

All-Paths Algorithm

Roland Backhouse
October 22, 2002

Overview

- *Goal*: derive a single generic path-finding algorithm.
- Exploit algebraic properties common to a variety of path-finding problems.
- *Key idea*: property of being a Kleene algebra extends to graphs/matrices.
- Develop algorithm in two steps.
 - skeleton using operations on graphs,
 - detailed implementation using operations on edge labels.

Reachability Problem

Given is an $n \times n$ matrix G where g_{ij} is **true** if there is an edge from node numbered i to the node numbered j , and **false** otherwise.

Determine, for each i and j , whether there is a path from i to j .

```
for k := 0 to n-1
do for i := 0 to n-1
  do for j := 0 to n-1
    do  $g_{ij} := g_{ij} \vee (g_{ik} \wedge g_{kj})$ 
  end_for
end_for
end_for
```

Least-Cost Paths

g_{ij} represents the least *cost* of traversing an edge from node i to node j . Determine, for each i and j , the least cost of a path from i to j .

```
for k := 0 to n-1
do for i := 0 to n-1
  do for j := 0 to n-1
    do  $g_{ij} := g_{ij} \downarrow (g_{ik} + g_{kj})$ 
    end_for
  end_for
end_for
end_for
```

Bottleneck Problem

g_{ij} represents the height of the lowest underpass on the road connecting i and j . Determine, for each i and j , the height of the lowest underpass on the best route from i to j .

```
for k := 0 to n-1
do for i := 0 to n-1
  do for j := 0 to n-1
    do  $g_{ij} := g_{ij} \uparrow (g_{ik} \downarrow g_{kj})$ 
  end_for
end_for
end_for
```

All Paths

Edge labels are letters of some alphabet and a path spells out a word. Determine, for each i and j , a regular expression representing all words spelt out by a path from i to j .

```
for k := 0 to n-1
do for i := 0 to n-1
  do for j := 0 to n-1
    do  $g_{ij} := g_{ij} + g_{ik}(g_{kk})^*g_{kj}$ 
    end_for
  end_for
end_for
end_for
```

Selectors

\mathbf{N} is the set of nodes of the graph.

i , j and k denote individual nodes of the graph.

L , M and P denote sets of nodes (i.e. subsets of \mathbf{N}).

$1 \times |\mathbf{N}|$ and $|\mathbf{N}| \times 1$ matrices are called *vectors*, 1×1 matrices will be called *scalars* and $|\mathbf{N}| \times |\mathbf{N}|$ matrices will be called *matrices*.

$\langle k|$ is the $1 \times |\mathbf{N}|$ vector that differs from $\mathbf{0}$ only in its k th component which is $\mathbf{1}$. Such a vector is called a *primitive selector vector*. The transpose of $\langle k|$ (thus an $|\mathbf{N}| \times 1$ vector) is denoted by $|k\rangle$.

We define the $|\mathbf{N}| \times |\mathbf{N}|$ *primitive selector matrix* \underline{k} by the equation

$$\underline{k} = |k\rangle \cdot \langle k| \quad . \quad (1)$$

Properties of Selectors

$$\underline{M} = \langle \Sigma k:k \in M : \underline{k} \rangle \quad (2)$$

$$\underline{\{k\}} = \underline{k} \quad (3)$$

$$X \cdot \underline{\phi} = \underline{\phi} \cdot X = \underline{\phi} \quad (4)$$

$$X \cdot \underline{\phi} = \underline{\phi} \cdot X = \underline{\phi} \quad (5)$$

$$\underline{\phi}^* = \mathbf{1} \quad (6)$$

$$\underline{LUM} = \underline{L} + \underline{M} \quad (7)$$

$$X \cdot \underline{N} = X = \underline{N} \cdot X \quad (8)$$

Identifying an Invariant

Assume $N = L \cup M$. Then,

$$G^* = (\underline{N} \cdot G)^* = (\underline{L \cup M} \cdot G)^* = (\underline{L} \cdot G + \underline{M} \cdot G)^* = (\underline{L} \cdot G)^* \cdot (\underline{M} \cdot G \cdot (\underline{L} \cdot G)^*)^* .$$

Thus,

$$G \cdot G^* = G \cdot (\underline{L} \cdot G)^* \cdot (\underline{M} \cdot G \cdot (\underline{L} \cdot G)^*)^* .$$

Define $f(X, P)$ by

$$f(X, P) = X \cdot (\underline{P} \cdot X)^* . \quad (9)$$

Then the above calculation establishes first that

$$G \cdot G^* = f(G, N) \quad (10)$$

and

$$f(X, L \cup M) = f(X \cdot (\underline{L} \cdot X)^* , M) . \quad (11)$$

Moreover, since $\underline{\phi} \cdot X = \underline{\phi}$ and $\underline{\phi}^* = \mathbf{1}$,

$$f(X, \phi) = X . \quad (12)$$

Skeleton Algorithm (Continued)

Assume nodes are numbered from 0 through $n-1$. Thus, $N = [0..n)$.

$X, k := G, 0$

{ *Invariant*: $G^+ = X \cdot (\underline{P} \cdot X)^*$ where $P = [k..n)$ }

; do $k \neq n \rightarrow X, k := X \cdot (\underline{k} \cdot X)^*$, $k+1$

od

{ $G^+ = X$ } .

Reducing to Primitive Operations

$$\begin{aligned}
 & X \cdot (\underline{k} \cdot X)^* \\
 = & \quad \{ \text{unfolding } (\underline{k} \cdot X)^* \} \\
 & X \cdot (\mathbf{1} + (\underline{k} \cdot X)^* \cdot \underline{k} \cdot X) \\
 = & \quad \{ \text{distributivity, unit} \} \\
 & X + X \cdot (\underline{k} \cdot X)^* \cdot \underline{k} \cdot X \\
 = & \quad \{ \underline{k} = |k\rangle \cdot \langle k| \} \\
 & X + X \cdot (|k\rangle \cdot \langle k| \cdot X)^* \cdot |k\rangle \cdot \langle k| \cdot X \\
 = & \quad \{ \text{mirror rule for } |k\rangle \} \\
 & X + X \cdot |k\rangle \cdot \langle k| X |k\rangle^* \cdot \langle k| \cdot X .
 \end{aligned}$$

Reducing to Primitive Operations

$X, k := G, 0$

{ *Invariant*: $G^+ = X \cdot (\underline{P} \cdot X)^*$ where $P = [k..n)$ }

; do $k \neq n \rightarrow X, k := X + X \cdot |k\rangle \cdot \langle k|X|k\rangle^* \cdot \langle k| \cdot X, k+1$

od

{ $G^+ = X$ } .

Reducing to Primitive Operations

$$X := X + X \cdot |k\rangle \cdot \langle k|X|k\rangle^* \cdot \langle k| \cdot X$$

is directly implemented as the *simultaneous* assignment

```
simultaneously_for (i := 0 to n-1) and (j := 0 to n-1)
do  $\langle i|X|j\rangle := \langle i|X|j\rangle + \langle i|X|k\rangle \cdot \langle k|X|k\rangle^* \cdot \langle k|X|j\rangle$ 
end_for
```

Writing $\langle i|X|j\rangle$ conventionally as x_{ij}

```
simultaneously_for (i := 0 to n-1) and (j := 0 to n-1)
do  $x_{ij} := x_{ij} + x_{ik} \cdot (x_{kk})^* \cdot x_{kj}$ 
end_for
```

Exploiting Idempotence

Mapping

$$X := X + X \cdot |k\rangle \cdot \langle k|X|k\rangle^* \cdot \langle k| \cdot X$$

is a closure operator. Hence, it can be implemented using a destructive assignment:

```

X,k := G,0
{ Invariant:  $G^+ = X \cdot (\underline{P} \cdot X)^*$  where  $P = [k..n)$  }
; do k  $\neq$  n  $\rightarrow$  for each pair (i,j),  $0 \leq i,j < n$ 
    do  $x_{ij} := x_{ij} + x_{ik} \cdot (x_{kk})^* \cdot x_{kj}$ 
    end_for
; k := k+1
od
{  $G^+ = X$  } .

```