QUELQUES UTILISATIONS DE Prover9-Mace4

FRIEDRICH WEHRUNG

1. Introduction (très) rapide à Prover9-Mace4 (version GUI)

Les deux logiciels manipulent les "énoncés du premier ordre", c'est à dire les expressions "bien formées" à partir de symboles de variables, constantes, relations, fonctions, connecteurs logiques ("et", "ou", "négation"), et quantificateurs existentiel ($\exists$) et universel ($\forall$).

Codages: "A et B" est `A & B.`; "A ou B" est `A | B.`; "non A" est `-A.` ; "pour tout x, A(x)" est `all x(A(x)).`; "il existe x tel que A(x)" est `exists x(A(x)).`

En théorie, `Prover9` peut trouver une preuve de n'importe quel énoncé prouvable du premier ordre, et `Mace4` peut trouver un contrexemple fini, s'il existe, à n'importe quel énoncé du premier ordre.

En raison des limitations physiques des machines, `Prover9-Mace4` fonctionne très bien avec certains énoncés ("toute algèbre de Robbins et une algèbre de Boole"), et plutôt mal avec d'autres ("Tout anneau satisfaisant l'identité $x^3 = x$ est commutatif").

Dans la fenêtre "Assumptions", entrer les hypothèses. Dans la fenêtre "Goals", entrer les conclusions désirées. L'une des deux fenêtres (en général la seconde) peut être laissée vide.

En cas de doute, il est possible de faire fonctionner `Prover9` et `Mace4` simultanément.

Les *variables* du langage sont des lettres de l'alphabet (`a`, `b`, `c`, ..., `x`, `y`, `z`), ou même des combinaisons de lettres et de chiffres (`chaise1`, `Table2` ...).

Les nombres (0, 1, 2, ...) sont automatiquement des *constantes*, qui peuvent ne rien avoir à voir avec les nombres auxquels nous sommes habitués. Cependant, ces constantes sont supposées distinctes ($0 \neq 1$, etc.).

UN POINT TRÈS IMPORTANT: Toujours terminer un énoncé avec un point final!

UN CONSEIL: En cas de grosse entrée (surtout dans "Assumptions"), préparer le texte au préalable avec l'éditeur de texte de votre choix.

Exemple 1.1. *Principe du tiers exclu*: on veut montrer que toute "proposition" `p` est soit vraie, soit fausse. La notation pour la négation de `p` est `-p`.

Laisser la fenêtre "Assumptions" vide, entrer dans "Goals" le texte

```
p | -p.
```

Lancer `Prover9`. On doit obtenir quelques préliminaires d'identification (date, heure, machine...), puis quelque chose comme

---

Date: Mars 2014.

```
============================== PROOF =================================

% -------- Comments from original proof --------
% Proof 1 at 0.00 (+ 0.01) seconds.
% Length of proof is 2.
% Level of proof is 1.
% Maximum clause weight is 0.
% Given clauses 0.

1 p | -p # label(non_clause) # label(goal).  [goal].
2 $F.  [deny(1)].

============================== end of proof ==========================
```

Exemple 1.2. Entrer dans “Assumptions” le texte

```
homme(x)->mortel(x).
homme(Socrate).
```

et dans “Goals” le texte

```
mortel(Socrate).
```

Lancer `Prover9`. Hors préliminaires, on obtient

```
============================== PROOF =================================

% -------- Comments from original proof --------
% Proof 1 at 0.00 (+ 0.01) seconds.
% Length of proof is 7.
% Level of proof is 3.
% Maximum clause weight is 0.
% Given clauses 0.

1 homme(x) -> mortel(x) # label(non_clause).  [assumption].
2 mortel(Socrate) # label(non_clause) # label(goal).  [goal].
3 homme(Socrate).  [assumption].
4 -homme(x) | mortel(x).  [clausify(1)].
5 mortel(Socrate).  [resolve(3,a,4,a)].
6 -mortel(Socrate).  [deny(2)].
7 $F.  [resolve(5,a,6,a)].

============================== end of proof ==========================
```

Quand `Prover9` (ou `Mace4`) lit `homme(x)`, il en conclut automatiquement que `homme` est une relation unaire, et `x` est une variable. Quand il lit `homme(Socrate)`, il en conclut automatiquement que `Socrate` est une variable (et n’a aucun préjugé sur l’appartenance de Socrate à la race humaine).

Exercice 1.1. Remplacer les “Assumptions” ci-dessus par

```
homme(a)->mortel(a).
homme(Socrate).
```

(1) Lancer `Prover9`, et observer la différence avec l’Exemple 1.2.
(2) Lancer `Mace4`; observer le résultat.

Explication: `Prover9-Mace4` suppose, par défaut, que les lettres `a`–`t` sont des *constantes*, alors que les lettres `u`–`z` sont des *variables*.

Si nous tenons vraiment à la lettre `a` au lieu de la lettre `x`, alors il faut écrire

```
all a(homme(a)->mortel(a)).
homme(Socrate).
```

Le résultat est alors le même qu'avec

```
homme(x)->mortel(x).
homme(Socrate).
```

L'exemple suivant est mathématiquement plus intéressant.

Exemple 1.3. Dans "Assumptions", entrer

```
x+(y+z)=(x+y)+z.
x+0=x.
0+x=x.
x+i(x)=0.

a+b!=b+a.
```

Ne rien écrire dans "Goals". Lancer `Mace4`. On doit obtenir, en un instant,

```
interpretation( 6, [number = 1,seconds = 0], [
    function(+(_,_), [
        0,1,2,3,4,5,
        1,0,3,2,5,4,
        2,4,0,5,1,3,
        3,5,1,4,0,2,
        4,2,5,0,3,1,
        5,3,4,1,2,0]),
    function(a, [1]),
    function(b, [2]),
    function(i(_), [0,1,2,4,3,5])]).
```

Explication: Quand il lit le symbole `+`, `Prover9-Mace4` ne présuppose pas que c'est l'addition de tous les jours, donc ne présuppose pas la commutativité $x+y=y+x$. De même, quand il lit `0`, il ne présuppose pas que c'est le 0 de tous les jours, donc il ne présuppose pas que $x+0=x$.

La chaîne de caractères `!=` est une abréviation pour "différent de" ($\neq$). Ainsi, `a!=b` veut dire "$a \neq b$". Dans cet exemple, `Mace4` trouve un groupe non commutatif à 6 éléments, nécessairement isomorphe au groupe $\mathfrak{S}_3$ des permutations d'un ensemble à 3 éléments.

Exercice 1.2. Trouver un anneau non commutatif.

Exercice 1.3. Trouver un monoïde commutatif (opération binaire + commutative $[x+y=y+x]$, associative $[x+(y+z)=(x+y)+z]$, élément neutre $[x+0=0+x=x]$, satisfaisant l'identité $3x=2x$, avec un élément a tel que $2a \neq a$.

2. Prise d'envol

Exercice 2.1. La notation mathématique usuelle pour l'inverse dans un groupe (`i(x)` dans l'exemple ci-dessus) est x^{-1}. Montrer que dans tout groupe, tous éléments a, b, c avec $aba^{-1}=b^2$, $bcb^{-1}=c^2$ et $cac^{-1}=a^2$ sont égaux à l'élément neutre du groupe.

Remarque 2.1. Sans `Prover9`, cet exercice est pratiquement inaccessible à qui n'est pas un expert confirmé en théorie des groupes. La preuve, obtenue par `Prover9` en une seconde, est déjà très longue et difficile à interpréter.

Exercice 2.2.

(1) Est-ce que l'une des identités d'associativité $(x * (y * z) = (x * y) * z)$ et d'associativité "tordue" $(x * (y * z) = (z * x) * y)$ implique l'autre? (Utiliser `Mace4`.) Varier les hypothèses: demander plusieurs idempotents, d'abord $0 * 0 = 0$ et $1 * 1 = 1$, puis $0 * 0 = 0$, $1 * 1 = 1$ et $2 * 2 = 2$. **La seconde entrée donne un message d'erreur**. Aller dans `Mace4 Options` et fixer une "start size" strictement plus grande que 2, par exemple 3.

Si vous en avez assez d'attendre le résultat, cliquer sur "kill". Pour cet exemple, `Mace4` a du mal (notez le "`seconds = 268`"):

```
interpretation( 8, [number = 1,seconds = 268], [
    function(*(_,_), [
        0,0,0,0,0,0,0,0,
        0,1,0,0,0,0,0,0,
        0,0,2,0,0,0,0,0,
        0,0,0,0,5,6,0,0,
        0,0,0,7,0,0,0,0,
        0,0,0,0,0,0,0,0,
        0,0,0,0,0,0,0,0,
        0,0,0,6,0,0,0,0]),
    function(a, [3]),
    function(b, [3]),
    function(c, [4])]).
```

(2) Montrer que l'associativité "tordue" `x*(y*z)=(z*x)*y.`, avec un élément neutre d'un côté (`1*x=x.`) ou de l'autre (`x*1=x.`) implique aussi bien l'associativité `x*(y*z)=(x*y)*z.` que la commutativité `x*y=y*x.`

Exercice 2.3. Un *corps* est un anneau dans lequel pour tout élément non nul admet un inverse multiplicatif $(x \neq 0 \Rightarrow x \cdot i(x) = i(x) \cdot x = 1)$.

(1) (Nécessite d'aller dans `Mace4 options` et d'ajuster la "start size" et la "end size".) Existe-t-il des corps avec 2, 3, 4, 5, 6, 7, 8, 9, 10 éléments?

L'entrée de base est

```
x*(y*z)=(x*y)*z.
x*1=x.
1*x=x.

x+(y+z)=(x+y)+z.
x+y=y+x.
x+0=x.
x+o(x)=0.

x*(y+z)=(x*y)+(x*z).
(x+y)*z=(x*z)+(y*z).

i(0)=0.
x!=0->x*i(x)=1.
```

(La ligne " `i(0)=0.` " n'est pas indispensable, mais elle permet de réduire considérablement le "search space".)

(2) En mettant "start size" et "end size" à 8 et "max models" à -1 (en informatique, -1 veut dire l'infini), trouver tous les corps à 8 éléments. Une fois le résultat obtenu, appliquer "Isofilter" pour éliminer les paires de corps isomorphes. Qu'observe-t-on?

(3) Lancer simultanément `Prover9` et `Mace4` pour savoir si oui ou non, il existe un corps non commutatif. Qu'observe-t-on?

(4) Supprimer l'une des deux hypothèses de distributivité, par exemple la ligne `(x+y)*z=(x*z)+(y*z).`, et trouver un exemple non-commutatif ($a * b \neq b * a$). (*Remarque*: pour supprimer une ligne, il est souvent indiqué de non pas l'effacer, mais de la faire précéder d'un signe `%`, par exemple, ici, remplacer `(x+y)*z=(x*z)+(y*z).` par `%(x+y)*z=(x*z)+(y*z).`)

Exercice 2.4. On considère le "graphe de Herschel", représenté dans la Figure 1. Montrer que ce graphe n'a pas de circuit visitant chaque point exactement une fois.

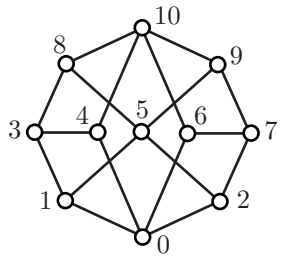

FIGURE 1. Le graphe de Herschel

Solution: Dans "`Mace4 Options` ", fixer toutes les "sizes" à 11, puis cocher "integer ring". Du coup, $+$ et $*$ sont interprétées comme les "vraies" addition et multiplication modulo `domain-size` (ici, 11). L'entrée ci-dessous va un peu plus loin que le but recherché.

```
G(x,0)<->(x=1 | x=2 | x=4 | x=6).
G(x,1)<->(x=0 | x=3 | x=5).
G(x,2)<->(x=0 | x=5 | x=7).
G(x,3)<->(x=1 | x=4 | x=8).
G(x,4)<->(x=3 | x=0 | x=10).
G(x,5)<->(x=1 | x=2 | x=8 | x=9).
G(x,6)<->(x=0 | x=7 | x=10).
G(x,7)<->(x=2 | x=6 | x=9).
G(x,8)<->(x=3 | x=5 | x=10).
G(x,9)<->(x=5 | x=7 | x=10).
G(x,10)<->(x=4 | x=6 | x=8 | x=9).

G(f(x),f(x+1)).
```

Exercice 2.5. On considère l'ensemble ordonné $(P, \leq)$ représenté dans la Figure 2. Ainsi, $0 \leq 0$, $0 \not\leq 1$, $0 \leq 3$, etc.

Une application $f\colon P \times P \to P$ (*à un couple* (x, y)*, avec* x *et* y *entre* 0 *et* 5*, on associe un nombre* $f(x, y)$ *compris entre* 0 *et* 5) est *croissante* si ($x \leq u$ et $y \leq v$)

implique que $f(x,y) \leq f(u,v)$. Montrer que si f est croissante et $f(x,x) \leq x$ pour tout x, alors soit $f(x,y) = x$ pour tous x et y, ou bien $f(x,y) = y$ pour tous x et y.

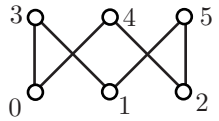

FIGURE 2. Le cube tronqué

Solution: Dans `Mace4 Options`, fixer la taille à 6, demander 3 solutions, et entrer

```
x<=0<->x=0.
x<=1<->x=1.
x<=2<->x=2.
x<=3<->(x=0 | x=1 | x=3).
x<=4<->(x=0 | x=2 | x=4).
x<=5<->(x=1 | x=2 | x=5).

f(x,x)<=x.

(x<=u & y<=v)->f(x,y)<=f(u,v).
```

Combien de solutions obtient-on?

3. Une preuve que tout anneau satisfaisant l'identité $x^3 = x$ est commutatif

(*Source*: Math Stack Exchange.)

On commence par observer que $2x = (2x)^3 = 8x^3 = 8x$, donc $6x = 0$ pour tout x.

De plus,

$$x + y = (x + y)^3 = x^3 + x^2y + xyx + yx^2 + xy^2 + yxy + y^2x + y^3$$

et

$$x - y = (x - y)^3 = x^3 - x^2y - xyx - yx^2 + xy^2 + yxy + y^2x - y^3\,.$$

En soustrayant, on obtient

$$2(x^2y + xyx + yx^2) = 0$$

En multipliant cette dernière relation par x à gauche et à droite on obtient

$$2(xy + x^2yx + xyx^2) = 0 \text{ et } 2(x^2yx + xyx^2 + yx) = 0\,.$$

En soutrayant les deux dernières relations on obtient

$$2(xy - yx) = 0\,.$$

On montre alors que $3(x + x^2) = 0$. Ceci est une conséquence de

$$x + x^2 = (x + x^2)^3 = x^3 + 3x^4 + 3x^5 + x^6 = 4(x + x^2)\,.$$

En particulier,

$$3(x + y + (x + y)^2) = 3(x + x^2 + y + y^2 + xy + yx) = 0$$

et on termine avec $3(xy + yx) = 0$. Mais comme $6xy = 0$, nous obtenons $3(xy - yx) = 0$. En soustrayant $2(xy - yx) = 0$, nous obtenons $xy - yx = 0$.

Références

[1] W. McCune, *Prover9 and Mace4*, logiciel disponible en ligne à http://www.cs.unm.edu/~mccune/prover9/, 2005–2010.

Friedrich Wehrung, LMNO, CNRS UMR 6139, Département de Mathématiques, Université de Caen, 14032 Caen Cedex, France
E-mail address: friedrich.wehrung01@unicaen.fr
URL: http://www.math.unicaen.fr/~wehrung