T.M.E. Logique

Machine : sous WINDOWS, ENSEIGNEMENT
Tarski se trouve dans *tout le réseau / Enseignement / C/ Program Files*
Faire une copie du fichier d'exercices dans *votre directory personnel* **h** : vous ne pourrez travailler que sur cette copie.

Tarski's world est un petit logiciel illustrant de façon simple et récréative les notions de *modèle* et d'*interprétation* des formules du calcul des prédicats.

0.1 Présentation

Domaine d'interprétation

Le "monde" servant de support au domaine d'interprétation de formules est une grille carrée de 8 sur 8 sur laquelle on peut disposer trois sortes de figures géométriques de tailles différentes. Les figures sont : la pyramide, le cube et le dodécaèdre. Les tailles sont : petit, moyen et grand.

Langage

Pour décrire ce monde géométrique, le logiciel propose un *langage* réduit (ensemble de symboles de variables, constantes et prédicats).

- 6 symboles de constantes : `a b c d e f`
- 6 symboles de variables : `u v w x y z`
- 13 symboles de prédicats dont
  - 6 symboles unaires : `Tet Cube Dodec Small Medium Large`
  - 6 symboles binaires : `Smaller Larger LeftOf RightOf BackOf FrontOf`
  - 1 symbole ternaire : `Between`

La sémantique attendue des symboles est indiquée par la signification du mot anglais utilisé pour les désigner. Pour en être sûr, vous pouvez déterminer la sémantique des symboles de prédicat en saisissant pour chacun d'eux une formule atomique et en construisant un monde où elle est valide et un monde où elle est fausse. A-t-on par exemple `Smaller(a,b)` si `a,b` sont tous deux *small*, ou bien `FrontOf(a,b)` si `a,b` sont sur une même horizontale ?

Ergonomie

Le logiciel offre une interface graphique aux fonctionnalités usuelles (usage de la souris, menus, etc.). Il présente à l'utilisateur quatre fenêtres :

1. une représentation graphique de la grille offrant, sur la gauche, les trois figures possibles. Cette fenêtre sert à construire un domaine d'interprétation. Au lancement du programme, cette fenêtre est intitulée `NEW0.WLD`.

2. un "clavier" dont chaque touche permet l'édition des symboles logiques (quantificateurs, connecteurs) et des symboles du langage (variables, constantes, prédicats).

3. une zone de saisie des formules séparées l'une de l'autre par un trait horizontal. L'utilisateur peut éditer une formule à l'aide du clavier du logiciel ou directement (clavier de sa console). Au lancement du programme, cette fenêtre est intitulée `NEW0.SEN`.

4. la dernière fenêtre, intitulée `Inspector` permet deux catégories d'opérations : l'édition des caractéristiques d'une figure ; la vérification de la validité d'une formule (fenêtre 3) dans le monde représenté dans la fenêtre 1. Le choix de l'une ou l'autre de ces opérations est accessible par un petit menu en haut de la fenêtre. L'option `Block` correspond à l'édition des figures ; l'option `Sentence` correspond à la vérification des formules.

Complément : la figure suivante illustre l'interprétation du prédicat `Between`, les cases grisées sont celles qui satisfont `Between`(z, x, y) si x et y sont représentés par les points noirs.

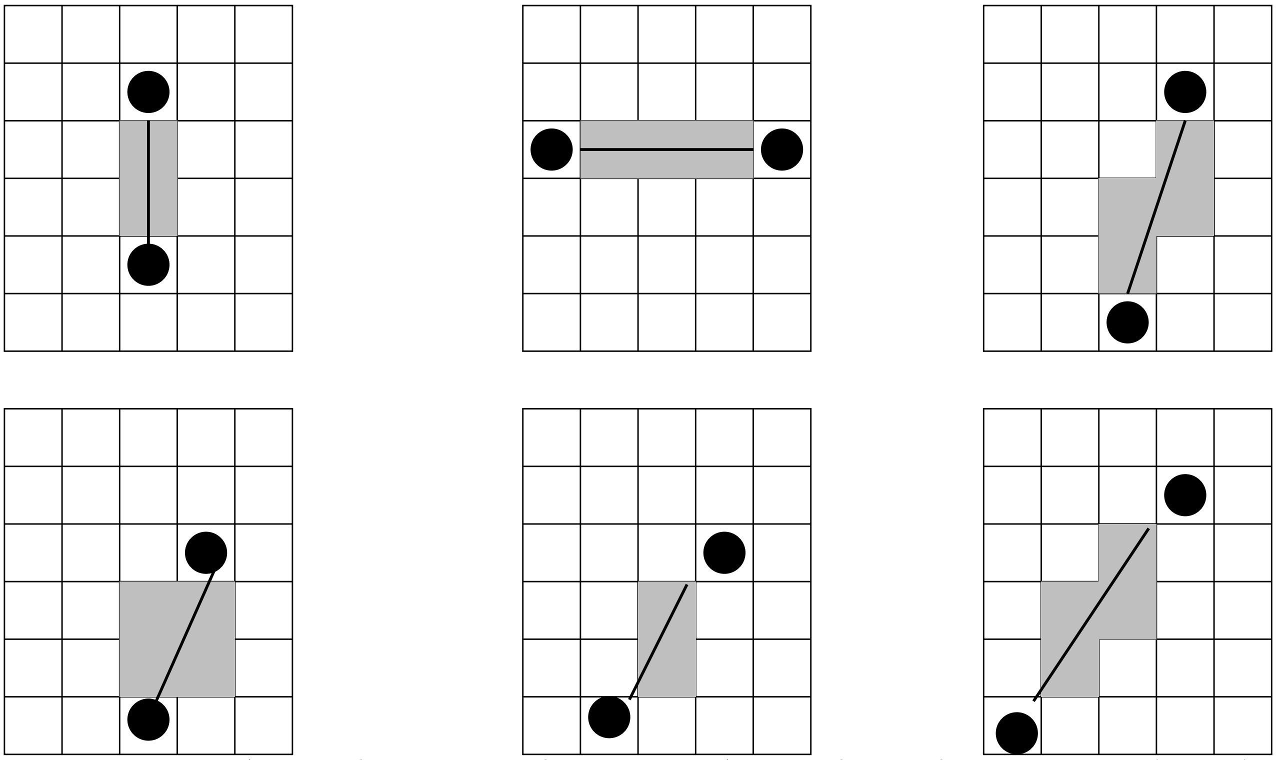

Figure 1 Interprétation de `Between` : les cases grisées sont les z tels que `Between`(z, x, y)

Construction d'un monde

- En cliquant sur l'une des figures à gauche de la fenêtre, vous créez une nouvelle instance de cette figure sur la grille.
- Par "glisser-déposer" vous pouvez déplacer les figures d'une case à l'autre. Deux figures ne peuvent cohabiter sur une même case.
- En mode `Block` de la fenêtre `Inspector`, vous pouvez modifier les attributs d'une figure préalablement sélectionnée par un clic. On peut modifier la nature des figures, leur taille ou nommer une figure en utilisant un des symboles de constante (`a`, `b`, `c`, `d`, `e` ou `f`)

REMARQUE 0.1 Une même figure peut recevoir plusieurs noms (symboles de constantes).

Validation d'une formule

Il y a trois niveaux de vérifications :
- correction syntaxique : `wff` *i.e.* *Well-Formed-Formula* ou encore formule bien formée selon les règles d'écriture des formules ;
- énoncé clos (ou non) : `sentence` (ou non) ;
- vérité : la formule est vraie dans le modèle considéré.

Pour chacun de ces niveaux, deux réponses sont possibles (`yes` ou `no`) que l'on sélectionne en cochant la réponse attendue. Un bouton `Verify` permet de demander au logiciel de contrôler la validité de vos choix. Une croix est affichée sur la ligne correspondante lorsque la réponse proposée est erronée et un signe ressemblant à un V est affiché sur la ligne correspondante lorsque la réponse proposée est correcte.

SYNTAXE les expressions sont *complètement* parenthésées - sauf lorsque le connecteur est associatif (conjonction ou disjonction)

VALIDITÉ seuls les *énoncés clos* peuvent être vrais ou faux.

Le jeu de la vérité

Le logiciel propose un mode "pas-à-pas" pour vérifier la validité d'un formule : le bouton `Game` de la fenêtre `Inspector`. À chaque étape de validation (ou d'invalidation) on peut vous demander de sélectionner une sous-formule, de sélectionner ou de nommer une figure du monde. La succession des étapes constitue une *preuve* de la vérité ou la fausseté de l'énoncé traité. En fait, cette preuve est partielle : par exemple, tous les cas d'une conjonction ne serons pas examinés.

REMARQUE 0.2 avant de pouvoir utiliser cette fonctionnalité, il faut avoir fait vérifier (bouton `Verify`) la vérité ou la fausseté de la formule visée.

Mondes prédéfinis

Le logiciel propose un certain nombre de mondes ou d'ensemble de formules prédéfinis accessibles par le menu `File`, option `Open`, puis `World` ou `Sentence`. Les mondes prédéfinis sont dans le sous répertoire `exercise`.

0.2 Exercices - Calcul propositionnel

Les 3 exercices suivants sont tirés du livre Tarski's World de Barwise et Etchemendy

Exercice 1 Chargez le fichier `DeMorg.sen`. Réaliser un monde qui rende toutes les formules vraies. ♢

Exercice 2 1. Chargez le fichier `Boole.wld`. Créez un fichier `formules.sen` où vous mettrez des formules exprimant que :

1. **f** (le grand dodécaèdre au fond) n'est pas devant **a**
2. **f** est à droite de **a** et à gauche de **b**
3. **f** est soit derrière **a** soit plus petit que **a**.
4. **e** et **d** sont entre **c** et **a**.
5. Ni **e** ni **d** ne sont plus grands que **c**.
6. **e** n'est ni plus grand ni plus petit que **d**.
7. **c** est plus grand que **e** et plus petit que **a**.
8. **c** est plus petit que **f** et de plus il est devant **f**.

2. Changez le monde de manière à ce que toutes vos formules $1-8$ deviennent fausses dans ce nouveau monde. Vous pourrez commencer par déplacer **f** dans le coin avant de la grille. Ensuite déplacez **e** dans le coin arrière gauche de la grille et faites-en un grand objet. Vérifiez toutes vos formules $1-8$ dans ce monde. Si vous les avez formulées correctement elles devraient être toutes fausses. Si l'une d'elles est encore vraie, pouvez vous trouver votre erreur. ♢

Exercice 3 1. Créez un nouveau fichier de formules et inscrivez-y la traduction en logique du premier ordre des phrases suivantes :

1. **a** est petit ou **c** et **d** sont grands.
2. **d** et **e** sont tous deux derrière **b**.
3. **d** et **e** sont tous deux derrière **b** et sont tous deux plus grands que **b**.
4. **c** et **d** sont tous deux des cubes et de plus aucun des deux n'est petit.
5. Ni **e** ni **a** n'est à droite de **c** et à gauche de **b**.
6. Soit **e** n'est pas grand, soit **e** est derrière **a**.
7. **c** n'est ni entre **a** et **b**, ni en avant de **a** , ni en avant de **b**.
8. Soit **a** et **e** sont des tétraèdres, soit **a** et **f** sont des tétraèdres.
9. Ni **c** ni **d** n'est devant soit **c**, soit **b**.
10. **c** est entre **d** et **f** ou plus petit que **d** et plus petit que **f**.

Enregistrez cet ensemble de propositions sous le nom exo3.sentences.

2. Chargez le fichier `Wittgenstein.wld`. Vérifiez que toutes les propositions écrites en français dans la question 1. sont vraies dans ce monde. Vérifiez que vos traductions sont toutes valides ; si ce n'est pas le cas corrigez-les.

3. Modifiez le monde de `Wittgenstein.wld` comme suit : réduisez à la taille *small* tous les objets moyens ou grands, et agrandissez à la taille maximale tous les petits objets. Après ces changements, les formules 1, 3, 4 et 10 deviennent fausses, et toutes les autres restent vraies. Vérifiez qu'il en est bien ainsi dans votre fichier exo3.sentences (si ce n'était pas le cas, corrigez vos formules). Ensuite faites pivoter de 90 degrés vers la droite le monde ainsi modifié : les formules 5, 6, 8 et 9 devraient être les seules à être vraies. Corrigez vos formules si ce n'était pas le cas.

Chargez le fichier `Boole.wld` : seule la formule 6 doit être vraie dans ce monde. Corrigez vos formules si ce n'était pas le cas. Modifiez `Boole.wld` en échangeant les positions de **b** et **c**. Les formules 2, 5, 6 et 7 deviennent vraies et les autres restent fausses. Corrigez vos formules si ce n'était pas le cas. ♢

0.3 Exercices - Calcul des prédicats

Les 12 exercices suivants sont tirés du livre Tarski's World de Barwise et Etchemendy

Exercice 4
1. Ouvrir `Edgar.sen`. Evaluer les formules dans `Edgar.wld`.
2. Quelle formule traduit "il y a un grand tétraèdre" ?
3. Quelle formule traduit "il y a un cube entre **a** et **b**" ?
4. Exprimer en français les assertions faites par les formules 5 et 6. ♢

Exercice 5
1. Ouvrir `Allan.sen`.
2. Quelle formule traduit "un dodécaèdre est grand" ? Construisez des mondes pour vérifier votre choix.
3. Quelle formule traduit "tous les tétraèdres sont petits" ? Construisez des mondes pour vérifier votre choix.
4. Construisez un monde où la formule 3 soit fausse et la formule 4 soit vraie.
5. Construisez un monde où les formules 2 et 4 soient vraies et les formules 1 et 3 soient fausses.
6. Peut-on construire un monde où la formule 1 soit vraie et la formule 2 soit fausse. Si oui construisez-le, si non expliquez pourquoi. ♢

Exercice 6 Ouvrir un nouveau fichier de suffixe `.sen` et y mettre les formules traduisant les assertions suivantes (chaque formule contiendra un seul quantificateur qui sera universel) :
1. Tous les cubes sont petits.
2. Chaque petit cube est à droite de **a**.
3. Tous les dodécaèdres sont grands.
4. **a** est à gauche de tout dodécaèdre.
5. Tout tétraèdre moyen est devant **b**.
6. Chaque cube est devant **b** ou derrière **a**.
7. Tout cube est à gauche de **b** et à droite de **a**.
8. Tout ce qui se trouve entre **b** et **a** est un cube.
9. Tout ce qui est plut petit que **a** est un cube.
10. Tous les dodécaèdres ne sont pas petits. Remarquer que cette assertion est ambiguë : on peut la traduire par une formule commençant par $\forall$ ou bien par une formule commençant par $\neg$. Vous choisirez la première formule, celle qui affirme que tous les dodécaèdres sont soit moyens soit grands.
11. Aucun dodécaèdre n'est petit.
12. **a** n'est pas à droite de tout objet. Remarquer que cette assertion est ambiguë : la traduire comme la négation de "**a** est à droite de tout".
13. **a** n'est à droite d'aucun objet. Remarquer la différence avec l'assertion précédente.
14. **a** n'est pas à droite d'un cube.
15. Si un objet est un cube, alors cet objet est à gauche de **b** et à droite de **a**.
16. Un objet est un cube, si et seulement si cet objet est à gauche de **b** et à droite de **a**.

Sauvegardez votre fichier dans `exo6.sen`.

1. Ouvrez `Claire.wld`. Vérifiez que toutes vos formules sont vraies (sinon il y une erreur dans vos formules que vous corrigerez).

2. Changez le fichier `Claire.wld` en déplaçant **a** dans l'angle droit au premier rang. Vérifiez que les formules 2, 4, 7, 13, 14, 15, 16 sont fausses et les autres vraies (sinon il y une erreur dans vos formules que vous corrigerez).

3. Chargez le fichier `Wittgenstein.wld`. Remarquez que les assertions 2, 4, 8, 9, 12, 14 sont vraies et les autres fausses. S'il n'en est pas de même de vos formules, vous les corrigerez.

4. Chargez le fichier `Leibniz.wld`. Remarquez que les assertions 3, 4, 8, 9, 10, 11, 12, 13, 14 sont vraies et les autres fausses. S'il n'en est pas de même de vos formules, vous les corrigerez. ♢

Exercice 7 1. Chargez le fichier `Leibniz.wld`. Ouvrez un nouveau fichier `exo7.sen` où vous mettrez des formules traduisant les assertions 1 à 5 suivantes. Chaque assertion que vous allez traduire est vraie dans ce monde. Vérifiez qu'il en est de même des formules que vous écrirez.

1. **b** est un tétraèdre plus petit que **e**.
2. Il n'y a pas de cube moyen.
3. Rien n'est devant **b**.
4. Tout cube est soit devant soit derrière **e**.
5. Aucun cube n'est entre deux objets (Il faut 3 quantificateurs pour traduire cette assertion).

1. Changez le monde de sorte que toutes les assertions deviennent fausses. Pour ce faire, transformez **b** en cube moyen, supprimez le tétraèdre le plus à gauche et mettez **b** à sa place, puis mettez un petit cube à l'ancienne place de **b**. Toutes vos formules doivent être fausses dans ce modèle; si ce n'est pas le cas, vous les corrigerez.

2. Faites divers changements au monde pour que certaines assertions soient vraies et d'autres fausses, et vérifiez que vos formules prennent la même valeur de vérité que les assertions. ♢

Exercice 8 1. Chargez le fichier `Ramsey.sen`. Les formules sont soit existentielles soit sans quantificateur : on peut donc les rendre vraies en ajoutant des objets nouveaux pour chaque formule, mais un des buts de l'exercice est d'ajouter le moins possible d'objets. Réaliser un monde qui rende les formules 1–10 vraies. On essaiera de faire un tout petit monde : 6 objets doivent suffire (il faudra que l'un des objets ait 2 noms).

2. Si le monde réalisé à la question 1) ne satisfait pas les formules 11–20 (formules universelles), modifiez-le *sans ajouter d'objet* pour que toutes les formules 1–20 soient vraies.

3. Changez le monde de la question 2), *uniquement en ajoutant* des objets pour falsifier le plus grand nombre possible de formules de `Ramsey.sen`. Quelles sont les formules que vous arrivez à falsifier, et quelles sont celles que vous n'arriverez pas à falsifier ? ♢

Exercice 9 1. Chargez le fichier `Peano.wld`. Créez un fichier `formules.sen` où vous mettrez des formules exprimant que :

1. *Tout dodécaèdre est petit.*
2. *Il y a un cube moyen.*
3. *Il y a au moins deux cubes.*
4. *Il y a un tétraèdre entre deux cubes.*
5. *Tout cube n'est pas devant un dodécaèdre.* Remarquer que cette assertion est ambiguë : la traduire comme la négation de “Tout cube est devant un dodécaèdre”.

2. Changez le monde en remplaçant le cube moyen par un dodécaèdre moyen. Toutes vos formules doivent devenir fausses dans ce nouveau monde. ♢

Exercice 10 1. Chargez le fichier `Buridan.sen`. Construire un monde où toutes les formules sont vraies et sauvegardez-le dans `exo10.wld`.

2. Les assertions suivantes sont des conséquences des formules de `Buridan.sen`, elles doivent donc être vraies dans `exo10.wld`.

1. *Il n'y a pas de cube.*
2. *Un tétraèdre n'est pas grand.* remarquer que cette assertion est ambiguë pour certains, vous choisirez de l'interpréter comme “il y a tétraèdre qui n'est pas grand”
3. *Rien n'est derrière* **a**.
4. *Il n'y que des grands objets derrière* **b**.

Traduisez les assertions 1 à 4 en formules, ajoutez-les à `Buridan.sen` et sauvegardez le tout dans `exo10.sen` et vérifiez que toutes les formules de `exo10.sen` sont vraies dans `exo10.wld`.

3. Pour chaque formule traduisant une des assertions 1 à 4, modifiez le modèle `exo10.wld` pour qu'elle devienne fausse, et cherchez celle(s) des formules initiales de `Buridan.sen` qui sont devenues fausses.

4. Montrez que la formule 9 n'est pas conséquence des 13 autres formules : il suffit de construire un monde (que vous appelerez `exo10b.wld`) où 9 est fausse et où toutes les autres sont vraies.

5. Montrez que la formule F suivante (affirmant qu'il y a au moins 2 tétrèdres moyens) est indépendante des formules de `Buridan.sen` : *i.e.* F n'est pas conséquence de formules de `Buridan.sen` et $\neg F$ non plus.

$$\exists x \exists y\ ((x \neq y) \wedge Tet(x) \wedge Tet(y) \wedge Medium(x) \wedge Medium(y))$$

Il suffit de construire deux mondes, l'un où F est fausse et où toutes les formules de `Buridan.sen` sont vraies, et l'autre où $\neg F$ est fausse et où toutes les formules de `Buridan.sen` sont vraies. ♢

Exercice 11 1. Chargez le fichier `Whitehead.sen`.

1. Testez les formules 1 à 7 dans des mondes à 1, 2, 3, ou 4 objets. Quelles formules affirment qu'il y a au moins (au plus, exactement) 2 objets ? Quelles formules affirment qu'il y a au moins (au plus, exactement) 3 objets ?

2. Quelle est la différence entre ce qu'affirment les formules 8 et 9 ?

3. Construire un monde où la formule 10 est vraie.

4. Construire un monde où les formules 11 et 12 sont vraies.

5. A quelle formule est équivalente la formule 13 ? Traduisez en français. Vérifiez en construisant des modèles.

5. Qu'affirme la formule 14 ? Vérifiez en construisant des modèles. ♢

Exercice 12 1. Chargez le fichier `Montague.sen`. Complétez les formules. Les assertions faites par les formules sont les suivantes :

1. *Tout cube est à gauche de tout tétraèdre :* il faut donc compléter par

$$\forall y (Tet(y) \longrightarrow LeftOf(x, y))$$

2. *Tout petit cube est derrière un grand cube.*
3. *Un cube est devant tous les tétrèdres (*assertion ambiguë*).*
4. *Un grand cube est devant un petit cube (*assertion ambiguë*).*
5. *Rien n'est plus grand que tout.*
6. *Tout cube qui est devant tous les tétraèdres est grand.*
7. *Tout ce qui est à droite d'un grand cube est petit.*
8. *Rien de ce qui est derrière un cube et devant un cube n'est grand.*
9. *Tout objet qui n'a rien derrière lui est grand.*
10. *Tout dodécaèdre est plus petit qu'un tétraèdre.*

2. Chargez le fichier `Pierce.wld`. Vérifiez que toutes vos formules sont vraies.

3. Chargez le fichier `Claire.wld`. Vérifiez que seules les formules 3, 5, 7, 9 sont vraies (sinon il y une erreur dans vos formules que vous corrigerez).

4. Chargez le fichier `Leibniz.wld`. Vérifiez que seules les formules 5, 6, 8, 10 sont vraies (sinon il y une erreur dans vos formules que vous corrigerez).

5. Chargez le fichier `Ron.wld`. Vérifiez que seules les formules 2, 3, 4, 5, 8 sont vraies. ♢

Exercice 13 1. Créez un fichier `exo13.sen` où vous mettrez des formules traduisant les assertions suivantes :

1. *Tout tétraèdre est devant tout dodécaèdre.*
2. *Aucun dodécaèdre n'a quelque chose derrière lui.*

3. *Aucun tétraèdre n'a la même taille qu'un cube.*
4. *Tout dodécaèdre a la même taille qu'un cube.*
5. *Tout objet entre deux tétraèdres est un petit cube.*
6. *Tout cube est entre deux objets.*
7. *Tout cube qui a quelque chose derrière lui est petit.*
8. *Tout dodécaèdre qui n'a rien à sa droite est petit.*
9. *Tout dodécaèdre qui n'a rien à sa droite a quelque chose à sa gauche.*
10. *Tout dodécaèdre à gauche d'un cube est grand.*

2. Chargez le fichier `Bolzano.wld`. Vérifiez que toutes vos formules sont vraies.

3. Chargez le fichier `Ron.wld`. Vérifiez que seules les formules 4, 5, 8, 9, 10 sont vraies (sinon il y une erreur dans vos formules que vous corrigerez).

4. Chargez le fichier `Claire.wld`. Vérifiez que seules les formules 1, 3, 5, 7, 9, 10 sont vraies(sinon il y une erreur dans vos formules que vous corrigerez).

5. Chargez le fichier `Peano.wld`. Vérifiez que seules les formules 8 et 9 sont vraies (sinon il y une erreur dans vos formules que vous corrigerez). ♢

EXERCICE 14 1. Créer un fichier `exo14.sen` qui contienne des formules traduisant les propriétés suivantes :

1. *Seuls les grands objets n'ont rien devant eux.*
2. *Si un cube a quelque chose devant lui, alors il est petit*
3. *Tout cube qui est derrière un dodécaèdre est aussi plus petit que ce dodécaèdre.*
4. *Si* **e** *est entre deux objets, alors ils sont tous les deux petits.*
5. *Si un tétraèdre est entre deux objets, alors ils sont tous les deux petits.*
6. *Tout dodécaèdre est au moins aussi grand que tout cube.*
7. *Si un cube est à droite d'un dodécaèdre mais pas derrière lui, alors il est aussi grand que le dodécaèdre.*
8. *Aucun cube n'ayant rien à sa gauche n'est entre deux cubes.*
9. *Les seuls grands cubes sont* **b** *et* **c**.
10. *Au plus* **b** *et* **c** *sont des grands cubes.*

2. Chargez le fichier `Ron.wld`. Vérifiez que toutes vos formules sont vraies (attention, il y a des objets cachés).

3. Chargez le fichier `Bolzano.wld`. Vérifiez que seules les formules 3, 8, 10 sont vraies (sinon il y une erreur dans vos formules que vous corrigerez).

4. Chargez le fichier `Wittgenstein.wld`. Vérifiez que seules les formules 5, 7, 8 sont vraies. ♢

EXERCICE 15 Ouvrez `Bozo2.sen` et `Leibniz.wld`. La plupart des expressions de ce fichier ne sont pas des formules bien formées ; quelques unes sont des formules bien formées mais ne sont pas des formules closes (sentence). Lisez et examinez chaque expression, si ce n'est pas une formule bien formée, corrigez-la. Si ce n'est pas une sentence faites-en une sentence vraie, et ce en apportant le moins possible de changements. Si c'est une sentence fausse, essayez de la rendre vraie, ici aussi en apportant le moins possible de changements. Voyez si vous pouvez reconstituer ce qu'on cherchait initialement à dire. Enregistrez votre ensemble de sentences sous `exo15.sen`. ♢

EXERCICE 16 1. Créer un fichier `quant1.sen` qui contienne des formules traduisant les propriétés suivantes : (on interprète la phrase *le cube est grand* par : il y a exactement un cube, et il est grand ; cette phrase sera supposée fausse si par exemple il n'y a pas de cube, ou s'il y en a deux)

1. *Il y a au moins deux dodécaèdres.*
2. *Il y a au plus deux tétraèdres.*
3. *Il y a exactement deux cubes.*
4. *Il y a seulement trois objets qui ne sont pas petits.*
5. *Le petit tétraèdre n'a rien devant lui.*
6. *Le tétraèdre qui a quelque chose devant lui est grand.*

7. *Aucun dodécaèdre n'est derrière le grand cube.*
8. *Le cube moyen est à droite du grand cube.*
9. *Le seul objet n'ayant rien à sa droite est le cube moyen.*
10. *Le plus petit cube est moyen.*

2. Chargez le fichier `Peano.wld`. Vérifiez que toutes vos formules sont vraies.
3. Chargez le fichier `Bolzano.wld`. Vérifiez que seules les formules 1, 3, 7 sont vraies (sinon il y une erreur dans vos formules que vous corrigerez).
4. Chargez le fichier `Skolem.wld`. Vérifiez que seules les formules 5, 7 sont vraies.
5. Chargez le fichier `Montague.wld`. Vérifiez que seules les formules 2, 3, 5, 7, 10 sont vraies. ♢

Dans les deux exercices suivants, nous dirons qu'une formule est T-valide *si toute interprétation où les prédicats* `Cube,... , Between` *sont fixés comme dans Tarski's world est un modèle de cette formule. Noter qu'on ne fixe pas l'interprétation des six constantes.*

Exercice 17 1. Chargez les fichiers `Carnap.sen` et `Bolzano.wld`. Traduisez les formules en français et vérifiez qu'elles sont vraies dans `Bolzano.wld`.

2. Quelles sont les formules qui sont T-valides ? (Il y en a cinq). Pour chaque formule qui n'est pas T-valide construire un monde qui falsifie la formule.
3. Quelle formule est une tautologie ?
4. Pour chaque formule qui est T-valide mais qui n'est pas une tautologie, dites comment changer l'interprétation des prédicats pour falsifier la formule. ♢

Exercice 18 1. Chargez le fichier `Post.sen`.

2. Quelles sont les formules qui sont T-valides ?
3. Pour chaque formule qui n'est pas T-valide essayez de construire un monde qui falsifie la formule.
4. Y a-t-il des formules non T-valides mais pour lesquelles vous ne pouvez pas construire de contrexemple à l'aide de Tarski's world à cause des contraintes sur les mondes de Tarski's world ? ♢

Exercice 19 Traduire l'ensemble d'énoncés suivants en formules et vérifiez que l'ensemble de formules obtenu est satisfaisable en construisant un monde `monde.wld` qui en est un modèle.

1. *Tout cube est à gauche de tout tétraèdre.*
2. *Il n'y a pas de dodécaèdre.*
3. *Il y a exactement quatre cubes.*
4. *Il y a exactement quatre tétraèdres.*
5. *Aucun tétraèdre n'est grand.*
6. *Aucun objet n'est plus grand que tous les objets à sa droite.*
7. *Un objet est à gauche d'un autre objet uniquement si le second est derrière le premier.* ♢

Exercice 20 1. Chargez le fichier `Padoa.sen`. Il contient quatre formules, trois quelconques d'entre elles sont satisfaisables, mais pas les quatre. Construire quatre mondes `123.wld`, `124.wld`, `134.wld`, `234.wld` qui sont modèles des sous-ensembles de trois formules.

2. Réinterprétez les prédicats `Tet` et `Dodec` pour que la formule 3 devienne vraie dans `124.wld`.
3. Réinterprétez le prédicat `Between` pour que la formule 4 devienne vraie dans `123.wld`. ♢

Exercice 21 Cet exercice explore les limites posées par le nombre de variables qui est de six dans Tarski's world.

1. Traduire *Il y a au moins deux objets*. Combien de variables avez-vous utilisées ?
2. Traduire *Il y a au moins trois objets*. Combien de variables avez-vous utilisées ? On peut montrer qu'on ne peut pas exprimer *Il y a au moins sept objets* en n'utilisant que les six variables et le symbole =.

3. Chargez les fichiers `Robinson.sen` et `Robinson.wld`. Il y a six cubes, vérifiez la validité de la formule. Ajoutez un petit cube à droite des autres et revérifiez. Jouez le jeu en prétendant (faussement) que la formule est fausse. Observez la stratégie de Tarski pour choisir un objet pour la variable x. Pouvez-vous en déduire comment on a réutilisé les variables ?

4. Supprimez un des cubes, et rejouez en prétendant que la formule est vraie. Voyez-vous pourquoi on ne peut pas gagner ?

5. Dans la question 2), on a dit qu'il était impossible d'exprimer l'existence de 7 objets avec 6 variables ; ici on exprime l'existence de 7 objets avec 2 variables. Pouvez-vous expliquer ce paradoxe apparent ? (Indication : faites tourner la figure de 90^o et réévaluez la formule de Robinson. Reste-t-elle vraie ?)

6. Ecrivez une formule qui dit qu'il y a au moins 4 objets, les uns devant les autres, en utilisant uniquement les variables x et y. $\diamondsuit$

Exercice 22 On définit un ordre de complexité sur les figures en combinant *lexicographiquement* les deux ordres suivants :

- une figure a est plus complexe qu'une figure b si a contient plus de faces que b ;
- une figure a est plus complexe qu'une figure b si a est plus grande que b.

Ainsi une pyramide est moins complexe qu'un cube, un petit cube est moins complexe qu'un grand cube et une grande pyramide est moins complexe qu'un cube moyen, qui est lui-même moins complexe qu'un petit dodécaèdre.

1. Construire un monde contenant 6 figures rangées de gauche à droite par ordre croissant de complexité.
2. Saisir un ensemble de formules décrivant la propriété de rangement de ce monde.
3. Vérifier. $\diamondsuit$

Exercice 23

1. Charger l'ensemble de formules du fichier `austin.sen`
2. Réaliser un monde qui satisfasse toutes les formules $\diamondsuit$

Exercice 24

1. Charger le fichier `montague.sen`
2. Compléter les formules
3. Réaliser un monde pour les formules obtenues $\diamondsuit$

Exercice 25

1. Créer un monde contenant au moins une figure de chaque espèce.
2. Saisir les deux formules

$$\forall x(Tet(x) \lor Cube(x) \lor Dodec(x))$$
$$\forall x(Tet(x) \to (Cube(x) \to Dodec(x)))$$

3. Vérifier automatiquement puis interactivement (bouton `Game`) la validité de ces formules.

Variante : construire un monde avec *une seule figure* qui valide la formule

$$\forall x(Tet(x) \to (Cube(x) \to Dodec(x)))$$

Est-ce le seul monde validant cette formule ? $\diamondsuit$

Exercice 26

1. Saisir la formule

$$\forall x((Tet(x) \lor Cube(x)) \to Dodec(x))$$

2. Construire un monde à *une seule figure* validant cette formule.

3. Vérifier interactivement la validité. ♢

EXERCICE 27
1. Charger le fichier `morecnf.sen`
2. Donner les formes normales conjontives de chacune des formules
3. Réaliser un monde pour les formules obtenues ♢

EXERCICE 28 Existence et unicité.
1. Charger le fichier `russel.sen`
2. Réaliser un monde pour les formules obtenues ♢

EXERCICE 29
1. Charger le fichier `skolem.sen`
2. Réaliser un monde pour les formules obtenues ♢

EXERCICE 30
1. Charger le fichier `post.sen`
2. Réaliser un monde pour les formules obtenues ♢

EXERCICE 31 Paradoxe apparent.
1. Charger le fichier `dodgson.sen`
2. Réaliser un monde pour les formules obtenues ♢

EXERCICE 32
1. Charger le fichier `bozo.sen`
2. Dire (et vérifier) quelles formules sont correctes et lesquelles ne le sont pas.
3. Corriger les formules incorrectes
4. Réaliser (si possible) un monde pour les formules obtenues ♢

Exercices suivants

enjoy yourself on your own !