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 ∨ (g_ik ∧ 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 ↓ (g_ik + g_kj)
        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 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
```

Selectors

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 N).

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

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

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

$$\underline{\mathsf{k}} = |\mathsf{k}\rangle \cdot \langle \mathsf{k}| \quad . \tag{1}$$

Properties of Selectors

$$\underline{M} = \langle \Sigma k : k \in M : \underline{k} \rangle \tag{2}$$

$$\underline{\{k\}} = \underline{k} \tag{3}$$

$$X \cdot \underline{\phi} = \underline{\phi} \cdot X = \underline{\phi} \tag{4}$$

$$X \cdot \underline{\phi} = \underline{\phi} \cdot X = \underline{\phi} \tag{5}$$

$$\underline{\phi}^* = \mathbf{1} \tag{6}$$

$$\underline{L \cup M} = \underline{L} + \underline{M} \tag{7}$$

$$X \cdot \underline{N} = X = \underline{N} \cdot X \tag{8}$$

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

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

Then the above calculation establishes first that

$$G \cdot G^* = f(G,N) \tag{10}$$

and

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

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

$$f(X,\phi) = X \ . \tag{12}$$

Skeleton Algorithm (Continued)

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

$$X,k \ := \ G,0$$

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

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

$$\mathsf{od}$$

$$\{ \ G^{+}=X \ \} \ .$$

Reducing to Primitive Operations

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

Reducing to Primitive Operations

$X,k := G,0$

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

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

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

$\{\; G^{+} = X \;\}$.