



Contrôle continu du 22 mars 2023

Tous les documents et notes de cours sont autorisés. La documentation de Coq en ligne est également autorisée. Aucun autre document n'est autorisé. L'épreuve dure 1h30. Le sujet comporte 2 pages et il y a 1 exercice.

Toutes les réponses sont à réaliser en Coq dans un unique fichier que vous rendrez sous Moodle. Attention à bien sélectionner le rendu correspondant à votre groupe.

Exercice

On considère les nombres entiers naturels codés sous la forme de nombres binaires. En Coq, ces nombres pourraient être représentés par le type inductif suivant :

```
1 Inductive bin : Set :=  
2   | C0 : bin  
3   | C1 : bin  
4   | C0S : bin → bin  
5   | C1S : bin → bin.
```

Dans cette représentation, le constructeur de tête représente le bit de poids faible du nombre. Ainsi, le chiffre 6, codé comme 110, est représenté par le terme suivant : C0S (C1S C1).

1. Écrire la relation `is_even`, qui modélise l'ensemble des nombres binaires pairs. On rappelle que le bit de poids faible est toujours à 0 pour les nombres binaires pairs. La signature de cette relation est la suivante :

```
1 Inductive is_even : bin → Prop := ...
```

Démontrer que le chiffre 6 (codé comme 110) est pair.

2. Écrire la fonction `head_0`, qui étant donné un nombre binaire rajoute un 0 en tête de ce nombre. Par exemple, `head_0` appliquée à 110 rendra 0110. La signature de cette fonction est la suivante :

```
1 Fixpoint head_0 (b : bin) : bin := ...
```

Tester cette fonction sur le chiffre 6 (codé comme 110).

3. Écrire la relation `is_equal`, qui modélise l'égalité sur les nombres binaires. Dans cette relation, on considérera comme égaux un nombre et le même nombre où un nombre arbitraire de 0 a été ajouté en tête du nombre. Par exemple, 110 et 0110 sont égaux dans cette relation. Plus précisément, on considérera que :

- Un nombre binaire b et b avec un 0 en tête sont égaux dans cette relation.
- La relation est réflexive.
- La relation est symétrique.
- La relation est transitive.

La signature de cette relation est la suivante :

```
1 Inductive is_equal : bin → bin → Prop := ...
```

Dans la définition de cette relation, on pourra utiliser la fonction `head_0`, définie à la question précédente.

Démontrer que 110 et 00110 sont égaux dans cette relation.

4. Écrire la fonction `mult_2`, qui étant donné un nombre binaire multiplie ce nombre par 2. On rappelle qu'en binaire, une multiplication d'un nombre par 2 est un décalage à gauche de ce nombre. Cette fonction pourra rendre des nombres binaires éventuellement non canoniques, i.e. avec des 0 en tête inutiles. La signature de cette fonction est la suivante :

```
1 Fixpoint mult_2 (b : bin) := ...
```

Tester cette fonction avec le chiffre 6 (codé comme 110).

5. Démontrer que si un nombre binaire b est pair alors il existe un nombre binaire b' tel que b et $2 \times b'$ sont égaux (dans la relation `is_equal`). On utilisera la fonction `mult_2`, définie à la question précédente.