Voir la page <https://thor.enseirb-matmeca.fr/sites/if107-main/> pour toute information complémentaire sur cet enseignement.

1 Programmes itératifs

Exercice 1

Soit la fonction f ci-dessous.

1. Définir la machine à états de f .
2. Pouvez-vous prouver la terminaison de la fonction f ci-dessous avec la méthodes des ensembles bien fondés ? Justifiez et expliquez.
3. Prouvez la terminaison de la fonction f en prenant la précondition $x \geq 0$.

```

1 function f(x : int) :
2   while x != 0:
3     if x < -1:
4       x = 0;
5     x = x - 1

```

Exercice 2

La suite de Fibonacci est définie par :

$$F_0 = 0$$

$$F_1 = 1$$

$$F_n = F_{n-1} + F_{n-2} \quad n \geq 2$$

La fonction `fib` ci-dessous calcule le terme de rang n de la suite de Fibonacci.

```

1 function fib(n : nat) return (f : nat) :
2   f = 0; a = 1; b = 1;
3   i = 1;
4   while i <= n:
5     b = f + a;
6     f = a;
7     a = b;
8     i = i+1

```

1. Écrire une spécification de `fib`.
2. Décrire la machine à états de la fonction `fib`
3. Montrer que la fonction `fib` termine.
4. Prouver que `fib` est correcte pour votre spécification.

Exercice 3

Soit la fonction `fast_exp` ci-dessous qui calcule a^b .

```

1 function fast_exp(a : real, b : int) returns (y : real):
2     x = a; y = 1; z = b;
3     while z ≠ 0:
4         if z % 2 = 1:
5             y = x * y
6             z = z / 2;           // quotient
7             x = x * x

```

1. Écrire la spécification de la fonction `fast_exp`.
2. Définir la machine à états de la fonction `fast_exp` (on considèrera que les séquences d'instructions sans boucle forment une seule instruction).
3. Prouver que la fonction `fast_exp` termine.
4. Prouver que la fonction `fast_exp` est correcte pour votre spécification.

Exercice 4

On considère la fonction `gcd` ci-dessous qui calcule le PGCD g de deux entiers naturels a et b .

```

1 function gcd(a : nat, b : nat) returns (g : nat):
2     x = a; y = b;
3     while y > 0:
4         r = x % y;
5         x = y;
6         y = r
7     g = x

```

1. Spécifier la fonction mathématique $pgcd(x, y)$ qui associe à x et y leur PGCD (avec $x, y \in \mathbb{N}$ et $x > 0$ ou $y > 0$), par une formule de la logique des prédicats.
2. Écrire la spécification de la fonction `gcd` en utilisant la fonction `pgcd` introduite ci-dessus.
3. Définir la machine à états de la fonction `gcd` (on considèrera que les séquences d'instructions sans boucle forment une seule instruction).
4. Prouver que la fonction `gcd` termine.
5. Prouver que la fonction `gcd` est correcte pour votre spécification.

Exercice 5

La fonction `search` ci-dessous indique si l'entier x apparaît dans la liste l .

Le type `List` correspond au type inductif vu en cours : $\langle \rangle$ est une liste, et pour toute liste l et tout entier n , $\langle n, l \rangle$ est une liste. Pour toute liste non vide $\langle n, l \rangle$ on définit $\text{head}(\langle n, l \rangle) = n$ et $\text{tail}(\langle n, l \rangle) = l$.

```

1 function search(l : List, x : int) returns (found : bool)
2     p := l
3     found := false
4     while p ≠  $\langle \rangle$  and not (found):
5         if x = head(p):
6             found := true
7         p := tail(p)

```

1. Écrire une spécification de la fonction `search`
2. Définir la machine à états de la fonction `search`
3. Prouver la terminaison de la fonction `search`
4. Prouver la correction de la fonction `search`