Modélisation et Programmation – Sessions 1 – 6/11/2017

Prénom : Nom : Groupe :

Les exercices sont indépendants les uns des autres. Lisez le sujet avant de commencer. Traitez les exercices dans votre ordre de préférence.
Dans tous les exercices, vous veillerez à justifier vos résultats, par exemple en faisant apparaître le nom des règles utilisées, il en sera tenu compte dans la notation.

Exercice 1 Modéliser en logique des propositions, sous la forme de formules bien formées, l'énoncé suivant : « Lorsque je dors, je fais des rêves ou des cauchemars. Je ne suis reposé que lorsque j'ai fait des rêves. Or je suis reposé donc je n'ai pas fait de cauchemars. »

Exercice 2 Modéliser en logique des prédicats, sous la forme de formules bien formées, l'énoncé suivant : « Tous les hommes sont mortels, or Socrate est un homme donc Socrate est mortel. » (citation d'Aristote pour introduire les Syllogismes)

Exercice 3 Soient A et B des variables propositionnelles, montrer, **en utilisant les propriétés des opérateurs de logique des propositions** rappelées en annexe A, que la formule suivante est une tautologie :

$$((\neg A) \vee B) \rightarrow (\neg((\neg B) \wedge A))$$

Exercice 4 Soient A et B des variables propositionnelles, montrer, **en utilisant la déduction naturelle constructive (c'est-à-dire sans les règles du tiers-exclu (TE dans le résumé de cours) et de l'absurde classique (A dans le résumé de cours))**, que la formule suivante est une tautologie :

$$((\neg A) \vee B) \rightarrow (\neg((\neg B) \wedge A))$$

Exercice 5 Soient A et B des variables propositionnelles, montrer, **en utilisant la déduction naturelle classique (y compris les règles du tiers-exclu ou de l'absurde classique)**, que la formule suivante est une tautologie :

$$(A \rightarrow B) \rightarrow ((\neg A) \vee B)$$

Exercice 6 Prouver la correction totale du triplet de Hoare suivant pour un programme calculant le logarithme en base 2 entier R d'un entier strictement positif N (la partie entière du logarithme en base 2 de N). Nous vous suggérons d'exploiter $X \leq 2 \times N \wedge X = 2^{R}$ comme invariant et $2 \times N - X$ comme variant. Vous compléterez l'invariant si nécessaire pour construire la preuve. Vous détaillerez sur la page suivante la preuve des obligations obtenues.

$\{N > 0\}$

```
R := 1;

X := 2;

while (X <= N) do

        R := R + 1;

        X := 2 * X

od;

R := R - 1
```

$\{R \geq 0 \wedge 2^{R} \leq N < 2^{R+1}\}$

Exercice 7 Nous considérons la spécification des listes munie des équations étudiées en cours et travaux dirigés :

(a) $\forall l \in \texttt{liste}(A).\ \texttt{append}(\texttt{Nil},l) = l$
(b) $\forall t \in A.\forall l, q \in \texttt{liste}(A).\ \texttt{append}(\texttt{Cons}(t,\ q),l) = \texttt{Cons}(t,\ \texttt{append}(q,l))$
(c) $\forall l \in \texttt{liste}(A).\ \texttt{append}(l,\texttt{Nil}) = l$
(d) $\forall l_1, l_2, l_3 \in \texttt{liste}(A).\ \texttt{append}(l_1,\texttt{append}(l_2,l_3)) = \texttt{append}(\texttt{append}(l_1,l_2),l_3)$
(e) $\text{rev}(\texttt{Nil}) = \texttt{Nil}$
(f) $\forall t \in A.\forall q \in \texttt{liste}(A).\ \texttt{rev}(\texttt{Cons}(t,\ q)) = \texttt{append}(\texttt{rev}(q),\texttt{Cons}(t,\ \texttt{Nil}))$
(g) $\forall l_1, l_2 \in \texttt{liste}(A).\ \texttt{rev}(\texttt{append}(l_1,l_2)) = \texttt{append}(\texttt{rev}(l_2),\texttt{rev}(l_1))$
(h) $\forall l \in \texttt{liste}(A).\ \texttt{rev}(\texttt{rev}(l)) = l$

Nous complétons cette spécification par la fonction $\texttt{mapf}(l)$ qui applique l'opérateur `f` sur chaque élément de la liste l et renvoie une liste contenant les résultats de l'appel de `f`.
Le comportement de cette fonction peut être modélisé par les équations suivantes :

(i) $\texttt{mapf}(\texttt{Nil}) = \texttt{Nil}$
(j) $\forall t \in E.\forall q \in \texttt{liste}(E).\ \texttt{mapf}(\texttt{Cons}(t,\ q)) = \texttt{Cons}(\texttt{f}(t),\ \texttt{mapf}(q))$

En utilisant ces équations comme des axiomes et en introduisant si nécessaire des lemmes intermédiaires, montrer que ces opérations vérifient la propriété suivante :

$$\text{(k) } \forall l \in \texttt{liste}(E).\ \ \texttt{mapf}(\texttt{rev}(l)) = \texttt{rev}(\texttt{mapf}(l))$$

Exercice 8 La représentation canonique des dates sous Unix est composée de :

- Un préfixe optionnel représentant le numéro du siècle (soit `19`, soit `20`).
- Le numéro de l'année dans le siècle indiqué par 2 chiffres (le préfixe `0` est obligatoire si le numéro est inférieur à dix).
- Le numéro du mois dans l'année compris entre `1` et `12` (le préfixe `0` est interdit si le numéro est inférieur à dix).
- Le numéro du jour dans le mois compris entre `1` et `31` (le préfixe `0` est interdit si le numéro est inférieur à dix).

1. Donner une expression régulière modélisant ce langage.

Exercice 9 Le langage LISP (dont sont dérivés de nombreux autres langages comme Clojure) est un langage conçu pour écrire des programmes en Intelligence Artificielle Symbolique avec une syntaxe très simple. Un programme est une S-expression. Une S-expression est :

- soit un identificateur représentant un symbole
- soit le symbole `nil` également noté `()`
- soit une paire pointée composée du caractère parenthèse ouvrante `(` suivie d'une S-expression suivie du caractère point `.` suivie d'une S-expression, suivie du caractère parenthèse fermante `)`
- soit une liste de n S-expressions notée `(` $\mathtt{s}_1$ `...` $\mathtt{s}_n$ `)` équivalente à `(`$\mathtt{s}_1$`.(`$\mathtt{s}_2$`.(... . nil)...))`

Voici un exemple de programme LISP :

```
((a . (b (() . d) e)) . (f . nil))
```

1. Peut on modéliser ce langage par une expression régulière ? Pourquoi ?

2. Donner une grammaire au format graphique de Conway modélisant ce langage.

3. Donner une grammaire au format des règles de production modélisant ce langage.

4. Donner l'arbre de dérivation de l'exemple précédent pour la grammaire au format règles de production que vous venez de proposer.

A Logique des propositions : Vision sémantique

A.1 Tables de vérité

La sémantique de $\top$ (respectivement $\bot$) est représenté par V (respectivement F).

A	B	$\neg A$	$A \wedge B$	$A \vee B$	$A \rightarrow B$	$A \leftrightarrow B$
F	F	V	F	F	V	V
F	V	V	F	V	V	F
V	F	F	F	V	F	F
V	V	F	V	V	V	V

A.2 Relation d'équivalence sémantique des formules bien formées

$\varphi = \psi$ si et seulement si les deux formules bien formées φ et ψ ont la même table de vérité.

	$A \rightarrow B = \neg A \vee B$ $A \leftrightarrow B = (A \rightarrow B)\ \wedge\ (B \rightarrow A)$
Idempotence	$A \wedge A = A$ $A \vee A = A$
	$A \wedge \neg A = \bot$ $A \vee \neg A = \top$ $A \wedge \bot = \bot$ $A \wedge \top = A$ $A \vee \bot = A$ $A \vee \top = \top$ $\neg\neg A = A$
Commutativité	$A \wedge B = B \wedge A$ $A \vee B = B \vee A$
Associativité	$(A \wedge B) \wedge C = A \wedge (B \wedge C)$ $(A \vee B) \vee C = A \vee (B \vee C)$
Distributivité	$A \wedge (B \vee C) = (A \wedge B) \vee (A \wedge C)$ $A \vee (B \wedge C) = (A \vee B) \wedge (A \vee C)$
De Morgan	$\neg(A \wedge B) = \neg A \vee \neg B$ $\neg(A \vee B) = \neg A \wedge \neg B$
Simplification	$A \vee (\neg A \wedge B) = A \vee B$ $A \wedge (\neg A \vee B) = A \wedge B$ $A \vee (A \wedge B) = A$ $A \wedge (A \vee B) = A$

B Logique des propositions : Vision syntaxique

B.1 Déduction naturelle constructive

Hypothèse	$\dfrac{}{\Gamma \cup \{\varphi\} \vdash \varphi}$ Hyp	
Opérateur	Introduction	Elimination
$\rightarrow$	$\dfrac{\Gamma \cup \{\varphi\} \vdash \psi}{\Gamma \vdash \varphi \rightarrow \psi} I_{\rightarrow}$	$\dfrac{\Gamma \vdash \varphi \rightarrow \psi \quad \Gamma \vdash \varphi}{\Gamma \vdash \psi} E_{\rightarrow}$
$\wedge$	$\dfrac{\Gamma \vdash \varphi \quad \Gamma \vdash \psi}{\Gamma \vdash \varphi \wedge \psi} I_{\wedge}$	$\dfrac{\Gamma \vdash \varphi \wedge \psi}{\Gamma \vdash \varphi} E_{\wedge}^{G} \quad \dfrac{\Gamma \vdash \varphi \wedge \psi}{\Gamma \vdash \psi} E_{\wedge}^{D}$
$\vee$	$\dfrac{\Gamma \vdash \varphi}{\Gamma \vdash \varphi \vee \psi} I_{\vee}^{G} \quad \dfrac{\Gamma \vdash \psi}{\Gamma \vdash \varphi \vee \psi} I_{\vee}^{D}$	$\dfrac{\Gamma \vdash \varphi \vee \psi \quad \Gamma \cup \{\varphi\} \vdash \chi \quad \Gamma \cup \{\psi\} \vdash \chi}{\Gamma \vdash \chi} E_{\vee}$
$\neg$	$\dfrac{\Gamma \cup \{\varphi\} \vdash \bot}{\Gamma \vdash \neg\varphi} I_{\neg}$	$\dfrac{\Gamma \vdash \varphi \quad \Gamma \vdash \neg\varphi}{\Gamma \vdash \bot} E_{\neg}$
$\bot$	$\dfrac{\Gamma \vdash \varphi \quad \Gamma \vdash \neg\varphi}{\Gamma \vdash \bot} I_{\bot}$	$\dfrac{\Gamma \vdash \bot}{\Gamma \vdash \varphi} E_{\bot}$

B.2 Déduction naturelle classique

Tiers-exclu	Preuve par l'absurde
$\dfrac{}{\Gamma \vdash \varphi \vee \neg\varphi}$ TE	$\dfrac{\Gamma \cup \{\neg\varphi\} \vdash \bot}{\Gamma \vdash \varphi}$ A

C Logique des prédicats : Vision sémantique

C.1 Relation d'équivalence sémantique des formules bien formées

$\varphi = \psi$ si et seulement si les deux formules bien formées φ et ψ ont la même sémantique, c'est à dire sont valides sémantiquement pour les mêmes modèles $(\forall \mathcal{M}, (\mathcal{M} \models \varphi) \leftrightarrow (\mathcal{M} \models \psi))$.

$\mathcal{U} \neq \emptyset$	$\forall x\ \varphi = \bigwedge_{x \in \mathcal{U}} \varphi$	$\exists x\ \varphi = \bigvee_{x \in \mathcal{U}} \varphi$
$\mathcal{U} = \emptyset$	$\forall x\ \varphi = \top$	$\exists x\ \varphi = \bot$

$$
\begin{array}{rl}
\forall x\ \varphi = \varphi & x \notin VL(\varphi) \wedge \mathcal{U} \neq \emptyset \\
\exists x\ \varphi = \varphi & x \notin VL(\varphi) \wedge \mathcal{U} \neq \emptyset \\
\forall x\ \varphi = \forall y\ [y/x]\,\varphi & y \text{ inutilisée} \\
\exists x\ \varphi = \exists y\ [y/x]\,\varphi & y \text{ inutilisée} \\
\forall x\ (\forall y\ \varphi) = \forall y\ (\forall x\ \varphi) & \\
\exists x\ (\exists y\ \varphi) = \exists y\ (\exists x\ \varphi) & \\
\neg(\forall x\ \varphi) = \exists x\ (\neg\varphi) & \\
\neg(\exists x\ \varphi) = \forall x\ (\neg\varphi) & \\
\forall x\ (\varphi \rightarrow \psi) = (\exists x\ \varphi) \rightarrow \psi & x \notin VL(\psi) \\
\forall x\ (\varphi \rightarrow \psi) = \varphi \rightarrow (\forall x\ \psi) & x \notin VL(\varphi) \\
\exists x\ (\varphi \rightarrow \psi) = (\forall x\ \varphi) \rightarrow \psi & x \notin VL(\psi) \wedge \mathcal{U} \neq \emptyset \\
\exists x\ (\varphi \rightarrow \psi) = \varphi \rightarrow (\exists x\ \psi) & x \notin VL(\varphi) \wedge \mathcal{U} \neq \emptyset \\
\forall x\ (\varphi \wedge \psi) = (\forall x\ \varphi) \wedge (\forall x\ \psi) & \\
\forall x\ (\varphi \vee \psi) = (\forall x\ \varphi) \vee \psi & x \notin VL(\psi) \\
\exists x\ (\varphi \vee \psi) = (\exists x\ \varphi) \vee (\exists x\ \psi) & \\
\exists x\ (\varphi \wedge \psi) = (\exists x\ \varphi) \wedge \psi & x \notin VL(\psi)
\end{array}
$$

D Logique des prédicats : Vision syntaxique

D.1 Déduction naturelle constructive

Opérateur	Introduction	Elimination
$\forall$	$\dfrac{\Gamma \vdash \varphi}{\Gamma \vdash \forall x\ \varphi}\ I_\forall\ (x \notin VL(\Gamma))$	$\dfrac{\Gamma \vdash \forall x\ \varphi}{\Gamma \vdash [t/x]\,\varphi}\ E_\forall$
$\exists$	$\dfrac{\Gamma \vdash [t/x]\,\varphi}{\Gamma \vdash \exists x\ \varphi}\ I_\exists$	$\dfrac{\Gamma \vdash \exists x\ \varphi}{\Gamma \vdash [f(\vec{x})/x]\,\varphi}\ E_\exists^{SK}\ (\vec{x} = VL(\Gamma) \cup VL(\exists x\ \varphi))$ $\dfrac{\Gamma \vdash \exists x\ \varphi \qquad \Gamma \cup \{\varphi\} \vdash \psi}{\Gamma \vdash \psi}\ E_\exists^{MP}\ (x \notin VL(\Gamma) \cup VL(\psi))$

E Logique de Floyd/Hoare

$$\dfrac{}{\{\psi\}\ skip\ \{\psi\}}\ \text{skip} \qquad \dfrac{}{\{[E/x]\,\psi\}\ x := E\ \{\psi\}}\ \text{assign}$$

$$\dfrac{\{\varphi\}\ P\ \{\chi\} \qquad \{\chi\}\ Q\ \{\psi\}}{\{\varphi\}\ P\ ;Q\ \{\psi\}}\ \text{sequence}$$

$$\dfrac{\{\varphi \wedge C\}\ P\ \{\psi\} \qquad \{\varphi \wedge \neg C\}\ Q\ \{\psi\}}{\{\varphi\}\ if\ C\ then\ P\ else\ Q\ fi\ \{\psi\}}\ \text{conditional}$$

$$\dfrac{\{\varphi \wedge C\}\ P\ \{\varphi\}}{\{\varphi\}\ while\ C\ invariant\ \varphi\ do\ P\ od\ \{\varphi \wedge \neg C\}}\ \text{partial loop}$$

$$\dfrac{\{\varphi \wedge C \wedge E \in \mathbb{N} \wedge V = E\}\ P\ \{\varphi \wedge E \in \mathbb{N} \wedge V > E\}}{\{\varphi \wedge E \in \mathbb{N}\}\ while\ C\ invariant\ \varphi\ variant\ E\ do\ P\ od\ \{\varphi \wedge \neg C\}}\ \text{total loop}$$

$$\dfrac{\varphi \to \chi \qquad \{\chi\}\ P\ \{\psi\}}{\{\varphi\}\ P\ \{\psi\}}\ \text{weaken} \qquad \dfrac{\{\varphi\}\ P\ \{\chi\} \qquad \chi \to \psi}{\{\varphi\}\ P\ \{\psi\}}\ \text{strenghten}$$

F Expressions régulières : Equivalence sémantique

L'opérateur de concaténation/juxtaposition . est implicite pour ne pas surcharger l'écriture des expressions : $e_1.e_2$ est notée $e_1\,e_2$.

$$\begin{array}{ll}
\emptyset\, e = e\, \emptyset = \emptyset & \Lambda\, e = e\, \Lambda = e \\
e \mid \emptyset = \emptyset \mid e = e & e \mid e = e \\
e_1\,(e_2\,e_3) = (e_1\,e_2)\,e_3 & e_1 \mid (e_2 \mid e_3) = (e_1 \mid e_2) \mid e_3 \\
e_1\,(e_2 \mid e_3) = (e_1\,e_2) \mid (e_1\,e_3) & (e_1 \mid e_2)\,e_3 = (e_1\,e_2) \mid (e_1\,e_3) \\
e_1 \mid e_2 = e_2 \mid e_1 & \emptyset^\star = \Lambda^\star = \Lambda \\
e^\star = \Lambda \mid e^+ & e^+ = e\,e^\star = e^\star\,e \\
e^\star\,e^\star = e^\star & e^{\star\star} = e^\star \\
e = e^\star \Leftrightarrow e = e\,e \wedge e \neq \emptyset & e\,e^\star = e^\star \Leftrightarrow \Lambda \in L(e) \\
(e_1^\star\,e_2^\star)^\star = (e_1 \mid e_2)^\star = (e_1^\star \mid e_2^\star)^\star & \\
(e_1^\star\,e_2)^\star\,(e_1^\star) = (e_1 \mid e_2)^\star = e_1^\star\,(e_2\,(e_1^\star))^\star &
\end{array}$$