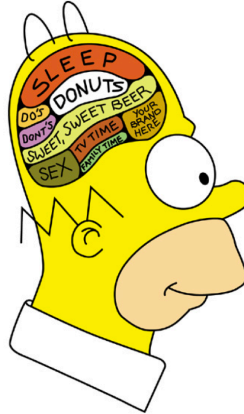


Langage simple



15 avril 2019

Damien Morard

Au cours de cette séance, vous verrez un langage simple : sa syntaxe concrète, sa syntaxe abstraite, sa sémantique, et des preuves à partir de celle-ci.

On s'intéresse au cours de cette séance à un véritable langage de programmation minimaliste : [brainfuck](#)¹. Les informations en anglais données ci-dessous sont reprises de [Wikipedia](#).

A brainfuck program has an implicit byte pointer, called "the pointer", which is free to move around within an array of 30000 bytes, initially all set to zero. The pointer itself is initialized to point to the beginning of this array.

The brainfuck programming language consists of eight commands, each of which is represented as a single character.

>	Increment the pointer.
<	Decrement the pointer.
+	Increment the byte at the pointer.
-	Decrement the byte at the pointer.
.	Output the byte at the pointer.
[Jump forward past the matching] if the byte at the pointer is zero.
]	Jump backward to the matching [unless the byte at the pointer is zero.

Cell-clear A simple program fragment that sets the current location to 0, by iteratively decrementing until it is equal to 0.

[~]

Clear previous cells Clears the current cell and all those before it, stopping at the first 0 value.

[[~]<]

1. <http://en.wikipedia.org/wiki/Brainfuck>, <http://www.muppetlabs.com/~breadbox/bf/>

Add This adds the current location (destructively, as its value is zeroed in the process) to the next location.

```
[->+<]
```

Rewind Decrements the data pointer until 0 is found, then increments it so the current cell is non-zero.

```
[<]>
```

Simple loop A continuous loop that takes text input from the keyboard and echoes it to the screen (similar to the Unix cat program). Note that this assumes the cell is set to 0 when a , command is executed after the end of input (sometimes called end-of-file or "EOF").

```
,[.,]
```

Seek Moves the pointer forward until it lands on a byte with a value of 1, preserving all bytes it passes; then it stops. To find other values, repeat each + and - that number of times.

```
-[+>-]+
```

Copying a byte There is no operation for copying bytes. This must be done with the looping construct and arithmetical operators.

```
>[-]>[-]<<[->+<<]>>[-<<+>>]<<
```

Exercise 1 : Syntaxe Concrète



Donnez la grammaire EBNF de ce langage.

Exercise 2 : Syntaxe Abstraite



1. Donnez une définition inductive, à l'aide de règles d'inférence, des programmes brainfuck.
2. Écrivez le code Prolog permettant de définir des programmes de ce langage.

Exercise 3 : Sémantique



Donnez la sémantique des programmes définis précédemment. Pour cela, répondez aux questions suivantes.

1. Quel est le domaine sémantique ?
2. Quelles sont les règles pour les opérateurs définis dans les questions précédentes ?
3. Donnez le code Prolog correspondant.

Exercise 4 : Évaluation

1. Évaluez à la main le programme suivant : `>+>+>+<[<]>`
2. Évaluez à la main le programme suivant : `+>+>+>+[[-<]<]`