

# Module Sémantique

## TD 2 : The Folk Theorem

Le But de ce TD est de démontrer le *Folk Theorem* classique : Tout programme TOY est équivalent à un programme constitué d'une seule boucle `while`.

**Définition 1:** *Forme Normale.*

Un programme TOY est en *forme normale* s'il est de la forme

```
x1 := e1; x2 := e2; ...; xn := en;
while b do
  S
end
```

où `S` est un programme TOY ne contenant aucune boucle `while`

1. Montrer que le programme `Skip` peut être mis sous forme normale. (NB : il faut donc ajouter une boucle `while`)

2. Même question pour `x := e`.

On suppose que `S` et `T` sont des programmes TOY sous forme normale.

3. Mettre `S; T` sous forme normale.

4. Mettre `if b then S else T` sous forme normale.

5. Mettre `while b do S end` sous forme normale.

6. Les formes normales ci-dessus sont-elles équivalentes (au sens de l'équivalence fonctionnelle dérivée de la SOS) aux programmes originaux ?

On ajoute aux instructions du langage TOY l'instruction `release x` qui a la sémantique opérationnelle suivante :

$$\langle \text{release } x, \sigma \rangle \rightarrow \langle \text{skip}, \sigma[x \leftarrow \perp] \rangle$$

7. Modifier les traductions sous forme normale pour pouvoir prouver l'équivalence des

formes normales des programmes ci dessus avec les programmes originaux (au sens de l'équivalence fonctionnelle dérivée de la SOS).

NB : Les formes normales sont maintenant de la forme :

```
x1 := e1; x2 := e2; ...; xn := en;
while b do
  S
end
release v1; release v2; ...; release vk;
```