TD14 - Déduction naturelle (2)

Rappels de logique du premier ordre

La *logique du premier ordre* diffère de la *logique propositionnelle* en ce qu'elle ne manipule pas uniquement des propositions, mais également des objets, appelés *termes*, qui ne sont pas eux-mêmes de nature logique, mais à propos desquels on peut exprimer des propriétés, par l'intermédiaire de *prédicats*. Les *formules logiques du premier ordre* combinent ces propriétés à l'aide des connecteurs déjà connus, mais utilisent également de nouveaux symboles, appelés *quantificateurs*, permettant d'exprimer que certains énoncés s'appliquent à tout ou partie des termes.

Domaine, termes et prédicats

Considérons un tableau a de 4 entiers $a[0]$, $a[1]$, $a[2]$ et $a[3]$. Ce tableau, les valeurs qu'il contient et les indices de ses différentes cases constituent le *domaine* dont les formules vont parler. La notation $a[i]$ peut être interprétée comme l'application d'un *symbole de fonction* $\texttt{pos}$ à a et à i de sorte que $\texttt{pos}(a, i)$ renvoie $a[i]$. Considérons un premier *symbole de prédicat* noté $\texttt{even}$, d'arité 1, qui exprime la parité d'un entier : $\texttt{even}(x)$ se traduit par x *est pair*. Un deuxième *symbole de prédicat* noté $\texttt{leq}$, d'arité 2, désigne la relation d'ordre *est inférieur ou égal à* : $\texttt{leq}(x, y)$ se traduit par x *est inférieur ou égal à* y. Ces prédicats sont définis indépendamment de toute valeur de vérité, en appliquant un symbole à des arguments.

Définition 1

Un *prédicat* est un énoncé dépendant d'une ou plusieurs variables et dont la valeur de vérité dépend du choix des valeurs de vérités de ses variables. Le nombre de ses variables est son *arité*.

Les symboles précédents permettent d'écrire des expressions de la forme $\texttt{even}(\texttt{pos}(a, 0))$, $\texttt{leq}(\texttt{pos}(a, 2), \texttt{pos}(a, 0))$ qui expriment de propriétés élémentaires à propos des éléments du tableau. Ce sont des formules logiques atomiques qu'on peut combiner à l'aide de connecteurs logiques. Par exemple, la *formule* suivante exprime que tous les éléments du tableau a sont pairs.

$$\texttt{even}(\texttt{pos}(a, 0)) \land \texttt{even}(\texttt{pos}(a, 1)) \land \texttt{even}(\texttt{pos}(a, 2)) \land \texttt{even}(\texttt{pos}(a, 3))$$

Avec seulement quatre éléments dans le tableau, l'écriture de la formule est aisée. Avec un plus grand nombre d'éléments, on peut lui préférer une notation plus compacte comme la suivante, exprimant une conjonction sur un ensemble d'indices.

$$\bigwedge_{i=0}^{3} \texttt{even}(\texttt{pos}(a, i))$$

Mais il ne s'agit ici que d'une simple ré-écriture : la formule est toujours une grande conjonction. Pour traduire de manière plus directe qu'être pair est une *propriété universelle* des éléments du tableau, on introduit un nouvel élément : le *quantificateur universel* $\forall$. On peut ainsi écrire une nouvelle version de la formule.

$$\forall i \in [0, 3]\ \texttt{even}(\texttt{pos}(a, i))$$

Une autre écriture de cette même formule explicite le lien entre i et l'intervalle $[0, 3]$.

$$\forall i\ ((\texttt{leq}(0, i) \land \texttt{leq}(i, 3)) \to \texttt{even}(\texttt{pos}(a, i)))$$

Cette dernière comporte une phrase $(\texttt{leq}(0, i) \land \texttt{leq}(i, 3)) \to \texttt{even}(\texttt{pos}(a, i))$ à propos d'une variable i désignant un élément indéterminé du domaine et est considérée comme vraie dès que tous les éléments du domaine valident effectivement cette phrase. Dans cette formule, on distingue :

- les *symboles de constantes* 0, 3, a ;
- le *symbole de fonction* $\texttt{pos}$;
- les *symboles de prédicats* $\texttt{leq}$, $\texttt{even}$;
- le *symbole de variable* i ;
- les *connecteurs* $\land$, $\to$;
- le *quantificateur* $\forall$.

Dans la suite, on définit un *domaine* en fournissant un ensemble pour chacun de ces types d'objets.

Définition 2

On désigne par :

- X l'ensemble infini dénombrable des *symboles de variables*.
- $\mathcal{F}$ l'ensemble des *symboles de fonctions*, c'est-à-dire les désignations de fonctions d'arité quelconque.
- $\mathcal{P}$ l'ensemble non vide de *symboles de prédicats*.

Ces trois ensembles sont supposés disjoints.

On note $\mathcal{F}_k$ l'ensemble des symboles de fonctions d'arité k. Les *symboles de constantes* peuvent être vus comme des fonctions d'arité 0, c'est-à-dire des éléments de $\mathcal{F}_0$. On note $\mathcal{P}_k$ l'ensemble des symboles de prédicats d'arité k. Les éléments de $\mathcal{P}_0$ sont appelés *propositions* et jouent le rôle des variables propositionnelles de la logique des propositions. On a :

$$\mathcal{F} = \bigcup_{k\in\mathbb{N}} \mathcal{F}_k \qquad \mathcal{P} = \bigcup_{k\in\mathbb{N}} \mathcal{P}_k$$

Exemple 1

La formule

$$\forall i.\ ((\texttt{leq}(0,i) \wedge \texttt{leq}(i,3)) \to \texttt{even}(\texttt{pos}(a,i)))$$

fait intervenir les ensembles suivants.

- $X = \{i\}$;
- $\mathcal{F} = \mathcal{F}_0 \cup \mathcal{F}_2$ avec $\mathcal{F}_0 = \{0, 3, a\}$ et $\mathcal{F}_2 = \{\texttt{pos}\}$
- $\mathcal{P} = \mathcal{P}_1 \cup \mathcal{P}_2$ avec $\mathcal{P}_1 = \{\texttt{even}\}$ et $\mathcal{P}_2 = \{\texttt{leq}\}$

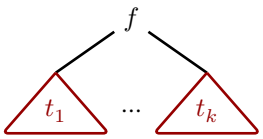
Les *termes* sont les expressions syntaxiques qui désignent les objets d'un discours.

Définition 3 – terme

Étant donnée une *signature* $(X, \mathcal{F})$, l'ensemble des *termes* est défini par induction comme suite.

- Tout symbole de variable de X est un terme.
- Tout symbole de constante de $\mathcal{F}_0$ est un terme.
- Si $t_1, \dots, t_k$ sont des termes et si $f \in \mathcal{F}_k$ alors $f(t_1, \dots, t_k)$ est un terme.

Il est possible de définir directement les *termes* sur $\mathcal{F}$ et X à l'aide d'arbres. Si $x \in X$, $t_1, \dots, t_k$ sont des termes et si $f \in \mathcal{F}_k$, c'est l'ensemble des arbres suivants.

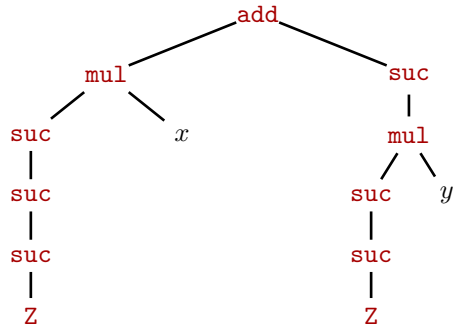
x ou 

Exemple 2

Soit le symbole de constante Z (arité 0), le symbole de fonction succ (arité 1) qui représente la fonction *successeur*, les symboles des fonction add et mul d'arité 2 qui représentent les fonctions d'*addition* et de *multiplication*. Ainsi $\mathcal{F} = \{\texttt{Z}^{(0)}, \texttt{suc}^{(1)}, \texttt{add}^{(2)}, \texttt{mul}^{(2)}\}$. L'exposant à coté de chaque symbole est son arité. Alors $(X, \mathcal{F})$ définit une signature sur les entiers naturels.
Si $x, y \in X$, l'expression suivante est un terme sur $(X, \mathcal{F})$:

$$\texttt{add}(\texttt{mul}(\texttt{suc}(\texttt{suc}(\texttt{suc}(\texttt{Z}))), x), \texttt{suc}(\texttt{mul}(\texttt{suc}(\texttt{suc}(\texttt{Z})), y)))$$

qui représente l'expression mathématique $(3x + (2y + 1))$. On peut lui associer l'arbre suivant.



Formules du premier ordre

Les prédicats, appliqués à des éléments concrets ou non du domaine, sont des formules atomiques pouvant servir de base à la construction des formules.

Définition 4 – formule atomique

Une *formule atomique* sur $(X, \mathcal{F}, \mathcal{P})$ est la donnée d'une expression de la forme $p(t_1, \dots, t_k)$ où $p \in \mathcal{P}_k$ et $t_1, \dots, t_k$ sont des termes.

En plus des connecteurs déjà connus, la logique du premier ordre utilise deux constructions supplémentaires, les quantificateurs, qui permettent de préciser la manière dont doivent être comprises les variables faisant référence à des éléments du domaine.

Définition 5 – quantificateurs

En logique du premier ordre, on a deux *quantificateurs* :

- le *quantificateur universel*, noté $\forall$, exprime qu'une propriété est vraie pour tous les éléments du domaine ;
- le *quantificateur existentiel*, noté $\exists$, qui décrit l'existence d'au moins un objet qui a une certaine propriété.

Dans une formule logique, les quantificateurs sont prioritaires sur les connecteurs logiques.

Définition 6 – formule du premier ordre

Une *formule du premier ordre* sur $(X, \mathcal{F}, \mathcal{P})$ est définie inductivement par :

- toute formule atomique sur $(X, \mathcal{F}, \mathcal{P})$;
- si φ est une formule alors $\neg\varphi$ est une formule ;
- si φ et ψ sont deux formules alors $(\varphi \wedge \psi)$, $(\varphi \vee \psi)$, $(\varphi \to \psi)$ sont des formules ;
- si $x \in X$ et si φ est une formule alors $(\forall x.\varphi)$ et $(\exists x.\varphi)$ sont des formules.

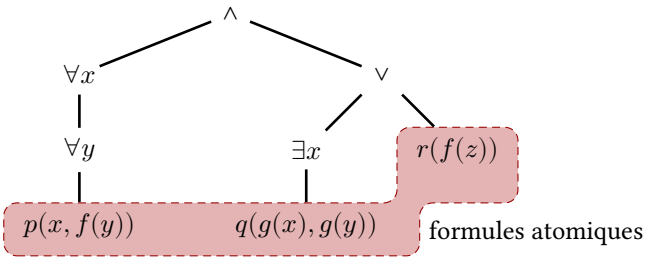
Toute formule logique peut être représentée par un arbre. Les éléments situés aux feuilles sont des formules atomiques.

Exemple 3

La formule :

$$\varphi = (\forall x.\forall y.p(x, f(y))) \wedge ((\exists x.q(g(x), g(y))) \vee r(f(z)))$$

peut être représenté par l'arbre suivant.



Parmi les variables de X présentes dans une formule, certaines apparaissent à la suite de quantificateurs, d'autres sont isolées.

Définition 7 – variables libres et liées

Un variable $x \in X$ qui apparaît à la suite d'un quantificateur est dite *liée*. Sinon, elle est dite *libre*.

On peut faire un parallèle entre variables liées et libres en logique et variables *locales* et *globales* en programmation.

Dans l'exemple 2, la première formule atomique $p(x, f(y))$ contient deux variables liées x et y. La formule est précédée de $\forall x$ et de $\forall y$ qui n'agissent que sur cette formule atomique. On dit que leur *portée* est $p(x, f(y))$. La formule atomique $q(g(x), g(y))$ contient une variable liée x et une variable libre y. La formule atomique $r(f(z))$ contient une variable libre z.

Définition 8 – portée

Dans une formule $\forall x.\varphi$ ou $\exists x.\varphi$, la *portée* de x est la formule φ.

Substitution d'une variable

Certaines formules logiques présentant un caractère universel, on peut souhaiter les utiliser pour traiter des cas particuliers. Ceci est possible en substituant, dans une formule, un *variable libre* par un terme. Par exemple, on peut substituer la variable libre x par $f(z)$, où $f \in \mathcal{F}_1$ dans la formule suivante où $p \in \mathcal{P}_2$: $\exists y.p(x, y)$. Le résultat de la substitution est la formule : $\exists y.p(f(z), y)$. On note :

$$(\exists y.p(x, y))^{\{x \leftarrow f(z)\}} = (\exists y.p(f(z), y))$$

Il est important que la variable substituée soit libre.

- Tenter une substitution sur une variable liée est un non sens.
- Si une formule comporte des symboles de variables liées qu'on souhaite utiliser dans une substitution, il faut procéder en deux temps : *renommer* les variables liées qui le nécessitent puis *substituer*.

Dans l'exemple précédent, il n'y avait aucune difficulté pour effectuer la substitution : x est une variable libre, y est une variable liée et on substitue x par $f(x)$ qui ne contient pas le symbole y. Mais on aurait pu vouloir réaliser la substitution de x par $f(y)$, opération qui aurait d'abord nécessité de *renommer* la variable liée y présente dans la formule pour éviter toute ambiguité. Ainsi :

$$\begin{aligned}(\exists y.p(x, y))^{\{x \leftarrow f(y)\}} &= (\exists z.p(x, z))^{\{x \leftarrow f(y)\}} && \text{(renommage)}\\ &= (\exists z.p(f(y), z)) && \text{(substitution)}\end{aligned}$$

Définition 9 – substitution d'une variable libre

Soit φ et ψ deux formules logiques, t un terme et x une variable libre susceptible d'être présente dans les deux formules.

$$\begin{aligned}p(t_1, \dots, t_k)^{\{x \leftarrow t\}} &= p(t_1^{\{x \leftarrow t\}}, \dots, t_k^{\{x \leftarrow t\}}) && (t_i \text{ termes}, p \in \mathcal{P}_k)\\ (\neg\varphi)^{\{x \leftarrow t\}} &= \neg(\varphi^{\{x \leftarrow t\}})\\ (\varphi \diamond \psi)^{\{x \leftarrow t\}} &= \varphi^{\{x \leftarrow t\}} \diamond \psi^{\{x \leftarrow t\}} && (\diamond \text{ connecteur binaire})\\ (\forall y.\varphi)^{\{x \leftarrow t\}} &= \forall y.(\varphi^{\{x \leftarrow t\}})\\ (\exists y.\varphi)^{\{x \leftarrow t\}} &= \exists y.(\varphi^{\{x \leftarrow t\}})\end{aligned}$$

Sémantique des quantificateurs.

Donner formellement une sémantique aux formules du premier ordre est assez technique, et hors programme en MP2I/MPI. On se basera sur les interprétations intuitives suivantes.

Quantification universelle. Une formule $\forall x.\varphi$ est considérée comme valide si φ est vraie en tous les points du domaine, c'est-à-dire si $\varphi^{\{x \leftarrow v\}}$ est vraie pour toutes les valeurs v que peut représenter la variable du premier ordre x.

Quantification existentielle. Une formule $\exists x.\varphi$ est considérée comme valide si φ est vraie en au moins un point du domaine, c'est-à-dire s'il y a au moins une valeur v telle que $\varphi^{\{x \leftarrow v\}}$.

On peut comprendre $\forall x.\varphi$ comme la conjonction gigantesque, voire infinie, de toutes les instanciations possibles de la formule φ. Inversement, on peut voir $\exists x.\varphi$ comme la disjonction de ces mêmes instanciations. En conséquence, les lois de de Morgan, qui liaient conjonction, disjonction et négation, ont encore un équivalent avec les quantificateurs.

$$\begin{array}{rcl} \neg(\forall x.\varphi) & \equiv & \exists x.\neg\varphi \\ \neg(\exists x.\varphi) & \equiv & \forall x.\neg\varphi \end{array}$$

Exercice 1

On donne la formule logique φ suivante.

$$\forall x.\forall y.\exists z.\,(\neg(x<y)\vee((x<z)\wedge(z<y)))$$

Question 1. Quel est son arbre de syntaxe abstraite ?

Question 2. Quelles sont ses formules atomiques ?

Question 3. Comporte-t-elle des variables liées ? des variables libres ? lesquelles ?

Question 4. Quels sont ses termes ?

Exercice 2

Question 1. Proposer une formule φ et dessiner son arbre de syntaxe abstraite où φ est telle que :

- φ ne comporte que trois symboles de variable notés x, y, z tels que :
  - x admet uniquement deux occurrences libres et un occurrents liée dans φ ;
  - y admet uniquement une occurrence liées par le quantificateur $\forall$ et une occurrence liées par le quantificateur $\exists$ dans φ
- φ ne comporte que deux symboles de fonction : $k\in\mathcal{F}_0$ (constante) et $f\in\mathcal{F}_2$ (arité 2), chacun de ces symboles pouvant apparaître plusieurs dans la formule ;
- φ contient exactement trois formules atomiques.

Question 2. Une *formule close* est une formule ψ qui ne contient aucune variable libre. Si une formule ψ contient les variables libres $x_1,\dots,x_n$, il est possible de construire une formule close en quantifiant chacune des variables libres : $\forall x_1.\dots\forall x_n.\psi$. Cette dernière formule est appelée *clôture universelle* de ψ.
□ **2.1.** Proposer une clôture universelle noté φ' de φ définie à la question précédente.
□ **2.2.** Renommer certains symboles de variable de φ' pour obtenir une formule φ'' logiquement équivalente à φ et dans laquelle les quantificateurs portent sur des symboles de variable différents.

Exercice 3

Soit x, y, z, w des symboles de variable, f, g, h des symboles de fonction d'arité respective 2, 3, 1 et p, q des symboles de prédicat d'arité 2. Calculer $\varphi^{\{x\leftarrow f(y,z)\}}$ lorsque :

Question 1. $\varphi=\forall x.\exists z.p(f(y,z),x)$

Question 2. $\varphi=\forall y.\exists z.p(g(y,z,x),x)$

Question 3. $\varphi=(\forall w.p(h(w),x))\to(\forall x.\forall z.q(x,f(z,z)))$

Exercice 4

On considère le problème de la recherche d'un motif m dans un texte t, et on se donne la fonction C suivante. La fonction renvoie l'indice de début d'une occurrence de m dans t s'il en existe une, et -1 sinon.

```
int search(char *m, char *t) {
  int lm = strlen(m), lt = strlen(t);
  for (int i = 0; i <= lt - lm; i++) {
    int j = 0;
    while (j < lm) {
      if (m[j] != t[i+j]) break;
      j++;
    }
    if (j == lm) return i;
  }
  return -1;
}
```

Question 1. Donner des formules de logique du premier ordre exprimant la spécification du problème. Dans ces formules, on dénotera le résultat renvoyé par r.

Question 2. Donner des formules de logique du premier ordre exprimant les invariants des deux boucles.

Exercice 5

On considère la fonction `longest_repetition` suivante.

```
int longest_repetition(char *t) {
  int lt = strlen(t);
  int i = 0, r = 0;
  while (i < lt) {
    int k = 1;
    while (i + k < lt && t[i + k] == t[i]) { k++; }
    if (k > r) { r = k; }
    i += k;
  }
  return r;
}
```

Donner des invariants pour ses deux boucles, sous la forme de formules de logique du premier ordre.

Exercice 6

Donner des *dérivations* pour les séquents suivants.

Question 1. $(\forall x\ \varphi) \vdash (\neg\exists x\ \neg\varphi)$

Question 2. $\vdash \forall x\ \forall y\ (x = y) \to (y = x)$

Question 3. $\vdash \forall x\ \forall y\ (\exists z\ (x = z) \wedge (y = z)) \to (x = y)$