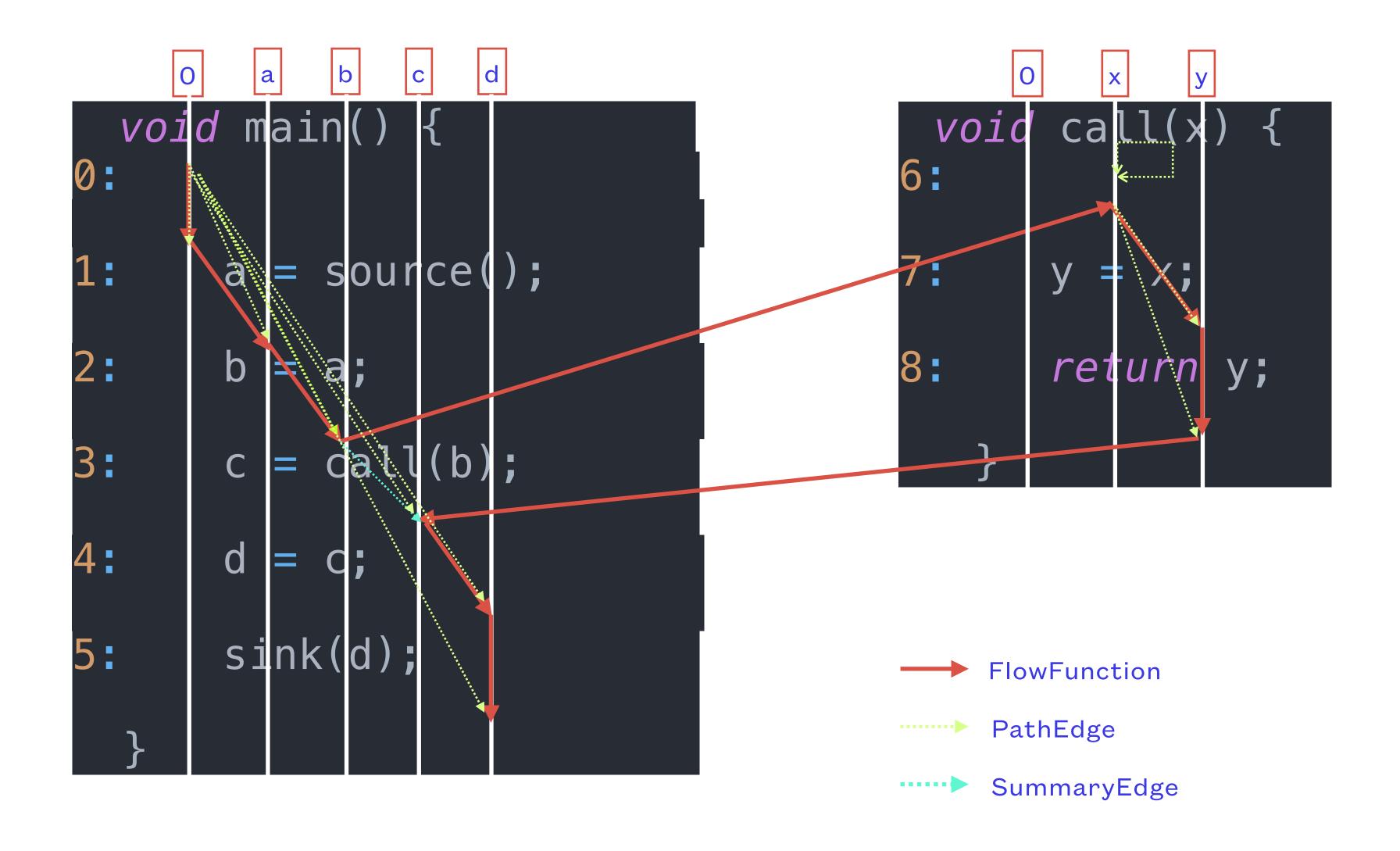
```
void main() {
0:
     a = source();
      b = a;
     c = call(b);
     d = c;
      sink(d);
```

```
void call(x) {
6:
7:      y = x;
8:      return y;
}
```



```
declare PathEdge, WorkList, SummaryEdge: global edge set
          algorithm Tabulate(G_{IP}^{\#})
          begin
             Let (N^{\#}, E^{\#}) = G_{IP}^{\#}
[1]
[2]
[3]
[4]
[5]
[6]
[7]
[8]
            PathEdge := \{\langle s_{main}, \mathbf{0} \rangle \rightarrow \langle s_{main}, \mathbf{0} \rangle\}
WorkList := \{\langle s_{main}, \mathbf{0} \rangle \rightarrow \langle s_{main}, \mathbf{0} \rangle\}
SummaryEdge := \emptyset
             ForwardTabulateSLRPs() for each n \in N^* do
                 X_n := \{ d_2 \in D \mid \exists d_1 \in (D \cup \{ \mathbf{0} \}) \text{ such that } \langle s_{procOf(n)}, d_1 \rangle \rightarrow \langle n, d_2 \rangle \in \text{PathEdge} \}
          end
          procedure Propagate(e)
          begin
            if e \notin PathEdge then Insert e into PathEdge; Insert e into WorkList fi
          end
          procedure ForwardTabulateSLRPs() SLRPs: Same Level Realizable Paths
          begin
            while WorkList \neq \emptyset do
                 Select and remove an edge \langle s_p, d_1 \rangle \rightarrow \langle n, d_2 \rangle from WorkList
[11]
[12]
                 switch n
                     case n \in Call_n:
[13]
                        for each d_3 such that \langle n, d_2 \rangle \rightarrow \langle s_{calledProc(n)}, d_3 \rangle \in E^{\#} do
                             Propagate(\langle s_{calledProc(n)}, d_3 \rangle \rightarrow \langle s_{calledProc(n)}, d_3 \rangle)
[16]
                        for each d_3 such that \langle n, d_2 \rangle \rightarrow \langle returnSite(n), d_3 \rangle \in (E^\# \cup SummaryEdge) do
[17]
[18]
                             Propagate(\langle s_p, d_1 \rangle \rightarrow \langle returnSite(n), d_3 \rangle)
[19]
[20]
                     end case
                                                                                  CallToReturnFlowFunction
[21]
[22]
[23]
                      case n = e_n:
                        for each c \in callers(p) do_
                                                                                                    CallFlowFunction
                                                                                                                                                                                              ReturnFlowFunction
                           for each d_4, d_5 such that \langle c, d_4 \rangle \to \langle s_p, d_1 \rangle \in E^{\#} and \langle e_p, d_2 \rangle \to \langle returnSite(c), d_5 \rangle \in E^{\#} do if \langle c, d_4 \rangle \to \langle returnSite(c), d_5 \rangle \notin SummaryEdge then
[24]
[25]
[26]
[27]
[28]
[29]
[30]
[31]
[32]
                                  Insert \langle c, d_4 \rangle \rightarrow \langle returnSite(c), d_5 \rangle into SummaryEdge for each d_3 such that \langle s_{procOf(c)}, d_3 \rangle \rightarrow \langle c, d_4 \rangle \in PathEdge do
Propagate(\langle s_{procOf(c)}, d_3 \rangle \rightarrow \langle returnSite(c), d_5 \rangle)
                     end case
                                                                                                               NormalFlowFunction
                    case n \in (N_p - Call_p - \{e_p\}):

for each \langle m, d_3 \rangle such that \langle n, d_2 \rangle \rightarrow \langle m, d_3 \rangle \in E^\# do

Propagate(\langle s_p, d_1 \rangle \rightarrow \langle m, d_3 \rangle)
[34]
[35]
[36]
[37]
                         od
                     end case
[38]
[39]
                 end switch
          end
```

```
void main() {
0:
1:     a = source();
2:     b = a;
3:     c = call(b);
4:     d = c;
5:     sink(d);
     } o a b c d
```

```
declare PathEdge, WorkList, SummaryEdge: global edge set
          algorithm Tabulate(G_{IP}^{\#})
          begin
              Let (N^{\#}, E^{\#}) = G_{IP}^{\#}
[1]
[2]
[3]
[4]
[5]
[6]
[7]
[8]
             PathEdge := \{\langle s_{main}, \mathbf{0} \rangle \rightarrow \langle s_{main}, \mathbf{0} \rangle\}

WorkList := \{\langle s_{main}, \mathbf{0} \rangle \rightarrow \langle s_{main}, \mathbf{0} \rangle\}

SummaryEdge := \emptyset
             Forward Tabulate SLRPs() for each n \in N^* do
                 X_n := \{ d_2 \in D \mid \exists d_1 \in (D \cup \{ \mathbf{0} \}) \text{ such that } \langle s_{procOf(n)}, d_1 \rangle \rightarrow \langle n, d_2 \rangle \in \text{PathEdge} \}
          end
          procedure Propagate(e)
          begin
             i\bar{f} e \not\in PathEdge then Insert e into PathEdge; Insert e into WorkList fi
           end
           procedure ForwardTabulateSLRPs()
          begin
             while WorkList \neq \emptyset do
[10]
                 Select and remove an edge \langle s_p, d_1 \rangle \rightarrow \langle n, d_2 \rangle from WorkList
[11]
                                                                                                                                                                                                                          PathEdge
[12]
                  switch n
[13]
                      case n \in Call_n:
                        for each d_3 such that \langle n, d_2 \rangle \rightarrow \langle s_{calledProc(n)}, d_3 \rangle \in E^{\#} do Propagate(\langle s_{calledProc(n)}, d_3 \rangle \rightarrow \langle s_{calledProc(n)}, d_3 \rangle)
 [15]
[16]
[17]
                         for each d_3 such that \langle n, d_2 \rangle \rightarrow \langle returnSite(n), d_3 \rangle \in (E^\# \cup SummaryEdge) do
[18]
                              Propagate(\langle s_n, d_1 \rangle \rightarrow \langle returnSite(n), d_3 \rangle)
[19]
                         od
[20]
                      end case
                                                                                                                                                                                                                           WorkList
[21]
                      case n = e_n:
[22]
[23]
                         for each c \in callers(p) do
                            for each d_4, d_5 such that \langle c, d_4 \rangle \to \langle s_p, d_1 \rangle \in E^{\#} and \langle e_p, d_2 \rangle \to \langle returnSite(c), d_5 \rangle \in E^{\#} do if \langle c, d_4 \rangle \to \langle returnSite(c), d_5 \rangle \notin SummaryEdge then
[24]
[25]
[26]
[27]
[28]
[29]
[30]
[31]
[32]
                                   Insert \langle c, d_4 \rangle \rightarrow \langle returnSite(c), d_5 \rangle into SummaryEdge

for each d_3 such that \langle s_{procOf(c)}, d_3 \rangle \rightarrow \langle c, d_4 \rangle \in PathEdge do

Propagate(\langle s_{procOf(c)}, d_3 \rangle \rightarrow \langle returnSite(c), d_5 \rangle)
                                    od
                             od
                      end case
[33]
[34]
[35]
                      case n \in (N_p - Call_p - \{e_p\}):

for each \langle m, d_3 \rangle such that \langle n, d_2 \rangle \rightarrow \langle m, d_3 \rangle \in E^{\#} do
                             Propagate(\langle s_p, d_1 \rangle \rightarrow \langle m, d_3 \rangle)
[36]
[37]
                      end case
[38]
[39]
                  end switch
             od
          end
```

```
void main() {
      a = source();
2:
      b = a;
      c = call(b);
3:
      d = c;
      sink(d);
   O
```

void call(x) {

y = x;

return y;

У

6:

Q

```
\langle s_{main}, zero \rangle \rightarrow \langle 1, zero \rangle
\langle s_{main}, zero \rangle \rightarrow \langle 1, zero \rangle
```

```
void main() {
         declare PathEdge, WorkList, SummaryEdge: global edge set
         algorithm Tabulate(G_{IP}^{\#})
         begin
            Let (N^{\#}, E^{\#}) = G_{IP}^{\#}
[1]
[2]
[3]
[4]
[5]
[6]
[7]
[8]
           PathEdge := \{\langle s_{main}, \mathbf{0} \rangle \rightarrow \langle s_{main}, \mathbf{0} \rangle\}

WorkList := \{\langle s_{main}, \mathbf{0} \rangle \rightarrow \langle s_{main}, \mathbf{0} \rangle\}

SummaryEdge := \emptyset
                                                                                                                                                                                                                                                      a = source();
                                                                                                                                                                                                                       2:
                                                                                                                                                                                                                                                      b = a;
            ForwardTabulateSLRPs()
            for each n \in N^* do
                                                                                                                                                                                                                                                      c = call(b);
                                                                                                                                                                                                                       3:
               X_n := \{ d_2 \in D \mid \exists d_1 \in (D \cup \{ \mathbf{0} \}) \text{ such that } \langle s_{procOf(n)}, d_1 \rangle \rightarrow \langle n, d_2 \rangle \in \text{PathEdge} \}
                                                                                                                                                                                                                                                      d = c;
         end
         procedure Propagate(e)
                                                                                                                                                                                                                                                      sink(d);
         begin
           i\bar{f} e \not\in PathEdge then Insert e into PathEdge; Insert e into WorkList fi
         end
                                                                                                                                                                                                                                        O
         procedure ForwardTabulateSLRPs()
         begin
           while WorkList \neq \emptyset do
               Select and remove an edge \langle s_p, d_1 \rangle \rightarrow \langle n, d_2 \rangle from WorkList
[11]
                                                                                                                                                                                               PathEdge
                                                                                                                                                                                                                                 \langle s_{main}, zero \rangle \rightarrow \langle 1, zero \rangle
[12]
               switch n
                   case n \in Call_n:
[13]
                     for each d_3 such that \langle n, d_2 \rangle \rightarrow \langle s_{calledProc(n)}, d_3 \rangle \in E^{\#} do Propagate(\langle s_{calledProc(n)}, d_3 \rangle \rightarrow \langle s_{calledProc(n)}, d_3 \rangle)
[16]
[17]
                      for each d_3 such that \langle n, d_2 \rangle \rightarrow \langle returnSite(n), d_3 \rangle \in (E^\# \cup SummaryEdge) do
[18]
                          Propagate(\langle s_p, d_1 \rangle \rightarrow \langle returnSite(n), d_3 \rangle)
[19]
                      od
[20]
                   end case
                                                                                                                                                                                                WorkList
[21]
                   case n = e_n:
[22]
[23]
                      for each c \in callers(p) do
                         for each d_4, d_5 such that \langle c, d_4 \rangle \to \langle s_p, d_1 \rangle \in E^{\#} and \langle e_p, d_2 \rangle \to \langle returnSite(c), d_5 \rangle \in E^{\#} do if \langle c, d_4 \rangle \to \langle returnSite(c), d_5 \rangle \notin SummaryEdge then
[24]
[25]
[26]
[27]
[28]
[29]
[30]
[31]
[32]
                               Insert \langle c, d_4 \rangle \rightarrow \langle returnSite(c), d_5 \rangle into SummaryEdge

for each d_3 such that \langle s_{procOf(c)}, d_3 \rangle \rightarrow \langle c, d_4 \rangle \in PathEdge do

Propagate(\langle s_{procOf(c)}, d_3 \rangle \rightarrow \langle returnSite(c), d_5 \rangle)
                                                                                                                                                                                                \langle s_{main}, zero \rangle \rightarrow \langle 1, zero \rangle
                                od
                   end case
[33]
[34]
[35]
                   case n \in (N_p - Call_p - \{e_p\}):

for each \langle m, d_3 \rangle such that \langle n, d_2 \rangle \rightarrow \langle m, d_3 \rangle \in E^{\#} do
                          Propagate(\langle s_p, d_1 \rangle \rightarrow \langle m, d_3 \rangle)
[36]
[37]
                   end case
[38]
[39]
               end switch
           od
         end
```

```
declare PathEdge, WorkList, SummaryEdge: global edge set
          algorithm Tabulate(G_{IP}^{\#})
          begin
             Let (N^{\#}, E^{\#}) = G_{IP}^{\#}
[1]
[2]
[3]
[4]
[5]
[6]
[7]
[8]
            PathEdge := \{\langle s_{main}, \mathbf{0} \rangle \rightarrow \langle s_{main}, \mathbf{0} \rangle\}
WorkList := \{\langle s_{main}, \mathbf{0} \rangle \rightarrow \langle s_{main}, \mathbf{0} \rangle\}
SummaryEdge := \emptyset
             ForwardTabulateSLRPs()
             for each n \in N^* do
                 X_n := \{ d_2 \in D \mid \exists d_1 \in (D \cup \{ \mathbf{0} \}) \text{ such that } \langle s_{procOf(n)}, d_1 \rangle \rightarrow \langle n, d_2 \rangle \in \text{PathEdge} \}
          end
          procedure Propagate(e)
          begin
             if e \notin PathEdge then Insert e into PathEdge; Insert e into WorkList fi
          end
          procedure ForwardTabulateSLRPs()
          begin
            while WorkList \neq \emptyset do
                 Select and remove an edge \langle s_p, d_1 \rangle \rightarrow \langle n, d_2 \rangle from WorkList
[11]
[12]
                 switch n
                    case n \in Call_n:
[13]
                        for each d_3 such that \langle n, d_2 \rangle \rightarrow \langle s_{calledProc(n)}, d_3 \rangle \in E^{\#} do Propagate(\langle s_{calledProc(n)}, d_3 \rangle \rightarrow \langle s_{calledProc(n)}, d_3 \rangle)
 [15]
[16]
                        for each d_3 such that \langle n, d_2 \rangle \rightarrow \langle returnSite(n), d_3 \rangle \in (E^\# \cup SummaryEdge) do
 [17]
[18]
                            Propagate(\langle s_p, d_1 \rangle \rightarrow \langle returnSite(n), d_3 \rangle)
[19]
                        od
[20]
                     end case
                                                     source污点的生成: 特殊处理成call-to-return-flow-function
[21]
                     case n = e_n:
 [22]
[23]
                        for each c \in callers(p) do
                           for each d_4, d_5 such that \langle c, d_4 \rangle \to \langle s_p, d_1 \rangle \in E^{\#} and \langle e_p, d_2 \rangle \to \langle returnSite(c), d_5 \rangle \in E^{\#} do if \langle c, d_4 \rangle \to \langle returnSite(c), d_5 \rangle \notin SummaryEdge then
[24]
[25]
[26]
[27]
[28]
[29]
[30]
[31]
[32]
                                  Insert \langle c, d_4 \rangle \rightarrow \langle returnSite(c), d_5 \rangle into SummaryEdge

for each d_3 such that \langle s_{procOf(c)}, d_3 \rangle \rightarrow \langle c, d_4 \rangle \in PathEdge do

Propagate(\langle s_{procOf(c)}, d_3 \rangle \rightarrow \langle returnSite(c), d_5 \rangle)
                            od
                     end case
                     case n \in (N_p - Call_p - \{e_p\}):

for each \langle m, d_3 \rangle such that \langle n, d_2 \rangle \rightarrow \langle m, d_3 \rangle \in E^{\#} do
[34]
[35]
                            Propagate(\langle s_p, d_1 \rangle \rightarrow \langle m, d_3 \rangle)
[36]
[37]
                     end case
[38]
[39]
                 end switch
             od
          end
```

```
void main() {
0:
      a = source();
2:
        = a;
      c = call(b);
      d = c;
      sink(d);
   О
```

```
void call(x) {
6:
      y = x;
      return y;
   Q
```

```
PathEdge
```

```
\langle s_{main}, zero \rangle \rightarrow \langle 1, zero \rangle
```

```
\langle s_{main}, zero \rangle \rightarrow \langle 1, zero \rangle
```

```
<1, zero>\rightarrow<2, a>
```

```
传递性: a -> b且b -> c =>
a -> c
```

```
Propagate(\langle s_{main}, zero \rangle \rightarrow \langle 2, a \rangle)
```

```
void call(x) {
                                                                                                                                                                                                                      void main() {
         declare PathEdge, WorkList, SummaryEdge: global edge set
         algorithm Tabulate(G_{IP}^{\#})
                                                                                                                                                                                                                                                                                                                         6:
                                                                                                                                                                                                             0:
        begin
           Let (N^{\#}, E^{\#}) = G_{IP}^{\#}
[1]
[2]
[3]
[4]
[5]
[6]
[7]
[8]
           PathEdge := \{\langle s_{main}, \mathbf{0} \rangle \rightarrow \langle s_{main}, \mathbf{0} \rangle\}

WorkList := \{\langle s_{main}, \mathbf{0} \rangle \rightarrow \langle s_{main}, \mathbf{0} \rangle\}

SummaryEdge := \emptyset
                                                                                                                                                                                                                                          a = source();
                                                                                                                                                                                                                                                                                                                                                        y = x;
                                                                                                                                                                                                                                                                                                                                                       return y;
                                                                                                                                                                                                              2:
                                                                                                                                                                                                                                                    = a;
           ForwardTabulateSLRPs() for each n \in N^* do
                                                                                                                                                                                                                                                                                                                                          Q
                                                                                                                                                                                                                                            c = call(b);
              X_n := \{ d_2 \in D \mid \exists d_1 \in (D \cup \{ \mathbf{0} \}) \text{ such that } \langle s_{procOf(n)}, d_1 \rangle \rightarrow \langle n, d_2 \rangle \in \text{PathEdge} \}
                                                                                                                                                                                                                                            d = c;
        end
         procedure Propagate(e)
                                                                                                                                                                                                                                            sink(d);
         begin
           if e \notin PathEdge then Insert e into PathEdge; Insert e into WorkList fi
         end
                                                                                                                                                                                                                               О
         procedure ForwardTabulateSLRPs()
         begin
           while WorkList \neq \emptyset do
[10]
               Select and remove an edge \langle s_p, d_1 \rangle \rightarrow \langle n, d_2 \rangle from WorkList
[11]
                                                                                                                                                                                                                                                                                                \langle s_{main}, zero \rangle \rightarrow \langle 2, a \rangle
                                                                                                                                                                                       PathEdge
                                                                                                                                                                                                                        \langle s_{main}, zero \rangle \rightarrow \langle 1, zero \rangle
[12]
               switch n
                  case n \in Call_n:
[13]
                    for each d_3 such that \langle n, d_2 \rangle \rightarrow \langle s_{calledProc(n)}, d_3 \rangle \in E^{\#} do Propagate(\langle s_{calledProc(n)}, d_3 \rangle \rightarrow \langle s_{calledProc(n)}, d_3 \rangle)
[15]
[16]
[17]
                     for each d_3 such that \langle n, d_2 \rangle \rightarrow \langle returnSite(n), d_3 \rangle \in (E^\# \cup SummaryEdge) do
[18]
                         Propagate(\langle s_p, d_1 \rangle \rightarrow \langle returnSite(n), d_3 \rangle)
[19]
                     od
[20]
                  end case
                                                                                                                                                                                                                            \langle s_{main}, zero \rangle \rightarrow \langle 2, a \rangle
                                                                                                                                                                                        WorkList
[21]
                   case n = e_n:
[22]
[23]
                     for each c \in callers(p) do
                        for each d_4, d_5 such that \langle c, d_4 \rangle \rightarrow \langle s_p, d_1 \rangle \in E^{\#} and \langle e_p, d_2 \rangle \rightarrow \langle returnSite(c), d_5 \rangle \in E^{\#} do if \langle c, d_4 \rangle \rightarrow \langle returnSite(c), d_5 \rangle \notin SummaryEdge then
[24]
[25]
[26]
[27]
[28]
[29]
[30]
[31]
[32]
                              Insert \langle c, d_4 \rangle \rightarrow \langle returnSite(c), d_5 \rangle into SummaryEdge

for each d_3 such that \langle s_{procOf(c)}, d_3 \rangle \rightarrow \langle c, d_4 \rangle \in PathEdge do

Propagate(\langle s_{procOf(c)}, d_3 \rangle \rightarrow \langle returnSite(c), d_5 \rangle)
                               od
                         od
                                                                                                                                                                                        Propagate(\langle s_{main}, zero \rangle \rightarrow \langle 2, a \rangle)
                  end case
[33]
[34]
[35]
                  case n \in (N_p - Call_p - \{e_p\}):

for each \langle m, d_3 \rangle such that \langle n, d_2 \rangle \rightarrow \langle m, d_3 \rangle \in E^{\#} do
                         Propagate(\langle s_p, d_1 \rangle \rightarrow \langle m, d_3 \rangle)
[36]
[37]
                  end case
[38]
[39]
               end switch
           od
        end
```

```
declare PathEdge, WorkList, SummaryEdge: global edge set
           algorithm Tabulate(G_{IP}^{\#})
           begin
              Let (N^{\#}, E^{\#}) = G_{IP}^{\#}
[1]
[2]
[3]
[4]
[5]
[6]
[7]
[8]
             PathEdge := \{\langle s_{main}, \mathbf{0} \rangle \rightarrow \langle s_{main}, \mathbf{0} \rangle\}

WorkList := \{\langle s_{main}, \mathbf{0} \rangle \rightarrow \langle s_{main}, \mathbf{0} \rangle\}

SummaryEdge := \emptyset

ForwardTabulateSLRPs()

for each n \in N^* do
                  X_n := \{ d_2 \in D \mid \exists d_1 \in (D \cup \{ \mathbf{0} \}) \text{ such that } \langle s_{procOf(n)}, d_1 \rangle \rightarrow \langle n, d_2 \rangle \in \text{PathEdge} \}
           end
           procedure Propagate(e)
           begin
              if e \notin PathEdge then Insert e into PathEdge; Insert e into WorkList fi
            end
            procedure ForwardTabulateSLRPs()
           begin
              while WorkList \neq \emptyset do
[10]
                  Select and remove an edge \langle s_p, d_1 \rangle \rightarrow \langle n, d_2 \rangle from WorkList
[11]
[12]
                   switch n
[13]
                       case n \in Call_n:
                          for each d_3 such that \langle n, d_2 \rangle \rightarrow \langle s_{calledProc(n)}, d_3 \rangle \in E^{\#} do Propagate(\langle s_{calledProc(n)}, d_3 \rangle \rightarrow \langle s_{calledProc(n)}, d_3 \rangle)
 [15]
 [16]
                           for each d_3 such that \langle n, d_2 \rangle \rightarrow \langle returnSite(n), d_3 \rangle \in (E^{\#} \cup SummaryEdge) do
 [17]
 [18]
                               Propagate(\langle s_p, d_1 \rangle \rightarrow \langle returnSite(n), d_3 \rangle)
[19]
                           od
[20]
                       end case
[21]
                        case n = e_n:
[22]
[23]
                           for each c \in callers(p) do
                              for each d_4, d_5 such that \langle c, d_4 \rangle \to \langle s_p, d_1 \rangle \in E^{\#} and \langle e_p, d_2 \rangle \to \langle returnSite(c), d_5 \rangle \in E^{\#} do if \langle c, d_4 \rangle \to \langle returnSite(c), d_5 \rangle \notin SummaryEdge then
 [24]
[25]
[26]
[27]
[28]
[29]
[30]
[31]
[32]
                                     Insert \langle c, d_4 \rangle \rightarrow \langle returnSite(c), d_5 \rangle into SummaryEdge

for each d_3 such that \langle s_{procOf(c)}, d_3 \rangle \rightarrow \langle c, d_4 \rangle \in PathEdge do

Propagate(\langle s_{procOf(c)}, d_3 \rangle \rightarrow \langle returnSite(c), d_5 \rangle)
                                       od
                               od
                       end case
[33]
[34]
[35]
                       case n \in (N_p - Call_p - \{e_p\}):

for each \langle m, d_3 \rangle such that \langle n, d_2 \rangle \rightarrow \langle m, d_3 \rangle \in E^{\#} do
                               Propagate(\langle s_p, d_1 \rangle \rightarrow \langle m, d_3 \rangle)
 [36]
[37]
                       end case
[38]
[39]
                   end switch
              od
           end
```

```
void main() {
0:
1:     a = source();
2:     b = a;
3:     c = call(b);
4:     d = c;
5:     sink(d);
}     o     a     b     c     d
```

$$\langle s_{main}, zero \rangle \rightarrow \langle 1, zero \rangle$$

$$\langle s_{main}, zero \rangle \rightarrow \langle 2, a \rangle$$

```
\langle s_{main}, zero \rangle \rightarrow \langle 2, a \rangle
```

```
void call(x) {
                                                                                                                                                                                                        void main() {
        declare PathEdge, WorkList, SummaryEdge: global edge set
        algorithm Tabulate(G_{IP}^{\#})
                                                                                                                                                                                                                                                                                                    6:
                                                                                                                                                                                                0:
        begin
           Let (N^{\#}, E^{\#}) = G_{IP}^{\#}
[1]
[2]
[3]
[4]
[5]
[6]
[7]
[8]
          PathEdge := \{\langle s_{main}, \mathbf{0} \rangle \rightarrow \langle s_{main}, \mathbf{0} \rangle\}

WorkList := \{\langle s_{main}, \mathbf{0} \rangle \rightarrow \langle s_{main}, \mathbf{0} \rangle\}

SummaryEdge := \emptyset
                                                                                                                                                                                                                                                                                                    7:
                                                                                                                                                                                                                                  = source();
                                                                                                                                                                                                                                                                                                                                 y = x;
                                                                                                                                                                                                                                                                                                                                return y;
                                                                                                                                                                                                2:
          ForwardTabulateSLRPs() for each n \in N^* do
                                                                                                                                                                                                                                                                                                                    } o
                                                                                                                                                                                                                            c = call(b);
                                                                                                                                                                                                3:
              X_n := \{ d_2 \in D \mid \exists d_1 \in (D \cup \{ \mathbf{0} \}) \text{ such that } \langle s_{procOf(n)}, d_1 \rangle \rightarrow \langle n, d_2 \rangle \in \text{PathEdge} \}
                                                                                                                                                                                                                            d = c;
        end
        procedure Propagate(e)
                                                                                                                                                                                                                            sink(d);
        begin
          if e \notin PathEdge then Insert e into PathEdge; Insert e into WorkList fi
         end
                                                                                                                                                                                                                О
         procedure ForwardTabulateSLRPs()
        begin
          while WorkList \neq \emptyset do
[10]
              Select and remove an edge \langle s_p, d_1 \rangle \rightarrow \langle n, d_2 \rangle from WorkList
[11]
                                                                                                                                                                                                                                                                            \langle s_{main}, zero \rangle \rightarrow \langle 2, a \rangle
                                                                                                                                                                           PathEdge
                                                                                                                                                                                                          \langle s_{main}, zero \rangle \rightarrow \langle 1, zero \rangle
[12]
              switch n
                 case n \in Call_n:
[13]
                   for each d_3 such that \langle n, d_2 \rangle \rightarrow \langle s_{calledProc(n)}, d_3 \rangle \in E^{\#} do Propagate(\langle s_{calledProc(n)}, d_3 \rangle \rightarrow \langle s_{calledProc(n)}, d_3 \rangle)
 [15]
[16]
[17]
                    for each d_3 such that \langle n, d_2 \rangle \rightarrow \langle returnSite(n), d_3 \rangle \in (E^\# \cup SummaryEdge) do
[18]
                       Propagate(\langle s_n, d_1 \rangle \rightarrow \langle returnSite(n), d_3 \rangle)
[19]
                    od
[20]
                 end case
                                                                                                                                                                           WorkList
[21]
                  case n = e_n:
 [22]
[23]
                    for each c \in callers(p) do
                      for each d_4, d_5 such that \langle c, d_4 \rangle \rightarrow \langle s_p, d_1 \rangle \in E^{\#} and \langle e_p, d_2 \rangle \rightarrow \langle returnSite(c), d_5 \rangle \in E^{\#} do if \langle c, d_4 \rangle \rightarrow \langle returnSite(c), d_5 \rangle \not\in SummaryEdge then
[24]
[25]
[26]
[27]
[28]
[29]
[30]
[31]
[32]
                            Insert \langle c, d_4 \rangle \rightarrow \langle returnSite(c), d_5 \rangle into SummaryEdge

for each d_3 such that \langle s_{procOf(c)}, d_3 \rangle \rightarrow \langle c, d_4 \rangle \in PathEdge do

Propagate(\langle s_{procOf(c)}, d_3 \rangle \rightarrow \langle returnSite(c), d_5 \rangle)
                                                                                                                                                                              \langle s_{main}, zero \rangle \rightarrow \langle 2, a \rangle
                                                                                                                                                                                                                                                         <2, a> \rightarrow <3, b>
                             od
                       od
                                                                                                                                                                             Propagate (\langle s_{main}, zero \rangle \rightarrow \langle 3, b \rangle)
                 end case
[33]
                 case n \in (N_n - Call_n - \{e_n\}):
                    for each \langle m, d_3 \rangle such that \langle n, d_2 \rangle \rightarrow \langle m, d_3 \rangle \in E^{\#} do
 [34]
 [35]
                       Propagate(\langle s_p, d_1 \rangle \rightarrow \langle m, d_3 \rangle)
 [36]
                    od
[37]
                 end case
[38]
              end switch
[39]
                                                                                          NormalFlowFunction
                                                                                                                             实际上当前Normal Flow Function会存在另一条数据流<2, a> → <3, a>, 导致生成PathEdge<s<sub>main</sub>, zero> → <3, a>
        end
                                                                                                                                          这里限于篇幅以及算法演示的时间,对于这类边不予演示;对于最终Source->Sink并不影响
```

```
void call(x) {
                                                                                                                                                                                                                         void main() {
         declare PathEdge, WorkList, SummaryEdge: global edge set
         algorithm Tabulate(G_{IP}^{\#})
                                                                                                                                                                                                                                                                                                                             6:
                                                                                                                                                                                                                0:
        begin
           Let (N^{\#}, E^{\#}) = G_{IP}^{\#}
[1]
[2]
[3]
[4]
[5]
[6]
[7]
[8]
           PathEdge := \{\langle s_{main}, \mathbf{0} \rangle \rightarrow \langle s_{main}, \mathbf{0} \rangle\}

WorkList := \{\langle s_{main}, \mathbf{0} \rangle \rightarrow \langle s_{main}, \mathbf{0} \rangle\}

SummaryEdge := \emptyset
                                                                                                                                                                                                                                                        = source();
                                                                                                                                                                                                                                                                                                                                                             y = x;
                                                                                                                                                                                                                                                                                                                                                            return y;
                                                                                                                                                                                                                 2:
           ForwardTabulateSLRPs() for each n \in N^* do
                                                                                                                                                                                                                                                                                                                                               Q
                                                                                                                                                                                                                                               c = call(b);
               X_n := \{ d_2 \in D \mid \exists d_1 \in (D \cup \{ \mathbf{0} \}) \text{ such that } \langle s_{procOf(n)}, d_1 \rangle \rightarrow \langle n, d_2 \rangle \in \text{PathEdge} \}
                                                                                                                                                                                                                                               d = c;
        end
         procedure Propagate(e)
                                                                                                                                                                                                                                               sink(d);
        begin
           if e \notin PathEdge then Insert e into PathEdge; Insert e into WorkList fi
                                                                                                                                                                                                                                  О
         procedure ForwardTabulateSLRPs()
         begin
           while WorkList \neq \emptyset do
[10]
               Select and remove an edge \langle s_p, d_1 \rangle \rightarrow \langle n, d_2 \rangle from WorkList
[11]
                                                                                                                                                                                         PathEdge
                                                                                                                                                                                                                                                                                                    \langle s_{main}, zero \rangle \rightarrow \langle 2, a \rangle
                                                                                                                                                                                                                           \langle s_{main}, zero \rangle \rightarrow \langle 1, zero \rangle
[12]
               switch n
                  case n \in Call_n:
[13]
                                                                                                                                                                                                                            \langle s_{main}, zero \rangle \rightarrow \langle 3, b \rangle
                     for each d_3 such that \langle n, d_2 \rangle \rightarrow \langle s_{calledProc(n)}, d_3 \rangle \in E^{\#} do Propagate(\langle s_{calledProc(n)}, d_3 \rangle \rightarrow \langle s_{calledProc(n)}, d_3 \rangle)
[15]
[16]
[17]
                     for each d_3 such that \langle n, d_2 \rangle \rightarrow \langle returnSite(n), d_3 \rangle \in (E^\# \cup SummaryEdge) do
[18]
                         Propagate(\langle s_n, d_1 \rangle \rightarrow \langle returnSite(n), d_3 \rangle)
[19]
                     od
[20]
                  end case
                                                                                                                                                                                                                            \langle s_{main}, zero \rangle \rightarrow \langle 3, b \rangle
                                                                                                                                                                                          WorkList
[21]
                   case n = e_n:
[22]
[23]
                     for each c \in callers(p) do
                        for each d_4, d_5 such that \langle c, d_4 \rangle \rightarrow \langle s_p, d_1 \rangle \in E^{\#} and \langle e_p, d_2 \rangle \rightarrow \langle returnSite(c), d_5 \rangle \in E^{\#} do if \langle c, d_4 \rangle \rightarrow \langle returnSite(c), d_5 \rangle \notin SummaryEdge then
[24]
[25]
[26]
[27]
[28]
[29]
[30]
[31]
[32]
                              Insert \langle c, d_4 \rangle \rightarrow \langle returnSite(c), d_5 \rangle into SummaryEdge

for each d_3 such that \langle s_{procOf(c)}, d_3 \rangle \rightarrow \langle c, d_4 \rangle \in PathEdge do

Propagate(\langle s_{procOf(c)}, d_3 \rangle \rightarrow \langle returnSite(c), d_5 \rangle)
                               od
                         od
                                                                                                                                                                                            Propagate(\langle s_{main}, zero \rangle \rightarrow \langle 3, b \rangle)
                  end case
[33]
[34]
[35]
                  case n \in (N_p - Call_p - \{e_p\}):
for each \langle m, d_3 \rangle such that \langle n, d_2 \rangle \rightarrow \langle m, d_3 \rangle \in E^{\#} do
                         Propagate(\langle s_p, d_1 \rangle \rightarrow \langle m, d_3 \rangle)
[36]
[37]
                  end case
[38]
[39]
               end switch
           od
        end
```

```
void call(x) {
                                                                                                                                                                                                     void main() {
        declare PathEdge, WorkList, SummaryEdge: global edge set
        algorithm Tabulate(G_{IP}^{\#})
                                                                                                                                                                                                                                                                                                6:
                                                                                                                                                                                             0:
        begin
          Let (N^{\#}, E^{\#}) = G_{IP}^{\#}
[1]
[2]
[3]
[4]
[5]
[6]
[7]
[8]
          PathEdge := \{\langle s_{main}, \mathbf{0} \rangle \rightarrow \langle s_{main}, \mathbf{0} \rangle\}
WorkList := \{\langle s_{main}, \mathbf{0} \rangle \rightarrow \langle s_{main}, \mathbf{0} \rangle\}
SummaryEdge := \emptyset
                                                                                                                                                                                                                                  = source();
                                                                                                                                                                                                                                                                                                                             y = x;
                                                                                                                                                                                                                                                                                                8:
                                                                                                                                                                                                                                                                                                                            return y;
                                                                                                                                                                                              2:
                                                                                                                                                                                                                                  a;
          ForwardTabulateSLRPs()
          for each n \in N^* do
                                                                                                                                                                                                                         c = call(b);
                                                                                                                                                                                                                                                                                                                } o
             X_n := \{ d_2 \in D \mid \exists d_1 \in (D \cup \{ \mathbf{0} \}) \text{ such that } \langle s_{procOf(n)}, d_1 \rangle \rightarrow \langle n, d_2 \rangle \in \text{PathEdge} \}
                                                                                                                                                                                                                         d = c;
        end
        procedure Propagate(e)
                                                                                                                                                                                                                         sink(d);
        begin
          if e \notin PathEdge then Insert e into PathEdge; Insert e into WorkList fi
        end
                                                                                                                                                                                                             О
        procedure ForwardTabulateSLRPs()
        begin
          while WorkList \neq \emptyset do
             Select and remove an edge \langle s_p, d_1 \rangle \rightarrow \langle n, d_2 \rangle from WorkList
[11]
                                                                                                                                                                        PathEdge
                                                                                                                                                                                                                                                                         \langle s_{main}, zero \rangle \rightarrow \langle 2, a \rangle
                                                                                                                                                                                                       \langle s_{main}, zero \rangle \rightarrow \langle 1, zero \rangle
[12]
              switch n
                 case n \in Call_n:
[13]
                                                                                                                        CallFlowFunction
                                                                                                                                                                                                        \langle s_{main}, zero \rangle \rightarrow \langle 3, b \rangle
                   for each d_3 such that \langle n, d_2 \rangle \rightarrow \langle s_{calledProc(n)}, d_3 \rangle \in E^{\#} do Propagate(\langle s_{calledProc(n)}, d_3 \rangle \rightarrow \langle s_{calledProc(n)}, d_3 \rangle)
[15]
[16]
                   od
                   for each d_3 such that \langle n, d_2 \rangle \rightarrow \langle returnSite(n), d_3 \rangle \in (E^{\#} \cup SummaryEdge) do
[17]
[18]
                       Propagate(\langle s_p, d_1 \rangle \rightarrow \langle returnSite(n), d_3 \rangle)
[19]
                   od
[20]
                 end case
                                                                                                                                                                         WorkList
[21]
                 case n = e_n:
[22]
[23]
                   for each c \in callers(p) do
                      for each d_4, d_5 such that \langle c, d_4 \rangle \to \langle s_p, d_1 \rangle \in E^{\#} and \langle e_p, d_2 \rangle \to \langle returnSite(c), d_5 \rangle \in E^{\#} do if \langle c, d_4 \rangle \to \langle returnSite(c), d_5 \rangle \not\in SummaryEdge then
[24]
[25]
[26]
[27]
[28]
[29]
[30]
[31]
[32]
                           Insert \langle c, d_4 \rangle \rightarrow \langle returnSite(c), d_5 \rangle into SummaryEdge for each d_3 such that \langle s_{procOf(c)}, d_3 \rangle \rightarrow \langle c, d_4 \rangle \in PathEdge do Propagate(\langle s_{procOf(c)}, d_3 \rangle \rightarrow \langle returnSite(c), d_5 \rangle)
                                                                                                                                                                                                                                            <3, b> \rightarrow <s_{call}, x>
                                                                                                                                                                          \langle s_{main}, zero \rangle \rightarrow \langle 3, b \rangle
                       od
                                                                                                                                                                          Propagate(
                                                                                                                                                                                                                  \langle s_{call}, x \rangle \rightarrow \langle s_{call}, x \rangle
                 end case
                 case n \in (N_n - Call_n - \{e_n\}):
                   for each \langle m, d_3 \rangle such that \langle n, d_2 \rangle \rightarrow \langle m, d_3 \rangle \in E^{\#} do
[34]
                      Propagate(\langle s_p, d_1 \rangle \rightarrow \langle m, d_3 \rangle)
[35]
                                                                                                                                                                                                                                       导致被调用函数内的数据流传播是上下文无关的
                                                                                                                                                                                                         自环边!!!
[36]
[37]
                 end case
[38]
              end switch
[39]
```

end

```
declare PathEdge, WorkList, SummaryEdge: global edge set
          algorithm Tabulate(G_{IP}^{\#})
          begin
              Let (N^{\#}, E^{\#}) = G_{IP}^{\#}
[1]
[2]
[3]
[4]
[5]
[6]
[7]
[8]
             PathEdge := \{\langle s_{main}, \mathbf{0} \rangle \rightarrow \langle s_{main}, \mathbf{0} \rangle\}

WorkList := \{\langle s_{main}, \mathbf{0} \rangle \rightarrow \langle s_{main}, \mathbf{0} \rangle\}

SummaryEdge := \emptyset
             ForwardTabulateSLRPs() for each n \in N^* do
                 X_n := \{ d_2 \in D \mid \exists d_1 \in (D \cup \{ \mathbf{0} \}) \text{ such that } \langle s_{procOf(n)}, d_1 \rangle \rightarrow \langle n, d_2 \rangle \in \text{PathEdge} \}
          end
          procedure Propagate(e)
          begin
             if e \notin PathEdge then Insert e into PathEdge; Insert e into WorkList fi
           end
           procedure ForwardTabulateSLRPs()
          begin
             while WorkList \neq \emptyset do
[10]
                 Select and remove an edge \langle s_p, d_1 \rangle \rightarrow \langle n, d_2 \rangle from WorkList
[11]
[12]
                  switch n
                      case n \in Call_n:
[13]
                         for each d_3 such that \langle n, d_2 \rangle \rightarrow \langle s_{calledProc(n)}, d_3 \rangle \in E^{\#} do
Propagate(\langle s_{calledProc(n)}, d_3 \rangle \rightarrow \langle s_{calledProc(n)}, d_3 \rangle)
 [15]
                                                                                                                                           CallToReturnFlowFunction
[16]
                         for each d_3 such that \langle n, d_2 \rangle \rightarrow \langle returnSite(n), d_3 \rangle \in (E^{\#} \cup SummaryEdge) do
 [17]
[18]
                             Propagate(\langle s_p, d_1 \rangle \rightarrow \langle returnSite(n), d_3 \rangle)
[19]
                         od
[20]
                      end case
[21]
                      case n = e_n:
 [22]
[23]
                         for each c \in callers(p) do
                             for each d_4, d_5 such that \langle c, d_4 \rangle \to \langle s_p, d_1 \rangle \in E^{\#} and \langle e_p, d_2 \rangle \to \langle returnSite(c), d_5 \rangle \in E^{\#} do if \langle c, d_4 \rangle \to \langle returnSite(c), d_5 \rangle \notin SummaryEdge then
[24]
[25]
[26]
[27]
[28]
[29]
[30]
[31]
[32]
                                    Insert \langle c, d_4 \rangle \rightarrow \langle returnSite(c), d_5 \rangle into SummaryEdge

for each d_3 such that \langle s_{procOf(c)}, d_3 \rangle \rightarrow \langle c, d_4 \rangle \in PathEdge do

Propagate(\langle s_{procOf(c)}, d_3 \rangle \rightarrow \langle returnSite(c), d_5 \rangle)
                                     od
                              od
                      end case
[33]
[34]
[35]
                      case n \in (N_p - Call_p - \{e_p\}):

for each \langle m, d_3 \rangle such that \langle n, d_2 \rangle \rightarrow \langle m, d_3 \rangle \in E^{\#} do
                              Propagate(\langle s_p, d_1 \rangle \rightarrow \langle m, d_3 \rangle)
[36]
[37]
                      end case
[38]
[39]
                  end switch
             od
          end
```

```
void main() {
0:
1:     a = source();
2:     b = a;
3:     c = call(b);
4:     d = c;
5:     sink(d);
}     o     a     b     c     d
```

$$\langle s_{main}, zero \rangle \rightarrow \langle 1, zero \rangle$$
  $\langle s_{main}, zero \rangle \rightarrow \langle 2, a \rangle$   $\langle s_{main}, zero \rangle \rightarrow \langle 3, b \rangle$   $\langle s_{call}, x \rangle \rightarrow \langle s_{call}, x \rangle$ 

WorkList

$$\langle s_{call}, x \rangle \rightarrow \langle s_{call}, x \rangle$$

 $\langle s_{main}, zero \rangle \rightarrow \langle 3, b \rangle$ 

```
void main() {
                                                                                                                                                                                                                                                                                                                                            void call(x) {
         declare PathEdge, WorkList, SummaryEdge: global edge set
         algorithm Tabulate(G_{IP}^{\#})
                                                                                                                                                                                                                                                                                                                                     6:
                                                                                                                                                                                                                     0:
         begin
            Let (N^{\#}, E^{\#}) = G_{IP}^{\#}
[1]
[2]
[3]
[4]
[5]
[6]
[7]
[8]
           PathEdge := \{\langle s_{main}, \mathbf{0} \rangle \rightarrow \langle s_{main}, \mathbf{0} \rangle\}
WorkList := \{\langle s_{main}, \mathbf{0} \rangle \rightarrow \langle s_{main}, \mathbf{0} \rangle\}
SummaryEdge := \emptyset
                                                                                                                                                                                                                                                               = source();
                                                                                                                                                                                                                                                                                                                                                                    return y;
                                                                                                                                                                                                                                                                                                                                     8:
                                                                                                                                                                                                                      2:
                                                                                                                                                                                                                                                               a;
           ForwardTabulateSLRPs() for each n \in N^* do
                                                                                                                                                                                                                                                     c = call(b);
                                                                                                                                                                                                                                                                                                                                                       } o
               X_n := \{ d_2 \in D \mid \exists d_1 \in (D \cup \{ \mathbf{0} \}) \text{ such that } \langle s_{procOf(n)}, d_1 \rangle \rightarrow \langle n, d_2 \rangle \in \text{PathEdge} \}
                                                                                                                                                                                                                                                     d = c;
         end
         procedure Propagate(e)
                                                                                                                                                                                                                                                     sink(d);
         begin
           if e \notin PathEdge then Insert e into PathEdge; Insert e into WorkList fi
         end
                                                                                                                                                                                                                                       О
         procedure ForwardTabulateSLRPs()
         begin
           while WorkList \neq \emptyset do
[10]
               Select and remove an edge \langle s_n, d_1 \rangle \rightarrow \langle n, d_2 \rangle from WorkList
[11]
                                                                                                                                                                                              PathEdge
                                                                                                                                                                                                                                                                                                           \langle s_{main}, zero \rangle \rightarrow \langle 2, a \rangle
[12]
                                                                                                                                                                                                                                 \langle s_{main}, zero \rangle \rightarrow \langle 1, zero \rangle
               switch n
[13]
                   case n \in Call_n:
                                                                                                                                                                                                                                  \langle s_{main}, zero \rangle \rightarrow \langle 3, b \rangle |\langle s_{call}, x \rangle \rightarrow \langle s_{call}, x \rangle
                      for each d_3 such that \langle n, d_2 \rangle \rightarrow \langle s_{calledProc(n)}, d_3 \rangle \in E^{\#} do Propagate(\langle s_{calledProc(n)}, d_3 \rangle \rightarrow \langle s_{calledProc(n)}, d_3 \rangle)
[15]
[16]
[17]
                      for each d_3 such that \langle n, d_2 \rangle \rightarrow \langle returnSite(n), d_3 \rangle \in (E^\# \cup SummaryEdge) do
[18]
                          Propagate(\langle s_p, d_1 \rangle \rightarrow \langle returnSite(n), d_3 \rangle)
[19]
                      od
[20]
                   end case
                                                                                                                                                                                               WorkList
[21]
                   case n = e_n:
[22]
[23]
                      for each c \in callers(p) do
                         for each d_4, d_5 such that \langle c, d_4 \rangle \rightarrow \langle s_p, d_1 \rangle \in E^{\#} and \langle e_p, d_2 \rangle \rightarrow \langle returnSite(c), d_5 \rangle \in E^{\#} do if \langle c, d_4 \rangle \rightarrow \langle returnSite(c), d_5 \rangle \notin SummaryEdge then
[24]
[25]
[26]
[27]
[28]
[29]
[30]
[31]
[32]
                               Insert \langle c, d_4 \rangle \rightarrow \langle returnSite(c), d_5 \rangle into SummaryEdge

for each d_3 such that \langle s_{procOf(c)}, d_3 \rangle \rightarrow \langle c, d_4 \rangle \in PathEdge do

Propagate(\langle s_{procOf(c)}, d_3 \rangle \rightarrow \langle returnSite(c), d_5 \rangle)
                                                                                                                                                                                                                                                                 \langle s_{call}, x \rangle \rightarrow \langle 7, x \rangle
                                                                                                                                                                                               \langle s_{call}, x \rangle \rightarrow \langle s_{call}, x \rangle
                                od
                          od
                                                                                                                                                                                                Propagate(\langle s_{call}, x \rangle \rightarrow \langle 7, x \rangle)
                   end case
                  case n \in (N_p - Call_p - \{e_p\}):

for each \langle m, d_3 \rangle such that \langle n, d_2 \rangle \rightarrow \langle m, d_3 \rangle \in E^{\#} do

Propagate(\langle s_p, d_1 \rangle \rightarrow \langle m, d_3 \rangle)
[34]
[35]
[36]
[37]
                      od
                  end case
[38]
[39]
               end switch
                                                                                                          NormalFlowFunction
         end
```

```
declare PathEdge, WorkList, SummaryEdge: global edge set
           algorithm Tabulate(G_{IP}^{\#})
          begin
              Let (N^{\#}, E^{\#}) = G_{IP}^{\#}
[1]
[2]
[3]
[4]
[5]
[6]
[7]
[8]
            PathEdge := \{\langle s_{main}, \mathbf{0} \rangle \rightarrow \langle s_{main}, \mathbf{0} \rangle\}

WorkList := \{\langle s_{main}, \mathbf{0} \rangle \rightarrow \langle s_{main}, \mathbf{0} \rangle\}

SummaryEdge := \emptyset

ForwardTabulateSLRPs()

for each n \in N^* do
                  X_n := \{ d_2 \in D \mid \exists d_1 \in (D \cup \{ \mathbf{0} \}) \text{ such that } \langle s_{procOf(n)}, d_1 \rangle \rightarrow \langle n, d_2 \rangle \in \text{PathEdge} \}
          end
          procedure Propagate(e)
          begin
             if e \notin PathEdge then Insert e into PathEdge; Insert e into WorkList fi
           end
           procedure ForwardTabulateSLRPs()
           begin
              while WorkList \neq \emptyset do
[10]
                  Select and remove an edge \langle s_p, d_1 \rangle \rightarrow \langle n, d_2 \rangle from WorkList
[11]
[12]
                  switch n
[13]
                      case n \in Call_n:
                          for each d_3 such that \langle n, d_2 \rangle \rightarrow \langle s_{calledProc(n)}, d_3 \rangle \in E^{\#} do Propagate(\langle s_{calledProc(n)}, d_3 \rangle \rightarrow \langle s_{calledProc(n)}, d_3 \rangle)
 [15]
[16]
[17]
                          for each d_3 such that \langle n, d_2 \rangle \rightarrow \langle returnSite(n), d_3 \rangle \in (E^\# \cup SummaryEdge) do
                               Propagate(\langle s_p, d_1 \rangle \rightarrow \langle returnSite(n), d_3 \rangle)
[18]
[19]
                          od
[20]
                      end case
[21]
                       case n = e_n:
[22]
[23]
                          for each c \in callers(p) do
                              for each d_4, d_5 such that \langle c, d_4 \rangle \to \langle s_p, d_1 \rangle \in E^{\#} and \langle e_p, d_2 \rangle \to \langle returnSite(c), d_5 \rangle \in E^{\#} do if \langle c, d_4 \rangle \to \langle returnSite(c), d_5 \rangle \notin SummaryEdge then
[24]
[25]
[26]
[27]
[28]
[29]
[30]
[31]
[32]
                                     Insert \langle c, d_4 \rangle \rightarrow \langle returnSite(c), d_5 \rangle into SummaryEdge for each d_3 such that \langle s_{procOf(c)}, d_3 \rangle \rightarrow \langle c, d_4 \rangle \in PathEdge do Propagate(\langle s_{procOf(c)}, d_3 \rangle \rightarrow \langle returnSite(c), d_5 \rangle)
                                      od
                              od
                      end case
[33]
[34]
[35]
                      case n \in (N_p - Call_p - \{e_p\}):

for each \langle m, d_3 \rangle such that \langle n, d_2 \rangle \rightarrow \langle m, d_3 \rangle \in E^{\#} do
                              Propagate(\langle s_p, d_1 \rangle \rightarrow \langle m, d_3 \rangle)
[36]
[37]
                      end case
[38]
[39]
                  end switch
              od
          end
```

```
void main() {
0:
1:     a = source();
2:     b = a;
3:     c = call(b);
4:     d = c;
5:     sink(d);
}     o     a     b     c     d
```

$$\langle s_{main}, zero \rangle \rightarrow \langle 1, zero \rangle$$
  $\langle s_{main}, zero \rangle \rightarrow \langle 2, a \rangle$   $\langle s_{main}, zero \rangle \rightarrow \langle 3, b \rangle$   $\langle s_{call}, x \rangle \rightarrow \langle s_{call}, x \rangle$   $\langle s_{call}, x \rangle \rightarrow \langle 7, x \rangle$ 

$$\langle s_{call}, x \rangle \rightarrow \langle 7, x \rangle$$

```
void main() {
                                                                                                                                                                                                                                                                                                                                            void call(x) {
         declare PathEdge, WorkList, SummaryEdge: global edge set
         algorithm Tabulate(G_{IP}^{\#})
                                                                                                                                                                                                                                                                                                                                     6:
                                                                                                                                                                                                                     0:
         begin
            Let (N^{\#}, E^{\#}) = G_{IP}^{\#}
[1]
[2]
[3]
[4]
[5]
[6]
[7]
[8]
           PathEdge := \{\langle s_{main}, \mathbf{0} \rangle \rightarrow \langle s_{main}, \mathbf{0} \rangle\}

WorkList := \{\langle s_{main}, \mathbf{0} \rangle \rightarrow \langle s_{main}, \mathbf{0} \rangle\}

SummaryEdge := \emptyset
                                                                                                                                                                                                                                                              = source();
                                                                                                                                                                                                                                                                                                                                     8:
                                                                                                                                                                                                                                                                                                                                                                     return y;
                                                                                                                                                                                                                      2:
                                                                                                                                                                                                                                                               a;
           ForwardTabulateSLRPs() for each n \in N^* do
                                                                                                                                                                                                                                                             = call(b);
                                                                                                                                                                                                                                                                                                                                                       } o
               X_n := \{ d_2 \in D \mid \exists d_1 \in (D \cup \{ \mathbf{0} \}) \text{ such that } \langle s_{procOf(n)}, d_1 \rangle \rightarrow \langle n, d_2 \rangle \in \text{PathEdge} \}
                                                                                                                                                                                                                                                              = C;
         end
         procedure Propagate(e)
                                                                                                                                                                                                                                                     sink(d);
         begin
           if e \notin PathEdge then Insert e into PathEdge; Insert e into WorkList fi
         end
                                                                                                                                                                                                                                       О
         procedure ForwardTabulateSLRPs()
         begin
           while WorkList \neq \emptyset do
[10]
               Select and remove an edge \langle s_p, d_1 \rangle \rightarrow \langle n, d_2 \rangle from WorkList
[11]
                                                                                                                                                                                              PathEdge
                                                                                                                                                                                                                                                                                                           \langle s_{main}, zero \rangle \rightarrow \langle 2, a \rangle
                                                                                                                                                                                                                                 \langle s_{main}, zero \rangle \rightarrow \langle 1, zero \rangle
[12]
               switch n
                   case n \in Call_n:
[13]
                                                                                                                                                                                                                                  \langle s_{main}, zero \rangle \rightarrow \langle 3, b \rangle |\langle s_{call}, x \rangle \rightarrow \langle s_{call}, x \rangle
                                                                                                                                                                                                                                                                                                                                                               \langle s_{call}, x \rangle \rightarrow \langle 7, x \rangle
                     for each d_3 such that \langle n, d_2 \rangle \rightarrow \langle s_{calledProc(n)}, d_3 \rangle \in E^{\#} do Propagate(\langle s_{calledProc(n)}, d_3 \rangle \rightarrow \langle s_{calledProc(n)}, d_3 \rangle)
[15]
[16]
[17]
                      for each d_3 such that \langle n, d_2 \rangle \rightarrow \langle returnSite(n), d_3 \rangle \in (E^\# \cup SummaryEdge) do
[18]
                          Propagate(\langle s_p, d_1 \rangle \rightarrow \langle returnSite(n), d_3 \rangle)
[19]
                      od
[20]
                   end case
                                                                                                                                                                                               WorkList
[21]
                   case n = e_n:
[22]
[23]
                      for each c \in callers(p) do
                         for each d_4, d_5 such that \langle c, d_4 \rangle \rightarrow \langle s_p, d_1 \rangle \in E^{\#} and \langle e_p, d_2 \rangle \rightarrow \langle returnSite(c), d_5 \rangle \in E^{\#} do if \langle c, d_4 \rangle \rightarrow \langle returnSite(c), d_5 \rangle \notin SummaryEdge then
[24]
[25]
[26]
[27]
[28]
[29]
[30]
[31]
[32]
                               Insert \langle c, d_4 \rangle \rightarrow \langle returnSite(c), d_5 \rangle into SummaryEdge

for each d_3 such that \langle s_{procOf(c)}, d_3 \rangle \rightarrow \langle c, d_4 \rangle \in PathEdge do

Propagate(\langle s_{procOf(c)}, d_3 \rangle \rightarrow \langle returnSite(c), d_5 \rangle)
                                                                                                                                                                                                \langle s_{call}, x \rangle \rightarrow \langle 7, x \rangle
                                                                                                                                                                                                                                                           <7, x> \rightarrow <8, y>
                                od
                          od
                                                                                                                                                                                                Propagate(\langle s_{call}, x \rangle \rightarrow \langle 8, y \rangle)
                   end case
                  case n \in (N_p - Call_p - \{e_p\}):

for each \langle m, d_3 \rangle such that \langle n, d_2 \rangle \rightarrow \langle m, d_3 \rangle \in E^{\#} do

Propagate(\langle s_p, d_1 \rangle \rightarrow \langle m, d_3 \rangle)
[33]
[34]
[35]
[36]
[37]
                      od
                  end case
[38]
[39]
               end switch
                                                                                                          NormalFlowFunction
            od
         end
```

```
declare PathEdge, WorkList, SummaryEdge: global edge set
           algorithm Tabulate(G_{IP}^{\#})
          begin
              Let (N^{\#}, E^{\#}) = G_{IP}^{\#}
[1]
[2]
[3]
[4]
[5]
[6]
[7]
[8]
            PathEdge := \{\langle s_{main}, \mathbf{0} \rangle \rightarrow \langle s_{main}, \mathbf{0} \rangle\}

WorkList := \{\langle s_{main}, \mathbf{0} \rangle \rightarrow \langle s_{main}, \mathbf{0} \rangle\}

SummaryEdge := \emptyset

ForwardTabulateSLRPs()

for each n \in N^* do
                  X_n := \{ d_2 \in D \mid \exists d_1 \in (D \cup \{ \mathbf{0} \}) \text{ such that } \langle s_{procOf(n)}, d_1 \rangle \rightarrow \langle n, d_2 \rangle \in \text{PathEdge} \}
           end
           procedure Propagate(e)
           begin
             if e \notin PathEdge then Insert e into PathEdge; Insert e into WorkList fi
           end
           procedure ForwardTabulateSLRPs()
           begin
              while WorkList \neq \emptyset do
[10]
                  Select and remove an edge \langle s_p, d_1 \rangle \rightarrow \langle n, d_2 \rangle from WorkList
[11]
[12]
                  switch n
[13]
                      case n \in Call_n:
                          for each d_3 such that \langle n, d_2 \rangle \rightarrow \langle s_{calledProc(n)}, d_3 \rangle \in E^{\#} do Propagate(\langle s_{calledProc(n)}, d_3 \rangle \rightarrow \langle s_{calledProc(n)}, d_3 \rangle)
 [15]
[16]
                           od
                          for each d_3 such that \langle n, d_2 \rangle \rightarrow \langle returnSite(n), d_3 \rangle \in (E^{\#} \cup SummaryEdge) do
[17]
                               Propagate(\langle s_p, d_1 \rangle \rightarrow \langle returnSite(n), d_3 \rangle)
[18]
[19]
                          od
[20]
                      end case
[21]
                       case n = e_n:
[22]
[23]
                          for each c \in callers(p) do
                              for each d_4, d_5 such that \langle c, d_4 \rangle \to \langle s_p, d_1 \rangle \in E^{\#} and \langle e_p, d_2 \rangle \to \langle returnSite(c), d_5 \rangle \in E^{\#} do if \langle c, d_4 \rangle \to \langle returnSite(c), d_5 \rangle \notin SummaryEdge then
[24]
[25]
[26]
[27]
[28]
[29]
[30]
[31]
[32]
                                     Insert \langle c, d_4 \rangle \rightarrow \langle returnSite(c), d_5 \rangle into SummaryEdge

for each d_3 such that \langle s_{procOf(c)}, d_3 \rangle \rightarrow \langle c, d_4 \rangle \in PathEdge do

Propagate(\langle s_{procOf(c)}, d_3 \rangle \rightarrow \langle returnSite(c), d_5 \rangle)
                                      od
                              od
                      end case
[33]
[34]
[35]
                      case n \in (N_p - Call_p - \{e_p\}):

for each \langle m, d_3 \rangle such that \langle n, d_2 \rangle \rightarrow \langle m, d_3 \rangle \in E^{\#} do
                              Propagate(\langle s_p, d_1 \rangle \rightarrow \langle m, d_3 \rangle)
[36]
[37]
                      end case
[38]
[39]
                  end switch
              od
          end
```

```
void main() {
0:
1:     a = source();
2:     b = a;
3:     c = call(b);
4:     d = c;
5:     sink(d);
}     o     a     b     c     d
```

```
void call(x) {
6:
7: y = x;
8: return y;
}o x y
```

$$\langle s_{main}, zero \rangle \rightarrow \langle 1, zero \rangle$$
  $\langle s_{main}, zero \rangle \rightarrow \langle 2, a \rangle$   $\langle s_{main}, zero \rangle \rightarrow \langle 3, b \rangle$   $\langle s_{call}, x \rangle \rightarrow \langle s_{call}, x \rangle$   $\langle s_{call}, x \rangle \rightarrow \langle 8, y \rangle$ 

$$\langle s_{call}, x \rangle \rightarrow \langle 8, y \rangle$$

```
void main() {
                                                                                                                                                                                                                                                                                                                                                              void call(x) {
         declare PathEdge, WorkList, SummaryEdge: global edge set
         algorithm Tabulate(G_{IP}^{\#})
                                                                                                                                                                                                                                                                                                                                                      6:
                                                                                                                                                                                                                                 0:
         begin
            Let (N^{\#}, E^{\#}) = G_{IP}^{\#}
[1]
[2]
[3]
[4]
[5]
[6]
[7]
[8]
           PathEdge := \{\langle s_{main}, \mathbf{0} \rangle \rightarrow \langle s_{main}, \mathbf{0} \rangle\}

WorkList := \{\langle s_{main}, \mathbf{0} \rangle \rightarrow \langle s_{main}, \mathbf{0} \rangle\}

SummaryEdge := \emptyset

ForwardTabulateSLRPs()

for each n \in N^* do
                                                                                                                                                                                                                                                                            = source();
                                                                                                                                                                                                                                                                                                                                                      8:
                                                                                                                                                                                                                                                                                                                                                                                        return y;
                                                                                                                                                                                                                                 2:
                                                                                                                                                                                                                                                                             a;
                                                                                                                                                                                                                                                                   c = call(b);
                                                                                                                                                                                                                                                                                                                                                                         } o
                X_n := \{ d_2 \in D \mid \exists d_1 \in (D \cup \{ \mathbf{0} \}) \text{ such that } \langle s_{procOf(n)}, d_1 \rangle \rightarrow \langle n, d_2 \rangle \in \text{PathEdge} \}
                                                                                                                                                                                                                                                                            = C;
         end
         procedure Propagate(e)
                                                                                                                                                                                                                                                                  sink(d);
         begin
            if e \notin PathEdge then Insert e into PathEdge; Insert e into WorkList fi
          end
                                                                                                                                                                                                                                                    О
          procedure ForwardTabulateSLRPs()
         begin
            while WorkList \neq \emptyset do
                Select and remove an edge \langle s_p, d_1 \rangle \rightarrow \langle n, d_2 \rangle from WorkList
[11]
                                                                                                                                                                                                        PathEdge
                                                                                                                                                                                                                                                                                                                           \langle s_{main}, zero \rangle \rightarrow \langle 2, a \rangle
[12]
                                                                                                                                                                                                                                             \langle s_{main}, zero \rangle \rightarrow \langle 1, zero \rangle
                switch n
                    case n \in Call_n:
[13]
                                                                                                                                                                                                                                              \langle s_{main}, zero \rangle \rightarrow \langle 3, b \rangle \langle s_{call}, x \rangle \rightarrow \langle s_{call}, x \rangle
                                                                                                                                                                                                                                                                                                                                                                                  \langle s_{call}, x \rangle \rightarrow \langle 7, x \rangle
                       for each d_3 such that \langle n, d_2 \rangle \rightarrow \langle s_{calledProc(n)}, d_3 \rangle \in E^{\#} do Propagate(\langle s_{calledProc(n)}, d_3 \rangle \rightarrow \langle s_{calledProc(n)}, d_3 \rangle)
 [15]
[16]
                                                                                                                                                                                                                                             \langle s_{call}, x \rangle \rightarrow \langle 8, y \rangle
                       for each d_3 such that \langle n, d_2 \rangle \rightarrow \langle returnSite(n), d_3 \rangle \in (E^{\#} \cup SummaryEdge) do
 [17]
 [18]
                           Propagate(\langle s_p, d_1 \rangle \rightarrow \langle returnSite(n), d_3 \rangle)
[19]
                       od
[20]
                    end case
                                                                                                                                                                                                         WorkList
[21]
                    case n = e_p:
 [22]
[23]
                       for each c \in callers(p) do
                         for each d_4, d_5 such that \langle c, d_4 \rangle \rightarrow \langle s_p, d_1 \rangle \in E^{\#} and \langle e_p, d_2 \rangle \rightarrow \langle returnSite(c), d_5 \rangle \in E^{\#} do

if \langle c, d_4 \rangle \rightarrow \langle returnSite(c), d_5 \rangle \notin SummaryEdge then

Insert \langle c, d_4 \rangle \rightarrow \langle returnSite(c), d_5 \rangle into SummaryEdge

for each d_3 such that \langle s_{procOf(c)}, d_3 \rangle \rightarrow \langle c, d_4 \rangle \in PathEdge do

Propagate(\langle s_{procOf(c)}, d_3 \rangle \rightarrow \langle returnSite(c), d_5 \rangle)
 [24]
[25]
[26]
                                                                                                                                                                                                         \langle s_{call}, x \rangle \rightarrow \langle 8, y \rangle
[27]
[28]
[29]
 [30]
                           od
                                                                                                                                                                                                                                                                                                                                                                                       <s<sub>call</sub>, x>
[31]
[32]
                    end case
                    case n \in (N_p - Call_p - \{e_p\}):

for each \langle m, d_3 \rangle such that \langle n, d_2 \rangle \rightarrow \langle m, d_3 \rangle \in E^{\#} do
[34]
[35]
                           Propagate(\langle s_p, d_1 \rangle \rightarrow \langle m, d_3 \rangle)
[36]
[37]
                                                                                                                                                                                                                                                                                                                                                                                                       <8, y>
                    end case
[38]
[39]
                end switch
            od
```

end

```
void main() {
                                                                                                                                                                                                                                                                                                                                                         void call(x) {
         declare PathEdge, WorkList, SummaryEdge: global edge set
         algorithm Tabulate(G_{IP}^{\#})
                                                                                                                                                                                                                                                                                                                                                  6:
                                                                                                                                                                                                                              0:
         begin
            Let (N^{\#}, E^{\#}) = G_{IP}^{\#}
[1]
[2]
[3]
[4]
[5]
[6]
[7]
[8]
           PathEdge := \{\langle s_{main}, \mathbf{0} \rangle \rightarrow \langle s_{main}, \mathbf{0} \rangle\}

WorkList := \{\langle s_{main}, \mathbf{0} \rangle \rightarrow \langle s_{main}, \mathbf{0} \rangle\}

SummaryEdge := \emptyset

ForwardTabulateSLRPs()

for each n \in N^* do
                                                                                                                                                                                                                                                                         = source();
                                                                                                                                                                                                                                                                                                                                                  8:
                                                                                                                                                                                                                                                                                                                                                                                   retürn y;
                                                                                                                                                                                                                               2:
                                                                                                                                                                                                                                                                         a;
                                                                                                                                                                                                                                                               c = call(b);
                                                                                                                                                                                                                                                                                                                                                                     } o
                X_n := \{ d_2 \in D \mid \exists d_1 \in (D \cup \{ \mathbf{0} \}) \text{ such that } \langle s_{procOf(n)}, d_1 \rangle \rightarrow \langle n, d_2 \rangle \in \text{PathEdge} \}
                                                                                                                                                                                                                                                                        = C;
         end
         procedure Propagate(e)
                                                                                                                                                                                                                                                               sink(d);
         begin
            if e \notin PathEdge then Insert e into PathEdge; Insert e into WorkList fi
          end
                                                                                                                                                                                                                                                 О
          procedure ForwardTabulateSLRPs()
         begin
            while WorkList \neq \emptyset do
                Select and remove an edge \langle s_p, d_1 \rangle \rightarrow \langle n, d_2 \rangle from WorkList
[11]
                                                                                                                                                                                                      PathEdge
                                                                                                                                                                                                                                                                                                                       \langle s_{main}, zero \rangle \rightarrow \langle 2, a \rangle
[12]
                                                                                                                                                                                                                                          \langle s_{main}, zero \rangle \rightarrow \langle 1, zero \rangle
                switch n
                   case n \in Call_n:
[13]
                                                                                                                                                                                                                                                                                                                                                                              \langle s_{call}, x \rangle \rightarrow \langle 7, x \rangle
                                                                                                                                                                                                                                           \langle s_{main}, zero \rangle \rightarrow \langle 3, b \rangle \langle s_{call}, x \rangle \rightarrow \langle s_{call}, x \rangle
                      for each d_3 such that \langle n, d_2 \rangle \rightarrow \langle s_{calledProc(n)}, d_3 \rangle \in E^{\#} do Propagate(\langle s_{calledProc(n)}, d_3 \rangle \rightarrow \langle s_{calledProc(n)}, d_3 \rangle)
 [15]
                                                                                                                                                                                                                                          \langle s_{call}, x \rangle \rightarrow \langle 8, y \rangle
[16]
                       for each d_3 such that \langle n, d_2 \rangle \rightarrow \langle returnSite(n), d_3 \rangle \in (E^{\#} \cup SummaryEdge) do
 [17]
 [18]
                           Propagate(\langle s_p, d_1 \rangle \rightarrow \langle returnSite(n), d_3 \rangle)
[19]
                       od
[20]
                    end case
                                                                                                                                                                                                       WorkList
[21]
                    case n = e_n:
 [22]
[23]
                      for each c \in callers(p) do
                                                                                         CallFlowFunction
                         for each d_4, d_5 such that \langle c, d_4 \rangle \rightarrow \langle s_p, d_1 \rangle \in E^\# and \langle e_p, d_2 \rangle \rightarrow \langle returnSite(c), d_5 \rangle \in E^\# do

if \langle c, d_4 \rangle \rightarrow \langle returnSite(c), d_5 \rangle \notin SummaryEdge then

Insert \langle c, d_4 \rangle \rightarrow \langle returnSite(c), d_5 \rangle into SummaryEdge

for each d_3 such that \langle s_{procOf(c)}, d_3 \rangle \rightarrow \langle c, d_4 \rangle \in PathEdge do

Propagate(\langle s_{procOf(c)}, d_3 \rangle \rightarrow \langle returnSite(c), d_5 \rangle)
 [24]
[25]
[26]
                                                                                                                                                                                                       \langle s_{call}, x \rangle \rightarrow \langle 8, y \rangle
[27]
[28]
[29]
 [30]
                                                                                                                                                                                                                                                                                                                                                                                  <s<sub>call</sub>, x>
[31]
[32]
                                                                                                                                                                                                                                                                                                                                            CallFlowFunction
                    end case
                    case n \in (N_p - Call_p - \{e_p\}):

for each \langle m, d_3 \rangle such that \langle n, d_2 \rangle \rightarrow \langle m, d_3 \rangle \in E^{\#} do
                                                                                                                                                                                                                                                                                                                            <3, b>
[34]
[35]
                           Propagate(\langle s_p, d_1 \rangle \rightarrow \langle m, d_3 \rangle)
                                                                                                                                                                                                                                                                                                                                                                                                 <8, y>
[36]
[37]
                    end case
[38]
[39]
                end switch
         end
```

```
void main() {
                                                                                                                                                                                                                                                                                                                                                             void call(x) {
         declare PathEdge, WorkList, SummaryEdge: global edge set
         algorithm Tabulate(G_{IP}^{\#})
                                                                                                                                                                                                                                                                                                                                                     6:
                                                                                                                                                                                                                                0:
         begin
            Let (N^{\#}, E^{\#}) = G_{IP}^{\#}
[1]
[2]
[3]
[4]
[5]
[6]
[7]
[8]
           PathEdge := \{\langle s_{main}, \mathbf{0} \rangle \rightarrow \langle s_{main}, \mathbf{0} \rangle\}

WorkList := \{\langle s_{main}, \mathbf{0} \rangle \rightarrow \langle s_{main}, \mathbf{0} \rangle\}

SummaryEdge := \emptyset

ForwardTabulateSLRPs()

for each n \in N^* do
                                                                                                                                                                                                                                                                           = source();
                                                                                                                                                                                                                                                                                                                                                     8:
                                                                                                                                                                                                                                                                                                                                                                                       return y;
                                                                                                                                                                                                                                 2:
                                                                                                                                                                                                                                                                            = a;
                                                                                                                                                                                                                                                                           = call(b);
                X_n := \{ d_2 \in D \mid \exists d_1 \in (D \cup \{ \mathbf{0} \}) \text{ such that } \langle s_{procOf(n)}, d_1 \rangle \rightarrow \langle n, d_2 \rangle \in \text{PathEdge} \}
                                                                                                                                                                                                                                                                           = C;
         end
         procedure Propagate(e)
                                                                                                                                                                                                                                                                 sink(d);
         begin
            if e \notin PathEdge then Insert e into PathEdge; Insert e into WorkList fi
          end
                                                                                                                                                                                                                                                   О
          procedure ForwardTabulateSLRPs()
         begin
            while WorkList \neq \emptyset do
                Select and remove an edge \langle s_p, d_1 \rangle \rightarrow \langle n, d_2 \rangle from WorkList
[11]
                                                                                                                                                                                                        PathEdge
                                                                                                                                                                                                                                            \langle s_{main}, zero \rangle \rightarrow \langle 1, zero \rangle
                                                                                                                                                                                                                                                                                                                          \langle s_{main}, zero \rangle \rightarrow \langle 2, a \rangle
[12]
                switch n
                    case n \in Call_n:
[13]
                                                                                                                                                                                                                                             \langle s_{main}, zero \rangle \rightarrow \langle 3, b \rangle \langle s_{call}, x \rangle \rightarrow \langle s_{call}, x \rangle
                                                                                                                                                                                                                                                                                                                                                                                 \langle s_{call}, x \rangle \rightarrow \langle 7, x \rangle
                       for each d_3 such that \langle n, d_2 \rangle \rightarrow \langle s_{calledProc(n)}, d_3 \rangle \in E^{\#} do Propagate(\langle s_{calledProc(n)}, d_3 \rangle \rightarrow \langle s_{calledProc(n)}, d_3 \rangle)
 [15]
                                                                                                                                                                                                                                            \langle s_{call}, x \rangle \rightarrow \langle 8, y \rangle
[16]
                       for each d_3 such that \langle n, d_2 \rangle \rightarrow \langle returnSite(n), d_3 \rangle \in (E^{\#} \cup SummaryEdge) do
 [17]
 [18]
                           Propagate(\langle s_p, d_1 \rangle \rightarrow \langle returnSite(n), d_3 \rangle)
[19]
                       od
[20]
                    end case
                                                                                                                                                                                                        WorkList
[21]
                    case n = e_n:
 [22]
[23]
                       for each c \in callers(p) do
                                                                                                                                                           ReturnFlowFunction
                                                                                          CallFlowFunction
                         for each d_4, d_5 such that \langle c, d_4 \rangle \rightarrow \langle s_p, d_1 \rangle \in E^\# and \langle e_p, d_2 \rangle \rightarrow \langle returnSite(c), d_5 \rangle \in E^\# do

if \langle c, d_4 \rangle \rightarrow \langle returnSite(c), d_5 \rangle \notin SummaryEdge then

Insert \langle c, d_4 \rangle \rightarrow \langle returnSite(c), d_5 \rangle into SummaryEdge

for each d_3 such that \langle s_{procOf(c)}, d_3 \rangle \rightarrow \langle c, d_4 \rangle \in PathEdge do

Propagate(\langle s_{procOf(c)}, d_3 \rangle \rightarrow \langle returnSite(c), d_5 \rangle)
 [24]
[25]
[26]
                                                                                                                                                                                                         \langle s_{call}, x \rangle \rightarrow \langle 8, y \rangle
[27]
[28]
[29]
 [30]
                                                                                                                                                                                                                                                                                                                                                                                     <s<sub>call</sub>, x>
[31]
                                                                                                                                                                                                                                                                                                                                               CallFlowFunction
[32]
                    end case
                   case n \in (N_p - Call_p - \{e_p\}):

for each \langle m, d_3 \rangle such that \langle n, d_2 \rangle \rightarrow \langle m, d_3 \rangle \in E^\# do

Propagate(\langle s_p, d_1 \rangle \rightarrow \langle m, d_3 \rangle)
                                                                                                                                                                                                                                                                                                                               <3, b>
[34]
[35]
                                                                                                                                                                                                                                                                                                                                                                                                     <8, y>
[36]
[37]
                    end case
[38]
[39]
                end switch
             od
                                                                                                                                                                                                                                                                                                                                                                        ReturnFlowFunction
         end
                                                                                                                                                                                                                                                                                                                                            <4, c>
```

```
void main() {
                                                                                                                                                                                                                                                                                                               void call(x) {
        declare PathEdge, WorkList, SummaryEdge: global edge set
        algorithm Tabulate(G_{IP}^{\#})
                                                                                                                                                                                                                                                                                                         6:
                                                                                                                                                                                                   0:
        begin
           Let (N^{\#}, E^{\#}) = G_{IP}^{\#}
[1]
[2]
[3]
[4]
[5]
[6]
[7]
[8]
          PathEdge := \{\langle s_{main}, \mathbf{0} \rangle \rightarrow \langle s_{main}, \mathbf{0} \rangle\}

WorkList := \{\langle s_{main}, \mathbf{0} \rangle \rightarrow \langle s_{main}, \mathbf{0} \rangle\}

SummaryEdge := \emptyset
                                                                                                                                                                                                                                         = source();
                                                                                                                                                                                                                                                                                                         8:
                                                                                                                                                                                                   2:
                                                                                                                                                                                                                                                                                                                                      return y;
                                                                                                                                                                                                                                                 a;
          ForwardTabulateSLRPs() for each n \in N^* do
                                                                                                                                                                                                                                                   call(b);
              X_n := \{ d_2 \in D \mid \exists d_1 \in (D \cup \{ \mathbf{0} \}) \text{ such that } \langle s_{procOf(n)}, d_1 \rangle \rightarrow \langle n, d_2 \rangle \in \text{PathEdge} \}
        end
        procedure Propagate(e)
                                                                                                                                                                                                                                sink(d);
        begin
          if e \notin PathEdge then Insert e into PathEdge; Insert e into WorkList fi
         end
                                                                                                                                                                                                                   О
        procedure ForwardTabulateSLRPs()
        begin
          while WorkList \neq \emptyset do
              Select and remove an edge \langle s_p, d_1 \rangle \rightarrow \langle n, d_2 \rangle from WorkList
[11]
                                                                                                                                                                              PathEdge
                                                                                                                                                                                                             \langle s_{main}, zero \rangle \rightarrow \langle 1, zero \rangle
                                                                                                                                                                                                                                                                                 \langle s_{main}, zero \rangle \rightarrow \langle 2, a \rangle
[12]
              switch n
[13]
                 case n \in Call_n:
                                                                                                                                                                                                              \langle s_{main}, zero \rangle \rightarrow \langle 3, b \rangle \langle s_{call}, x \rangle \rightarrow \langle s_{call}, x \rangle
                                                                                                                                                                                                                                                                                                                                 \langle s_{call}, x \rangle \rightarrow \langle 7, x \rangle
                    for each d_3 such that \langle n, d_2 \rangle \rightarrow \langle s_{calledProc(n)}, d_3 \rangle \in E^{\#} do Propagate(\langle s_{calledProc(n)}, d_3 \rangle \rightarrow \langle s_{calledProc(n)}, d_3 \rangle)
 [15]
                                                                                                                                                                                                             \langle s_{call}, x \rangle \rightarrow \langle 8, y \rangle
[16]
                    for each d_3 such that \langle n, d_2 \rangle \rightarrow \langle returnSite(n), d_3 \rangle \in (E^{\#} \cup SummaryEdge) do
[17]
[18]
                        Propagate(\langle s_p, d_1 \rangle \rightarrow \langle returnSite(n), d_3 \rangle)
[19]
                    od
[20]
                 end case
                                                                                                                                                                              WorkList
[21]
                 case n = e_n:
[22]
[23]
                    for each c \in callers(p) do
                                                                              CallFlowFunction
                      for each d_4, d_5 such that \langle c, d_4 \rangle \to \langle s_p, d_1 \rangle \in E^{\#} and \langle e_p, d_2 \rangle \to \langle returnSite(c), d_5 \rangle \in E^{\#} do if \langle c, d_4 \rangle \to \langle returnSite(c), d_5 \rangle \notin SummaryEdge then
[24]
[25]
[26]
                             Insert \langle c, d_4 \rangle \rightarrow \langle returnSite(c), d_5 \rangle into SummaryEdge for each d_3 such that \langle s_{procOf(c)}, d_3 \rangle \rightarrow \langle c, d_4 \rangle \in PathEdge do
                                                                                                                                                                               \langle s_{call}, x \rangle \rightarrow \langle 8, y \rangle
[27]
[28]
[29]
                                Propagate(\langle s_{procOf(c)}, d_3 \rangle \rightarrow \langle returnSite(c), d_5 \rangle)
[30]
                                                                                                                                                                                                                                                                                                                                    <s<sub>call</sub>, x>
[31]
                                                                                                                                                                                                                                                                                                   CallFlowFunction
[32]
                 end case
                 case n \in (N_p - Call_p - \{e_p\}):
                                                                                                                                                                                                                                                                                      <3, b>
                    for each \langle m, d_3 \rangle such that \langle n, d_2 \rangle \rightarrow \langle m, d_3 \rangle \in E^{\#} do
[34]
                       Propagate(\langle s_p, d_1 \rangle \rightarrow \langle m, d_3 \rangle)
[35]
[36]
[37]
                                                                                                                                                                                                                                                                                                                                                    <8, y>
                    od
                 end case
                                                                                                                                                                                                                                                                                                      SummaryEdge
                                                                                                                                      SummaryEdge U= <3,b> → <4,c>
[38]
              end switch
[39]
                                                                                                                                                                                                                                                                                                                          ReturnFlowFunction
        end
                                                                                                                                                                                                                                                                                                 <4, c>
```

```
void main() {
                                                                                                                                                                                                                                                                                                                     void call(x) {
        declare PathEdge, WorkList, SummaryEdge: global edge set
        algorithm Tabulate(G_{IP}^{\#})
                                                                                                                                                                                                                                                                                                               6:
                                                                                                                                                                                                       0:
        begin
           Let (N^{\#}, E^{\#}) = G_{IP}^{\#}
[1]
[2]
[3]
[4]
[5]
[6]
[7]
[8]
          PathEdge := \{\langle s_{main}, \mathbf{0} \rangle \rightarrow \langle s_{main}, \mathbf{0} \rangle\}

WorkList := \{\langle s_{main}, \mathbf{0} \rangle \rightarrow \langle s_{main}, \mathbf{0} \rangle\}

SummaryEdge := \emptyset
                                                                                                                                                                                                                                             = source();
                                                                                                                                                                                                                                                                                                               8:
                                                                                                                                                                                                       2:
                                                                                                                                                                                                                                                                                                                                             return y;
                                                                                                                                                                                                                                              = a;
           ForwardTabulateSLRPs() for each n \in N^* do
                                                                                                                                                                                                                                                       call(b);
              X_n := \{ d_2 \in D \mid \exists d_1 \in (D \cup \{ \mathbf{0} \}) \text{ such that } \langle s_{procOf(n)}, d_1 \rangle \rightarrow \langle n, d_2 \rangle \in \text{PathEdge} \}
        end
        procedure Propagate(e)
                                                                                                                                                                                                                                    sink(d);
        begin
           if e \notin PathEdge then Insert e into PathEdge; Insert e into WorkList fi
         end
                                                                                                                                                                                                                        О
         procedure ForwardTabulateSLRPs()
        begin
           while WorkList \neq \emptyset do
              Select and remove an edge \langle s_p, d_1 \rangle \rightarrow \langle n, d_2 \rangle from WorkList
[11]
                                                                                                                                                                                 PathEdge
                                                                                                                                                                                                                  \langle s_{main}, zero \rangle \rightarrow \langle 1, zero \rangle
                                                                                                                                                                                                                                                                                       \langle s_{main}, zero \rangle \rightarrow \langle 2, a \rangle
[12]
              switch n
[13]
                 case n \in Call_n:
                                                                                                                                                                                                                  \langle s_{main}, zero \rangle \rightarrow \langle 3, b \rangle \langle s_{call}, x \rangle \rightarrow \langle s_{call}, x \rangle
                                                                                                                                                                                                                                                                                                                                        \langle s_{call}, x \rangle \rightarrow \langle 7, x \rangle
                    for each d_3 such that \langle n, d_2 \rangle \rightarrow \langle s_{calledProc(n)}, d_3 \rangle \in E^{\#} do Propagate(\langle s_{calledProc(n)}, d_3 \rangle \rightarrow \langle s_{calledProc(n)}, d_3 \rangle)
 [15]
                                                                                                                                                                                                                  \langle s_{call}, x \rangle \rightarrow \langle 8, y \rangle
[16]
                    for each d_3 such that \langle n, d_2 \rangle \rightarrow \langle returnSite(n), d_3 \rangle \in (E^{\#} \cup SummaryEdge) do
[17]
[18]
                        Propagate(\langle s_p, d_1 \rangle \rightarrow \langle returnSite(n), d_3 \rangle)
[19]
                    od
[20]
                  end case
                                                                                                                                                                                  WorkList
[21]
                  case n = e_n:
[22]
[23]
                    for each c \in callers(p) do
                                                                                CallFlowFunction
                                                                                                                                          ReturnFlowFunction
                       for each d_4, d_5 such that \langle c, d_4 \rangle \to \langle s_p, d_1 \rangle \in E^{\#} and \langle e_p, d_2 \rangle \to \langle returnSite(c), d_5 \rangle \in E^{\#} do if \langle c, d_4 \rangle \to \langle returnSite(c), d_5 \rangle \notin SummaryEdge then
[24]
[25]
[26]
                             Insert \langle c, d_4 \rangle \rightarrow \langle returnSite(c), d_5 \rangle into SummaryEdge

for each d_3 such that \langle s_{procOf(c)}, d_3 \rangle \rightarrow \langle c, d_4 \rangle \in PathEdge do

Propagate(\langle s_{procOf(c)}, d_3 \rangle \rightarrow \langle returnSite(c), d_5 \rangle)
                                                                                                                                                                                  \langle s_{call}, x \rangle \rightarrow \langle 8, y \rangle
[27]
[28]
[29]
                                                                                                                                                                                                                                                <s<sub>main</sub>, zero>
[30]
                                                                                                                                                                                                                                                                                                                                           <s<sub>call</sub>, x>
[31]
                                                                                                                                                                                                                                                                                                         CallFlowFunction
[32]
                 end case
                  case n \in (N_p - Call_p - \{e_p\}):
                                                                                                                                                                                                                                                                                            <3, b>
                    for each \langle m, d_3 \rangle such that \langle n, d_2 \rangle \rightarrow \langle m, d_3 \rangle \in E^{\#} do
[34]
                        Propagate(\langle s_p, d_1 \rangle \rightarrow \langle m, d_3 \rangle)
[35]
[36]
[37]
                                                                                                                                                                                                                                                                                                                                                          <8, y>
                     od
                 end case
                                                                                                                                                                                                                                                                                                            SummaryEdge
                                                                                                                                         SummaryEdge U= <3,b> → <4,c>
[38]
              end switch
[39]
                                                                                                                                                                                                                                                                                                                                ReturnFlowFunction
        end
                                                                                                                                                                                                                                                                                                      <4, c>
```

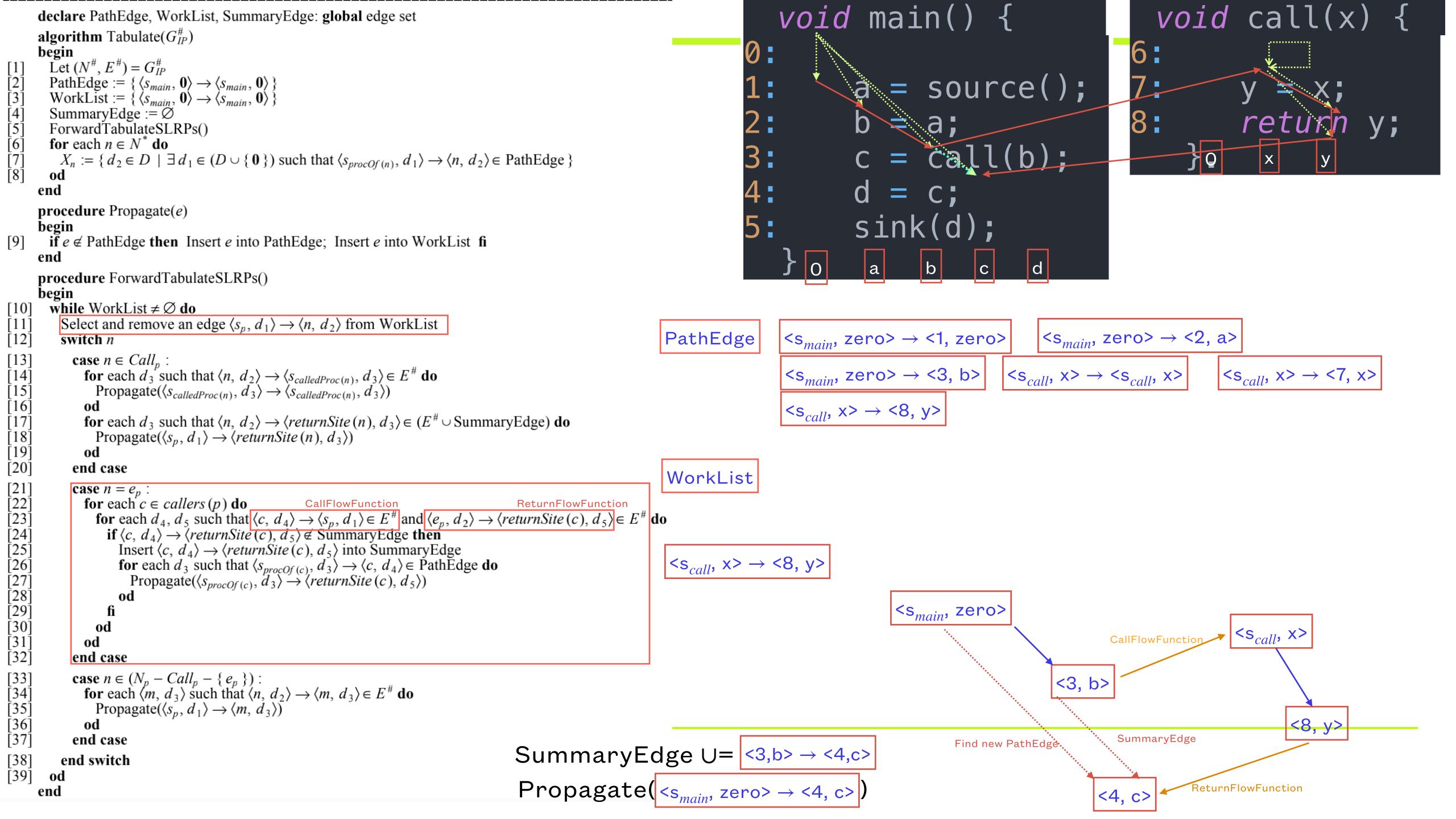
```
void main() {
                                                                                                                                                                                                                                                                                                                       void call(x) {
        declare PathEdge, WorkList, SummaryEdge: global edge set
        algorithm Tabulate(G_{IP}^{\#})
                                                                                                                                                                                                                                                                                                                 6:
                                                                                                                                                                                                        0:
        begin
           Let (N^{\#}, E^{\#}) = G_{IP}^{\#}
[1]
[2]
[3]
[4]
[5]
[6]
[7]
[8]
          PathEdge := \{\langle s_{main}, \mathbf{0} \rangle \rightarrow \langle s_{main}, \mathbf{0} \rangle\}

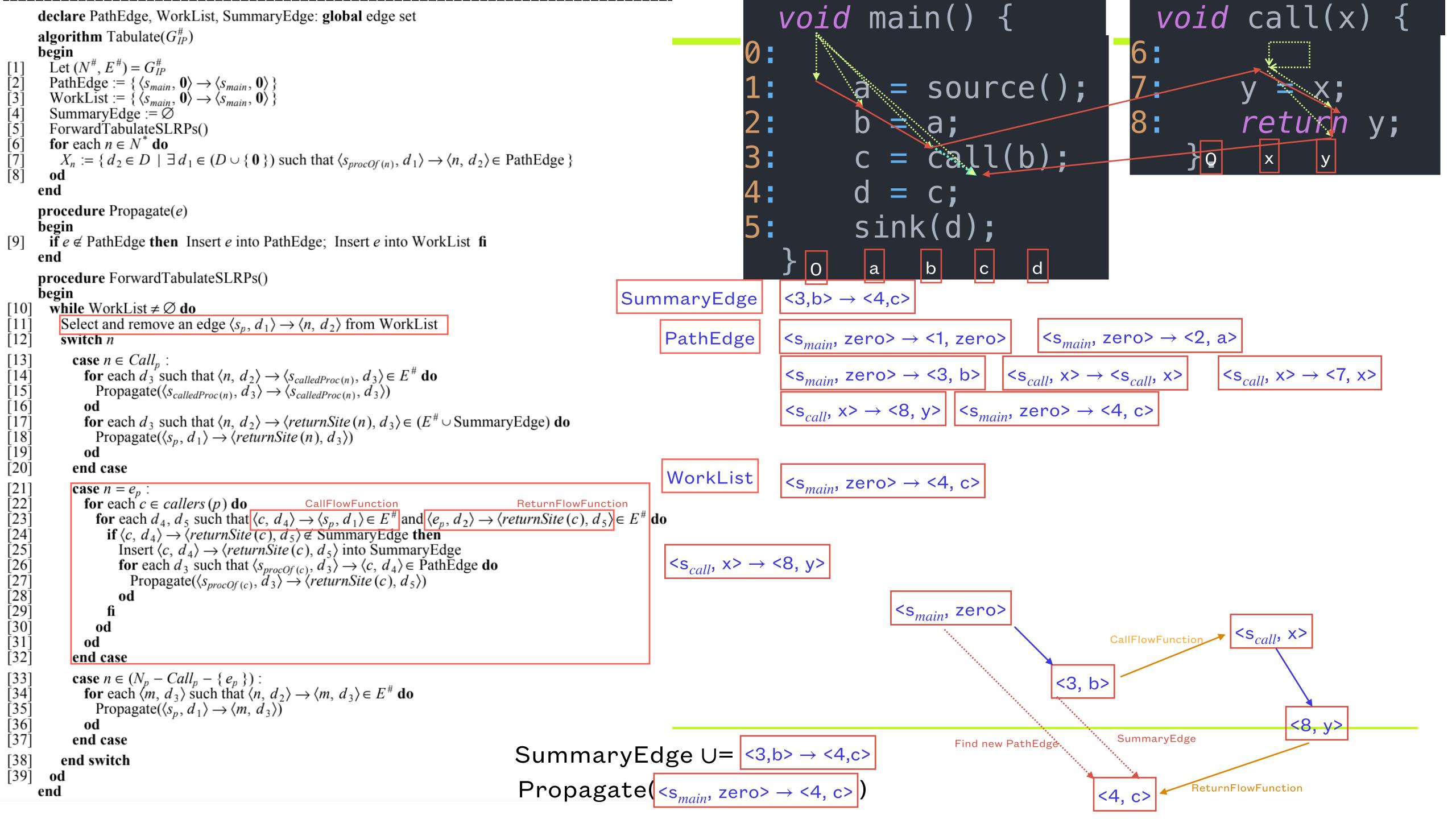
WorkList := \{\langle s_{main}, \mathbf{0} \rangle \rightarrow \langle s_{main}, \mathbf{0} \rangle\}

SummaryEdge := \emptyset
                                                                                                                                                                                                                                                         source();
                                                                                                                                                                                                                                                                                                                 8:
                                                                                                                                                                                                                                                                                                                                               return y;
                                                                                                                                                                                                         2:
           ForwardTabulateSLRPs() for each n \in N^* do
                                                                                                                                                                                                                                                        call(b);
              X_n := \{ d_2 \in D \mid \exists d_1 \in (D \cup \{ \mathbf{0} \}) \text{ such that } \langle s_{procOf(n)}, d_1 \rangle \rightarrow \langle n, d_2 \rangle \in \text{PathEdge} \}
        end
        procedure Propagate(e)
                                                                                                                                                                                                                                      sink(d);
        begin
           if e \notin PathEdge then Insert e into PathEdge; Insert e into WorkList fi
         end
                                                                                                                                                                                                                         О
         procedure ForwardTabulateSLRPs()
        begin
           while WorkList \neq \emptyset do
              Select and remove an edge \langle s_p, d_1 \rangle \rightarrow \langle n, d_2 \rangle from WorkList
[11]
                                                                                                                                                                                  PathEdge
                                                                                                                                                                                                                   \langle s_{main}, zero \rangle \rightarrow \langle 1, zero \rangle
                                                                                                                                                                                                                                                                                        \langle s_{main}, zero \rangle \rightarrow \langle 2, a \rangle
[12]
              switch n
[13]
                  case n \in Call_n:
                                                                                                                                                                                                                                                                                                                                          \langle s_{call}, x \rangle \rightarrow \langle 7, x \rangle
                                                                                                                                                                                                                    \langle s_{main}, zero \rangle \rightarrow \langle 3, b \rangle \langle s_{call}, x \rangle \rightarrow \langle s_{call}, x \rangle
                    for each d_3 such that \langle n, d_2 \rangle \rightarrow \langle s_{calledProc(n)}, d_3 \rangle \in E^{\#} do Propagate(\langle s_{calledProc(n)}, d_3 \rangle \rightarrow \langle s_{calledProc(n)}, d_3 \rangle)
 [15]
                                                                                                                                                                                                                   \langle s_{call}, x \rangle \rightarrow \langle 8, y \rangle
[16]
                    for each d_3 such that \langle n, d_2 \rangle \rightarrow \langle returnSite(n), d_3 \rangle \in (E^{\#} \cup SummaryEdge) do
[17]
[18]
                        Propagate(\langle s_p, d_1 \rangle \rightarrow \langle returnSite(n), d_3 \rangle)
[19]
                    od
[20]
                  end case
                                                                                                                                                                                   WorkList
[21]
                  case n = e_n:
[22]
[23]
                    for each c \in callers(p) do
                                                                                 CallFlowFunction
                                                                                                                                          ReturnFlowFunction
                       for each d_4, d_5 such that \langle c, d_4 \rangle \to \langle s_p, d_1 \rangle \in E^{\#} and \langle e_p, d_2 \rangle \to \langle returnSite(c), d_5 \rangle \in E^{\#} do if \langle c, d_4 \rangle \to \langle returnSite(c), d_5 \rangle \notin SummaryEdge then
[24]
[25]
[26]
                             Insert \langle c, d_4 \rangle \rightarrow \langle returnSite(c), d_5 \rangle into SummaryEdge

for each d_3 such that \langle s_{procOf(c)}, d_3 \rangle \rightarrow \langle c, d_4 \rangle \in PathEdge do

Propagate(\langle s_{procOf(c)}, d_3 \rangle \rightarrow \langle returnSite(c), d_5 \rangle)
                                                                                                                                                                                   \langle s_{call}, x \rangle \rightarrow \langle 8, y \rangle
[27]
[28]
[29]
                                                                                                                                                                                                                                                  <s<sub>main</sub>, zero>
[30]
                                                                                                                                                                                                                                                                                                                                             <s<sub>call</sub>, x>
[31]
                                                                                                                                                                                                                                                                                                           CallFlowFunction
[32]
                  end case
                  case n \in (N_n - Call_n - \{e_n\}):
                                                                                                                                                                                                                                                                                             <3, b>
                    for each \langle m, d_3 \rangle such that \langle n, d_2 \rangle \rightarrow \langle m, d_3 \rangle \in E^{\#} do
[34]
[35]
                        Propagate(\langle s_p, d_1 \rangle \rightarrow \langle m, d_3 \rangle)
[36]
[37]
                                                                                                                                                                                                                                                                                                                                                             <8, y>
                     od
                  end case
                                                                                                                                                                                                                                                                                                             SummaryEdge
                                                                                                                                                                                                                                                                 Find new PathEdge.
                                                                                                                                          SummaryEdge U= <3,b> → <4,c>
[38]
[39]
              end switch
                                                                                                                                                                                                                                                                                                                                  ReturnFlowFunction
        end
                                                                                                                                                                                                                                                                                                        <4, c>
```





```
declare PathEdge, WorkList, SummaryEdge: global edge set
          algorithm Tabulate(G_{IP}^{\#})
                                                                                                                                                                                                                                           0:
         begin
             Let (N^{\#}, E^{\#}) = G_{IP}^{\#}
[1]
[2]
[3]
[4]
[5]
[6]
[7]
[8]
            PathEdge := \{\langle s_{main}, \mathbf{0} \rangle \rightarrow \langle s_{main}, \mathbf{0} \rangle\}

WorkList := \{\langle s_{main}, \mathbf{0} \rangle \rightarrow \langle s_{main}, \mathbf{0} \rangle\}

SummaryEdge := \emptyset
                                                                                                                                                                                                                                            2:
             ForwardTabulateSLRPs() for each n \in N^* do
                 X_n := \{ d_2 \in D \mid \exists d_1 \in (D \cup \{ \mathbf{0} \}) \text{ such that } \langle s_{procOf(n)}, d_1 \rangle \rightarrow \langle n, d_2 \rangle \in \text{PathEdge} \}
          end
          procedure Propagate(e)
          begin
             if e \notin PathEdge then Insert e into PathEdge; Insert e into WorkList fi
                                                                                                                                                                                                                                                         } o
          end
          procedure ForwardTabulateSLRPs()
          begin
                                                                                                                                                                                                    SummaryEdge
                                                                                                                                                                                                                                                        <3,b> \to <4,c>
             while WorkList \neq \emptyset do
[10]
                 Select and remove an edge \langle s_p, d_1 \rangle \rightarrow \langle n, d_2 \rangle from WorkList
[11]
                                                                                                                                                                                                                 PathEdge
[12]
                 switch n
[13]
                     case n \in Call_n:
                        for each d_3 such that \langle n, d_2 \rangle \rightarrow \langle s_{calledProc(n)}, d_3 \rangle \in E^{\#} do Propagate(\langle s_{calledProc(n)}, d_3 \rangle \rightarrow \langle s_{calledProc(n)}, d_3 \rangle)
 [15]
[16]
                         od
                        for each d_3 such that \langle n, d_2 \rangle \rightarrow \langle returnSite(n), d_3 \rangle \in (E^{\#} \cup SummaryEdge) do
 [17]
[18]
                             Propagate(\langle s_p, d_1 \rangle \rightarrow \langle returnSite(n), d_3 \rangle)
[19]
                        od
[20]
                     end case
                                                                                                                                                                                                                   WorkList
                                                                                                                                                                                                                                                        \langle s_{main}, zero \rangle \rightarrow \langle 4, c \rangle
[21]
                     case n = e_n:
 [22]
[23]
                        for each c \in callers(p) do
                           for each d_4, d_5 such that \langle c, d_4 \rangle \to \langle s_p, d_1 \rangle \in E^{\#} and \langle e_p, d_2 \rangle \to \langle returnSite(c), d_5 \rangle \in E^{\#} do if \langle c, d_4 \rangle \to \langle returnSite(c), d_5 \rangle \notin SummaryEdge then
[24]
[25]
[26]
[27]
[28]
[29]
[30]
[31]
[32]
                                  Insert \langle c, d_4 \rangle \rightarrow \langle returnSite(c), d_5 \rangle into SummaryEdge

for each d_3 such that \langle s_{procOf(c)}, d_3 \rangle \rightarrow \langle c, d_4 \rangle \in PathEdge do

Propagate(\langle s_{procOf(c)}, d_3 \rangle \rightarrow \langle returnSite(c), d_5 \rangle)
                                   od
                            od
                     end case
[33]
[34]
[35]
                     case n \in (N_p - Call_p - \{e_p\}):

for each \langle m, d_3 \rangle such that \langle n, d_2 \rangle \rightarrow \langle m, d_3 \rangle \in E^{\#} do
                            Propagate(\langle s_p, d_1 \rangle \rightarrow \langle m, d_3 \rangle)
[36]
[37]
                     end case
[38]
[39]
                 end switch
             od
         end
```

```
void main() {
                                                                            void call(x) {
                                                                       6:
                              source();
                                                                       8:
                                                                                             return y;
                              call(b);
               sink(d);
 \langle s_{main}, zero \rangle \rightarrow \langle 1, zero \rangle
                                                     \langle s_{main}, zero \rangle \rightarrow \langle 2, a \rangle
  \langle s_{main}, zero \rangle \rightarrow \langle 3, b \rangle |\langle s_{call}, x \rangle \rightarrow \langle s_{call}, x \rangle
                                                                                         \langle s_{call}, x \rangle \rightarrow \langle 7, x \rangle
  \langle s_{call}, x \rangle \rightarrow \langle 8, y \rangle \langle s_{main}, zero \rangle \rightarrow \langle 4, c \rangle
```

```
void main() {
                                                                                                                                                                                                                                                                                                                                           void call(x) {
         declare PathEdge, WorkList, SummaryEdge: global edge set
         algorithm Tabulate(G_{IP}^{\#})
                                                                                                                                                                                                                                                                                                                                    6:
                                                                                                                                                                                                                     0:
         begin
            Let (N^{\#}, E^{\#}) = G_{IP}^{\#}
[1]
[2]
[3]
[4]
[5]
[6]
[7]
[8]
           PathEdge := \{\langle s_{main}, \mathbf{0} \rangle \rightarrow \langle s_{main}, \mathbf{0} \rangle\}

WorkList := \{\langle s_{main}, \mathbf{0} \rangle \rightarrow \langle s_{main}, \mathbf{0} \rangle\}

SummaryEdge := \emptyset
                                                                                                                                                                                                                                                                         source();
                                                                                                                                                                                                                                                                                                                                    8:
                                                                                                                                                                                                                      2:
                                                                                                                                                                                                                                                                                                                                                                    return y;
            ForwardTabulateSLRPs()
            for each n \in N^* do
                                                                                                                                                                                                                                                                         call(b);
               X_n := \{ d_2 \in D \mid \exists d_1 \in (D \cup \{ \mathbf{0} \}) \text{ such that } \langle s_{procOf(n)}, d_1 \rangle \rightarrow \langle n, d_2 \rangle \in \text{PathEdge} \}
         end
         procedure Propagate(e)
                                                                                                                                                                                                                                                    sink(d);
         begin
           if e \notin PathEdge then Insert e into PathEdge; Insert e into WorkList fi
                                                                                                                                                                                                                                      О
         end
         procedure ForwardTabulateSLRPs()
         begin
                                                                                                                                                                                                                                 <3,b> \to <4,d>
                                                                                                                                                                                  SummaryEdge
           while WorkList \neq \emptyset do
               Select and remove an edge \langle s_p, d_1 \rangle \rightarrow \langle n, d_2 \rangle from WorkList
[11]
                                                                                                                                                                                              PathEdge
                                                                                                                                                                                                                                                                                                          \langle s_{main}, zero \rangle \rightarrow \langle 2, a \rangle
                                                                                                                                                                                                                                 \langle s_{main}, zero \rangle \rightarrow \langle 1, zero \rangle
[12]
                switch n
                   case n \in Call_n:
[13]
                                                                                                                                                                                                                                 \langle s_{main}, zero \rangle \rightarrow \langle 3, b \rangle |\langle s_{call}, x \rangle \rightarrow \langle s_{call}, x \rangle
                                                                                                                                                                                                                                                                                                                                                               \langle s_{call}, x \rangle \rightarrow \langle 7, x \rangle
                      for each d_3 such that \langle n, d_2 \rangle \rightarrow \langle s_{calledProc(n)}, d_3 \rangle \in E^{\#} do Propagate(\langle s_{calledProc(n)}, d_3 \rangle \rightarrow \langle s_{calledProc(n)}, d_3 \rangle)
[15]
[16]
                                                                                                                                                                                                                                 \langle s_{call}, x \rangle \rightarrow \langle 8, y \rangle | \langle s_{main}, zero \rangle \rightarrow \langle 4, c \rangle
[17]
                      for each d_3 such that \langle n, d_2 \rangle \rightarrow \langle returnSite(n), d_3 \rangle \in (E^\# \cup SummaryEdge) do
[18]
                          Propagate(\langle s_n, d_1 \rangle \rightarrow \langle returnSite(n), d_3 \rangle)
[19]
                      od
[20]
                   end case
                                                                                                                                                                                              WorkList
[21]
                   case n = e_n:
[22]
[23]
                      for each c \in callers(p) do
                         for each d_4, d_5 such that \langle c, d_4 \rangle \rightarrow \langle s_p, d_1 \rangle \in E^{\#} and \langle e_p, d_2 \rangle \rightarrow \langle returnSite(c), d_5 \rangle \in E^{\#} do if \langle c, d_4 \rangle \rightarrow \langle returnSite(c), d_5 \rangle \not\in SummaryEdge then
[24]
[25]
[26]
[27]
[28]
[29]
[30]
[31]
[32]
                               Insert \langle c, d_4 \rangle \rightarrow \langle returnSite(c), d_5 \rangle into SummaryEdge for each d_3 such that \langle s_{procOf(c)}, d_3 \rangle \rightarrow \langle c, d_4 \rangle \in PathEdge do Propagate(\langle s_{procOf(c)}, d_3 \rangle \rightarrow \langle returnSite(c), d_5 \rangle)
                                                                                                                                                                                              \langle s_{main}, zero \rangle \rightarrow \langle 4, c \rangle
                                                                                                                                                                                                                                                                                  <4, c> \rightarrow <5, d>
                         od
                                                                                                                                                                                             Propagate (\langle s_{main}, zero \rangle \rightarrow \langle 5, d \rangle)
                   end case
                  case n \in (N_p - Call_p - \{e_p\}):

for each \langle m, d_3 \rangle such that \langle n, d_2 \rangle \rightarrow \langle m, d_3 \rangle \in E^{\#} do

Propagate(\langle s_p, d_1 \rangle \rightarrow \langle m, d_3 \rangle)
[34]
[35]
[36]
[37]
                      od
                  end case
[38]
[39]
               end switch
                                                                                                         NormalFlowFunction
         end
```

```
void call(x) {
                                                                                                                                                                                                                             void main() {
         declare PathEdge, WorkList, SummaryEdge: global edge set
         algorithm Tabulate(G_{IP}^{\#})
                                                                                                                                                                                                                                                                                                                                   6:
                                                                                                                                                                                                                    0:
        begin
            Let (N^{\#}, E^{\#}) = G_{IP}^{\#}
[1]
[2]
[3]
[4]
[5]
[6]
[7]
[8]
           PathEdge := \{\langle s_{main}, \mathbf{0} \rangle \rightarrow \langle s_{main}, \mathbf{0} \rangle\}
WorkList := \{\langle s_{main}, \mathbf{0} \rangle \rightarrow \langle s_{main}, \mathbf{0} \rangle\}
SummaryEdge := \emptyset
                                                                                                                                                                                                                                                                        source();
                                                                                                                                                                                                                                                                                                                                   8:
                                                                                                                                                                                                                    2:
                                                                                                                                                                                                                                                                                                                                                                   return y;
            ForwardTabulateSLRPs()
            for each n \in N^* do
                                                                                                                                                                                                                                                                        call(b);
               X_n := \{ d_2 \in D \mid \exists d_1 \in (D \cup \{ \mathbf{0} \}) \text{ such that } \langle s_{procOf(n)}, d_1 \rangle \rightarrow \langle n, d_2 \rangle \in \text{PathEdge} \}
         end
         procedure Propagate(e)
                                                                                                                                                                                                                                                   sink(d);
         begin
           if e \notin PathEdge then Insert e into PathEdge; Insert e into WorkList fi
                                                                                                                                                                                                                                } o
         end
         procedure ForwardTabulateSLRPs()
         begin
                                                                                                                                                                                                                                <3,b> \to <4,d>
                                                                                                                                                                                 SummaryEdge
           while WorkList \neq \emptyset do
               Select and remove an edge \langle s_p, d_1 \rangle \rightarrow \langle n, d_2 \rangle from WorkList
[11]
                                                                                                                                                                                             PathEdge
                                                                                                                                                                                                                                                                                                         \langle s_{main}, zero \rangle \rightarrow \langle 2, a \rangle
                                                                                                                                                                                                                                \langle s_{main}, zero \rangle \rightarrow \langle 1, zero \rangle
[12]
               switch n
                   case n \in Call_n:
[13]
                                                                                                                                                                                                                                \langle s_{main}, zero \rangle \rightarrow \langle 3, b \rangle |\langle s_{call}, x \rangle \rightarrow \langle s_{call}, x \rangle
                                                                                                                                                                                                                                                                                                                                                             \langle s_{call}, x \rangle \rightarrow \langle 7, x \rangle
                     for each d_3 such that \langle n, d_2 \rangle \rightarrow \langle s_{calledProc(n)}, d_3 \rangle \in E^{\#} do Propagate(\langle s_{calledProc(n)}, d_3 \rangle \rightarrow \langle s_{calledProc(n)}, d_3 \rangle)
[15]
[16]
                                                                                                                                                                                                                                \langle s_{call}, x \rangle \rightarrow \langle 8, y \rangle |\langle s_{main}, zero \rangle \rightarrow \langle 4, c \rangle |\langle s_{main}, zero \rangle \rightarrow \langle 5, d \rangle
[17]
                      for each d_3 such that \langle n, d_2 \rangle \rightarrow \langle returnSite(n), d_3 \rangle \in (E^\# \cup SummaryEdge) do
[18]
                          Propagate(\langle s_p, d_1 \rangle \rightarrow \langle returnSite(n), d_3 \rangle)
[19]
                      od
[20]
                   end case
                                                                                                                                                                                                                                \langle s_{main}, zero \rangle \rightarrow \langle 5, d \rangle
                                                                                                                                                                                              WorkList
[21]
                   case n = e_n:
[22]
[23]
                      for each c \in callers(p) do
                         for each d_4, d_5 such that \langle c, d_4 \rangle \rightarrow \langle s_p, d_1 \rangle \in E^{\#} and \langle e_p, d_2 \rangle \rightarrow \langle returnSite(c), d_5 \rangle \in E^{\#} do if \langle c, d_4 \rangle \rightarrow \langle returnSite(c), d_5 \rangle \not\in SummaryEdge then
[24]
[25]
[26]
[27]
[28]
[29]
[30]
[31]
[32]
                               Insert \langle c, d_4 \rangle \rightarrow \langle returnSite(c), d_5 \rangle into SummaryEdge

for each d_3 such that \langle s_{procOf(c)}, d_3 \rangle \rightarrow \langle c, d_4 \rangle \in PathEdge do

Propagate(\langle s_{procOf(c)}, d_3 \rangle \rightarrow \langle returnSite(c), d_5 \rangle)
                         od
                                                                                                                                                                                            Propagate (\langle s_{main}, zero \rangle \rightarrow \langle 5, d \rangle)
                   end case
                   case n \in (N_p - Call_p - \{e_p\}):

for each \langle m, d_3 \rangle such that \langle n, d_2 \rangle \rightarrow \langle m, d_3 \rangle \in E^{\#} do
[34]
[35]
                         Propagate(\langle s_p, d_1 \rangle \rightarrow \langle m, d_3 \rangle)
[36]
[37]
                   end case
[38]
[39]
               end switch
           od
        end
```

```
void main() {
                                                                                                                                                                                                                                                                                                                                             void call(x) {
         declare PathEdge, WorkList, SummaryEdge: global edge set
         algorithm Tabulate(G_{IP}^{\#})
                                                                                                                                                                                                                                                                                                                                      6:
                                                                                                                                                                                                                      0:
         begin
            Let (N^{\#}, E^{\#}) = G_{IP}^{\#}
[1]
[2]
[3]
[4]
[5]
[6]
[7]
[8]
           PathEdge := \{\langle s_{main}, \mathbf{0} \rangle \rightarrow \langle s_{main}, \mathbf{0} \rangle\}

WorkList := \{\langle s_{main}, \mathbf{0} \rangle \rightarrow \langle s_{main}, \mathbf{0} \rangle\}

SummaryEdge := \emptyset
                                                                                                                                                                                                                                                                           source();
                                                                                                                                                                                                                                                                                                                                      8:
                                                                                                                                                                                                                       2:
                                                                                                                                                                                                                                                                                                                                                                       return y;
            ForwardTabulateSLRPs()
            for each n \in N^* do
                                                                                                                                                                                                                                                                           call(b);
               X_n := \{ d_2 \in D \mid \exists d_1 \in (D \cup \{ \mathbf{0} \}) \text{ such that } \langle s_{procOf(n)}, d_1 \rangle \rightarrow \langle n, d_2 \rangle \in \text{PathEdge} \}
         end
         procedure Propagate(e)
                                                                                                                                                                                                                                                      sink(d);
         begin
           if e \notin PathEdge then Insert e into PathEdge; Insert e into WorkList fi
         end
                                                                                                                                                                                                                                        О
                                                                                                                                                                                                                                                                                         С
         procedure ForwardTabulateSLRPs()
         begin
                                                                                                                                                                                                                                  <3,b> \to <4,d>
                                                                                                                                                                                   SummaryEdge
            while WorkList \neq \emptyset do
               Select and remove an edge \langle s_p, d_1 \rangle \rightarrow \langle n, d_2 \rangle from WorkList
[11]
                                                                                                                                                                                               PathEdge
                                                                                                                                                                                                                                                                                                            \langle s_{main}, zero \rangle \rightarrow \langle 2, a \rangle
                                                                                                                                                                                                                                  \langle s_{main}, zero \rangle \rightarrow \langle 1, zero \rangle
[12]
               switch n
[13]
                   case n \in Call_n:
                                                                                                                                                                                                                                  \langle s_{main}, zero \rangle \rightarrow \langle 3, b \rangle |\langle s_{call}, x \rangle \rightarrow \langle s_{call}, x \rangle
                                                                                                                                                                                                                                                                                                                                                                 \langle s_{call}, x \rangle \rightarrow \langle 7, x \rangle
                      for each d_3 such that \langle n, d_2 \rangle \rightarrow \langle s_{calledProc(n)}, d_3 \rangle \in E^{\#} do Propagate(\langle s_{calledProc(n)}, d_3 \rangle \rightarrow \langle s_{calledProc(n)}, d_3 \rangle)
[15]
[16]
                                                                                                                                                                                                                                   \langle s_{call}, x \rangle \rightarrow \langle 8, y \rangle |\langle s_{main}, zero \rangle \rightarrow \langle 4, c \rangle |\langle s_{main}, zero \rangle \rightarrow \langle 5, d \rangle
[17]
                      for each d_3 such that \langle n, d_2 \rangle \rightarrow \langle returnSite(n), d_3 \rangle \in (E^\# \cup SummaryEdge) do
[18]
                          Propagate(\langle s_p, d_1 \rangle \rightarrow \langle returnSite(n), d_3 \rangle)
[19]
                      od
[20]
                   end case
                                                                                                                                                                                                WorkList
[21]
                   case n = e_n:
[22]
[23]
                      for each c \in callers(p) do
                         for each d_4, d_5 such that \langle c, d_4 \rangle \rightarrow \langle s_p, d_1 \rangle \in E^{\#} and \langle e_p, d_2 \rangle \rightarrow \langle returnSite(c), d_5 \rangle \in E^{\#} do if \langle c, d_4 \rangle \rightarrow \langle returnSite(c), d_5 \rangle \not\in SummaryEdge then
[24]
[25]
[26]
[27]
[28]
[29]
[30]
[31]
[32]
                               Insert \langle c, d_4 \rangle \rightarrow \langle returnSite(c), d_5 \rangle into SummaryEdge for each d_3 such that \langle s_{procOf(c)}, d_3 \rangle \rightarrow \langle c, d_4 \rangle \in PathEdge do Propagate(\langle s_{procOf(c)}, d_3 \rangle \rightarrow \langle returnSite(c), d_5 \rangle)
                                                                                                                                                                                                \langle s_{main}, zero \rangle \rightarrow \langle 5, d \rangle
                                                                                                                                                                                                                                                                              \langle 5, d \rangle \rightarrow \langle e_{main}, d \rangle
                                                                                                                                                                                              Propagate(\langle s_{main}, zero \rangle \rightarrow \langle e_{main}, d \rangle)
                   end case
                  case n \in (N_p - Call_p - \{e_p\}):

for each \langle m, d_3 \rangle such that \langle n, d_2 \rangle \rightarrow \langle m, d_3 \rangle \in E^{\#} do

Propagate(\langle s_p, d_1 \rangle \rightarrow \langle m, d_3 \rangle)
[33]
[34]
[35]
[36]
[37]
                      od
                   end case
[38]
[39]
               end switch
                                                                                                          NormalFlowFunction
         end
```

```
void main() {
                                                                                                                                                                                                                                                                                                                                        void call(x) {
         declare PathEdge, WorkList, SummaryEdge: global edge set
         algorithm Tabulate(G_{IP}^{\#})
                                                                                                                                                                                                                                                                                                                                 6:
                                                                                                                                                                                                                   0:
        begin
            Let (N^{\#}, E^{\#}) = G_{IP}^{\#}
[1]
[2]
[3]
[4]
[5]
[6]
[7]
[8]
           PathEdge := \{\langle s_{main}, \mathbf{0} \rangle \rightarrow \langle s_{main}, \mathbf{0} \rangle\}

WorkList := \{\langle s_{main}, \mathbf{0} \rangle \rightarrow \langle s_{main}, \mathbf{0} \rangle\}

SummaryEdge := \emptyset
                                                                                                                                                                                                                                                                      source();
                                                                                                                                                                                                                                                                                                                                 8:
                                                                                                                                                                                                                   2:
                                                                                                                                                                                                                                                                                                                                                                 return y;
            ForwardTabulateSLRPs()
            for each n \in N^* do
                                                                                                                                                                                                                                                                       call(b);
               X_n := \{ d_2 \in D \mid \exists d_1 \in (D \cup \{ \mathbf{0} \}) \text{ such that } \langle s_{procOf(n)}, d_1 \rangle \rightarrow \langle n, d_2 \rangle \in \text{PathEdge} \}
         end
         procedure Propagate(e)
                                                                                                                                                                                                                                                  sink(d)
         begin
           if e \notin PathEdge then Insert e into PathEdge; Insert e into WorkList fi
                                                                                                                                                                                                                                    О
         end
                                                                                                                                                                                                                                                                                      С
         procedure ForwardTabulateSLRPs()
         begin
                                                                                                                                                                                                                              <3,b> \to <4,d>
                                                                                                                                                                                SummaryEdge
           while WorkList \neq \emptyset do
[10]
               Select and remove an edge \langle s_p, d_1 \rangle \rightarrow \langle n, d_2 \rangle from WorkList
[11]
                                                                                                                                                                                            PathEdge
                                                                                                                                                                                                                                                                                                       \langle s_{main}, zero \rangle \rightarrow \langle 2, a \rangle
                                                                                                                                                                                                                              \langle s_{main}, zero \rangle \rightarrow \langle 1, zero \rangle
[12]
               switch n
                   case n \in Call_n:
[13]
                                                                                                                                                                                                                               \langle s_{main}, zero \rangle \rightarrow \langle 3, b \rangle |\langle s_{call}, x \rangle \rightarrow \langle s_{call}, x \rangle
                                                                                                                                                                                                                                                                                                                                                           \langle s_{call}, x \rangle \rightarrow \langle 7, x \rangle
                     for each d_3 such that \langle n, d_2 \rangle \rightarrow \langle s_{calledProc(n)}, d_3 \rangle \in E^{\#} do Propagate(\langle s_{calledProc(n)}, d_3 \rangle \rightarrow \langle s_{calledProc(n)}, d_3 \rangle)
[15]
[16]
                                                                                                                                                                                                                               \langle s_{call}, x \rangle \rightarrow \langle 8, y \rangle |\langle s_{main}, zero \rangle \rightarrow \langle 4, c \rangle |\langle s_{main}, zero \rangle \rightarrow \langle 5, d \rangle
[17]
                      for each d_3 such that \langle n, d_2 \rangle \rightarrow \langle returnSite(n), d_3 \rangle \in (E^\# \cup SummaryEdge) do
                         Propagate(\langle s_p, d_1 \rangle \rightarrow \langle returnSite(n), d_3 \rangle)
[18]
                                                                                                                                                                                                                                                                                                                                         \langle s_{main}, zero \rangle \rightarrow \langle e_{main}, d \rangle
[19]
                      od
[20]
                   end case
                                                                                                                                                                                             WorkList
                                                                                                                                                                                                                              \langle s_{main}, zero \rangle \rightarrow \langle e_{main}, d \rangle
[21]
                   case n = e_n:
[22]
[23]
                      for each c \in callers(p) do
                        for each d_4, d_5 such that \langle c, d_4 \rangle \to \langle s_p, d_1 \rangle \in E^{\#} and \langle e_p, d_2 \rangle \to \langle returnSite(c), d_5 \rangle \in E^{\#} do if \langle c, d_4 \rangle \to \langle returnSite(c), d_5 \rangle \notin SummaryEdge then
[24]
[25]
[26]
[27]
[28]
[29]
[30]
[31]
[32]
                               Insert \langle c, d_4 \rangle \rightarrow \langle returnSite(c), d_5 \rangle into SummaryEdge

for each d_3 such that \langle s_{procOf(c)}, d_3 \rangle \rightarrow \langle c, d_4 \rangle \in PathEdge do

Propagate(\langle s_{procOf(c)}, d_3 \rangle \rightarrow \langle returnSite(c), d_5 \rangle)
                                                                                                                                                                                           Propagate(\langle s_{main}, zero \rangle \rightarrow \langle e_{main}, d \rangle)
                   end case
                   case n \in (N_p - Call_p - \{e_p\}):

for each \langle m, d_3 \rangle such that \langle n, d_2 \rangle \rightarrow \langle m, d_3 \rangle \in E^{\#} do
[34]
[35]
                         Propagate(\langle s_p, d_1 \rangle \rightarrow \langle m, d_3 \rangle)
[36]
[37]
                   end case
[38]
[39]
               end switch
           od
        end
```

```
void call(x) {
                                                                                                                                                                                                                          void main() {
         declare PathEdge, WorkList, SummaryEdge: global edge set
         algorithm Tabulate(G_{IP}^{\#})
                                                                                                                                                                                                                                                                                                                               6:
                                                                                                                                                                                                                  0:
        begin
           Let (N^{\#}, E^{\#}) = G_{IP}^{\#}
[1]
[2]
[3]
[4]
[5]
[6]
[7]
[8]
           PathEdge := \{\langle s_{main}, \mathbf{0} \rangle \rightarrow \langle s_{main}, \mathbf{0} \rangle\}

WorkList := \{\langle s_{main}, \mathbf{0} \rangle \rightarrow \langle s_{main}, \mathbf{0} \rangle\}

SummaryEdge := \emptyset
                                                                                                                                                                                                                                                                     source();
                                                                                                                                                                                                                                                                                                                               8:
                                                                                                                                                                                                                  2:
                                                                                                                                                                                                                                                                                                                                                               return y;
            ForwardTabulateSLRPs()
           for each n \in N^* do
                                                                                                                                                                                                                                                                      call(b);
               X_n := \{ d_2 \in D \mid \exists d_1 \in (D \cup \{ \mathbf{0} \}) \text{ such that } \langle s_{procOf(n)}, d_1 \rangle \rightarrow \langle n, d_2 \rangle \in \text{PathEdge} \}
         end
         procedure Propagate(e)
                                                                                                                                                                                                                                                 sink(d)
         begin
           if e \notin PathEdge then Insert e into PathEdge; Insert e into WorkList fi
                                                                                                                                                                                                                                  О
         end
                                                                                                                                                                                                                                                                                    С
         procedure ForwardTabulateSLRPs()
         begin
                                                                                                                                                                                                                             <3,b> \to <4,d>
                                                                                                                                                                               SummaryEdge
           while WorkList \neq \emptyset do
               Select and remove an edge \langle s_p, d_1 \rangle \rightarrow \langle n, d_2 \rangle from WorkList
[11]
                                                                                                                                                                                           PathEdge
                                                                                                                                                                                                                                                                                                      \langle s_{main}, zero \rangle \rightarrow \langle 2, a \rangle
                                                                                                                                                                                                                             \langle s_{main}, zero \rangle \rightarrow \langle 1, zero \rangle
[12]
               switch n
[13]
                  case n \in Call_n:
                                                                                                                                                                                                                             \langle s_{main}, zero \rangle \rightarrow \langle 3, b \rangle |\langle s_{call}, x \rangle \rightarrow \langle s_{call}, x \rangle
                                                                                                                                                                                                                                                                                                                                                         \langle s_{call}, x \rangle \rightarrow \langle 7, x \rangle
                     for each d_3 such that \langle n, d_2 \rangle \rightarrow \langle s_{calledProc(n)}, d_3 \rangle \in E^{\#} do Propagate(\langle s_{calledProc(n)}, d_3 \rangle \rightarrow \langle s_{calledProc(n)}, d_3 \rangle)
 [15]
[16]
                                                                                                                                                                                                                             \langle s_{call}, x \rangle \rightarrow \langle 8, y \rangle |\langle s_{main}, zero \rangle \rightarrow \langle 4, c \rangle |\langle s_{main}, zero \rangle \rightarrow \langle 5, d \rangle
[17]
                     for each d_3 such that \langle n, d_2 \rangle \rightarrow \langle returnSite(n), d_3 \rangle \in (E^\# \cup SummaryEdge) do
                         Propagate(\langle s_p, d_1 \rangle \rightarrow \langle returnSite(n), d_3 \rangle)
[18]
                                                                                                                                                                                                                                                                                                                                       \langle s_{main}, zero \rangle \rightarrow \langle e_{main}, d \rangle
[19]
                     od
[20]
                   end case
                                                                                                                                                                                           WorkList
[21]
                  case n = e_n:
[22]
[23]
                     for each c \in callers(p) do
                        for each d_4, d_5 such that \langle c, d_4 \rangle \to \langle s_p, d_1 \rangle \in E^\# and \langle e_p, d_2 \rangle \to \langle returnSite(c), d_5 \rangle \in E^\# do if \langle c, d_4 \rangle \to \langle returnSite(c), d_5 \rangle \notin SummaryEdge then
[24]
[25]
[26]
[27]
[28]
[29]
[30]
[31]
[32]
                               Insert \langle c, d_4 \rangle \rightarrow \langle returnSite(c), d_5 \rangle into SummaryEdge

for each d_3 such that \langle s_{procOf(c)}, d_3 \rangle \rightarrow \langle c, d_4 \rangle \in PathEdge do

Propagate(\langle s_{procOf(c)}, d_3 \rangle \rightarrow \langle returnSite(c), d_5 \rangle)
                                                                                                                                                                                            \langle s_{main}, zero \rangle \rightarrow \langle e_{main}, d \rangle
                                                                                                                                                                                            Callers(main) = \emptyset;
                  end case
                                                                                                                                                                                                       算法终止!
[33]
[34]
[35]
                  case n \in (N_p - Call_p - \{e_p\}):

for each \langle m, d_3 \rangle such that \langle n, d_2 \rangle \rightarrow \langle m, d_3 \rangle \in E^{\#} do
                         Propagate (\langle s_p, d_1 \rangle \rightarrow \langle m, d_3 \rangle)
[36]
[37]
                  end case
[38]
[39]
               end switch
        end
```

```
void main() {
                                                                                                                                                                                                                                                                                                                                  void call(x) {
         declare PathEdge, WorkList, SummaryEdge: global edge set
         algorithm Tabulate(G_{IP}^{\#})
                                                                                                                                                                                                                                                                                                                           6:
                                                                                                                                                                                                               0:
         begin
            Let (N^{\#}, E^{\#}) = G_{IP}^{\#}
[1]
[2]
[3]
[4]
[5]
[6]
[7]
[8]
           PathEdge := \{\langle s_{main}, \mathbf{0} \rangle \rightarrow \langle s_{main}, \mathbf{0} \rangle\}
WorkList := \{\langle s_{main}, \mathbf{0} \rangle \rightarrow \langle s_{main}, \mathbf{0} \rangle\}
SummaryEdge := \emptyset
                                                                                                                                                                                                                                                                  source();
                                                                                                                                                                                                                                                                                                                           8:
                                                                                                                                                                                                                2:
                                                                                                                                                                                                                                                                                                                                                           return y;
            ForwardTabulateSLRPs()
           for each n \in N^* do
                                                                                                                                                                                                                                                                   call(b);
                                                                                                                                                                                                                3:
               X_n := \{ d_2 \in D \mid \exists d_1 \in (D \cup \{ \mathbf{0} \}) \text{ such that } \langle s_{procOf(n)}, d_1 \rangle \rightarrow \langle n, d_2 \rangle \in \text{PathEdge} \}
         end
         procedure Propagate(e)
                                                                                                                                                                                                                                              sink(d)
         begin
           if e \notin PathEdge then Insert e into PathEdge; Insert e into WorkList fi
                                                                                                                                                                                                                                О
          end
                                                                                                                                                                                                                                                                                 С
          procedure ForwardTabulateSLRPs()
         begin
                                                                                                                                                                                                                           <3,b> \to <4,d>
                                                                                                                                                                             SummaryEdge
           while WorkList \neq \emptyset do
[10]
               Select and remove an edge \langle s_p, d_1 \rangle \rightarrow \langle n, d_2 \rangle from WorkList
[11]
                                                                                                                                                                                         PathEdge
                                                                                                                                                                                                                                                                                                  \langle s_{main}, zero \rangle \rightarrow \langle 2, a \rangle
                                                                                                                                                                                                                           \langle s_{main}, zero \rangle \rightarrow \langle 1, zero \rangle
 [12]
               switch n
                   case n \in Call_n:
[13]
                                                                                                                                                                                                                           \langle s_{main}, zero \rangle \rightarrow \langle 3, b \rangle |\langle s_{call}, x \rangle \rightarrow \langle s_{call}, x \rangle
                                                                                                                                                                                                                                                                                                                                                     \langle s_{call}, x \rangle \rightarrow \langle 7, x \rangle
                     for each d_3 such that \langle n, d_2 \rangle \rightarrow \langle s_{calledProc(n)}, d_3 \rangle \in E^{\#} do Propagate(\langle s_{calledProc(n)}, d_3 \rangle \rightarrow \langle s_{calledProc(n)}, d_3 \rangle)
 [15]
 [16]
                                                                                                                                                                                                                           \langle s_{call}, x \rangle \rightarrow \langle 8, y \rangle |\langle s_{main}, zero \rangle \rightarrow \langle 4, c \rangle |\langle s_{main}, zero \rangle \rightarrow \langle 5, d \rangle
 [17]
                      for each d_3 such that \langle n, d_2 \rangle \rightarrow \langle returnSite(n), d_3 \rangle \in (E^\# \cup SummaryEdge) do
 [18]
                          Propagate(\langle s_n, d_1 \rangle \rightarrow \langle returnSite(n), d_3 \rangle)
                                                                                                                                                                                                                                                                                                                                    \langle s_{main}, zero \rangle \rightarrow \langle e_{main}, d \rangle
 [19]
                      od
 [20]
                   end case
                                                                                                                                                                                         WorkList
 [21]
                   case n = e_n:
 [22]
[23]
                      for each c \in callers(p) do
                        for each d_4, d_5 such that \langle c, d_4 \rangle \rightarrow \langle s_p, d_1 \rangle \in E^{\#} and \langle e_p, d_2 \rangle \rightarrow \langle returnSite(c), d_5 \rangle \in E^{\#} do if \langle c, d_4 \rangle \rightarrow \langle returnSite(c), d_5 \rangle \not\in SummaryEdge then
 [24]
[25]
[26]
[27]
[28]
[29]
[30]
[31]
[32]
                               Insert \langle c, d_4 \rangle \rightarrow \langle returnSite(c), d_5 \rangle into SummaryEdge for each d_3 such that \langle s_{procOf(c)}, d_3 \rangle \rightarrow \langle c, d_4 \rangle \in PathEdge do Propagate(\langle s_{procOf(c)}, d_3 \rangle \rightarrow \langle returnSite(c), d_5 \rangle)
                                                                                                                                                                                                                                                     X_5 \supseteq \{d\};
                                                                                                                                                                                                        定义sink第0个参数为sink点,则找到污点缺陷
                   end case
                   case n \in (N_p - Call_p - \{e_p\}):

for each \langle m, d_3 \rangle such that \langle n, d_2 \rangle \rightarrow \langle m, d_3 \rangle \in E^{\#} do
 [34]
[35]
                         Propagate(\langle s_p, d_1 \rangle \rightarrow \langle m, d_3 \rangle)
 [36]
[37]
                   end case
[38]
[39]
               end switch
           od
         end
```