LINGI 1122: Projet « Machine de Turing »

Groupe 13B: Rapport 1

Matthieu Baerts Pieter Hollevoet Hélène Verhaeghe

29 avril 2013

1 Sous-Machines

1.1 `findFirstBlankOnTheLeft()` (F1BL)

Pré : La tête de lecture se trouve sur un caractère quelconque du ruban.

```
-----------------------------------------------v--------------------
 suite de caractères|B|suite de caractères non B|C|suite de caractères
-----------------------------------------------------------------------
```

où C est un caractère quelconque

Post : La tête de lecture se trouve sur le premier blanc à gauche de l'ancien emplacement de la tête de lecture. Le ruban est inchangé.

```
--------------------v-------------------------------------------------
 suite de caractères|B|suite de caractères non B|C|suite de caractères
-----------------------------------------------------------------------
```

1.2 `findFirstBlankOnTheRight()` (F1BR)

Pré : La tête de lecture se trouve sur un caractère quelconque du ruban.

```
--------------------v-------------------------------------------------
 suite de caractères|C|suite de caractères non B|B|suite de caractères
-----------------------------------------------------------------------
```

où C est un caractère quelconque

Post : La tête de lecture se trouve sur le premier blanc à droite de l'ancien emplacement de la tête de lecture. Le ruban est inchangé.

```
-----------------------------------------------v--------------------
 suite de caractères|C|suite de caractères non B|B|suite de caractères
-----------------------------------------------------------------------
```

où C est un caractère quelconque

1.3 `convertToUnaryLeft()`

Pré : La tête de lecture se trouve sur un blanc et, à sa gauche, se trouve une suite valide en binaire délimitée par un blanc.

```
-------------------------------------v--------------------
 suite de caractères|B|suite de 0 et 1|B|suite de caractères
-------------------------------------------------------------
```

Post : La tête de lecture est de retour sur le même blanc. La suite binaire a été convertie en une suite unaire de valeur équivalente délimitée par un blanc et se trouve à gauche de la tête de lecture. La taille de cette suite peut être différente de la précédente, les changements se font vers la gauche, c'est-à-dire que le blanc à gauche peut avoir été décalé. La séquence de caractères à droite de la tête de lecture reste inchangée.

```
--------------------------------v-----------------------------
 suite de caractères|B|suite de 1|B|suite de caractères inchangés
-------------------------------------------------------------
```

1.4 `convertToUnaryRight()`

Pré : La tête de lecture se trouve sur un blanc et, à sa droite, se trouve une suite valide en binaire délimitée par un blanc.

```
--------------------v-------------------------------------
 suite de caractères|B|suite de 0 et 1|B|suite de caractères
-----------------------------------------------------------
```

Post : La tête de lecture est de retour sur le même blanc. La suite binaire a été convertie en une suite unaire de valeur équivalente délimitée par un blanc et se trouve à droite de la tête de lecture. La taille de cette suite peut être différente de la précédente, les changements se font vers la droite, c'est-à-dire que le blanc à droite peut avoir été décalé. La séquence de caractères à gauche de la tête de lecture reste inchangée.

```
-------------------------------v---------------------------------
 suite de caractères inchangés|B|suite de 1|B|suite de caractères
-----------------------------------------------------------------
```

1.5 `convertToBinaryLeft()`

Pré : La tête de lecture se trouve sur un blanc et, à sa gauche, se trouve une suite valide en unaire délimitée par un blanc.

```
--------------------------------v--------------------
 suite de caractères|B|suite de 1|B|suite de caractères
-------------------------------------------------------
```

Post : La tête de lecture est de retour sur le même blanc. La suite unaire a été convertie en une suite binaire de valeur équivalente délimitée par un blanc et se trouve à gauche de la tête de lecture. La taille de cette suite peut être différente de la précédente, les changements se font du coté gauche, c'est-à-dire que le blanc à gauche peut avoir été décalé. La séquence de caractères à droite de la tête de lecture reste inchangée.

```
-----------------------------------------v------------------------------
 suite de caractères |B|suite de 0 ou de 1|B|suite de caractères inchangée
-------------------------------------------------------------------------
```

1.6 `convertToBinaryRight()`

Pré : La tête de lecture se trouve sur un blanc et, à sa droite, se trouve une suite valide en unaire délimitée par un blanc.

```
--------------------v---------------------------------
 suite de caractères|B|suite de 1|B|suite de caractères
-----------------------------------------------------
```

Post : La tête de lecture est de retour sur le même blanc. La suite unaire a été convertie en une suite binaire de valeur équivalente délimitée par un blanc et se trouve à droite de la tête de lecture. La taille de cette suite peut être différente de la précédente, les changements se font du coté droit, c'est-à-dire que le blanc à droite peut avoir été décalé. La séquence de caractères à gauche de la tête de lecture reste inchangée.

```
------------------------------v----------------------------------------
 suite de caractères inchangée|B|suite de 0 ou de 1|B|suite de caractères
-------------------------------------------------------------------------
```

1.7 `addOneBlankAfterTheFirstBlankOnTheLeft()`

Pré : La tête de lecture est placée sur un blanc. À gauche se situe une suite de caractères délimitée par un blanc.

```
----------------------------------------v--------------------
 suite de caractères|B|suite de caractères|B|suite de caractères
-----------------------------------------------------------------
```

Post : La tête de lecture est de retour au même endroit. La suite de caractères à sa gauche est quant à elle inchangée et un blanc a été ajouté après le blanc délimitant cette suite.

```
-----------------------------------------------------v------------------------------
 suite de caractères|B|B|suite de caractères inchangée|B|suite de caractères inchangée
-------------------------------------------------------------------------------------
```

1.8 `copyLeftOfLeftSequence()`

Pré : La tête de lecture est placée sur un blanc. À gauche se situe une suite de 1 délimitée par un blanc.

```
-----------------------------------------------------v--------------------
 suite de caractères|B|suite de caractères|B|suite de 1|B|suite de caractères
-------------------------------------------------------------------------
```

Post : La tête de lecture est de retour au même endroit. La suite de 1 à sa gauche est quant à elle inchangée (elle a pu être modifiée entre temps mais on y retrouve toujours le même nombre de 1 l'un à la suite délimité par un blanc). Cette même suite de 1 à été recopiée du 2ème blanc à gauche vers la gauche (en décallant ce deuxième blanc vers la gauche et en le remplaçant par un caractère 1).

```
-------------------------------------------------------------------------v--------------------
 suite de caractères|B|suite de 1|suite de caractères inchangés|B|suite de 1|B|suite de caractères
-----------------------------------------------------------------------------------------------
```

1.9 `eraseRight()`

Pré : La tête de lecture se trouve sur un blanc, à sa droite se trouve une représentation d'un nombre délimitée par une blanc.

```
--------------------v-------------------------------------------------------
 suite de caractères|B|suite de caractères non blancs|B|suite de caractères
-----------------------------------------------------------------------------
```

Post : La tête de lecture se trouve au même endroit que précédemment. La partie gauche du ruban n'a pas été modifiée. La représentation du nombre à sa droite délimité par un blanc a été effacée (remplacée par des blancs).

```
---------------------------------v-------------------------------------------
 suite de caractères non modifiés|B|suite de B|B|suite de caractères non modifiés
-------------------------------------------------------------------------------
```

1.10 `copyRightOfLeftSequence()`

Pré : La tête de lecture est placée sur un blanc, à gauche duquel se situe une suite de 1.

```
---------------------------------v--------------------
 suite de caractères|suite de 1|B|suite de caractères
-----------------------------------------------------
```

Post : La tête de lecture est au même endroit. La suite de 1 quand à elle est inchangée. Cette suite de 1 à été recopiée du 2ème blanc à droite vers la droite.

```
---------------------------------v-----------------------------------------------
 suite de caractères|suite de 1|B|copie de la suite de suite de 1|suite de caractères
-------------------------------------------------------------------------------------
```

1.11 `incrementLeft()`

Pré : La tête de lecture se trouve sur un blanc du ruban. A sa droite se trouve une représentation unaire d'un nombre délimité par un blanc (si il n'y a que des blancs le nombre est 0).

```
-------------------------------------------v--------------------
 suite de caractères|B|réprésentation unaire|B|suite de caractères
-------------------------------------------------------------------
```

Post : La tête de lecture se trouve à la même position. Le nombre unaire a droite a été incrémenté de 1. Dans le cas ou il était nul (blanc), il est mis a 1.

```
-----------------------------------------------v----------------------
 suite de caractères|B|1|réprésentation unaire|B|suite de caractères
-------------------------------------------------------------------------
```

1.12 `shiftSubstract()`

Pré : La tête de lecture se trouve sur un blanc séparant deux nombres en représentation unaire. Ces deux nombres sont délimités par un blanc.

```
-------------------------------------------v-----------------------------------------
 suite de caractères|B|réprésentation unaire|B|réprésentation unaire|B|suite de caractères
-------------------------------------------------------------------------------------------
```

Post : La tête se trouve sur le blanc entre les 2 représentations unaire. Le premier nombre a été diminué du second et le deuxième conserve sa valeur mais a été décalé de facon a ce qu'il n'y ait toujours qu'un espace qui sépare les nombres unaires.

```
-----------------------------------------------------v-----------------------------------------
 suite de caractères inchangés|B|réprésentation unaire|B|réprésentation unaire|B|suite de caractères
-------------------------------------------------------------------------------------------------
```

2 Addition

```
-------------------------------------------------------v--------------------
 suite de caractères|B|suite de 0 et 1|B|suite de 0 et 1|B|suite de caractères
-------------------------------------------------------------------------------
```

Premièrement, on place la tête de lecture entre les deux nombres grâce à `findFirstBlankOnTheLeft()`.

```
-------------------------------------v-------------------------------------
 suite de caractères|B|suite de 0 et 1|B|suite de 0 et 1|B|suite de caractères
-------------------------------------------------------------------------------
```

Ensuite, on utilise `convertToUnaryRight()` et `convertToUnaryLeft()` pour transformer les deux nombres en unaire.Le blanc entre les deux nombres est conservé.

```
---------------------------------v---------------------------------
 suite de caractères|B|suite de 1|B|suite de n 1|B|suite de caractères
-------------------------------------------------------------------
```

Pour réaliser l'addition en elle-même, il ne reste plus qu'à retirer un bâton à l'extrême droite du nombre de droite (utiliser `findFirstBlankOnTheRight()`) et à le placer à la place du blanc séparent initialement les deux nombres.

```
---------------------------------v---------------------------------
 suite de caractères|B|suite de 1|1|suite de n-1 1|B|suite de caractères
-------------------------------------------------------------------
```

Pour finir, se replacer à l'extrême droite du nombre final et appliquer `convertToBinaryLeft()`. La tête de lecture sera déjà au bon endroit.

```
-------------------------------------v--------------------
 suite de caractères|B|suite de 0 et 1|B|suite de caractères
-----------------------------------------------------------
```

3 Soustraction

Premièrement, on place la tête de lecture entre les deux nombres grâce à `findFirstBlankOnTheLeft()`. Ensuite, on utilise `convertToUnaryRight()` et `convertToUnaryLeft()` pour transformer les deux nombres en unaire. Le blanc entre les deux nombres est conservé tout comme les deux premières étapes de la section 2. Un exemple ici la soustraction de 5 par 3.

```
----------------------------v-------------------------
 suite de caractères|B|11111|B|111|B|suite de caractères
-----------------------------------------------------
```

Pour réaliser la soustraction en elle-même, il ne reste plus qu'à retirer un bâton (en le remplaçant par un blanc) dans la suite de droite en partant de la droite de cette suite : c'est-à-dire en utilisant `findFirstBlankOnTheRight()` puis en se plaçant à gauche, en regardant si le caractère est bien à 1 (sinon la soustraction est terminée) avant de remplacer ce caractère par un blanc pour enfin revenir au blanc situé entre les deux nombres avec `findFirstBlankOnTheLeft()`.

```
----------------------------v-------------------------
 suite de caractères|B|11111|B|11|BB|suite de caractères
-----------------------------------------------------
```

On fait pareil dans la suite de gauche en partant de la gauche de cette suite : c'est-à-dire en utilisant `findFirstBlankOnTheLeft()` puis en se plaçant à droite, en regardant si le caractère est bien à 1 (sinon la soustraction vaut 0, c'est terminé) avant de remplacer ce caractère par un blanc pour enfin revenir au blanc situé entre les deux nombres avec `findFirstBlankOnTheRight()`.

```
----------------------------v-------------------------
 suite de caractères|BB|1111|B|11|BB|suite de caractères
-----------------------------------------------------
```

Ces deux opérations sont donc à réaliser tant que l'une des deux suites ne sont pas (2 blancs consécutifs) :

```
----------------------------v-------------------------
 suite de caractères|BBBB|11|B||BBBB|suite de caractères
-----------------------------------------------------
```

Pour finir, il ne reste plus qu'à appliquer `convertToBinaryLeft()`. La tête de lecture sera déjà au bon endroit.

```
----------------------------v-------------------------
 suite de caractères|BBBB|10|B||BBBB|suite de caractères
-----------------------------------------------------
```

4 Multiplication

Premièrement, on place la tête de lecture entre les deux nombres grâce à `findFirstBlankOnTheLeft()`. Ensuite, on utilise `convertToUnaryRight()` et `convertToUnaryLeft()` pour transformer les deux nombres en unaire. Le blanc entre les deux nombres est conservé tout comme les deux premières étapes de la section 2.
Il faut ensuite placer un blanc à gauche du premier blanc de gauche grâce à `addOneBlankAfterTheFirstBlankOnTheLeft()`.

```
-----------------------------------v-----------------------------------
 suite de caractères|B|B|suite de m 1|B|suite de n 1|B|suite de caractères
-------------------------------------------------------------------------
```

Ensuite, on décrémente le nombre de droite par la droite tant que cette suite n'est pas vide (deux blancs).

```
-----------------------------------v---------------------------------------
 suite de caractères|B|suite de m 1|B|suite de n-1 1|BB|suite de caractères
-----------------------------------------------------------------------------
```

À chaque décrémentation, on appelle `copyLeftOfLeftSequence()` pour recopier la suite de m 1 à droite du deuxième blanc en décalant ce blanc vers la gauche et en le remplaçant par un blanc.

```
-------------------------------------------------v-----------------------------------
 suite de caractères|B|suite de m 1|B|suite de m 1|B|suite de n-1 1|BB|suite de caractères
-------------------------------------------------------------------------------------
```

Ces opérations sont donc à réaliser tant que la suite de droite n'est pas vide (2 blancs).

```
-------------------------------------------------------------v-----------------------------------
 suite de caractères|B|suite de m 1|...|suite de m 1|B|suite de m 1|B|suite de n B|B|suite de caractères
-------------------------------------------------------------------------------------------------
```

Il faut maintenant placer la tête de lecture sur le blanc de gauche (avec `findFirstBlankOnTheLeft()`). Le résultat de la multiplication se trouve à gauche de ce blanc, on peut donc supprimer tout ce qu'il y a à droite en utilisant `eraseRight()`.

```
-------------------------------------------------v-------------------------------------------------
 suite de caractères|B|suite de m 1|...|suite de m 1|B|suite de m B|B|suite de n B|B|suite de caractères
-------------------------------------------------------------------------------------------------
```

On retransforme ensuite le nombre unaire représentant le résultat en binaire avec `convertToBinaryLeft()`.

```
---------------------------------------v---------------------------------------
 suite de caractères|B|suite de 0 et 1|B|suite de m+n+1 B|B|suite de caractères
-------------------------------------------------------------------------------
```

5 Division

On transforme les deux nombres à multiplier en unaire comme expliqué dans la section 2. Notre ruban est alors de la forme :

```
-------------------------------v-----------------------------
 suite de caractères|Dividende|B|Diviseur|suite de caractères
-----------------------------------------------------------
```

Ensuite on doit tester si le dividende est plus grand que le diviseur ou non. Dans le cas ou il est plus grand on effectue une boucle, on utilise la méthode `shiftSubstract()` pour soustraire le diviseur du dividente (qui sera notre reste).

```
-----------------------------------------v-----------------------------
 suite de caractères|B|Dividende-Diviseur|B|Diviseur|suite de caractères
-------------------------------------------------------------------------
```

Ensuite on doit augmenter notre quotient de 1. On utilise `findFirstBlankOnTheLeft()` suivi de `incrementLeft()`.

```
-----------------------v-------------------------------------------------
 suite de caractères|1|B|Dividende-Diviseur|B|Diviseur|suite de caractères
-------------------------------------------------------------------------
```

Un appel a `findFirstBlankOnTheRight()` replace notre curseur a la position entre les 2 unaires et la forme générale du ruban sera :

```
-------------------------------------v-----------------------------
 suite de caractères|Quotient|B|Reste|B|Diviseur|suite de caractères
-----------------------------------------------------------------
```

On peut recommencer cette boucle jusqu'a ce que notre reste soit inférieur au diviseur. A la terminaison de la boucle, `eraseRight()` permet d'effacer notre diviseur et d'obtenir :

```
-------------------------------------v-----------------------------
 suite de caractères|Quotient|B|Reste|B|suite de B|suite de caractères
-------------------------------------------------------------------
```

`findFirstBlankOnTheLeft()` pour positionner la tête sure le blanc entre le quotient et le reste en unaire.

```
-------------------------------------v---------------------------------
 suite de caractères|Quotient unaire|B|Reste unaire|suite de caractères
-------------------------------------------------------------------
```

On retransforme ensuite les nombre unaire en binaire avec `convertToBinaryLeft()` et `convertToBinaryRight()`.

```
-------------------------------------v---------------------------------
 suite de caractères|Quotient binaire|B|Reste binaire|suite de caractères
-------------------------------------------------------------------
```

Et finalement `findFirstBlankOnTheright()` pour positionner la tête de lecture à l'endroit souhaité.