Option Traduction de Programmes
TD Mercredi 7 mars

1. Donner des attributs et leurs règles de calcul pour des instructions conditionnelles ternaires:
   `if <expression> (etiq1,etiq2,etiq3)`
   (où l'exécution saute vers l'étiquette 1, 2 ou 3 selon si la valeur de l'expression est négative, nulle ou positive)

2. Pour quelles opérations de comparaison arithmétique de C est-il possible de générer du code intermédiaire pour l'instruction C
   `if (X op Y) goto L;`
   et qui consiste en une instruction `CMP` suivie immédiatement d'une instruction `Branche`?
   (On suppose que les valeurs de `X` et `Y` sont déjà disponibles dans deux registres.)

   Pour quelles comparaisons est-il possible de générer du code qui calcule la valeur de la comparaison dans un registre en deux instructions?

3. Un arbre binaire pour les switch/case:

   Donner des définitions et règles de calcul d'attributs permettant de générer du code dans la forme d'un arbre binaire équlibré pour le fragment de langage suivant. La sémantique est que, selon la valeur (entière) de l'`expression`, une des `instructions` est choisie et exécutée. Si la valeur de l'expression n'est pas celle d'une des `constantes`, l'instruction qui suit `else` est exécutée s'il y en a (et sinon, rien). Vous pouvez supposer que les valeurs de toutes les `constantes` sont distinctes.

```
switch --> "case" expression "of" cases "esac"
cases --> constant ":" instruction cases
cases --> "else" instruction
cases -->       {vide}
```