

Syntol: la source XML

Paul Feautrier

ENS de Lyon

Paul.Feautrier@ens-lyon.fr
perso.ens-lyon.fr/paul.feautrier

30 avril 2009



Structure générale

```
<program label="vadd">
  <declaration>
    <inside> vadd </inside>
    <identifier> n </identifier>
    <basic size="32"> __integer</basic>
  </declaration>
  ...
  <for label="lbl_2">
    ...
  </for>
  <statement label="S">
    <inside> vadd vadd lbl_2 </inside>
    ...
  </statement>
</program>
```

```
int n;
process vadd(){
  float a[100], b[100], c[100];
  int i;
  for(i=0; i<n; i++)
S:    a[i] = b[i] + c[i];
}
```

```
<basic size="32"> __integer</basic>
```

```
<array>  
  <basic size="32"> __float</basic>  
  <expression>  
    <constant> cst_1 </constant>  
  </expression>  
</array>
```

- ▶ un type entier
- ▶ un type tableau de flottants

```
<declaration>  
  <inside> vadd </inside>  
  <identifiant> n </identifiant>  
  <basic size="32"> __integer</basic>  
</declaration>
```

```
<declaration>  
  <inside> vadd vadd </inside>  
  <identifiant> a </identifiant>  
  <array>  
    <basic size="32"> __float</basic>  
    <expression>  
      <constant> cst_1 </constant>  
    </expression>  
  </array>  
</declaration>
```

```
<declaration>  
  <inside> vadd </inside>  
  <constant value="0"> cst_3 </constant>  
  <basic> __integer </basic>  
</declaration>
```

- ▶ déclaration d'une variable entière
- ▶ déclaration d'un tableau
- ▶ déclaration d'une constante

Expressions et Instructions

```
<statement label="S">
  <inside> vadd vadd lbl_2 </inside>
  <expression>
    <operator> __assignment </operator>
    <expression>
      <operator> __subscript </operator>
      <expression>
        <identifier> a </identifier>
      </expression>
      <expression>
        <identifier> i </identifier>
      </expression>
    </expression>
  <expression>
    <operator> __fplus </operator>
    <expression>
      <operator> __subscript </operator>
      <expression>
        <identifier> b </identifier>
      </expression>
      <expression>
        <identifier> i </identifier>
      </expression>
    </expression>
  <expression>
    .....
  </expression>
</expression>
</expression>
</statement>
```

```
<for label="lbl_2">
  <init>
    <expression>
      <operator> __assignment </operator>
      <expression>
        <identifiant> i </identifiant>
      </expression>
      <expression>
        <constant> cst_3 </constant>
      </expression>
    </expression>
  </init>
  <test>
    <expression>
      <operator> __lt </operator>
      <expression>
        <identifiant> i </identifiant>
      </expression>
      <expression>
        <identifiant> n </identifiant>
      </expression>
    </expression>
  </test>
  <next>
    <expression>
      <operator> _post_incr </operator>
      <expression>
        <identifiant> i </identifiant>
      </expression>
    </expression>
  </next>
</for>
```

On retrouve les trois composantes de la boucle for classique :

- ▶ initialisation
- ▶ test de continuation
- ▶ progression

Une étape de *pattern matching* est nécessaire pour les bornes inférieures et supérieures et le pas.

- ▶ Les divers éléments ne sont pas inclus les uns dans les autres
- ▶ La structure est spécifiée par les balises `inside`
- ▶ Autres constructions, peu ou pas utilisées par Syntol :
 - ▶ pointeurs,
 - ▶ structures,
 - ▶ tests,
 - ▶ fonctions et appels de fonctions