

TP 6 - MIU

1 L'énigme de Mu, de Douglas Hofstadter

Soit $A = M, U, I$ un alphabet à trois lettres. On décrit le langage L de mots de A^* par les règles suivantes.

- MI est dans L .
- Si x est un mot quelconque et si xI est dans L , alors xIU est dans L .
- Si x est un mot quelconque et si Mx est dans L , alors Mxx est dans L .
- Si x et y sont des mots quelconques et si $xIIIy$ est dans L , alors xUy est dans L .
- Si x et y sont des mots quelconques et si $xUUy$ est dans L , alors xy est dans L .

L'objectif est de formaliser en Coq la question suivante : MU est-il dans L et y répondre ?

Question 1 Définissez un type inductif énuméré `alpha` représentant l'alphabet M, U, I sur lequel les mots seront construits. Un mot sur `alpha` sera une liste d'éléments de type `alpha`.

```
Inductive alpha : Set := ...
```

```
Definition word := list alpha.
```

Question 2 Définissez un prédicat inductif d'appartenance `lang` d'un mot au langage. On rappelle que l'opération de concaténation sur les listes est noté `++` et l'opération d'ajout en tête de liste `::`.

```
Inductive lang : word -> Prop := ...
```

La réponse à la question posée : “*MU est-il dans L ?*” est non. En effet, le nombre d'occurrences de I des mots de L n'est pas congru à 0 modulo 3. Pour simplifier la preuve, on va construire le corps $Z3$. Les propriétés de $Z3$ se prouveront facilement.

Question 3 On définit $Z3$ avec le type inductif suivant :

```
Inductive Z3 : Set :=  
  | Z0 | Z1 | Z2.
```

Définissez l'opération `succ` qui calcule le successeur d'un entier de $Z3$. Définissez également l'opération d'addition sur $Z3$ (donc modulo 3).

```
Definition succ z := match z with ... end
```

```
Definition plus (x y : Z3) := ...
```

Question 4 Démontrez les trois propriétés suivantes dans Coq :

```
Lemma succ3 : forall x:Z3, succ (succ (succ x)) = x.
```

```
Lemma plus_succ : forall x y:Z3, plus (succ x) y = succ (plus x y).
```

```
Lemma double_non_0 : forall x:Z3, x <> Z0 -> plus x x <> Z0.
```

Question 5 Définissez une fonction `occur` : `word -> Z3` qui compte le nombre d'occurrences modulo 3 de I dans un mot.

Question 6 Prouvez en Coq le lemme suivant :

Lemma plus_plus : forall a b, occur (a ++ b) = plus (occur a) (occur b).

Question 7 Prouvez en Coq que le nombre de I dans un mot du langage est toujours différent de 0. Dédisez en le lemme recherché.

Lemma lang_I : forall l, lang l -> occur l <> Z0.

Lemma conclusion : ~ lang (M::U::nil).