

CS 374 Spring 2016  
Homework number: 7  
Problem number: 3  
My name: Jozsef Morrissey  
My NetID: jtmorri2  
Group member names: Princewill Ibe, Bhavdeep K. Patel  
Group member NetID: pibe2, bkpatel3

Abstract:

The idea behind our algorithm is eliminating nodes only after all observers of their lifetime can be eliminated. We accomplish this by constructing a directional graph of birth death relations combined with a separate overlap relation connection. Only a subset of the sources of a given set of nodes could be safely eliminated, the set containing not overlap relations or only overlap relations with only the set of sources.

Type(1):  $x$  died before  $y$

Type(2):  $x$  overlapped  $y$

$P$  := a list containing type(1) and type(2) information

$G$  := an empty list of graph nodes

Node structure:

List *diedB4*

List *bornB4*

List *overLapped*

*isSource* = false

Determine Consistency()

Build Graph( $G$ )

$S$  := empty list to contain source nodes

While  $G$  is not empty

Find Sources( $S, G$ )

if Break down graph( $S, G$ ) //returns true if breakdown failed to remove a node

return data inconsistent

//End while loop

return data consistent

//End Determine Consistency

Build Graph( $G$ )

For each  $p$  in  $P$

$n_x$  = find  $x$  in  $G$

if  $x$  not in  $G$

$n_x$  = create new node

```

     $n_y$  = find  $y$  in  $G$ 
    if  $y$  not in  $G$ 
         $n_y$  = create new node

    append  $n_x$  to  $G$ 
    append  $n_y$  to  $G$ 

    if  $p$  is of type(1)
        append  $n_y$  to  $n_x.diedB4$ 
        append  $n_x$  to  $n_y.bornB4$ 

    if  $p$  is of type(2)
        append  $n_y$  to  $n_x.overlapped$ 
        append  $n_x$  to  $n_y.overlapped$ 

//End for loop
//End of Build Graph

```

Find Sources( $S$ ,  $G$ )

```

    For each  $n$  in  $G$ 

        if  $n.bornB4$  is empty
            append  $n$  to sources
             $n.isSource$  = true

//End for loop
//End Find Sources

```

Break down graph( $S$ ,  $G$ )

```

     $removedNode$  = false

    for each  $n$  in  $S$ 

        if  $n.overlapped$  is empty
             $removedNode$  = true
            remove the references to  $n$  in  $n$ 's neighbors and remove  $n$ 

        else
            Check All Overlaps are Connected to Sources( $removedNode$ ,  $n$ )

    if not  $removedNode$ 
        return true

    return false

//End of break down

```

Check All Overlaps are Connected to Sources( $removedNode$ ,  $n$ )

```

clear = true
for each z in n.overlapped

    if not z.isSource
        clear = false
        break

if clear
    remove the references to n in n's neighbors and remove n
    removedNode = true

//End for loop
//End Check All Overlaps are Connected to Sources

```