



UFR 919 Informatique – Master Informatique

Spécialité STL – UE ILP

TD-TME5 — Boucles et échappements en ILP3

Objectif : Ajouter des nouveaux traits (équivalents de `break` et `continue`) à ILP2 permettant l'échappement d'une boucle courante. On consultera avec profit le memento indiquant comment étendre ILP2. <http://www-master.ufr-info-p6.jussieu.fr/site-annuel-courant/Ext/queinnec/ILP/refcard.pdf>

Ω

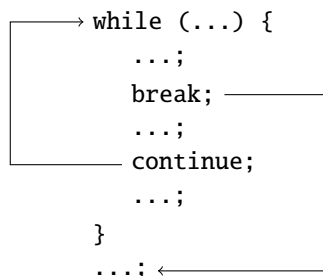
Buts :

- Implanter des opérateurs de contrôle avancés
- Connaître les changements d'implantation d'ILP2
- Apprendre à étendre ILP2

Présentation des nouveaux traits : C ou Java autorisent la rupture du flot de contrôle de l'évaluation avec les instruction `break` et `continue` (en Perl, il s'agit de `last` et `next`). Dans le corps d'une boucle, ces instructions ont l'effet suivant :

- `break` : sortir du corps de la boucle ;
- `continue` : revenir en début de boucle.

On peut illustrer ces comportements comme suit :



1 Échappements simples

Travail à réaliser : Introduire deux nouvelles instructions `break` et `continue` qui réalisent les deux échappements du corps de boucle courante.

- Écrire la grammaire étendue des deux mots-clefs sous-jacentes. Vous veillerez à l'écrire dans le fichier `fr/upmc/ilp/ilp2tme5/grammar2-tme5.rnc`.

- Identifiez ce que la grammaire ne permet pas de contraindre, et réfléchissez à des exemples de programmes respectant la grammaire mais n'ayant pas de sens. Notez bien que cette question est importante car de tels programmes devront être rejetés par votre implantation, pour l'évaluation comme pour la compilation.
- Écrire des programmes de test utilisant ces nouvelles fonctionnalités. Vous nommerez ces fichiers `u*-2tme5.xml`
- Implanter la compilation de ces instructions en C (notez bien que le schéma de compilation est très simple, bien plus que celui d'évaluation).
- Implanter l'évaluation de ces instructions.
- N'oubliez pas d'écrire les classes `Process` et `ProcessTest`, de vérifier que votre implantation se comporte correctement sur vos programmes de test et que vos changements ne cassent pas l'exécution des programmes existants.

Ω

2 Boucles nommées

En Java, on peut, par une instruction `break` ou `continue` rompre le flot de contrôle pour le réorienter vers un point déterminé par une étiquette. Ceci permet de s'échapper de plusieurs niveaux de boucles. Cette possibilité n'existe pas directement en C.

Pour cela, on place une étiquette en tête des boucles où l'on veut revenir ou dont on veut sortir. Les instructions d'échappement prennent en argument de telles étiquettes.

Travail à réaliser :

- Définir une nouvelle grammaire pour exprimer ces nouveaux traits.
- Écrire un nouveau programme montrant l'utilisation de ces nouvelles fonctionnalités, préciser les sorties désirées
- Ici encore, réfléchissez à des exemples de programmes acceptés par la grammaire mais dont l'interprétation et la compilation doivent échouer.
- Implanter la compilation de ces instructions en C.
- Implanter ces instructions dans l'évaluateur Java.

Ω