Parallel and Distributed Algorithms and Programs
TD n°5 - Nested Loops

Hadrien Croubois
hadrien.croubois@ens-lyon.fr

Aurélien Cavelan
aurelien.cavelan@ens-lyon.fr

4/12/2015

All documents are available on my website: `http://hadriencroubois.com/#Teaching`

Part 1

Dependency analysis

Question 1

a) Perform the dependency analysis on the following code (flow, anti, output):

$S_1 : A \leftarrow B + C$
$S_2 : D \leftarrow E + F$
$S_3 : C \leftarrow A + C$
$S_4 : E \leftarrow C + D$
$S_5 : E \leftarrow 1$

b) Perform the dependency analysis on the following program:

for $i = 1$ *to* n **do**
  for $j = 1$ *to* n **do**
    S_1: $a(i+1, j-1) \leftarrow b(i, j+4) + c(i-2, j-3) + 1$
    S_2: $b(i-1, j) \leftarrow a(i, j) - 1$
    S_3: $c(i, j) \leftarrow a(i, j-2) + b(i, j)$

Part 2

Removing dependencies

Question 2

a) Compute the RDG (Reduced Dependency Graph) for the following code. Try to break the cycle by introducing a new temporary variable.

for $i = 1$ *to* n **do**
  S_1: $a(i) \leftarrow b(i) + c(i)$
  S_2: $a(i+1) \leftarrow a(i) + 2d(i)$

In the general case, cutting vertices is done as follows (note that only flow dependencies are renamed to *temp*, otherwise the code would be wrong):

for $i = 1$ *to* n **do**
  $\dots S_k$: $lhs(f(i)) \leftarrow rhs(\dots)$
  $\dots$
  $\dots$
  S_i: $\cdots \leftarrow lhs(g(i))$

for $i = 1$ *to* n **do**
  $\dots$
  S'_k: $temp(f(i)) \leftarrow rhs(\dots)$
  S_k: $lhs(f(i)) \leftarrow temp(f(i))$
  $\dots$
  S_i: $\cdots \leftarrow temp(g(i))$

Question 3

a) Assume that the access function to lhs (*left hand side*) is injective. Show what the six possible dependence types become by cutting the vertices: flow dependency in input on S_k, anti and output, flow dependencies in output of S_k, anti and output.

b) Let G be the RDG of nested loops, and G' the graph obtained from G by cutting all its vertices. Show that a cycle in G' includes either only flow dependencies, or only output dependencies. In addition, show that any cycle of G' matches a cycle already in G.

c) Apply the cutting vertices method to the following code (we will only cut the vertices S_2 and S_3):

for $i = 4$ *à* N **do**
  S_1: $a(i+5) \leftarrow c(i-3) + b(2i+2)$
  S_2: $b(2i) \leftarrow a(i-1) + 1$
  S_3: $a(i) \leftarrow c(i+5) + 1$
  S_4: $c(i) \leftarrow b(2i-4)$

Part 3

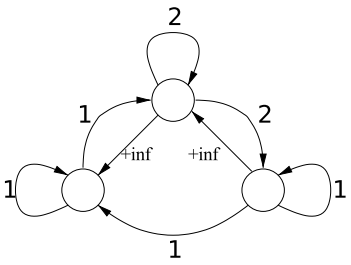
Allen and Kennedy algorithm

Question 4

a) Considering the following code, give the reduced dependency graph, and for each dependency, give its type (flow, anti, output) and its level. Apply the Allen and Kennedy algorithm to restore the nested loops and verify the nature of the obtained loops.

for $i = 1$ *à* N **do**
  for $j = 1$ *à* N **do**
    S_1: $a(i+1, j+1) \leftarrow a(i+1, j) + b(i, j+2)$
    S_2: $b(i+1, j) \leftarrow a(i+1, j-1) + b(i, j-1)$
    S_3: $a(i, j+2) \leftarrow b(i+1, j+1) - 1$

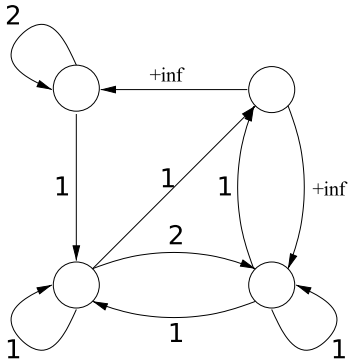
b) Give perfect nested loops, where all the dependencies are uniform, and of which the RDG is exactly the following:



Execute the Allen and Kennedy algorithm.

c) Give perfect nested loops, where all the dependencies are uniform, and which RDG is exactly the following:

Execute the Allen and Kennedy algorithm.


2
+inf
1
1
1
+inf
2
1
1