Disjoint Set Forests for Maintaining Connected Components

Gio Borje and Craig Steinke

UC Irvine

August 1, 2013
Outline

1. Motivation
2. Graph Theory Definitions
3. Algorithm for Connected Components
4. Disjoint-Set Forest
Motivation: Uniquely Labeling Connected Components

For each pair of vertices, if the pair is connected, union their components.

**Figure:** Unique labels of the connected components of a graph
Dynamic Solution

Disjoint-Set data structure for keeping tracking of partitions of a set. Given a partition of a set of elements, solve two problems dynamically:

- **Find(x)** Determine the subset for which an element belongs to
- **Union(S1, S2)** Merge two subsets into a single set
Definitions

Definition
For undirected graphs, vertex v is reachable from vertex u if there exists a path from u to v. A path from u to v is denoted as $u \rightsarrow v$.

Definition
A connected component of an undirected graph is a subgraph such that any vertex of this component can reach every other vertex and contains no other vertices of the supergraph.

Note that connected components are a type of equivalence class.
Examples

(a) $f \sim b$

(b) $\text{equiv}(f) = \{c, a, g, d, b\}$
General Algorithm for Connected Components

- Reduce the graph into a set of sets where each set contains only an individual element. Blank spaces are ignored.

- The second step is to merge sets if their components are adjacent to each other.

- We require a data structure to maintain connected components.

**Figure:** Reference Graph

```
<table>
<thead>
<tr>
<th></th>
<th>f</th>
<th>c</th>
<th>a</th>
</tr>
</thead>
<tbody>
<tr>
<td>g</td>
<td>d</td>
<td>b</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>q</th>
<th>s</th>
<th>r</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Gio Borje and Craig Steinke

Disjoint Set Forests for Maintaining Connected Components
procedure \textsc{Find}(x)
\begin{algorithmic}
  \If{x.parent \neq x}
    \State{x.parent \leftarrow \textsc{Find}(x.parent)}
  \EndIf
  \State \Return x.parent
\end{algorithmic}
end procedure

procedure \textsc{Union}(x, y)
\begin{algorithmic}
  \State \textsc{Find}(x).parent \leftarrow \textsc{Find}(y)
\end{algorithmic}
end procedure
Labeling Components with Disjoint Sets

(a) Before

(b) After
Thank you