

Exercice 73 — Modèle en LP1 (objectif 3.2.2)

On se place dans la logique des prédicats du premier ordre. On considère, outre le symbole de la relation binaire égalité, deux symboles de fonctions unaires f et g pouvant recevoir diverses interprétations. On définit les formules suivantes :

F1 : $\forall x \, (f(x) = g(x))$
F2 : $\forall x \, \forall y \, (f(x) = g(y))$
F3 : $\forall x \, \exists y \, (f(x) = g(y))$

Donner un modèle pour chacune des formules : $F1 \wedge \rceil F2$, F2, $\rceil F1 \wedge F3$.

Exercice 74 — Modèle d'un système logique (objectif 3.2.2)

Soit $\mathscr{A}$ les trois fbf suivantes :

(A1) $\forall x \forall y \forall z \; ((P(x,y) \wedge P(y,z) \Rightarrow P(x,z))$
(A2) $\forall x \; (P(A,x) \wedge P(x,B))$
(A3) $\forall x \; P(x,f(x))$

Donnez un modèle de $\mathscr{A}$.

LP1 — Mise sous forme clausale

Exercice 75 — Forme prenexe (objectif 3.1.5)

Mettre sous forme prenexe les formules suivantes

a — $(\neg\exists x\, P(x) \vee \forall x\, Q(x)) \wedge (R \Rightarrow \forall x\, S(x))$

b — $\neg ((\neg\exists x\, P(x) \vee \forall x\, Q(x)) \wedge (R \Rightarrow \forall x\, S(x)))$

c — $\forall x\, P(x) \Leftrightarrow \exists x\, Q(x)$

Exercice 76 — Formules closes et forme clausale (objectif 3.2.3)

Mettre sous forme clausale les fbf suivantes :

a — $\forall x\, \exists y\, \forall z\, [R(x,y,z) \Rightarrow \forall t\, \exists z\, S(t,z)]$

b — $\forall y\, \exists x\, R(x,y) \Leftrightarrow \forall z\, \forall x\, R(z,x)$

c — $\forall x\, \forall y\, (GP(x,y) \Rightarrow \exists z\, (P(x,z) \wedge P(y,z)))$

d — $\forall x\, [P(x) \wedge \forall y\, \exists t\, (\neg Q(t,y) \Rightarrow \forall z\, R(A,t,y))]$

e — $\forall x\, (\exists y\, [R(y,x) \Rightarrow \exists u\, R(u,x) \wedge \neg (\exists t\, (R(t,x) \wedge R(t,y)))])$

Unification

QUESTIONS DE COURS

Exercice 77 — Unification : cas d'école (objectif 3.1.6)

Trouvez, quand il existe, un unificateur de chacun des ensembles de clauses ci-dessous. Donnez en outre la fbf résultante de l'unification opérée sur ces clauses :

a —	G(x, f (A, y))	G(x, B)	G(x, f(A, g(z)))
b —	P(u, g(f(a, b)), u)	P(f(x, g(z)), x, f(y, g(b)))	
c —	G(x,y)	G(f(x), A)	

UNIFICATION : EXERCICES

Exercice 78 — Quelques unifications trop tranquilles... (objectif 3.1.6)

Pour chaque cas, dire si les deux formules atomiques sont unifiables et en donner le cas échéant un unificateur :

b —	A(x, g(x,y))	A(g(y, z), g (g(h(u), y) , h(u)))
b —	B(x, f(g(y)), f(x))	B(h(t,z),f(z),f(h(y,z)))
c —	P(x, f(x), g(f(x),x))	P(z, f(f(A)), g(f(g(A,z)), v))
d —	P(u, g(f(A,b)), u)	P(f(x, g(z)), x, f(y, g(B)))
e —	P(x, f(x), f(f(x)))	P(f(f(y)), y, f(y))

Réponse — formules unifiables : b et d.

Résolution LP1

QUESTIONS DE COURS

Exercice 79 — Résolution : cas d'école (objectif 3.1.6)

Préciser, en appliquant la méthode de résolution, si les ensembles de clauses suivantes sont contradictoires ou non.

1)	$S(z)$	$S(A) \vee S(t)$	$\neg S(A) \vee \neg S(y)$
2)	$\neg Q(A)$	$P(x,f(x))$	$\neg P(z,t) \vee Q(z)$
3)	$A(A)$	$N(B)$	$\neg A(x) \vee \neg N(x)$
4)	$\neg P(x) \vee Q(f(x))$	$\neg Q(y) \vee P(f(y))$	

Exercice 80 — Un petite résolution complète en LP1 (objectif 3.2.4)

Montrez, à l'aide du principe de résolution, la validité du raisonnement suivant (transitivité de l'implication) :

$$\forall x\,(P(x) \Rightarrow Q(x))\,,\ \forall x\,(\,Q(x) \Rightarrow R(x)\,) \quad \models \quad \forall x\,(\,P(x) \Rightarrow R(x)\,)$$

Exercice 81 — Unification et résolution. (objectifs 3.1.6 & 3.2.4)

Lorsque cela est possible, donnez la résolvante correspondant à l'application de l'unification sur les clauses ci-dessous. On précisera à chaque fois l'unificateur utilisé.

1 — $A(x,Jean) \vee B(Jean,y)$ et $A(z,Paul) \vee \neg B(Pierre,t)$
2 — $A(y,Jean) \vee B(Jean,y)$ et $A(Paul,z) \vee \neg B(t,Pierre)$
3 — $\neg A(Paul,Jean) \vee B(Jean,y)$ et $A(Paul,Jean) \vee \neg B(Jean,t)$
4 — $A(x,Jean) \vee B(Jean,f(x))$ et $A(x,y) \vee \neg B(Jean,g(t))$
5 — $A(x,Jean) \vee B(Jean,y)$ et $A(x,Jean) \vee \neg B(Jean,f(y))$

RESOLUTION : EXERCICES

Exercice 82 — Quelques résolutions un peu plus complexes (objectif 3.2.4)

Préciser, en appliquant la méthode de résolution, si les ensembles de clauses suivantes sont contradictoires ou non.

1)	$\neg P(x) \vee Q(s(x))$	$\neg Q(x) \vee P(s(x))$	$P(A)$	$\neg P(s(s(s(s(A)))))$	
2)	$P(A,y) \vee P(y,A)$	$\neg Q(z) \vee S(z)$	$\neg R(u) \vee Q(u)$	$\neg S(x)$	
3)	$\neg H(x) \vee P(x)$	$\neg D(y) \vee \neg P(y)$	$H(z) \vee \neg I(z)$	$D(F)$	$I(F)$

Exercice 83 — Toooh encore un exercice sur la résolution (objectifs 3.2.1 et 3.2.4)

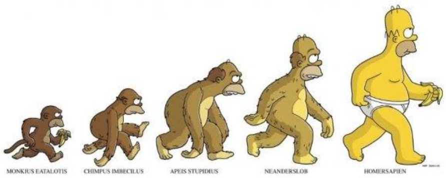
Démontrez la validité des raisonnements suivants à l'aide du principe de résolution. Nous avons déjà réalisé la traduction en LP1 du premier raisonnement.

a — *Tout homme est un singe supérieur*
Tout singe supérieur est un primate
Les dauphins ne sont pas des primates
Il y a des dauphines qui sont intelligents

Donc on peut ne pas être un homme et être intelligent

b — *Lisa Simpson est intelligente*
Bart Simpson a une intelligence très très limitée
Homer Simpson est moins intelligent qu'un donuts.
Lisa Simpson et Bart Simpson sont les enfants de Homer Simpson.

Donc l'intelligence n'est pas héréditaire.



HOMERSAPIEN

Exercice 84 — Résolution et validité de formule (objectif 3.2.4)

1. Démontrer la validité ou la contradiction de la formule suivante à l'aide de la méthode de résolution :

$$\exists x\ \forall y \quad (\neg(\, S(x) \Leftrightarrow S(y)\,) \wedge \forall x\ S(x))$$

2. Si la formule n'est pas valide, donnez un exemple d'interprétation qui la rend fausse.

Exercice 85 — Quelques raisonnements (objectif 3.2.4)

Montrer en utilisant le principe de résolution que les raisonnements suivants sont valides.

a —
$$\frac{\begin{array}{ll} \forall x\ \exists y & P(x,y) \\ \forall z \forall t & (\,P(z,t) \Rightarrow Q(z)\,) \end{array}}{\forall u \quad Q(u)}$$

b —
$$\frac{\begin{array}{ll} \forall x & (\,P(x) \Rightarrow P(\,f\,(f(x))\,)\,) \\ \forall x & (\,P(x) \Rightarrow R(\,f(x)\,)\,) \end{array}}{\forall x \quad (\,R(x) \Rightarrow P(\,f(x)\,)\,)}$$

c —
$$\frac{\begin{array}{ll} \forall x & (\,P(x) \Rightarrow Q(s(x))\,) \\ \forall x & (\,Q(x) \Rightarrow P(s(x))\,) \\ & P(A) \end{array}}{P(s(s(s(s(A)))))}$$

Exercice 86 — Un exemple d'indécidabilité en logique des prédicats (objectifs 3.2.4 et 4.1.1)

Nous avons vu en cours que la logique des prédicats du premier ordre était indécidable. Nous allons précisément étudier dans cet exercice un exemple flagrant d'indécidabilité.

Appliquez le principe de résolution sur le raisonnement suivant :

$$\forall x\ (\,P(x) \Rightarrow Q(f(x))\,)\ ,\ \forall x\ (\,Q(x) \Rightarrow P(f(x))\,)\ ,\ P(A) \quad \models \qquad \forall x\ P(x)$$

Peut-on conclure que le raisonnement est valide ? Si non, essayez de montrer que la conclusion n'est pas une conséquence logique des hypothèses en exhibant un contre-exemple.

Exercice 87 — Art et logique (objectifs 3.2.1 et 3.2.4)

On considère les deux raisonnements suivants :

a — *Certains peintres ont peint leur propre portrait.*
Il existe des peintres qui ont réalisé des natures mortes

Il existe des autoportraits de peintres de natures mortes.

b — *Cézanne a fait son autoportrait.*
Cézanne a peint de nombreuses natures mortes.

Il existe des autoportraits de peintres de natures mortes.

1 — Ces raisonnements vous paraissent-ils valides ?

2 — Donnez une représentation de ces raisonnements en logique des prédicats du premier ordre. On utilisera pour cela les prédicats `Peintre/1`, `Peinture_de/2` et `represente/2`.

3 — En utilisant la méthode de résolution, justifiez vos réponses à la première question.

Exercice 88 — De l'art de la contradiction (objectifs 3.2.2 et 3.2.4)

On considère l'ensemble des formules suivantes :

(1) $\forall x\ \forall y\ \forall z\ \ [\ (F(x,y) \wedge F(y,z)) \Rightarrow G(x,y)\ ]$

(2) $\forall x\ \exists y\ \ \ F(y,x)$

(3) $\neg[\ \forall x\ \exists y\ \ G(y,x)\ ]$

1 — En utilisant le principe de résolution, montrez que cet ensemble de clauses est contradictoire.

2 — La résolution montre que les 3 formules sonte toutes nécessaires à l'inconsistance de l'ensemble. Montrer que l'ensemble des clauses réduites à (1,2) est satisfaisable en en donnant un modèle. Faire de même avec (1,3).

Exercice 89 — Logique et religion (objectifs 3.2.1 et 3.2.4)

Suite à la découverte de l'Amérique et de ses habitants, une des questions qui agita l'église catholique fut de savoir si les amérindiens avaient une âme. Cette question, qui fut tranchée favorablement (controverse de Valladolid), n'était pas sans arrière pensée : elle justifiait ou non le recours à l'esclavage. La logique aurait-elle pu être mise à contribution dans ce débat religieux ? Considérons le raisonnement suivant :

Les amérindiens sont tous des êtres humains
Les êtres humains ont une âme

Il y a des amérindiens qui ont une âme

a — Donnez une représentation logique de ce raisonnement.

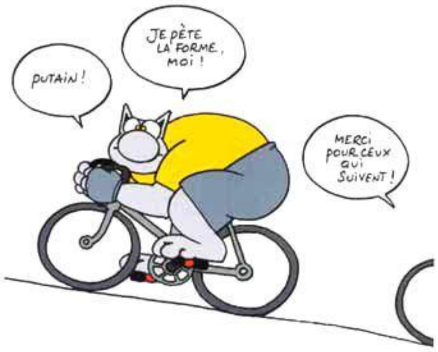
b — Donnez la liste des clauses correspondant à la formule de réfutation associée au raisonnement.

c— Ce raisonnement est-il valide ?

d— Ce résultat était-il prévisible ? Justifiez votre réponse dans le cadre ci-dessous

Exercice 90 — Les chargés de la route (objectifs 3.2.1 et 3.2.4)

La traduction sous forme logique des raisonnements humains est une science pleine de risque. Tout d'abord, nous avons souvent eu l'occasion de constater que la mise sous forme logique d'un énoncé en langue naturelle n'était le plus souvent ni triviale, ni directe. Mais en outre, le raisonnement humain recèle de considérations implicites qui, si elles ne sont pas explicitées, rendent le raisonnement logique totalement incohérent. L'exemple suivant, tiré des phénomènes de dopage dans le monde cycliste, va éclairer ce dernier point.



Bien que ses coéquipiers aient dans leur grande majorité avoué être aussi chargés que — dixit un enquêteur de la police judiciaire — "un bifteck de veau aux hormones", l'ancien cycliste Lance Armstrong a longtemps affirmé n'avoir eu recours à aucun produit prohibé

Le cycliste se révèle peut-être ici sur un terrain sur lequel on ne l'attendait pas, celui de la logique. S'il en connaît bien les principes, celui-ci peut en effet affirmer sans crainte que la conclusion du raisonnement suivant, à première vue correct, n'est pas une conséquence logique des hypothèses:

(1) Si tu te dopes, tu peux soit gagner, soit être pris.
(2) Si tu es un super coureur, tu peux gagner que tu sois dopé ou non.
(3) Lance Armstrong, qui n'est pas un super coureur, gagne des épreuves

---

(4) Lance Armstrong se dope

1 — Traduire ce raisonnement en utilisant la logique des prédicats du 1er ordre.

2 — En appliquant la méthode de résolution, montrez que ce raisonnement n'est pas valide.

3 — Quelle affirmation, non formulée dans le raisonnement ci-dessus, nous permettrait de conclure ? Quelle hypothèse complète-t-il implicitement ? Traduisez sous forme logique cette hypothèse implicite et montrez que le raisonnement est effectivement valide.

Exercice 91 — Les aliens sont parmi nous (objectifs 3.2.1 et 3.2.4)

Une étrange rumeur court au sein de la famille Patate et de ses semblables : des extra-légumes se seraient glissés au sein de la communauté. Afin de les démasquer, Monsieur Patate échafaude quelques raisonnements subtils. Pourriez-vous représenter ces derniers dans la logique des prédicats du 1er ordre, puis vérifier à l'aide de la méthode de résolution que Monsieur Patate ne raisonne pas comme une courge, c'est-à-dire que ses raisonnements sont valides… On utilisera pour cela les prédicats suivants :

- AL(X) est vrai si X a un sabre laser.
- P (X) est vrai si X est une (vraie) patate (i.e. pas un extra-légume)
- AM(X) est vrai si X a un masque.

Ainsi que la fonction v(X) qui désigne le voisin de tout individu X. Mr Patate s'appelle toujours Robert, et on supposera que tout non patate est un extra-légume (et inversement). **Attention** : les réponses sont à remettre sur la copie jointe

Raisonnement 2 *Les extra-légumes ont un masque*

Robert a un sable laser

Le voisin de Robert n'a pas un sabre laser

Il existe donc extra-légumes

Raisonnement 2 *Les extra-légumes ont un masque*

Robert n'a pas de masque

Donc Robert n'est pas un extra-légume

Raisonnement 3 *Les extra-légumes ont un masque*

Les patates n'ont pas de masque

Toute personne qui a un masque n'est pas une patate

RESOLUTION : PROBLEMES

Exercice 92 — Malvita (objectifs 3.2.1 et 3.2.4 ; 40')

Réfugié à Chollong-sur-Arve, petite bourgade de Normandie, Giovanni Manzoni, chef mafieux désormais repenti et recherché par toutes les familles de la mafia New-Yorkaise pour les avoir trahi, réfléchit sur sa vie, son passé. N'étant jamais allé à l'université, il peine à analyser de manière logique les idées qui lui viennent à la tête. Pourriez-vous l'aider en donnant la traduction, en logique des prédicats du 1er ordre, des énoncés ci-dessous ?

On utilisera pour cela les prédicats Voleur/1, Assassin/1, Criminel/1, et Famille/2 et on se placera dans le domaine d'interprétation de l'ensemble des êtres humains

1 — *Certaines personnes sont des assassins*
2 — *Tout assassin a déjà volé*
3 — *Les voleurs ne sont pas tous des assassins*
4 — *Un criminel ne peut être qu'un assassin ou un voleur (ou les deux)*
5 — *Il y a un criminel dans ma famille* (c'est-à-dire la famille de Giovanni Manzoni)
6 — *Tous les membres de ma famille* (c'est-à-dire la famille de Giovanni Manzoni) *sont des voleurs*

Giovanni en arrive alors à la conclusion suivante : *Il n'y a pas de voleurs dans ma famille*. Nous allons utiliser la méthode de résolution pour montrer validité (ou non) de son raisonnement.

7 — Donnez l'ensemble des clauses correspondant à la mise sous forme clausale de la formule de réfutation correspondant à ce raisonnement..

8 — Donnez l'arbre de résolution LP1 qui vous permet de montrer que le raisonnement est ou n'est pas valide

9 — Parmi les clauses que vous avez données en question 7, donnez celles qui ne sont pas des clauses de Horn.

Exercice 93 — Question(s) d'interprétation (objectifs 3.2.2, 3.2.3. et 3.2.4 ; 15')

Dans le cours sur la mise sous forme prénexe, nous avons montré un certain nombre de formules d'équivalence qui nous permettaient de passer en début de formules les quantificateurs présents en leur sein, tout en mettant en garde sur le fait que l'on ne pouvait pas toujours faire n'importe quoi. Ainsi, il n'y a pas équivalence entre la formule $\exists x\ A \lor \exists y\ B$ et la formule $\exists x\ (A \land B)$ où A et B sont des formules quelconques de la logique des prédicats du 1er ordre. Nous allons montrer cette absence d'équivalence de deux manières différentes.

1 — Construisez un exemple avec des formules A et B de votre choix, puis donnez en une interprétation pour montrer qu'il n'y a pas équivalence.

2 — En reprenant votre exemple pour A et B, utilisez la méthode de résolution pour montrer que les deux formules $\exists x\ A \lor \exists y\ B$ et $\exists x\ (A \land B)$ ne sont pas équivalentes.

Exercice 94 — Résolution (objectif 3.2.3 et 3.2.4)

On considère les trois hypothèses suivantes :

(H1) $\forall x\ \forall z\ (\exists y\ \ R(x,y,z) \Leftrightarrow S(z,x))$

(H2) $\forall x \forall y\ (\ S(x,y) \Rightarrow S(f(x),f(y)))$

(H3) $\forall x\ \forall y\ (\rceil R(x,y,x))$

On considère maintenant les trois conclusions suivantes :

(C1) $\exists x \neg S(f(x),f(x))$
(C2) $\forall x \exists y \neg S(f(x),y)$
(C3) $\exists y \forall x \neg S(f(x),y)$

Déterminer quelles sont les conséquences logiques de l'ensemble des trois hypothèses. Pour chacune des conclusions qui n'est pas conséquence logique des hypothèses, trouver un modèle des hypothèses où la conclusion est fausse

Exercice 95 — Quelques règles d'équivalence : exercice un peu plus théorique (objectif 3.1.2 et 3.2.2)

Démontrez la validité des formules suivantes en partant de la définition des interprétations des quantificateurs existentiels et universels.

a — $\models \forall x\, A(x) \Rightarrow A(x)$ *axiome de spécialisation*

b — $\models \forall x\, [\, A(x) \wedge B(x)] \Leftrightarrow \forall x\, A(x) \wedge \forall x\, B(x)$ *distributivité de $\forall$ sur $\wedge$*

c — $\models \exists x\, [\, A \wedge B(x)] \Leftrightarrow A \wedge \exists x\, B(x)$ *distributivité de $\exists$ sur $\wedge$ et A formule close pour x*

d — $\models \exists x\, \forall y\, R(x,y) \Rightarrow \forall y\, \exists x\, R(x,y)$ *commutativité partielle de $\exists$ et $\forall$*

e — $\models \exists y\, [\, R(x,y) \Rightarrow R(x,x)\,]$

Clauses de Horn et Langage PROLOG

QUESTIONS DE COURS

Exercice 96 — Le retour d'Aristote (objectifs 4.1.3, 4.1.4, 4.2.3 et 4.2.4)

Dans cet exercice, nous allons étudier le syllogisme aristotélicien suivant :

Les caniches sont des chiens
Les chiens sont des canidés

Donc il existe des caniches qui sont des canidés

On veut cette fois-ci vérifier expérimentalement la validité de ce raisonnement en utilisant un interpréteur Prolog.

1 — A quoi correspondront les hypothèses du raisonnement en Prolog ?

2 — A quoi correspondra la conclusion et comment vérifierons nous la validité du raisonnement ?

3 — Mettre sous forme de clauses Prolog, puis donnez les clauses de Horn correspondantes.

4 — Vérifiez alors la validité de ce raisonnement en appliquant la méthode de résolution.

5 — Comparez cette résolution avec la résolution Prolog équivalente.

Exercice 97 — Cap à l'Ouest (objectif 4.1.4)

On considère le programme ci-dessous, constitué de 4 clauses.

```
(1)     finistere(Brest);
(2)     morbihan(Vannes) , morbihan(Lorient);
(3)     bretagne(X) :- finistere(X);
(4)     bretagne(X) :- morbihan(X);
(5)     france(bretagne(X)).
```

1 — Ce programme présente plusieurs fautes de syntaxe. Pouvez vous les détecter et les corriger ?

2 — Quels sont les clauses qui correspondent à des faits dans ce programme ? à des règles ?

3 — Que représentent `brest`, `vannes` et `lorient` ? Sont-ils des variables ou des atomes (constantes) Prolog **?**

Exercice 98 — Bugs à tous les étages (5' ;objectif 4.1.4)

Ali Dentic a placé sur le serveur de l'université un squelette de programme correspondant à son TP sur les cursus d'informatique réalisables au sein de l'université. Or, ce programme ne respecte pas toujours la syntaxe du langage Prolog… Pouvez-vous l'aider à détecter ses erreurs de syntaxe (et non pas de sémantique !) ? Il est demandé de donner à chaque fois la ligne incriminée, l'erreur correspondante et comment vous la corrigeriez.

```
/**********************************************************/
/* FICHIER : di.pl                                        */
/* AUTEUR :  Dentic Ali                                   */
/**********************************************************/

/*-- suite(Avt,Apr) vrai si on peut suivre le cursus Apr juste apres Avt -----*/

suite(bac,l1_info1), suite(l1_info,l2_info), suite(l2_info,l3_info).
suite     (bac,iut_info1).
suite(iut_info1,dut_info).

/*-- cursus(Deb,Fin) vrai s'il existe un cursus entre Deb et Fin ------------*/

cursus(Deb,Fin) :- suite(Deb,Fin)
cursus(Deb,Fin) :- suite(Deb,Entre),
suite(Entre,Fin).
cursus(Deb,Fin) :- cursus(Deb,Entre,Fin).
```


EXERCICES : PROLOG ET LP1

Exercice 99 — Instruction civique européenne (10' ; objectifs 4.2.2 et 3.2.4)

Alors que des pays comme l'Irlande, le Royaume-Uni et, plus récemment, les pays scandinaves ont accordé le droit de vote aux élections locales à leurs résidents étrangers sans qu'aucune catastrophe ne s'ensuive, la classe politique française rechigne toujours a aborder cette question. Néanmoins, suite aux accords de Maastricht, tout ressortissant de l'union européenne dispose d'un tel droit de vote en France. On considère un petit programme Prolog illustrant cette situation :

```
francais( dupond ).
europeen( gert ).
habite( gert, France).
vote(X) :- francais(X).
vote(X) :- europeen(X), habite(X,france).
```

a) Donnez la traduction en logique des prédicats du 1[er] ordre de la question Prolog : `?- vote(X).`

b) Donnez l'ensemble des clauses de Horn correspondant à la traduction en logique des prédicats du 1[er] ordre du programme et de la question.

c) Donnez **toutes** les réponses obtenues par application de la méthode de résolution sur cet ensemble de clauses de Horn. Pour chaque solution, on donnera le nombre de démonstrations obtenues.

Exercice 100 — Do you speak clauses de Horn ? (objectifs 4.1.3 et 4.2.3)

Les clauses de Horn utilisées par Prolog imposent des contraintes non négligeables sur la traduction logique des raisonnements. En règle générale, ces contraintes restent cependant facilement satisfaisables. En particulier, l'opération de *renommage* que nous allons étudier dans cet exercice permettant une reformulation adéquates des clauses posant problème.

On considère ici l'ensemble des clauses suivantes, où P, Q, R et S désignent des formules atomiques :

(1) $P \vee Q \vee \neg R$

(2) $P \vee \neg S$

(3) $Q \vee T$

1 — Dire, parmi ces trois clauses, lesquelles ne sont pas des clauses de Horn.

2 — On appelle *renommage de P* l'opération, notée $[P \rightarrow \neg P1]$, qui consiste à remplacer dans toute clause P par $\neg P1$. Donner un ensemble de clauses de Horn équivalent à l'ensemble précédent en utilisant le renommage d'un nombre minimal de formules atomiques. On précisera l'ensemble des renommages effectués.

3 — Ecrire alors les affirmations suivantes sous forme de clauses Prolog

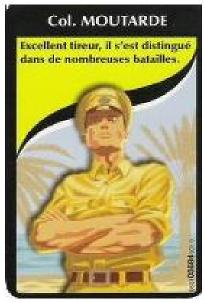
(1) *En cas de grève, je prendrai ma voiture*

(2) *Si je fais ce voyage, je prendrai le train ou la voiture*

(3) *Quelle calamité : il y aura grève ou bien ce sera encore la Coupe du Monde de football !*

EXERCICES : PROGRAMMATION PROLOG

Exercice 101 — Cluedo logique (objectifs 4.2.3 et 4.2.4)



Soit c*olonelMoutarde* et <u>mademoiselleRose</u> des *suspects*, *chandelier* un *objet contondant,* et le *bureau* et la *bibliothèque* une *salle*. Les empreintes de doigts appartenant au *colonelMoutarde* ont été trouvés dans le *bureau*, et celles de *mademoiselleRose* dans la *bibliothèque*. Un *cadavre* a été découvert dans le *bureau*. Le *chandelier* se trouve dans le *bureau*.

1 – En choisissant des prédicats adéquats, représentez ces connaissances par desclauses Prolog.

2 – Donner une clause Prolog utilisant les prédicats coupable/1, condamne/1 qui correspondent à l'assertion «*si un suspect est coupable d'un meurtre, alors il doit être condamné* »

3 – Donner une règle Prolog qui correspond à l'assertion « *si un cadavre a été découvert dans une salle et qu'on découvre les empreintes d'un suspect dans cette salle, alors le suspect est coupable* »

4 – Donner une règle Prolog qui correspond à l'assertion « *si un cadavre a été trouvé dans une salle et si un objet (contondant) se trouve dans cette salle, il s'agit indubitablement de l'objet qui a permis de commettre cet acte délictueux* »

5 – Donnez les arbres de dérivation qui correspondent aux question « Mademoiselle Rose est-elle coupable ?» et «Le Colonel Moutarde est-il coupable et quel objet a-t-il utilisé pour réaliser son forfait ?».

Exercice 102 — Progammation logique et littérature de l'absurde (objectifs 4.2.3 et 4.1.4)

Une bibliothèque possède un certain nombre d'ouvrages, tous écrits par des auteurs de l'absurde et classés en différentes catégories :

romans :	*La Peste*	*Camus*
essais :	*Le mythe de Sisyphe*	*Camus*
théâtre :	*Rhinocéros*	*Ionesco*
	En attendant Godot	*Beckett*
	Caligula	*Camus*

1 — Représenter les informations suivantes (par exemple : Godot est une pièce de théâtre dont l'auteur est Beckett) sous la forme de faits Prolog.

2 — Ecrire un prédicat `partout(Auteur)` qui réussit si la bibliothèque dispose d'œuvres de l'auteur `Auteur` dans tous les types d'ouvrages.

3 — Ecrire un prédicat `plusieurs(Auteur,Type)` qui réussit si la bibliothèque possède plusieurs œuvres de l'auteur `Auteur` dans la même catégorie `Type`. On suppose qu'on dispose d'un prédicat `diff(X,Y)` prédéfini qui réussit si `X` et `Y` sont différents.

Exercice 103 — Gastronomie Prologienne (objectifs 4.2.3 et 4.1.4)

On désire écrire un petit programme Prolog qui aide une personne donnée à choisir un menu au restaurant. Sur la carte de l'établissement, sont précisées les informations suivantes :

vin :	*cidre*
	côte du Rhône
plats :	*galettes de blé noir à l'andouille de Guéméné*
	ravioles du Royans
desserts :	*far aux pruneaux*
	pogne de Romans

1 — Représentation des connaissances — Représenter les informations ci-dessus (par exemple : *le far est un dessert*) sous la forme de faits Prolog.

2 — Pour chaque type de met, l'un est typique de Bretagne, l'autre de Rhône Alpes. Représenter les informations correspondantes en utilisant un prédicat d'arité 2 `lieu(Elt,Region)` qui réussit si `Elt` provient de `Region`.

3 — Ecrire un prédicat d'arité 4 `menu_reg(Vin,Plat,Dessert,Region)` qui réussit si le vin, le plat et le dessert considérés composent un menu typique de la `Region`.

4 — Fanch est breton et Pierre rhônalpin. Traduire cette information à l'aide du prédicat `lieu`. Ecrire un prédicat d'arité 4 `menu_habituel(Vin,Plat,Dessert,Personne)` qui réussit si le vin, le plat et le dessert composent un menu correspondant à la région d'où est originaire `Personne`.

5 — Par quel but obtiendrez vous le menu habituel qui sied à Fanch ?

Exercice 104 — Tous à Carrouf ! (15' ; objectifs 4.2.3 et 4.1.4)

On désire réaliser un programme Prolog qui modélise une base de données permettant de gérer le stock d'un supermarché. Dans cette base de données, chaque produit référencé est décrit par son nom, le rayon où il se trouve, son prix unitaire et la quantité (en nombre d'unités) présente en stock. Le tableau ci-dessous présente un extrait de cette BD.

On suppose que l'on dispose du prédicat prédéfini `between/2` défini comme suit : `between(Min,Val,Max)` réussit si `Val` est un entier ou un réel compris au sens large entre les valeurs `Min` et `Max`. Son seul mode d'utilisation est (+,+,+), c'est-à-dire que ses arguments doivent être instanciés par des valeurs lors de l'appel.

Nom	**Rayon**	**Prix (Euros)**	**Stock**
Marronsuisse Danone	Produits laitiers	1,6	0
BN Fraises	Biscuits	0,6	45
Huile olive Puget	Condiments	3,5	20

a) Représentation des connaissances – Comment représenterez-vous ces connaissances dans votre programme Prolog ? La représentation choisie devra vous permettre de répondre aisément aux problèmes ci-dessous.

b) Gestion de stock – Donnez l'écriture Prolog du prédicat `rupture/2` défini comme suit : `rupture(Prod,Rayon)` réussit si `Prod` est un produit présent normalement dans le rayon `Rayon` qui est actuellement en rupture de stock. Ce prédicat devra fonctionner dans tous les modes d'utilisation possibles.

c) Donnez des exemples de questions que vous poseriez pour tester la validité du prédicat `rupture/2` en mode (+,+).

d) Aide au client – Donnez l'écriture Prolog du prédicat `choix/3` défini comme suit : `choix(Rayon,Px_Max, Prod)` réussit si `Prod` est un produit présent dans le rayon `Rayon` et dont le prix est inférieur à `Px_Max`. Ce prédicat devra fonctionner dans les modes (+,+,-) et (+,+,+), c'est-à-dire que le client donne à chaque fois le rayon et le prix maximal recherché.

Exercice 105 — Conjonctions et disjonctions en Prolog : BD transport aérien

Conséquence de l'explosion d'Internet mais aussi de la mondialisation des échanges, les bases de données — et plus généralement les systèmes d'informations — occupent une place de plus en plus importante dans l'économie et constituent un domaine majeur de l'informatique. C'est pour cela que nous avons orienté notre Master Informatique vers les métiers liés au développement ou à la gestion des bases de données et leur interfaçage sur le WWW. Nous allons voir qu'il est également possible de définir des bases de données logiques en Prolog. Historiquement, le transport aérien a été un des premiers secteurs économiques faisant un usage intensif de grandes bases de données. Dans cet exercice, on considère une petite base de données donnant un ensemble d'informations sur les vols de différentes compagnies. La première table comprend les champs (attributs) décrits ci-dessous avec un exemple de contenu (enregistrements) :

- vol — numéro du vol
- compagnie — nom de la compagnie
- avion — type d'avion utilisé
- jour — jour de la semaine correspondant à ce vol
- horaire — horaire de départ du vol
- depart — code de l'aéroport de départ
- arrivee — code de l'aéroport d'arrivée

vol	compagnie	avion	jour	horaire	depart	arrivée
AF4056	Air France	Airbus	Me	9.45	NTE	LYS
AF7734	Air France	Fokker	Je	14.50	BRG	TLE
AL9712	Air Littoral	ATR	Lu	10.30	CDG	NIC

Par ailleurs, la table ci-dessous donne la liste des aéroports et de la ville principale qu'ils desservent :

- code — code de l'aéroport
- aéroport — nom de l'aéroport
- ville — ville desservie par l'aéroport

code	aéroport	ville
CDG	Roissy - Charles de Gaulle	Paris
BRG	Brest Guipavas	Brest
TLE	Toulouse Blagnac	Toulouse
NIC	Nice	Nice
NTE	Nantes Atlantique	Nantes Atlantique
LYS	Lyon Saint Saint-Exupéry	Lyon Saint Saint-Exupéry

Au cours de cet exercice, nous allons réaliser un petit programme qui nous donnera une idée de la réalisation en Prolog de bases de données simples ainsi que de leur interrogation.

1 — Représentation des connaissances — Comment représenterez-vous en Prolog les informations de cette base de données ? Donnez à titre d'illustration la représentation des vols figurant en exemple dans le tableau.

2 — On désire obtenir la liste des compagnies assurant une liaison entre deux villes. Définir le prédicat `liaison/3` tel que `liaison(Dep,Arr,Comp)` réussit si `Comp` est une compagnie assurant un vol entre `Dep` et `Arr`.

3 — On désire maintenir obtenir la liste des compagnies assurant une liaison entre deux villes *avec un départ matinal*. Définir un prédicat `liaison/4` tel que `liaison(Dep,Arr,Comp,Matin)` réussit si `Comp` est une compagnie assurant un vol entre `Dep` et `Arr` soit en matinée (`Matin` =1) soit en après-midi (`Matin` =0). Dans ce cas, il faut que le vol parte le matin ***ET*** qu'il assure la liaison demandée. On suppose que l'on dispose d'un prédicat `avant(X,Y)` qui réussit si l'heure X est inférieure à l'heure Y.

Afin de réduire leurs frais d'exploitations sur certaines liaisons intérieures peu rentables, les grandes compagnies aériennes ont de plus en plus recours à de petits opérateurs locaux assurant pour leur compte ces dessertes. C'est ainsi, par exemple, que la majeure partie des vols d'Air France au départ de la Bretagne sont assurées par BritAir. Cette information peut-être importante pour les clients, ces opérateurs régionaux proposant fréquemment un accueil à bord bien plus apprécié que celui des grandes compagnies.

On se propose donc de compléter notre base de données par une table complémentaire décrivant l'ensemble de ces vols réalisés par un opérateur autre que la compagnie commercialisant le vol :

- vol — numéro du vol
- compagnie — nom de la compagnie commercialisant le vol
- opérateur — nom de l'opérateur réalisant effectivement le vol

Le tableau ci-dessous donne un aperçu du contenu de cette base de données :

vol	compagnie	opérateur
AF4056	Air France	BritAir
AF7734	Air France	BritAir

4 — Représentation des connaissances — Comment représenterez-vous en Prolog ces nouvelles informations ? Donnez à titre d'exemple la représentation des accords commerciaux figurant ci-dessus.

5 — Modifier le prédicat `liaison/3` afin de considérer à la fois les compagnies commercialisant un vol ET celles l'assurant en tant qu'opérateur.

EXERCICES : SEMANTIQUE OPERATIONNELLE DE PROLOG

Exercice 106 — Paradis glacés (objectifs 4.1.5 et 4.2.4)

Si les îles sont le plus souvent synonymes de soleil, cocotiers et vahinés dans l'esprit des vacanciers occidentaux, il en est qui, balayées par les Quarantièmes Rugissants ou situées aux confins de l'Arctique, n'ont pour elles que leur solitude et leur caractère grandiose. On considère le programme Prolog suivant, correspondant à une mini base de données issue d'un SIG (système d'information géographique) consacré aux îles des régions arctiques :

```
(1)   ile(crozet,france,indien).
(2)   ile(georgie_sud,royaume_uni,atlantique_sud).
(3)   ile(francois_joseph,russie,arctique).
(4)   ile(ellesemere,canada,artique).
(5)   hemisphere(indien,sud).
(6)   hemisphere(atalantique_sud,sud).
(7)   hemisphere(arctique,nord).
(8)   possession(Pays,Hemisph):- ile(_,Pays,Ocean), hemisphere(Ocean,Hemisph).
```

1 — Donnez l'arbre de résolution correspondant à la question : `?- possession(france,sud).`

2 — Donnez l'arbre de résolution correspondant à la question : `?- possession(russie,sud).`

3 — Donnez l'arbre de résolution correspondant à la question : `?- possession(X,nord).`

Exercice 107 — Un petit tour à Roland Garros (objectifs 4.1.5 et 4.2.4)

On considère le programme Prolog suivant :

```
(1)   soleil.
(2)   lobes(ivanovic).
(3)   petit(safina).
(4)   meilleur(henin).
(5)   gagnant(X,Y)   :- meilleur(X).
(6)   gagnant(X,Y)   :- plus_malin(X,Y).
(7)   gagnant(X,Y)   :- plus_chanceux(X,Y).
(8)   plus_malin(X,_Y)      :- soleil, lobes(X).
(9)   plus_malin(X,Y)       :- petit(Y), lobes(X).
(10)  plus_chanceux(_X,safina).
```

Construire l'arbre de résolution Prolog correspond à la question ?- `gagnant(X,_Y)`. Quelle(s) réponse(s) obtient-on ?

Réponse indicative — On obtient 4 solutions possibles avec le même gagnant dans deux cas de figure.

Exercice 108 — Dès que le vent soufflera (objectifs 4.1.5 et 4.2.4)

On considère le programme Prolog suivant :

```
(1)   navigue(parlier,mardi).
(2)   navigue(desjoyeux,mercredi).
(3)   navigue(desjoyeux,jeudi).
(4)   navigue(ledam,J).

(5)   temps(mardi,calme).
(6)   temps(mercredi,calme).
(7)   temps(jeudi,grostemps).

(8)   aime(ledam,grostemps).
(9)   aime(X,calme).

(10)  chance(desjoyeux,mercredi).

(11)  gagne(X,J) :- navigue(X,J), temps(J,T), aime(X,T).
(12)  gagne(X,J) :- navigue(X,J), chance(X,J).
```

Question — Quelles sont **les** réponses données par Prolog au but : `?- gagne(X,mercredi)`
Donnez l'arbre de résolution correspondant.

Réponse indicative — On obtient 2 gagants possibles le mercredi ... mais trois démonstrations au total.

Exercice 109 — Récursivité par l'exemple: les amis de mes amis ... (objectifs 4.1.5 et 4.2.4)

Au chapitre suivant, nous allons découvrir les plaisirs de la programmation récursive, voir sa puissance mais aussi ses dangers. Rien ne nous empêche de découvrir par l'exemple ce type de programmation « se mordant la queue ». On considère le programme suivant :

```
(1)   ami(paul, pierre).
(2)   ami(pierre,jacques).
(3)   ami(X,Y) :- ami(X,Z), ami(Z,Y).
```

1 — Comment peut s'interpréter en français la règle (3) ? En quoi cette règle est-elle particulière ?

2 — Donnez l'arbre de résolution correspondant à la question : `?- ami(paul,jacques).`

3 — Donnez l'arbre de résolution correspondant à la question : `?- ami(paul,LesAutres).` Commentaire ?

PROBLEMES

Exercice 110 — Zoologie Prologienne (objectifs 4.1.3, 4.2.2., 4.2.3 et 4.2.4)

On considère la base de connaissances suivante, concernant des espèces alpines en voie de disparition :

Le tétras lyre est un oiseau (appelé aussi *coq de bruyère*)
Le gypaète barbu est un oiseau carnivore (appelé aussi *vautour des agneaux* car raffolant de ces tendres petites bêtes...)
Le lynx est un animal carnivore.
Un oiseau est un animal.
Les carnivores mangent de la viande.
Dans tout animal, il y a de la viande.
Tous les animaux boivent de l'eau.
Un animal consomme ce qu'il boit ou mange.

1 — Représenter cette base de connaissances en logique de prédicats du 1er ordre, puis en un programme Prolog.

2 — Les clauses Prolog du programme correspondent-elles bien à des clauses de Horn de la LP1 ?

3 — Comment interprétez-vous la question suivante : `?- carnivore(X), conso(X,Y).`

4 — Donnez l'arbre de résolution complet à cette question.

Exercice 111 — Erreurs de syntaxe et messages SWI-Prolog (15'; objectifs 4.1.4 et 4.2.5)

L'observation de la méthodologie de programmation des étudiants en séance de TP est une épreuve souvent éprouvante pour les enseignants qui s'attendent toujours à plus de réflexion de la part de leurs élèves. Il est ainsi rageant de vous voir modifier n'importe quoi au hasard dans votre programme suite à l'affichage d'une erreur de compilation, ou pire encore vous voir utiliser votre programme alors que des erreurs de compilation vous ont été signifiées… Ce petit exercice a précisément pour objectif de vous faire réfléchir à vos méthodes de programmation.

1 — Pourquoi est-il toujours possible d'utiliser un programme Prolog en dépit de l'affichage d'erreurs à la compilation ? Pourquoi est-on assuré que le comportement du système sera erroné ? Justifiez votre réponse.

2 — Pour compiler un programme sous SWI-Prolog, on appelle un prédicat prédéfini (`consult/1`). S'il y a des erreurs dans le programme compilé, l'appel à `consult` réussira-t-il (i .e. réponde Yes au final) ou non ?

On considère le message d'erreur suivant obtenu sur le compilateur SWI-PROLOG :

[WARNING: (/D:/toto.pl:2) Syntax error: Operator expected]

3 — Que signifie la partie « (/D:/toto.pl:2) » du message d'erreur ?

On considère le programme Prolog suivant (les numéros de lignes indiqués à gauche ne sont là que pour vous aider à vous repérer dans le fichier : ils n'apparaissent bien entendu pas dans le programme proprement dit) :

```
(1) /****************************************************************/
(2) /* FICHIER : syracuse.pl                                        */
(3) /* OBJET   : mise en oeuvre de l'algo de Syracuse               */
(4) /* AUTEUR  : J.Y. Antoine                                       */
(5) /* DATE    : 01/12/2001                                         */
(6) /****************************************************************/
```

```
(7) /* impair(N) reussit si N est un nombre entier impair ***********/
(8) /* pair (N) : idem avec parite */

(9)  impair(N) :- 1 is N mod 2.
(10) pair(N)   :- 0 is N mod 2.

(11) /* div2(NN,N) reussit si N est egal a la moitie de NN **********

(12) div2(NN,N) :- N is NN / 2. */

(13) /* suivant(Crt,Suivt) reussit si Suivt obtenu avec Crt par l'algo de Syracuse */

(14) suivant(Crt,Suivt) :- pair  (Crt), div2(Crt,Suivt).
(15) suivant(Crt,Suivt) :- impair(Crt), Suivant is Crt * 3 + 1.

(16) /* syra(N) execute l'algo de Syracuse a partir du nb N **********/

(17) syra(1)
(18) syra(N) :- write(N), tab(1), suivant(N,Suivt), syra(Suivt).
```

4 — Ce programme présente quatre erreurs. Il vous est demandé de les détecter en donnant à chaque fois la nature de l'erreur (i.e. quelle serait la bonne écriture), le message qu'affichera le compilateur Prolog en détectant cette erreur.

PROLOG : terminaison, récursivité

QUESTIONS DE COURS

Exercice 112 — Retour au port compromis (objectifs 4.2.4 et 5.2.2)

On désire réaliser un programme Prolog qui recherche, à partir de la donnée de relations maritimes directes (i.e. sans escales), l'ensemble des voyages possibles que l'on peut effectuer en bateau entre deux ports. Ce programme n'utilise qu'un seul prédicat d'arité 2 : `liaison(dep,arr)` qui réussit s'il existe une liaison maritime entre le port de départ `dep` et le port d'arrivée `arr`. Dans l'écriture proposée du programme, les relations directes sont alors représentées par un ensemble de faits, alors que les liaisons avec escale sont obtenues à l'aide d'une règle récursive :

```
(1)   liaison(brest,le_conquet).
(2)   liaison(le_conquet,molene).
(3)   liaison(molene,ouessant).
(4)   liaison(santander,brest).
(5)   liaison(brest,plymouth).
(6)   liaison(Dep,Arr) :- liaision(Dep,Esc), liaison(Esc,Arr).
```

1 — Donnez l'arbre de résolution Prolog correspondant à la question `?-liaison(brest,ouessant).`

2 — Donnez l'arbre de résolution Prolog correspondant à la question `?-liaison(brest,santander).` Quel problème rencontre-t-on ?

3 — Modifiez le programme pour résoudre le problème rencontré.

Exercice 113 — L'argent ne fait pas le bonheur : récursivité (mal) cachée (objectifs 4.2.4, 5.1.2. et 5.2.2)

On considère le programme Prolog suivant :

```
amoureux(jean).
heureux(X)    :- riche(X).
heureux(X)    :- amoureux(X).
riche(X)            :- actions(X).
actions(X)    :- argent(X).
argent(X)     :- riche(X).
```

1 — Donnez l'arbre de résolution Prolog correspondant à la question `?- heureux(jean).` Que se passe-t-il ?

2 — Comment résoudre le problème rencontré ?

Exercice 114 — Un peu de recul par rapport au cours… (objectif 5.1.2)

On considère un prédicat quelconque `lambda/2`. Sachant que ce prédicat n'est pas récursif et qu'il ne met pas en jeu de prédicats extralogiques, pouvez-vous dire quels sont ses modes d'utilisation ? Justifiez votre réponse.

Exercice 115 — Au suivant ! (10' ; objectifs 4.2.4 et 5.2.2)

On considère le programme Prolog suivant, définissant une relation d'ordre (supérieur) entre nombres entiers :

```
(1) suivant(1,0).
(2) suivant(2,1).
(3) suivant(3,2).
(4) sup(Sup,Inf) :- suivant(Sup,Inf).
(5) sup(Sup,Inf) :- sup(Sup,Entre), sup(Entre,Inf).
```

a) Donnez l'arbre de résolution correspondant à la question `?- sup(0,3).`

b) Qu'affichera Prolog en réponse à cette question `?- sup(0,3).`

c) Quels commentaires pouvez-vous faire sur le programme ?

Exercice 116 — Représentation des listes (objectif 5.1.3)

Ecrire la forme générale des listes suivantes :

1) liste quelconque à deux éléments
2) liste à un élément, telle que cet élément est une liste à deux éléments.
3) liste à deux éléments, telles que le deuxième élément est la liste vide.
4) liste ayant au moins deux éléments.
5) liste à au moins trois éléments, telles que le deuxième élément est une liste à trois éléments.

Exercice 117 — Listes : syntaxe (5' ; objectif 5.1.3)

Voici une fois encore a réalisé un programme présentant de nombreuses fautes de syntaxe… Pouvez-vous les corriger ?

```
/***************************************************************************/
/* FICHIER :      team.pl                                                  */
/* AUTEUR : Antoine Jean-Yves                                             */
/***************************************************************************/

/* faits */

equipe([nicolas, jean_pierre, jacques]).
equipe([lionel, dominique | laurent ]).
equipe([robert | [anne_marie , joseph]]).
equipe([arlette | alain , olivier]).
equipe([noel, [ dominique, alain]]).

/* oter(E,LE,L) reussit si L est la liste LE a laquelle on a ote E*/

oter(E,[E|Q],Q).
oter(E,[X|QE],[X|Q]) :- oter(E,[QE],[Q]).
```

a) On vous demande tout d'abord de donner l'ensemble des erreurs de syntaxe du programme, c'est-à-dire celles qui donneront lieu à des messages d'erreurs de la part de l'interpréteur Prolog lors de l'étape de consultation du programme (appel au prédicat `consult/1`). Il est demandé de donner à chaque fois la clause incriminée, l'erreur correspondante et comment vous la corrigeriez.

b) On vous demande tout d'abord de donner l'ensemble des erreurs de programmation qui correspondront à une sémantique non attendue du programme. C'est à dire aux erreurs qui ne seront pas détectées par l'interpréteur, mais qui conduisent visiblement à un comportement erroné du programme. Il est demandé de donner à chaque fois la clause incriminée, l'erreur correspondante et comment vous la corrigeriez.

Exercice 118 — Arbre de résolution avec liste (objectif 5.1.3 et 4.2.4)

On considère le prédicat d'arité 2 `elem(E,L)`, étudié en cours, qui réussit si E est un élément de la liste L :

```
(1) elem(E,[E|_]).
(2) elem(E,[_|Q]):- elem(E,Q).
```

Donnez les arbres de résolution complets qui correspondent aux questions suivantes :

1) `?- elem(1,[2,3]).`
2) `?- elem(X,[1,2]).`

EXERCICES : RECURSIVITE ET MODES D'UTILISATION

Exercice 119 — Les voisins de mes voisins sont mes voisins (5' ; objectifs 5.1.2 et 5.2.2)

Le programme Prolog suivant cherche à donner tous les couples de pays frontaliers aux abords de la péninsule ibérique. Il présente des problèmes de gestion de récursivité :

```
(1) frontalier(espagne,france).
(2) frontalier(espagne,portugal).
(3) frontalier(X,Y) :- frontalier(Y,X).
```

Réécrire le programme afin de rendre celui-ci utilisable dans tous les modes d'utilisation.

Exercice 120 — Vive la Royale (15' ; objectifs 4.2.4, 5.1.2 et 5.2.2)

On considère le programme Prolog suivant, qui décrit les relations de commandement au sein des officiers généraux et supérieurs de la marine nationale :

```
/* sup(Sup,Inf) reussit si Sup est le grade directement superieur à Inf */

    (1) sup(amiral,vice_amiral).
    (2) sup(vice_amiral,ctre_amiral).
    (3) sup(ctre_amiral,capt_vaisseau).
    (4) sup(capt_vaisseau,capt_fregate).
    (5) sup(capt_fregate,capt_corvette).

/* obeit(Inf,Sup) reussit si Sup a un grade superieur à Inf     */
/* modes d'utilisation recherches : (+,+) et (+,-)              */

    (6) obeit(Inf,Sup) :- obeit(Sup,Entre), obeit(Entre,Inf).
    (7) obeit(Inf,Sup) :- sup(Sup,Inf).
```

a – Qu'affichera l'interpréteur Prolog en réponse à la question `?- obeit(ctre_amiral,X).` Conclusion ?

b – Qu'affichera-t-il en réponse à la question `?- obeit(amiral,capt_corvette).` Conclusion ?

c – Modifiez le programme ci-dessus pour permettre un fonctionnement correct en modes (+,+) et (+,-).

d – En désire que le programme fonctionne en modes (+,+) et (-,+). Modifiez le programme en conséquence.

Exercice 121 — Factorielle (5' ; objectif 5.1.2)

On considère un programme Prolog ci-dessous réalisant le calcul de la factorielle FN d'un nombre N donné :

```
fact(0,1).
fact(N,FN) :- sup(N,0), plus(N,-1,N1), fact(N1,FN1), fois(N,FN1,FN).
```

On suppose que :

`sup(S,I)` est vrai si S entier est strictement supérieur à I. `sup/2` a pour mode d'utilisation (+,+).
`plus(A,B,AB)` est vrai si AB = A + B. `plus/3` a pour modes : (+,+,+), (+,+,-), (-,+,+), (+,-,+).
`fois(A,B,AB)` est vrai si AB = A * B. `fois/3` a pour modes : (+,+,+), (+,+,-), (-,+,+), (+,-,+).

Donnez les modes d'utilisation du prédicat `fact/2` ainsi défini.

Exercice 122 — Le programme inconnu (30' ; objectifs 4.2.4 et 5.1.2)

Etudiant en licence informatique, José Spéré-Ledis se voit proposer le programme inconnu suivant :

```
(0)   aez(A,B) :- jed(B,1,A).
(1)   jed (1,A,A).
(2)   jed (B,Q,A) :- QS is Q+1, 0 is B mod QS, BD is B div QS, jed(BD,QS,A).
```

On rappelle que les prédicats arithmétiques `mod` et `div` correspondent respectivement au calcul du reste et du quotient de la division entière.

1 — Donnez les arbres de résolution *complets* correspondant aux questions :

a) `?- aez(A,2).`
b) `?- aez(A,6).`
c) `?- aez(3,4).`

2 — Quels sont les modes d'utilisation de ce prédicat ? Justifiez votre réponse.

3 — Que réalise selon vous le prédicat `aez/2` ?

4 — Sauriez-vous donner une autre écriture de ce prédicat qui fonctionne en mode (+,-) ou (+,+) ?

5 — Dans la règle 2, que vérifie le sous-but `0 is B mod QS` ? On désire écrire un prédicat d'arité 2 `verif(B,QS)` qui réalise cette vérification. Donnez une écriture du prédicat qui réalise cette vérification en n'utilisant pas de prédicats extralogiques.

PROGRAMMATION RECURSIVE : EXERCICES SIMPLES

Exercice 123 — C'est pas moi, c'est l'autre (25' ; objectifs 5.2.1 et 5.2.2)

Souvenez-vous... La France a connu au cours de l'été 2002-2003 un épisode de canicule dont se serait bien passé Jean-Pierre Raffarin, le premier ministre de l'époque. Celui-ci fut en effet accusé par l'opposition, mais également par les observateurs politiques, de vacance de pouvoir alors même que des milliers de personnes mourraient d'hyperthermie dans des hôpitaux surchargés. L'explication qu'a donné le premier ministre à cette faillite de l'Etat se résume à la maxime « c'est pas moi, c'est l'autre » : il n'a pas été informé par le ministre de la Santé de l'époque (J-F. Mattei) qui lui même n'a pas été alerté par son Directeur Général de la Santé (L. Abenhaïm) qui lui même n'a pas reçu suffisamment d'informations du directeur (P. Brücker) de l'INSV (Institut National de Veille Sanitaire) qui lui-même…

Sans chercher à trancher ici sur les responsabilités de chacun, cet éprouvant épisode estival est l'occasion d'un exercice de rentrée portant sur la chaîne d'information qui doit remonter du terrain jusqu'aux cabinets gouvernementaux. Deux prédicats sont utilisés pour définir la liste des protagonistes de l'affaire et leur liens de subordination :

`fonction(P,F)` est vrai si la personne P a pour fonction (premier ministre, par exemple) F.

`superieur(F,FSup)` est vrai si la personne de fonction F est le subordonné direct de la personne de fonction Fsup (fonction hiérarchique directement supérieure).

1) Représentez les connaissances données dans l'énoncé ci-dessus à l'aide de ces prédicats.

Pour résoudre notre problème, on désire maintenant réaliser un prédicat `informe_par(P1,P2)`, qui réussit si la personne P1 doit être informée, directement ou indirectement (i.e. par voie hiérarchique) par la personne P2. Ce prédicat doit fonctionner en mode (+,+) et (+,-).

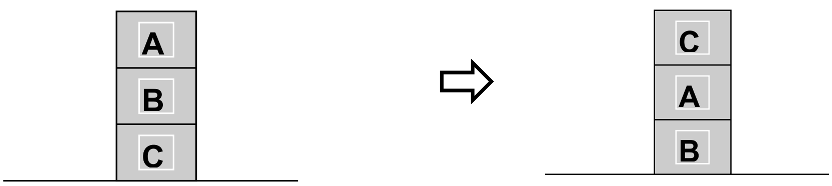
2) Donnez l'analyse du problème correspondant à la définition du prédicat.

3) Donner l'écriture du prédicat en Prolog.

4) Donnez un ensemble de tests permettant de vérifier **systématiquement** que le prédicat fonctionne bien dans les modes attendus.

Exercice 124 — Le monde des blocs (objectifs 5.2.1 et 5.2.2)

Se cherchant de grands défis intellectuels à solutionner, l'Intelligence Artificielle, à ses débuts dans les années 60, s'est passionnée pour ce qu'on appelait le *monde des blocs*. Il s'agissait en fait de résoudre un problème enfantin à la portée de tout nourrisson : déplacer, à l'aide des robots, des cubes empilés dans une configuration donnée pour les replacer dans une autre configuration, ceci en ne prenant à chaque fois qu'un seul cube. Par exemple :



La solution apparut rapidement, et ce grand classique trouva sa place dans tous les enseignements de base en Intelligence Artificielle. Le problème est d'ailleurs assez simple et peut être résolu en Prolog par un étudiant de 1er cycle.

Nous allons étudier ici une toute petite partie du problème. On représente tout empilement à l'aide d'un ensemble de faits `pose_sur(X,Y)`, prédicat qui réussit si X est posé directement Y. Ainsi, la pile à gauche de la figure ci-dessus se décrit comme suit :

```
pose_sur(c,sol).
pose_sur(b,c).
pose_sur(a,b).
```

Pour estimer si une position est correcte ou non, il peut être utile au robot de savoir d'une manière générale si un bloc donné est *au dessus* d'un autre, sans que ceux-ci soient posés l'un sur l'autre. Par exemple, sur la figure de gauche, A est au dessus de B, mais aussi de C et du sol.

Question — On demande précisément d'écrire le prédicat `sur(X,Y)` qui réussit sur X est au dessus de Y.

Exercice 125 — Histoire de l'Art (15' ; objectifs 5.2.1 et 5.2.2)

Si certains critiques d'art ont remis en cause une certaine vision déterministe de la modernité dans l'art, il n'en reste pas moins que l'évolution des courants artistiques s'est souvent faite de proche en proche : une école stylistique, une fois établie, influence de nouveaux artistes qui s'y intègrent puis vont plus encore de l'avant dans l'innovation pour créer finalement une nouvelle école artistique. En Prolog, on décide de représenter ces relations d'influences successives par une liste de faits utilisant le prédicat `prefigure(X,Y)`, qui réussit si l'école X a préfiguré l'apparition de l'école Y et est donc son prédécesseur direct dans l'histoire de l'art. On suppose (postulat très réducteur…) au passage que tout courant artistique n'a été influencé que par un prédécesseur unique. Par exemple :

```
prefigure(barbizon,impressionnisme).
prefigure(pont_aven,nabis).
prefigure(impressionnisme,fauvisme).
prefigure(cubisme,futurisme).
```

On veut écrire un prédicat d'arité 2 `apres(Apr,Avt)` qui réussit si `Avt` est une école stylistique apparue après `Apr`.

a) Donnez votre analyse du problème spécifié de la manière suivante : `apres(Apr,Avt)` réussit si l'école `Apr` est apparue après l'école artistique `Avt`

b) Donnez la programmation en Prolog du prédicat `apres`/2 conformément à cette analyse.

Exercice 126 — Wouèbe story, ou le mythe d'Ariane revisité (objectifs 4.2.4, 5.2.1 et 5.2.2)

Alec Tronic, l'ingénieur système du Département Informatique, est chargé de connecter l'ensemble des ordinateurs en libre service. Pour cela, il relie les ordinateurs deux à deux. Pour ne pas se tromper, il note à chaque fois le nom de chaque ordinateur relié, en tenant bien compte du sens de la connexion. Au bout d'un moment, il ne sait plus très bien où il en est. Sur son papier, il a noté les relations directes entre chaque ordinateur :

L'ordinateur a est relié à l'ordinateur f
L'ordinateur b est relié à l'ordinateur d
L'ordinateur d est relié à l'ordinateur c
L'ordinateur a est relié à l'ordinateur c
L'ordinateur c est relié à l'ordinateur e
L'ordinateur g est relié à l'ordinateur h

Pour l'aider, on va représenter chaque connexion directe entre des ordinateurs X et Y par le prédicat `lie(X,Y).` On suppose que ces connexions sont unidirectionnelles, c'est à dire que le fait que X est lié à Y n'implique pas que Y soit lié à X.

a — Définissez sous forme de clauses Prolog le prédicat `connecte(X,Y)` qui est vrai s'il existe une connexion, directe ou non, entre les ordinateurs X et Y. Là encore, il n'y a pas a priori de bidirectionnalité des connexions.

b — Etes-vous sûrs que votre programme ne boucle pas. Pour vous en persuader, écrivez l'arbre de preuve Prolog en réponse aux questions suivantes :

```
?- connecte(b,c).
?- connecte(c,b).
?- connecte(z,u).
```

Exercice 127 — Recherche de diviseur ... sans division (objectifs 5.2.1 et 5.2.2)

Pour l'ordinateur (et l'être humain !), la division est une opération plus difficile à réaliser qu'une simple addition ou soustraction. C'est ainsi qu'il n'est pas défini en Prolog de prédicat logique réalisant la division (cf. chapitre consacré à l'artithmétique Prolog et ses prédicat extra-logiques). Pour le programmeur désireux de n'utiliser qu'un langage Prolog pur, c'est à dire n'utilisant que des prédicats rigoureusement logiques, comment répondre alors au problème suivant : réaliser un prédicat d'arité `diviseur(N,Div)` qui réussit si l'entier `Div` est un diviseur de l'entier `N`.

La solution a ce problème est heureusement trouvée depuis l'antiquité : il suffiser de constater que si B est un diviseur de A, alors B est aussi un diviseur de B-A. Partant de cette propriété, sauez-vous réaliser ce prédicat ?

Indication — On pourra pour utliser le prédicat logique plus/3 défini comme suit : plus(X,Y,Z) réussit si Z est la somme de X et Y entiers relatifs.

Exercice 128 — Baby boom (45' ; objectifs 5.1.2.)

Le programme Prolog suivant cherche à donner l'ensemble des couples de jumeaux d'une maternité. Le prédicat `jumeaux/2` est sensé fonctionner en mode (+,+) et (+,-) :

```
/* faits : declarations des jumeaux */

(1) jumeaux(riri,fifi).
(2) jumeaux(igor,grichka).

/* regles : jumeaux(X,Y) est vrai si X et Y sont jumeaux et reciproquement */

(3) jumeaux(X,Y) :- jumeaux(Y,X).
```

1) Donnez l'arbre de résolution correspondant à la question : `?- jumeaux(fifi,riri).`

2) Donnez l'arbre de résolution correspondant à la question : `?- jumeaux(riri,X).`

3) Donnez l'arbre de résolution correspondant à la question : `?- jumeaux(cecilia,carla).`

4) A la lumière des questions précédentes, pouvez-vous expliquez quel(s) problème(s) présente ce programme ?

5) Réécrire le programme afin de rendre celui-ci utilisable dans tous les modes d'utilisation prévus.

On désire maintenant étendre la portée de ce programme à l'ensemble des triplés, quadruplés etc…

Représentation des connaissances — Dans ce cas, les bébés pris deux à deux sont considérés comme des jumeaux. On représentera donc les enfants multiples deux à deux, à l'aide du prédicat jumeaux/2. Par exemple, si Riri, Fifi et Loulou sont des triplés, on se contentera de définir les faits suivants :

```
jumeaux(riri,fifi).
jumeaux(fifi,loulou).
```

Spécification du problème — Il vous est demandé de réaliser un prédicat (qui peut-être `jumeaux/2`) d'arité 2 qui réussit pour tout couple de jumeaux, que ceux-ci fassent partie d'une « portée » de 2, 3, 4, 5 etc… enfants multiples.

6) Donnez l'analyse du problème correspondant à cette nouvelle définition du prédicat.

7) Donner l'écriture du prédicat correspondant à cette nouvelle définition

Exercice 129 — Calcul d'une fonction par méthode dichotomique (objectifs 5.2.1 et 5.2.2)

La recherche exacte des zéros d'une fonction est un problème mathématique parfois très complexe. L'ordinateur fournit alors une aide précieuse en trouvant une solution par approximation. La méthode la plus simple est de procéder par dichotomie : on considère la fonction sur un intervalle [a,b] sur lequel on sait que f change de signe (c'est à dire que $f(a)*f(b) \leq 0$). On cherche alors à réduire cet intervalle en le coupant en deux et en retravaillant ensuite sur le sous-intervalle vérifiant encore la propriété de changement de signe. On procède par itération jusqu'à ce que l'on tombe pile sur le zéro de la fonction, ou qu'on arrive à un intervalle dont on juge qu'il encadre de manière suffisamment précise le zéro. La valeur inférieure de l'intervalle est alors choisie comme approximation du zéro de la fonction.

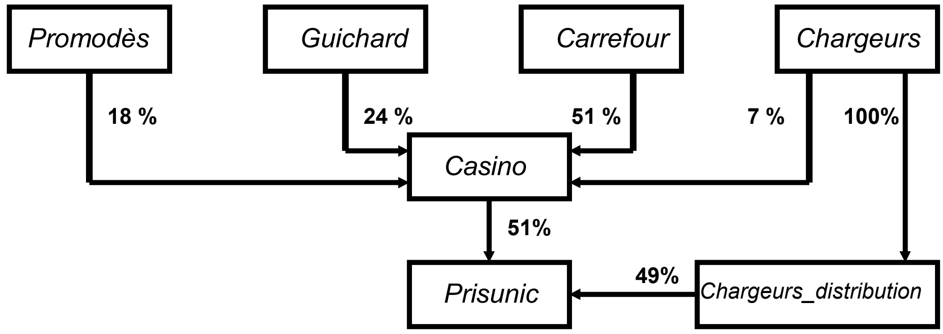
Question — On considère une fonction f d'arité 1définie en Prolog à l'aide du prédicat d'arité 2 `f(X,Y)` qui réussit si $Y = f(X)$. On demande de réaliser un prédicat d'arité 2 `zero(Zero,Min, Max, Seuil)` qui donne le zéro de cette fonction su l'intervalle [Min,Max] de manière exacte, ou approxime cette valeur avec un écart inférieur à `Seuil`. Le résultat étant représenté par le premier argument du prédicat.

Remarque — On suppose que l'on dispose pour cet exercice des prédicats prédéfinis suivants :

`plus(A,B,C)`	qui réussit si C est égal à la somme de A et de B
`fois(A,B,C)`	qui réussit si C est égal au produit de A et B
`div(A,B,C)`	qui réussit si C est égal à la division de A par B
`inf(A,B)`	qui réussit si A est inférieur à B

Exercice 130 — Mécano financier (objectifs 5.2.1 et 5.2.2)

Holding financières, participations croisées, O.P.A. amicales ou ennemies, intervention de fonds de pensions,... : les capital des grandes entreprises est de plus en plus complexe et fragmenté, à tel point que le petit actionnaire y perd le plus souvent ses petits. Dans cet exercice, nous allons nous intéresser au monde de la grande distribution, qui a connu ces dernières années plusieurs fusions / acquisitions spectaculaires.



Nous allons réaliser un petit programme qui permettra de répondre automatiquement a la question essentielle suivante : qui a le contrôle d'une société donnée. Pour cela, on étudie le cas — faussement artificiel — résumé par la figure ci-dessus.

Ce schéma se lit comme suit : la société *Promodès* détient 18% du capital de la société *Casino*. Les héritiers de Geoffroy *Guichard* détiennent 34 % de la société *Casino*, et ainsi de suite.

Remarque : dans cet exercice, on pourra utiliser les prédicats prédéfinis suivants :

`plus(A,B,C)`	qui réussit si C est égal à la somme de A et de B
`fois(A,B,C)`	qui réussit si C est égal au produit de A et B
`between(A,B,C)`	qui réussit si C est compris entre les valeurs de A et B

1 — Représentez en Prolog l'ensemble des faits illustrés sur la figure précédente. On utilisera pour cela le prédicat d'arité 3 `action(E1,E2,Part)` qui réussit si la société *E1* est un actionnaire de la société *E2* qui détient *Part* pour-cents du capital de cette dernière. On normalisera les pourcentages considérés, c'est à dire que 18% se représentera par exemple par le réel 0,18.

2 — Ecrire un prédicat d'arité 2 `filiale(E1,E2)` qui réussit si la société *E1* est une filiale de la société *E2*, c'est à dire que cette dernière détient une partie du capital de la première. On remarquera que cette détention de capital peut-être indirecte. Par exemple, votre programme doit répondre par l'affirmative à la question :

```
?- filiale(prisu,promodes).
```

3 — Etre actionnaire d'une société, c'est bien. En être l'actionnaire majoritaire, c'est mieux. Ecrire un prédicat d'arité 2 `controle(E1,E2)` qui réussit si la société *E1* est un actionnaire majoritaire de la société *E2* (nous dirons qu'elle la contrôle), ou bien qu'elle est actionnaire majoritaire d'une société qui contrôle la société *E2*. Ainsi, votre programme doit répondre par l'affirmative à la question : `?- controle(carrouf,prisu).`

Un des intérêts de ces prises de participation en cascade est qu'elles permettent de contrôler indirectement une société alors qu'on ne possède réellement qu'une minorité de son capital. C'est ce type de mécano financier qui permet, par exemple, à la famille Bouygues de contrôler à « peu » de frais un immense empire. Nous allons créer de nouveaux prédicats pour détecter ce genre de situation.

4 — Ecrire un prédicat `part_action(E1,E2,P)` qui réussit si la société *E1* possède en réalité *P* pour-cents du capital de la société *E2*. Par exemple, à la question

?- `part_action(carrouf,prisu,P).`

votre programme devra répondre par l'affirmative et donner `P = 0.2601`.

En effet, 51 % * 51 % = 26,01 %

On supposera défini un prédicat d'arit é 3 `fois(A,B,C)` qui réussit si C est égal à la multiplication de A par B.

5 — Enfin, écrire un prédicat d'arité 2 `bingo(E1,E2)` qui réussit si la société *E1* contrôle la société *E2* tout en ne possédant pas la majeure partie de son capital.

LISTES ET TERMINAISON

Exercice 131 — Listes et terminaison (objectif 4.2.4, 5.1.3 et 5.2.3)

On donne le programme Prolog suivant :

```
efface(X,[X|L],L).
efface(X,[H|L],[H|Leff]) :- efface(X,L,Leff).
```

Donner la réponse trouvée par Prolog à chacune des questions suivantes :

a — ?- `efface(2,[1,2,3,2,4],[1,3,2,4])`

b — ?- `efface(2,[1,2,3,2,4],[1,2,3,4])`

c — ?- `efface(2,[2,3],3)`

d — ?- `efface(X,[1,2,3,2,4],[1,2,3,4])`

e — ?- `efface(2,[1,2,3,2,4],L)`

Réponse indicative — On a une réponse positive pour toutes les questions sauf dans le cas c).

Exercice 132 — Problèmes d'utilisation (objectif 5.1.2. et 5.1.3)

Nous avons vu qu'il n'existait pas de méthode infaillible pour déterminer l'ordre des sous-buts d'une clause récursive. Cet exercice va précisément nous donner une illustration flagrante de cette impuissance : souvent, il n'existe d'ordre préférable de sous-buts. Simplement, à chaque ordre correspondra une utilisation différente du prédicat concerné.

a — On considère tout d'abord le prédicat `concat`, défini comme suit en cours :

```
(1)  concat([],L,L).
(2)  concat([T|Q],L,[T|QC]) :- concat(Q,L,QC).
```

Déterminez les modes d'utilisation de ce prédicat.

b — On considère maintenant le prédicat *inverse* (cf exercice 2-5). En fait, on peut définir ce prédicat de deux manières différant uniquement par l'ordre des sous-buts de la clause récursive :

```
(1)  inverse([],[]).
(2)  inverse([T|Q],L) :- inverse (Q,QR), concat(QR,[T],L).

(1') inverse([],[]).
(2') inverse([T|Q],L) :- concat(QR,[T],L), inverse (Q,QR).
```

Déterminez les modes d'utilisation du prédicat, pour les deux définitions. Conclusions ?

Réponse indicative — Les deux écritures se différencient par les modes d'utilisation (-,+) et (+,-).

Exercice 133 — Les Zuns et les Zotres (20' ; objectifs 5.2.3 et 5.2.4.)

Sur une planète éloignée règne Jeanmari Ier, le roi-despote des Zuns. Ce peuple belliqueux voue une haine séculaire aux Zotres, des étrangers n'ayant pas comme eux trois yeux, six antennes et la peau verte. Une loi ayant imposé aux Zotres de porter un nom comportant au moins un « z », le roi Jeanmari Ier a demandé à ses plus grands scientifiques de lui produire un programme Prolog permettant de détecter automatiquement tout nom Zotre, chaque nom étant représenté par une liste de lettres.

Après des années de recherche, le Haut-Comité Scientifique Zun propose le programme suivant : le prédicat d'arité 2 `type(Nom,Type)` doit être vrai si et seulement si le nom (rédigé uniquement en caractères minuscules) correspond à un patronyme réservé au type (Zun ou Zotre) adéquat.

```
(1) type( [ z |_] , zotre).
(2) type( [_|Q]   , Type ) :- type(Q,Type).
(3) type( []      , zun  ).
```

1 — Donnez les modes d'utilisation de ce programme, c'est-à-dire ici les modes pour lesquels le programme ne boucle pas, qu'il donne une réponse correcte ou erronée. Justifiez votre réponse.

2 — Il semble que le programme proposé ne soit pas correct. Afin de vous en persuader, donnez les arbres de résolution correspondant aux questions ci-dessous. Conclusions ?

a) `?- type([ j ,'O'],      Type).`
b) `?- type([ z , u , t ], Type).`
c) `?- type([ z , u , n ], zun ).`

Craignant le courroux de leur roi, les scientifiques proposent alors en désespoir de cause trois programmes alternatifs (`diff/2` est un prédicat d'arité 2, supposé prédéfini, qui réussit si ses deux arguments ne sont pas unifiables) :

Programme 1
```
(1) type( []      , zun  ) .
(2) type( [T|Q]   , Type ) :- type(Q,Type).
(3) type( [ z |_] , zotre).
```
Programme 2
```
(1) type( [ z |_] , zotre) .
(2) type( [T|Q]   , Type ) :- diff(T,z), type(Q,Type).
(3) type( []      , zun  ) .
```
Programme 3
```
(1) type( []      , zun  ) .
(2) type( [T|Q]   , Type ) :- diff(T,z), type(Q,Type).
(3) type( [ z |_] , zotre).
```

3 — Identifiez, parmi ces quatre propositions, les programmes incorrects. Justifiez vos réponses.

Exercice 134 — Le programme inconnu : modes d'utilisation et terminaison (contrôle 1999-2000 ; 25' ; objectif 5.1.2., 5.2.2. et 5.2.4.)

On considère le programme inconnu suivant :

```
(1) diwezh([T], T, []).
(2) diwezh([T|Q], S, [T|R]) :- diwezh(Q, S, R).
```

1 — Donnez les arbres de résolution *complets* correspondant aux questions suivantes :

1) `?- diwezh([], X, Y).`
2) `?- diwezh([1,2,3], X, Y).`

2 — Déduisez de la question précédente le rôle du prédicat `diwezh`. Donnez ses modes d'utilisation.

Indication — En breton : *diwezh* = *fin* en français.

3 — On désire maintenant utiliser `diwezh` pour programmer un second prédicat qui réalise l'inversion tête-queue des éléments d'une liste. Toujours facétieux, l'enseignant propose trois écritures de ce prédicat :

Programme A
```
(1a) renverse([], []).
(2a) renverse(L, [T|QR]) :- renverse(QL,QR), diwezh(L,T,QL).
```
Programme B
```
(1b) renverse([], []).
(2b) renverse(L, [T|QR]) :- diwezh(L,T,QL), renverse(QL,QR).
```
Programme C
```
(1c) renverse(L, [T|QR]) :- diwezh(L,T,QL), renverse(QL,QR).
(2c) renverse([], []).
```

Donnez, en justifiant vos réponses, les modes d'utilisation de chacun de ses programmes.

4 — En règle générale (liste de longueur quelconque) quel est le programme le plus efficace entre B et C ? Justifiez votre réponse.

Exercice 135 — Un autre programme inconnu : modes d'utilisation et terminaison (40' ; objectif 5.1.2., 5.2.2. et 5.2.4.)

Premier prédicat (contrôle continu) — Dans tout ce problème, on considèrera le programme (inconnu) suivant :

```
(1) dre_holl([],X,[]).
(2) dre_holl([T|Q],X,[T,X|QD]) :- dre_holl(Q,X,QD).
(3) dre_holl([T|Q],X,[T |QD]) :- dre_holl(Q,X,QD).
```

1 — Donnez les arbres de résolution correspondant aux questions suivantes :

1) `?- dre_holl([1,2,3],2,[1,2,2]).`
2) `?- dre_holl([1,2],3,X).`

2 — Quel est le rôle du prédicat `dre_holl/3` :

Indication — En breton : *dre-holl* = *partout* en français.

3 — Quel sont les modes d'utilisation du prédicat `dre_holl/3` (cochez les modes d'utilisation corrects) :

☐ (+,+,+) ☐ (+,+,-) ☐ (-,+,+) ☐ (-,+,-)

☐ (+,-,+) ☐ (+,-,-) ☐ (-,-,+) ☐ (-,-,-)

4 — Donnez la réponse à la question suivante :`?- dre_holl([[1,2],[3,4]],3, X).`

5 — On redéfinit le prédicat en plaçant en dernier les clauses non récursives du programme.

```
(1) dre_holl([T|Q],X,[T,X|QD]) :- dre_holl(Q,X,QD).
(2) dre_holl([T|Q],X,[T |QD]) :- dre_holl(Q,X,QD).
(3) dre_holl([],X,[]).
```

Quels sont alors les modes d'utilisation du prédicat ainsi redéfini .

Second prédicat (contrôle de septembre) — Dans tout ce problème, on considèrera le programme (inconnu) suivant :

```
(1) dizurzh([],[]).
(2) dizurzh([X,Y,Z |Q],[X,Z,Y|QG])     :- dizurzh(Q,QG).
(3) dizurzh([X,Y,Z |Q],[Y,X,Z|QG])     :- dizurzh(Q,QG).
(4) dizurzh([X,Y,Z |Q],[Y,Z,X|QG])     :- dizurzh(Q,QG).
```

1 — Donnez les arbres de résolution complets correspondant aux questions suivantes :

1) `?- dizurzh([1,3,2,4,5,6],[1,2,3,4,5,6]).`
2) `?- dizurzh([1,1,1],X).`

2 — Quel est le rôle du prédicat `dizurzh/3` ?

Indication — En breton : *dizurzhl* = *désordre* en français.

3 — Quel sont les modes d'utilisation du prédicat `dizurzh/3` ?

4 — Donnez la réponse du programme précédent aux questions suivantes. Commentaires ?

1) `?- dizurzh([1,2,3,4],X).`
2) `?- dizurzh([1,2,3,4,5],X).`

5 — Modifiez les cas d'arrêts du programme pour éviter les problèmes observés à la question précédente.

6 — Donnez alors la réponse du programme précédent à la question suivante :

```
?- dizurzh([[1,2,3]],X).
```

PROGRAMMATION AVEC LISTES

Exercice 136 — Quelques prédicats élémentaires (objectif 5.2.4.)

1 — Sans utiliser le prédicat `concat`, écrire le prédicat `ajout_en_fin(X,Ls,LsX)` vrai si et seulement si `LsX` est la liste obtenue en ajoutant l'élément `X` en queue de la liste `Ls`.

2 — Ecrire un prédicat `longueur(Ls,Long)` vrai si et seulement si `Long` est le nombre d'éléments de la liste `Ls` (utiliser `plus(X,Y,Z))`. Faire l'arbre de résolution du but : `?- longueur([1,2,3],N).`

3 — Ecrire le prédicat `double(L,LL)` vrai si et seulement si `LL` à la liste `L` dont on aurait doublé tous les éléments.

4 — Ecrire le prédicat `inverse(L,R)`, vrai si et seulement si *R* correspond à la liste `L` renversée. En déduire le prédicat `palindrome(L)` qui est vrai pour tout p alindrome (mot ou phrase) représenté sous par la liste `L` des lettres qui le compose.

5 — Ecrire le prédicat `adjacent(X,Y,L)` vrai si et seulement si `X` et `Y` sont deux éléments qui se suivent dans `L`.

Exercice 137 — Manipulation de listes et arithmétique (objectif 5.2.4.)

Dans cet exercice, on suppose que l'on dispose des prédicats prédéfinis suivants :

plus(A,B,C) qui réussit si C est égal à la somme de A et de B
fois(A,B,C) qui réussit si C est égal au produit de A et B
inf(A,B) qui réussit si A est inférieur à B

1 — Réaliser un prédicat d'arité 2 `somme(L,Somme)` qui réussit si `Somme` correspond à la somme des éléments de la liste `L`.

2 — Réaliser un prédicat d'arité 3 `somsign(L,SPlus,SMoins)` qui calcule les sommes `SPlus` et `SMoins` respectivement des éléments positifs et négatifs de la liste `L`.

3 — Réaliser un prédicat d'arité 3 `somspond(LVal,LPoids,Somme)` qui réussit si `Somme` est la somme des éléments de la liste `Lval` pondérés par les valeurs correspondantes de la liste `Lpoids`. Les 2 listes sont donc de même longueur.

Exercice 138 — Préfixes et suffixes (objectif 5.2.4.)

1 — On demande d'écrire de deux manières différentes (directement et en utilisant le prédicat `concat` de concaténation de listes) les prédicats suivants:

`prefixe(Prefs,Liste)` vrai si et seulement si la liste `Prefs` est un préfixe de la liste `Liste`
`suffixe(Suffs,Liste)` vrai si et seulement si la liste `Suffs` est un suffixe de la liste `Liste`

2 — Construire l'arbre de résolution des buts `prefixe(P,[1,2,3])` et `suffixe(L,[1,2])` pour chaque écriture.

3 — Ecrire 3 versions du prédicat `sous_liste(Sls,Ls)` vrai si et seulement si `Sls` est une sous liste de la liste `Ls`:

a) en utilisant prédicats `prefixe` et `suffixe` de la première question.

b) en utilisant uniquement le prédicat `prefixe`.

c) en utilisant le prédicat `concat`.

Quelle est l'écriture la plus efficace ? Pour vous aider à répondre, vous pouvez si nécessaire comparer les arbres de résolution de `sous_liste(Sls,[1,2])` avec ces différentes écritures.

4 — Mêmes questions avec le prédicat `dernier(X,Ls)` qui réussit ssi `X` est le dernier élément de la liste `Ls`.

Exercice 139 — Entre parenthèses (30' ; objectif 5.2.4.)

On se propose d'étudier des expressions arithmétiques composées exclusivement d'entiers et des deux opérateurs + et *, dans lesquelles les parenthésages sont complètement explicités. Par exemple, on veut écrire (3+(4*5))*6 et non pas l'écriture simplifiée (3 + 4*5)* 6 utilisant la connaissance des priorités entre opérateurs + et *.

On modélise ce type d'expression par une liste où *p* représente une parenthèse ouvrante et *q* une parenthèse fermante. L'expression donnée en exemple est donc modélisée par la liste [p,3,+,q,4,*,5,q,q,*,6].

Question 1 — Ecrire un prédicat d'arité 2 `nb_ouvrantes(L,NbO)` où `Nb0` est le nombre de parenthèses ouvrantes de l'expression. Ecrire un prédicat équivalent `nb_fermantes(L,NbF)` pour les parenthèses fermantes.

Question 2 — En déduire l'écriture du prédicat `nb_parentheses(Exp)` qui réussit si et seulement si l'expression `Exp` a le même nombre de parenthèses ouvrantes et fermantes.

Question 3 — Ecrire le prédicat `sous_liste(S,L)` vrai si et seulement si la liste `F` est une sous-liste de la liste `L` au sens des sous-expressions de la liste représentant l'expression arithmétique complète.

Question 4 — Ecrire le prédicat `valeur(Expr,N)` où `N` est l'entier qui correspond à l'évaluation de l'expression arithmétique que représente la liste `Expr`.

Indication — Une méthode possible est de remplacer récursivement des sous-listes de l'expression par leur valeur.

Exercice 140 — Un été de canicule (45' ; objectif 5.2.4.)

Par delà la question de la responsabilité du gouvernement dans la gestion de la canicule qu'a connu la France lors de l'été 2003, un autre débat anime aussi bien la communauté scientifique que le monde politique : cet épisode de grandes chaleurs a-t-il pour origine le réchauffement de la planète ? Les climatologues rappellent à juste titre que les accidents météorologiques exceptionnels ne sont pas nécessairement représentatifs du climat. 1921 reste ainsi l'année la plus sèche du XX° siècle, de même que les pics de chaleurs de l'été 1947 font toujours référence. Il n'empêche que, depuis une dizaine d'années, la plupart des indicateurs météorologiques se mettent au rouge et montrent que nos activités polluantes ont bien un effet sans précédent sur le climat planétaire.

Dans cet exercice, nous allons réaliser un ensemble de prédicats logiques permettant le calcul de ces indicateurs. A chaque fois, on effectuera l'analyse du problème étudié avant de donner la traduction en Prolog du prédicat. On suppose que l'on dispose des prédicats suivants :

`sup(Sup,Inf)`	est vrai si Sup est un nombre supérieur à Inf. Mode d'utilisation : (+,+)
`plus(A,B,APB)`	est vrai si APB = A + B. Modes d'utilisation : (+,+,+) et (+,+,-)
`div(A,B,Q)`	est vrai si Q, nombre réel, est égal à A divisé par B. Modes d'utilisation : (+,+,+) et (+,+,-)

Représentation des connaissances — Cet exercice porte sur les relevés de températures journalières moyennes (exprimées en degrés Celsius) de la planète, observées tout au long d'une année. Ces connaissances sont définies à l'aide du prédicat `releve/2` :

`releve(An,Temp)` est vrai si Temp est la liste des températures relevées chaque jour au cours de l'année An.

Le recours à une liste pour *Temp* est nécessaire du fait du nombre variable de jours dans une année (années bissextiles),.

1) Une température est qualifiée de caniculaire si elle dépasse les 35° C. Certains climatologues prévoient que le nombre de jours de canicule dans l'hexagone sera multiplié par 5 d'ici la fin du XXI° siècle du fait de l'effet de serre. On demande de créer un prédicat `Nb_canicule(An,Nb_Jours)` qui réussit si *Nb_Jours* est le nombre de jours caniculaires durant l'année *An*. Ce prédicat devra fonctionner en mode (+,+) et (+,-).

2) Les modifications du climat pourraient avoir d'autres effets inattendus. Par exemple, on redoute que la fonte des calottes polaires ne modifie l'équilibre des courants océaniques. La France ne bénéficierait ainsi plus du climat tempéré dû au Gulf Stream, mais pourrait connaître au contraire des hivers aussi rigoureux que ceux du Québec, qui est situé à notre latitude. Pour observer ce phénomène, on aimerait contrôler le nombre de jours de gels au cours d'une année. On demande donc de créer un prédicat `Nb_gel(An,Nb_Jours)` qui réussit si *Nb_Jours* est le nombre de jours de l'année *An* où la température a été inférieure à 0° C. Ce prédicat devra fonctionner en mode (+,+) et (+,-).

3) Si 1947 reste en moyenne l'année qui a connu les pics de chaleurs les plus élevés, certaines villes ont néanmoins battu ce record cette année. On demande donc de créer un prédicat `Max(An,TempMax)` qui réussit si *TempMax* correspond au maximum de température moyenne observé durant l'année *An*. Ce prédicat fonctionnera en mode (+, +) et (+,-).

4) L'indicateur le plus fiable de l'augmentation des températures reste la température moyenne au cours de l'année. On demande de créer un prédicat `Moy(An,Temp)` qui réussit si *TempMax* est la température moyenne observée durant toute l'année *An*. Ce prédicat fonctionnera en mode (+,+) et (+,-).

Exercice 141 — Casse-tête : constitution des groupes de TP (objectif 5.2.4.)

Chaque année, la constitution des groupes d'informatique est un vrai casse-tête à l'université. Le secrétariat de l'UFR Sciences a ainsi établi plusieurs listes correspondant à chacun des groupes constitués, ainsi qu'une liste regroupant l'ensemble des étudiants inscrits. Personne n'est cependant sûr que ces listes sont complètes et cohérentes entre elles. Nous allons écrire différents prédicats afin de répondre à ces questions.

1 — Prédicat *non_disjoint* : efficacité

Dans un premier temps, on désire savoir si les groupes sont cohérents, c'est à dire qu'ils ne possèdent pas d'élèves en commun. On définit alors le prédicat `non_disjoint(L1,L2)` qui réussit si au moins un élève a été placé dans les deux groupes.

Donnez une écriture de ce prédicat.

2 — Prédicats *intersect* et *reunion* : mode d'utilisation

On désire maintenant savoir quelle est la listes des étudiants qui sont effectivement dans plusieurs groupes, et si l'ensemble des étudiants de deuxième année est bien réunis dans ces groupes. Pour cela :

a — Ecrivez deux prédicats `inter(L1,L2,L)` et `reunion(L1,L2,L)` qui réussissent si la liste L est respectivement l'intersection et la réunion des ensembles d'élèves correspondants aux listes de groupe L1 et L2.

b — Précisez, en justifiant votre réponse, quels sont les modes d'utilisation de ces prédicats.

Exercice 142 — D'amour belle marquise vos beaux yeux me font mourir... (objectif 5.2.4.)

On désire réaliser un prédicat qui, à l'instar de Molière, donne toutes les permutations possibles au sein d'une liste L de mots (ou de tout autre chose). Pour cela, nous allons tout d'abord écrire quelques prédicats intermédiaires.

1 — Construire le prédicat `inserer(X,L,LX)` qui réussit si la liste LX correspond à la liste L, dans laquelle on a inséré l'élément X à un endroit quelconque.

2 — Construire le prédicat `supprimer(X,LX,L)` qui réussit si la liste L correspond à la liste LX à laquelle on aurait enlevé un élément quelconque X.

3 — Construire de plusieurs manières différentes le prédicat `permut(X,Y)` qui réussit si la liste X correspond à une permutation quelconque de la liste Y. Par exemple :

```
?- permut(['d amour','belle marquise','vos beaux yeux','me font mourir'], X).

 X =['belle marquise','vos beaux yeux','me font mourir','d amour']).
 Yes
```

4 — Votre programme ne boucle-t-il pas dans certains cas ? Construisez l'arbre de résolution Prolog de quelques questions bien choisies pour le vérifier.

Exercice 143 — Bienvenue à l'île des Pairs (Contrôle 2002-2003, 20') ; objectif 5.2.4.)

Au cours de nos TD de logique, nous nous sommes souvent aventurés sur les îles des Purs et des Pires. Ici, nous nous trouvons sur le royaume de Dubble-Dubble, le roi siamois de l'île des Pairs où tout marche par deux. Ne supportant pas la vue de nombre impairs, ce terrible despote vous demande de réaliser un programme qui ne conserve dans toute liste de nombres que la liste des nombres pairs la composant. Par exemple [1,5,6,9,4,8] donnera en réponse [6,4,8]. Saurez-vous relever ce défi, ou serez-vous condamné à subir le supplice du dédoublement de personnalité ?

a – Donnez l'analyse de ce problème. On précisera en particulier le découpage prédicatif envisagé pour réaliser le programme.

b – Donnez le programme Prolog correspondant. On pourra utiliser si nécessaire les prédicats `plus/3` et `between/3` de SWI-Prolog, en prenant garde à leur modes d'utilisations.

PROBLEMES

Exercice 144 — Mythe ou réalité ? (45' environ ; objectifs 5.2.1, 5.2.2)

Dans les temps reculés où Zeus, « assembleur des nuées » chanté par Homère, règnait sur l'Olympe, deux héros querelleurs, Atlas et Hercule, n'avaient de cesse de comparer leur force. Afin de les départager une fois pour toutes, le fils de Cronos leur proposa alors un défi : supporter la plus grande tour d'êtres humains montés les uns sur les autres. Grande était la force de nos deux héros. Si grande que douze aurores n'auraient suffit à compter le nombre d'hommes supportés par chacun. Assise auprès de Zeus, Héra « au trône d'or » lui conseilla alors de réaliser un programme Prolog pour l'aider dans ce décompte. Malheureusement, les Dieux de l'Olympe ont eux aussi leurs faiblesses, et jamais Zeus ne fut en mesure de satisfaire Héra. 3000 ans plus tard, saurez-vous surpasser le terrible « assembleur des nuées » et enfin réaliser ce programme ?

Représentation des connaissances — Les tours humaines seront décrites homme par homme, à l'aide du prédicat d'arité 2 `sur(X,Y)` qui est vrai si X supporte **directement** Y sur ses épaules. Par ailleurs, on définira des prédicat d'arité1 `base(X)` et `sommet(X)` qui sont vrais si et seulement si X est à la base (resp. au sommet) de la tour humaine. L'homme à la base supportant donc entièrement la tour.

On suppose la pyramide déjà complétement décrite, à l'aide d'une succession de faits. Par exemple :

```
base(atlas).
base(hercule).
sur(atlas,menelas).
sur(menelas,achille).
sur(achille,hector).
......
sommet(prias).
sommet(patrocle).
```

On demande d'écrire un prédicat d'arité 3 `gagnant(Heros1,Heros2,Gagnant)` qui étant donné deux *Héros* supportant une tour humaine, donne le plus fort des deux, au sens de celui qui supporte le plus grand nombre d'hommes. Ce prédicat devra être utilisable en mode (+,+,-). La mise en oeuvre de ce prédicat peut bien entendu supposer la définition d'autres sous-prédicats.

Question 1 : découpage prédicatif — On demande de donner le découpage prédicatif du programme attendu, c'est-à-dire de donner l'ensemble des prédicats nécessaires à sa réalisation, en précisant pour chacun :

- son rôle,
- son arité,
- le rôle de ses arguments,
- les modes d'utilisation de ce prédicat dans le programme.

Question 2 : programmation — Donner maintenant le programme complet correspondant à ce découpage.

Question 3 : amélioration — En réalité, les prédicats `base/1` et `sommet/1` peuvent être définis à partir du prédicat `sur/2` : une personne à la base n'est en effet sur personne, de même que celle au faîte de la tour ne supporte personne. Donnez donc une nouvelle écriture de ces prédicats, remplaçant leur définition précédente sous forme de listes de faits.

3 — Votre prédicat termine-t-il toujours ? Pour répondre à cette question, donnez l'arbre de résolution correspondant aux questions ci-dessous. Conclusion : quel est le problème ?

```
?- allies(france,russie).
?- allies(france,ru).
```

4 — Une solution pour éviter les bouclages constatés ci-dessus est de mémoriser dans une liste les pays déjà considérés lors de la recherche d'alliance et d'interdire toute reconsidération de ces pays dans la poursuite de la recherche. On attribue ainsi une arité 3 au prédicat d'alliance : `allies(X,Y,L)` réussit si X et Y sont alliés, la liste L correspondant aux pays considérés pour établir cette alliance.

a) Comment reformule-t-on les buts de la question 3 ?
b) Réecrire le prédicat en tenant compte de cette modification.

Même si les considérations géopolitiques de l'Autriche Hongrie étaient tournées ailleurs, l'origine de la guerre porte sur son conflit avec la Serbie. Dans le programme, cn décrit cet état par le fait :

```
est_ennemi(autriche_hongrie,serbie).
```

5 — La logique militaire se résume en un seul axiome : les ennemis de mes amis sont mes ennemis, et réciproquement... Ecrire un prédicat `ennemi` qui réussit pour tout pays X et Y membres respectivement des 2 alliances opposées. Là encore, il faudra tenir compte des éventuels bouclages intempestifs…

