

Sections 3.7-3.8

Situation - Problèmes

Yves Deville

2020-2021

Les situations – problèmes décrits ci-après doivent être lus avant de regarder les vidéos 5 et 6 du chapitre 3. Ce sont de questions à vous poser en visionnant ces vidéos. Les liens vers ces vidéos sont reprises sur le site Moodle du cours.

- Vidéo 5 : 3.7. Hoare-Allison
- Vidéo 6 : 3.8. Extension fonction totale

1. Reprenons la preuve du théorème de Hoare-Allison, mais avec le langage Java complet à la place du langage limité Q. On arrive alors à la conclusion que l'interpréteur Java n'est pas calculable en Java, ce que nous savons faux. Où se situe l'erreur dans cette démonstration modifiée ?
2. Pourquoi la non calculabilité de l'interpréteur de Q est-elle un signe de la limitation d'un langage comme Q ? Doit-on souvent écrire un interpréteur dans sa vie professionnelle ? Pour ceux qui doivent écrire un interpréteur, ils peuvent le faire en Java. Mais pour les autres problèmes, utilisons le langage Q ou il n'y a jamais de problème de terminaison.
3. Pourquoi l'ensemble  $S = \{n \mid \varphi_n \text{ est totale}\}$  n'est-il pas récursif ?
4. Pourquoi l'existence de fonctions partielles calculables ne pouvant pas être étendues en fonction totales calculables est-elle une limitation pour l'informatique ?

1. Car Java ne calcule pas que des fcts totales  
 $\Rightarrow$  si  $\text{resultat} = \perp$ ,  $\text{interpret}(p_n, x) + 1 = \perp + 1 = \perp$

le programme ne se termine pas (boucle)  
 $\Rightarrow \perp + 1$  ne sera pas calculé.

2. Voir dia 36 :

3. Si  $S$  récursif.

Alors  $\exists$  langage  $Q^*$ , avec fct totale et primitive  
reproduisant les fcts totales

↳ impossible Horewicz - Allison

Java<sup>\*</sup>  $J \rightarrow \text{pym Java}$

si  $J \in S$  Alors  $J \in \text{Java}^*$

sinon  $J \notin \text{Java}^*$

4.  $f(n) = \begin{cases} \text{résultat} & \text{si prénommé} \\ \perp & \text{sinon} \end{cases}$

↳  $f^*(n) = \begin{cases} \text{res. si prén} \\ \text{ERREUR sinon} \end{cases}$