

Quizz Programmation 1

Aliaume Lopez

25 octobre 2019

Langages et expressivité

Parmis ces « langages », lesquels sont « Turing Complets » ?

- | | | | |
|--------------|--------------------------|---------------|--------------------------|
| 1. Cobol | <input type="checkbox"/> | 2. Javascript | <input type="checkbox"/> |
| 3. HTML | <input type="checkbox"/> | 4. PHP | <input type="checkbox"/> |
| 5. XML | <input type="checkbox"/> | 6. Ouk ! | <input type="checkbox"/> |
| 7. Brainfuck | <input type="checkbox"/> | 8. JSON | <input type="checkbox"/> |

1.Vrai / 2.Vrai / 3.Faux / 4.Vrai / 5.Faux / 6.Vrai / 7.Vrai / 8.Faux

Machines et expressivité

Parmis les mécanisme suivants, lesquels sont « Turing Complets » ?

- | | | | |
|-------------------------------|--------------------------|----------------------|--------------------------|
| 1. La MMU d'un processeur X86 | <input type="checkbox"/> | 2. Les templates C++ | <input type="checkbox"/> |
| 3. Les « Generics » de Java | <input type="checkbox"/> | | |

1.Vrai / 2.Vrai / 3.Vrai

Unicode, mots et WC (Part I)

L'utilitaire WC (Word Count) permet de calculer le nombre de lignes, de mots et de caractères dans un document. Considérons une implémentation de WC correcte pour ASCII

- | | | | |
|--|--------------------------|---|--------------------------|
| 1. Elle compte le bon nombre de ligne en UTF-8 | <input type="checkbox"/> | 2. Elle compte le bon nombre de caractères en UTF-8 | <input type="checkbox"/> |
| 3. Elle compte le bon nombre de mots en UTF-8 | <input type="checkbox"/> | | |

1.Vrai / 2.Faux / 3.Faux

Unicode, mots et WC (Part II)

On suppose cette fois que WC est correct pour l'UTF-8. On souhaite paralléliser WC en calculant sur des bouts de fichiers. L'UTF-8 pouvant utiliser plus d'un octet pour représenter un caractère, on peut couper au milieu d'un caractère.

- | | | | |
|--|--------------------------|--------------------------------|--------------------------|
| 1. Ce n'est pas grave | <input type="checkbox"/> | 2. C'est grave pour les lignes | <input type="checkbox"/> |
| 3. Il faut faire attention aux caractères liants | <input type="checkbox"/> | | |

1.Faux / 2.Faux / 3.Vrai

Javascript (Part I)

Quel est le résultat de l'expression Javascript suivante : [] + []

- | | | | |
|-------|--------------------------|--------|--------------------------|
| 1. [] | <input type="checkbox"/> | 2. {} | <input type="checkbox"/> |
| 3. " | <input type="checkbox"/> | 4. NaN | <input type="checkbox"/> |

1.Faux / 2.Faux / 3.Vrai / 4.Faux

Javascript (Part II)

Quel est le résultat de l'expression Javascript suivante : {} + []

- | | | | |
|--------|--------------------------|-------|--------------------------|
| 1. [] | <input type="checkbox"/> | 2. 0 | <input type="checkbox"/> |
| 3. Nan | <input type="checkbox"/> | 4. {} | <input type="checkbox"/> |

1.Faux / 2.Vrai / 3.Faux / 4.Faux

Javascript (Part III)

Quel est le résultat de l'expression Javascript suivante : [] + {}

- | | | | |
|----------------------|--------------------------|-------|--------------------------|
| 1. '[object Object]' | <input type="checkbox"/> | 2. [] | <input type="checkbox"/> |
| 3. 0 | <input type="checkbox"/> | 4. {} | <input type="checkbox"/> |

1.Vrai / 2.Faux / 3.Faux / 4.Faux

Javascript (Part IV)

Quel est le résultat de l'expression Javascript suivante : {} + {}

- | | | | |
|----------------------|--------------------------|-------------------------------------|--------------------------|
| 1. '[object Object]' | <input type="checkbox"/> | 2. '[object Object][object Object]' | <input type="checkbox"/> |
| 3. [] | <input type="checkbox"/> | 4. {} | <input type="checkbox"/> |

1.Faux / 2.Vrai / 3.Faux / 4.Faux

Un peu de C

Lorsqu'une instruction possède un comportement indéfini en C, un compilateur reconstruit cette instruction peut

- | | | | |
|--|--------------------------|--|--------------------------|
| 1. Refuser de compiler | <input type="checkbox"/> | 2. Supprimer cette instruction | <input type="checkbox"/> |
| 3. Remplacer votre programme par une vidéo de chat | <input type="checkbox"/> | 4. Contacter Microsoft TM pour dénoncer vos pratiques | <input type="checkbox"/> |

1.Vrai / 2.Vrai / 3.Vrai / 4.Vrai

Un peu d'OCaml

En OCaml, il existe une fonction Obj.magic qui transforme un élément de type A en un élément de type B. Lors de la compilation vers du code machine, cette commande est traduite par

- | | | | |
|---------------------|--------------------------|-----------------------------|--------------------------|
| 1. Un cast en C | <input type="checkbox"/> | 2. Une conversion explicite | <input type="checkbox"/> |
| 3. Strictement rien | <input type="checkbox"/> | 4. Un type structure | <input type="checkbox"/> |

1.Faux / 2.Faux / 3.Vrai / 4.Faux

Un peu de Haskell

L'évaluation paresseuse (appel par nécessité) possède certains avantages, toutefois

- | | | | |
|---|--------------------------|---|--------------------------|
| 1. Il n'est pas possible de l'implémenter sur un ordinateur | <input type="checkbox"/> | 2. Le traitement des effets de bords est complexe | <input type="checkbox"/> |
| 3. Il ne permet pas de faire des programmes impurs | <input type="checkbox"/> | 4. Il peut engendrer une grande consommation de mémoire | <input type="checkbox"/> |

1.Faux / 2.Vrai / 3.Faux / 4.Vrai