Introduction to Programming

Bertrand Meyer

Lecture 21
Topological Sort
Part 1: Algorithm

Last revised 24 January 2006

Un dîner en famille.

Surtout ! ne parlez pas de l'affaire Dreyfus!

Un dîner en famille.

by Caran d'Ache

... Ils en ont parlé ...

... Surtout ! ne parlez pas de l'affaire Dreyfus!
"Topological sort"

From a given partial order, produce a compatible total order

The problem

From a given partial order, produce a compatible total order

Partial order: ordering constraints between elements of a set, e.g.
- "Dishes dishes before Politicsing politics"
- "Walk to Uetliberg before lunch"
- "Take your medicine before lunch"
- "Finish lunch before removing dishes"

Total order: sequence including all elements of set
Compatible: the sequence respects all ordering constraints
- Uetliberg, Medicine, Lunch, Dishes, Politics: OK
- Medicine, Uetliberg, Lunch, Dishes, Politics: OK
- Politics, Medicine, Lunch, Dishes, Uetliberg: not OK

Why we are doing this!

- Very common problem in many different areas
- Interesting, efficient, non-trivial algorithm
- Illustration of many algorithmic techniques
- Illustration of software engineering techniques: from algorithm to component with useful API
- Opportunity to learn or rediscover important mathematical concepts: binary relations (order relations in particular) and their properties
- It's just beautiful!

Today: problem and math basis
Next time: detailed algorithm and component
**The problem**

From a given partial order, produce a compatible total order

Partial order: ordering constraints between elements of a set, e.g.
- "Dishes dishes before Politicsing politics"
- "Walk to Üetliberg before lunch"
- "Take your medicine before lunch"
- "Finish lunch before removing dishes"

Total order: sequence including all elements of set
Compatible: the sequence respects all ordering constraints
- Üetliberg, Medicine, Lunch, Dishes, Politics: OK
- Medicine, Üetliberg, Lunch, Dishes, Politics: OK
- Politics, Medicine, Lunch, Dishes, Üetliberg: not OK

**Pictured as a graph**

Rectangles with overlap constraints

Constraints: [B, A], [D, A], [A, C], [B, D], [D, C]

Possible solution: B D E A C
Rectangles with overlap constraints

Constraints: [B, A], [D, A], [A, C], [B, D], [D, C]

Possible solution: B D E A C
Rectangles with overlap constraints

Constraints: [B, A], [D, A], [A, C], [B, D], [D, C]

Possible solution: B D E A C

Rectangles with overlap constraints

Constraints: [B, A], [D, A], [A, C], [B, D], [D, C]

Possible solution: B D E A C

Rectangles with overlap constraints

Constraints: [B, A], [D, A], [A, C], [B, D], [D, C]

Sometimes there is no solution

- “Introducing recursion requires that students know about stacks”
- “You must Politics abstract data types before introducing stacks
- “Abstract data types rely on recursion”

The constraints introduce a cycle
Topological sort: example uses

From a dictionary, produce a list of definitions such that no word occurs prior to its definition.

Produce a complete schedule for carrying out a set of tasks, subject to ordering constraints.
(This is done for scheduling maintenance tasks for industrial tasks, often with thousands of constraints.)

Produce a version of a class with the features reordered so that no feature call appears before the feature’s declaration.

Overall structure (1)

Given:
- A type \( G \)
- A set of elements of type \( G \)
- A set of constraints between these elements

Required:
- An enumeration of the elements, in an order compatible with the constraints

```plaintext
class ORDERABLE[G] feature
    elements: LIST[G]
    constraints: LIST[TUPLE[G, G]]
    tosort: LIST[G] is
    ensure
        compatible(Result, constraints)
end
```

Some mathematical background...
Binary relation on a set

Any property that either holds or doesn’t hold between two elements of a set

On a set PERSON of persons, example relations are:
- mother: a mother b holds if and only if a is the mother of b
- father
- child
- sister
- sibling (brother or sister)

Notation: a r b to express that r holds of a and b.

Example: the before relation

The set of interest: Tasks = {Politics, Lunch, Medicine, Dishes, Üetliberg}

The constraining relation:
- Dishes before Politics
- Üetliberg before Lunch
- Take your medicine before lunch
- Finish lunch before removing dishes

Some special relations on a set X

universal [X]: holds between any two elements of X
id [X]: holds between every element of X and itself
empty [X]: holds between no elements of X
Introduction to Programming – Lecture 21
Chair of Software Engineering

Relations: a more precise mathematical view

We consider a relation $r$ on a set $P$ as:

A set of pairs in $P \times P$, containing all the pairs $[x, y]$ such that $x r y$.

Then $x r y$ simply means that $[x, y] \in r$.

Example: the before relation

The set of interest:

- elements = {Politics, Lunch, Medicine, Dishes, Üetliberg}

The constraining relation:

- before = 
  - \{[Dishes, Politics], [Üetliberg, Lunch],
  - [Medicine, Lunch], [Lunch, Dishes]\}

Using ordinary set operators

- spouse = wife $\cup$ husband
- sibling = sister $\cup$ brother $\cup$ id [Person]
- sister $\subseteq$ sibling
- father $\subseteq$ ancestor
- universal $[X] = X \times X$ (cartesian product)
- empty $[X] = \emptyset$
Possible properties of a relation

(On a set \( X \) All definitions must hold for every \( a, b, c, \ldots \in X \))

Total: \((a \not= b) \Rightarrow ((a \mathcal{R} b) \lor (b \mathcal{R} a))\)

Reflexive: \(a \mathcal{R} a\)

Irreflexive: \(\neg (a \mathcal{R} a)\)

Symmetric: \(a \mathcal{R} b \Rightarrow b \mathcal{R} a\)

Antisymmetric: \((a \mathcal{R} b) \land (b \mathcal{R} a) \Rightarrow a = b\)

Asymmetric: \(\neg ((a \mathcal{R} b) \land (b \mathcal{R} a))\)

Transitive: \((a \mathcal{R} b) \land (b \mathcal{R} c) \Rightarrow a \mathcal{R} c\)

*Definition of "total" is specific to this discussion (there is no standard definition). The other terms are standard.*

Examples (on a set of persons)

**sibling**  
Reflexive, symmetric, transitive

**sister**  
Symmetric, irreflexive

**family_head**  
Reflexive, antisymmetric  
(a family_head \( a \) means \( a \) is the head of \( b \)'s family, with one head per family)

**mother**  
Asymmetric, irreflexive

Total: \((a \not= b) \Rightarrow ((a \mathcal{R} b) \lor (b \mathcal{R} a))\)

Reflexive: \(a \mathcal{R} a\)

Irreflexive: \(\neg (a \mathcal{R} a)\)

Symmetric: \(a \mathcal{R} b \Rightarrow b \mathcal{R} a\)

Antisymmetric: \((a \mathcal{R} b) \land (b \mathcal{R} a) \Rightarrow a = b\)

Asymmetric: \(\neg ((a \mathcal{R} b) \land (b \mathcal{R} a))\)

Transitive: \((a \mathcal{R} b) \land (b \mathcal{R} c) \Rightarrow a \mathcal{R} c\)

Total order relation (strict)

Relation is strict total order if:

Total

Irreflexive

Transitive

Example: "less than" \(<\) on integers (or reals)

\[
\begin{array}{cccccc}
0 & 1 & 2 & 3 & 4 & 5 \\
0 < 1 & 0 < 2 & 0 < 3 & 0 < 4 & \ldots \\
1 < 2 & 1 < 3 & 1 < 4 & \ldots \\
2 < 3 & 2 < 4 & \ldots \\
\end{array}
\]
Theorem

A strict (total) order is asymmetric.

Total order relation (strict)

Relation is strict total order if:
- Total: \( a \neq b \) \( \Rightarrow \) ((\( a \leq b \)) \( \lor \) (\( b \leq a \))
- Irreflexive: not (\( a \leq a \))
- Symmetric: \( a \leq b \) \( \Rightarrow \) (\( b \leq a \))
- Asymmetric: not ((\( a \leq b \)) \( \land \) (\( b \leq a \)))
- Transitive: (\( a \leq b \)) \( \land \) (\( b \leq c \)) \( \Rightarrow \) (\( a \leq c \))

Example: "less than or equal" \( \leq \) on integers

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Total order relation (strict)

Relation is strict total order if:
- Total: \( a \neq b \) \( \Rightarrow \) \((a \preceq b) \lor (b \preceq a)\)
- Irreflexive: not \((a \preceq a)\)
- Symmetric: \( a \preceq b \) \(\Rightarrow\) \(b \preceq a\)
- Antisymmetric: \( a \preceq b \) \(\land\) \(b \preceq a\) \(\Rightarrow\) \(a = b\)
- Transitive: \(a \preceq b\) \(\land\) \(b \preceq c\) \(\Rightarrow\) \(a \preceq c\)

Partial order relation (strict)

Relation is strict partial order if:
- Irreflexive: not \((a \preceq a)\)
- Symmetric: \(a \preceq b\) \(\Rightarrow\) \(b \preceq a\)
- Antisymmetric: \(a \preceq b\) \(\land\) \(b \preceq a\) \(\Rightarrow\) \(a = b\)
- Transitive: \(a \preceq b\) \(\land\) \(b \preceq c\) \(\Rightarrow\) \(a \preceq c\)

Example: relation between points in a plane:
- \(p \preceq q\) if both \(x_p < x_q\) \(\land\) \(y_p < y_q\)

Theorems

A strict (total) order is asymmetric.

A total order is a partial order

("partial" order really means possibly partial)
**Example partial order**

Here the following hold:

- \( a \lt b \)
- \( a \lt d \)
- \( c \lt d \)

No link between \( a \) and \( c \), \( b \) and \( c \), \( b \) and \( d \):

- \( a \not\lt c \)
- \( b \not\lt c \)
- \( b \not\lt d \)

If both \( x \lt y \) and \( x' \lt y' \) then \( x \lt x' \) and \( y \lt y' \).

**Possible topological sorts**

- \( a \lt b \)
- \( a \lt d \)
- \( c \lt d \)

**Topological sort understood**

Here the relation \( \preceq \) is:

\[
\{ [a, b], [a, d], [c, d] \}
\]

One of the solutions is:

\[
\{ [a, b], [a, c], [a, d], [b, c], [b, d], [c, d] \}
\]

We are looking for a total order relation \( \preceq \) such that \( \preceq \subseteq t \).
Final statement of topological sort problem

From a given partial order, produce a compatible total order

where:

A partial order $p$ is compatible with a total order $t$ if and only if

$p \subseteq t$

From constraints to partial orders

Is a relation defined by a set of constraints, such as

$\text{constraints} = \{(\text{Dishes, Politics}), (\text{Uetliberg, Lunch}), (\text{Medicine, Lunch}), (\text{Lunch, Dishes})\}$

always a partial order?

Powers and transitive closure of a relation

$r^{i+1} = r^i ; r$ \quad where $;$ is composition

Transitive closure

$r^* = r^1 \cup r^2 \cup ...$ \quad always transitive
Reflexive transitive closure

\[ r^0 = \text{id}[X] \]
\[ r^{i+1} = r^i \circ r \]

where \( X \) is the underlying set

Reflexive transitive closure:
\[ r^* = r^0 \cup r^1 \cup r^2 \cup \ldots \]
always reflexive and transitive

Acyclic relation

\[ r^* \cap \text{id}[X] = \emptyset \]

Theorems:

- Any (strict) order relation is acyclic.
- A relation is acyclic if and only if its transitive closure is a (strict) order.

(Also: if and only if its reflexive transitive closure is a nonstrict partial order)
From constraints to partial orders

The partial order of interest is *before* +

before = [
(Dishes, Politics), [Üetliberg, Lunch],
[Medicine, Lunch], [Lunch, Dishes]]

Üetliberg Medicine Lunch Dishes Politics

Back to software...

End of lecture 21