Lecture 24:
Topological Sort — 1: Background

“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.
  - "Remove dishes before discussing politics"
  - "Walk to Ütliberg 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
  - Ütliberg, Medicine, Lunch, Remove, Discuss : OK
  - Medicine, Ütliberg, Lunch, Remove, Discuss : OK
  - Discuss, Medicine, Lunch, Remove, Ütliberg : not OK
Pictured as a graph

- "Remove dishes before discussing politics"
- "Walk to Ütliberg before lunch"
- "Take your medicine before lunch"
- "Finish lunch before removing dishes"

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

Sometimes there is no solution

- "Introducing recursion requires that students know about stacks"
- "You must discuss 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]]
    topsort: LIST [G] is
        ensure compatible (Result, constraints)
end
```

Some mathematical background...

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

```
• "Remove dishes before discussing politics"
• "Walk to Ütliberg before lunch"
• "Take your medicine before lunch"
• "Finish lunch before removing dishes"
```

The set of interest:

Tasks = {Discuss, Lunch, Medicine, Remove, Ütliberg}

The constraining relation:

Dishes *before* Politics
Ütliberg *before* Lunch
Medicine *before* Lunch
Lunch *before* Dishes

---

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 \( X \)

---

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:
**Example: the before relation**

- "Remove dishes before discussing politics"
- "Walk to Ütliberg before lunch"
- "Take your medicine before lunch"
- "Finish lunch before removing dishes"

The set of interest:

\[ \text{elements} = \{ \text{Discuss, Lunch, Medicine, Remove, Ütliberg} \} \]

The constraining relation:

\[ \text{constraints} = \{ [\text{Dishes, Politics}], [\text{Ütliberg, Lunch}], [\text{Medicine, Lunch}], [\text{Lunch, Dishes}] \} \]

**Using ordinary set operators**

- \( \text{spouse} = \text{wife} \cup \text{husband} \)
- \( \text{sibling} = \text{sister} \cup \text{brother} \cup \text{id} [\text{Person}] \)
- \( \text{sibling} \subseteq \text{sibling} \)
- \( \text{universal} [X] = X \times X \) (cartesian product)
- \( \text{empty} [X] = \emptyset \)

**Possible properties of a relation**

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

- Total*: \( (a \neq b) \Rightarrow ((a \not\in b) \lor (b \not\in a)) \)
- Reflexive: \( a \in a \)
- Irreflexive: not \( (a \in a) \)
- Symmetric: \( a \in b \Rightarrow b \in a \)
- Antisymmetric: \( (a \in b) \land (b \in a) \Rightarrow a = b \)
- Asymmetric: not \( ((a \in b) \land (b \in a)) \)
- Transitive: \( (a \in b) \land (b \in c) \Rightarrow a \in c \)

*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 b means a is the head of b’s family, with one head per family)
mother    Asymmetric, irreflexive

Total order relation (strict)

Relation is strict total order if:
• Total
• Irreflexive
• Transitive

Example: “less than” < on integers (or reals)

Theorem

A strict (total) order is asymmetric
### Total order relation (strict)

Relation is strict total order if:
- **Total**
- **Irreflexive**
- **Transitive**

\[
\begin{align*}
\text{Total: } & (a \neq b) \Rightarrow ((a \leq b) \vee (b \leq a)) \\
\text{Irreflexive: } & \neg (a \leq a) \\
\text{Symmetric: } & \neg (a \leq b) \Rightarrow (b \leq a) \\
\text{Asymmetric: } & \neg ((a \leq b) \land (b \leq a)) \\
\text{Transitive: } & (a \leq b) \land (b \leq c) \Rightarrow (a \leq c)
\end{align*}
\]

### Total order relation (non-strict)

Relation is non-strict total order if:
- **Total**
- **Reflexive**
- **Transitive**
- **Antisymmetric**

\[
\begin{align*}
\text{Total: } & (a \neq b) \Rightarrow ((a \leq b) \vee (b \leq a)) \\
\text{Irreflexive: } & \neg (a \leq a) \\
\text{Symmetric: } & a \leq b \Rightarrow b \leq a \\
\text{Antisymmetric: } & (a \leq b) \land (b \leq a) \Rightarrow a = b \\
\text{Transitive: } & (a \leq b) \land (b \leq c) \Rightarrow a \leq c
\end{align*}
\]

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

<table>
<thead>
<tr>
<th>Integer</th>
<th>( \leq 0 )</th>
<th>( \leq 1 )</th>
<th>( \leq 2 )</th>
<th>( \leq 3 )</th>
<th>( \leq 4 )</th>
<th>( \leq 5 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
</tbody>
</table>

0 ≤ 0, 1 ≤ 1, 2 ≤ 2, 3 ≤ 3, 4 ≤ 4, 5 ≤ 5

\( \leq 0 \), \( \leq 1 \), \( \leq 2 \), \( \leq 3 \), \( \leq 4 \), 0 ≤ 0, 1 ≤ 1, 2 ≤ 2, 3 ≤ 3, 4 ≤ 4, ...
Partial order relation (strict)

Relation is strict partial order if:

- Irreflexive
- Transitive

\[ \text{Irreflexive: } \neg(a \preceq a) \]

\[ \text{Symmetric: } a \preceq b \Rightarrow b \preceq a \]

\[ \text{Antisymmetric: } (a \preceq b) \land (b \preceq a) \Rightarrow a = b \]

\[ \text{Transitive: } (a \preceq b) \land (b \preceq c) \Rightarrow a \preceq c \]

Example: relation between points in a plane:

\[ p \preceq q \text{ if both } \begin{align*} x_p < x_q \quad & \text{or} \quad x_p = x_q \text{ and } y_p < y_q \end{align*} \]

Theorems

A strict (total) order is asymmetric

A total order is a partial order

("partial" order really means possibly partial)

Example partial order

\[ p \preceq q \text{ if both } \begin{align*} x_p < x_q \quad & \text{or} \quad x_p = x_q \text{ and } y_p < y_q \end{align*} \]

Here the following hold:

\[ a \preceq b \quad c \preceq d \quad a \preceq d \]

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

\[ a \not\preceq c \quad \text{nor} \quad b \not\preceq c \quad \text{nor} \quad b \not\preceq d \]
Possible topological sorts

\[ a \preceq b \quad c \preceq d \quad a \preceq d \]

Topological sort understood

Here the relation \( \preceq \) is:
\[ \{(a, b), (a, d), (c, d)\} \]

One of the solutions is:
\[ \{(a, b), (a, c), (a, d), (c, d)\} \]

We are looking for a total order relation \( t \) such that \( \preceq \subset t \)

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

\[
\text{constraints} = \{(\text{Dishes, Politics}), (\text{Ütliberg, Lunch}), \\
(\text{Medicine, Lunch}), (\text{Lunch, Dishes})\}
\]

always a partial order?

Powers and transitive closure of a relation

- \( r^{i+1} = r^i \circ r \) where \( \circ \) is composition

Transitive closure
- \( r^* = r_1 \cup r_2 \cup \ldots \) always transitive

Reflexive transitive closure

- \( r^0 = \text{id} [X] \) where \( X \) is the underlying set
- \( r^{i+1} = r^i \circ r \)

Reflexive transitive closure
- \( r^* = r_1 \cup r_2 \cup \ldots \) always reflexive and transitive
Acyclic relation

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

\begin{align*}
M & \rightarrow L \rightarrow D \rightarrow P \\
D & \rightarrow M \rightarrow L \rightarrow D \\
L & \rightarrow D \rightarrow M \rightarrow L \\
D & \rightarrow L \rightarrow D \rightarrow M \\
\end{align*}

before:

id [X]

Acyclic relations and partial orders

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

\[ \text{before} = \{[\text{Dishes, Politics}], [\text{Utliberg, Lunch}], [\text{Medicine, Lunch}], [\text{Lunch, Dishes}]\} \]
Back to software...

End of lecture 24