Top-Down Application Design

Process Control Example — Middleware design

↑

Support for a large class of apps.

Today: Start on top-down design

Specification refinement ⇒ Algorithms
+ supporting data structures ⇒ Impl.
  (class hierarchy, design, component interaction)
Stable Marriage Problem:

**Input:**
- \( n \) men and \( n \) women
- each man ranks orders all women \( \forall \text{ total orderings} \)
- each woman ranks orders all men \( \forall \text{ orderings} \)

**Output:** Collection of engagements that is **stable**.

Stable — No two people will leave their betrothed and run off together.

More precisely, \( \forall \text{ a man } M \text{ and woman } W, \text{ s.t. } \)
- \( M \) ranks \( W \) higher than his fiancee \( F \)
- \( W \) ranks \( M \) higher than her fiancee \( F' \)
Algorithm ideas:

1. Is best rank

- See if anyone has an exact match for 1st choice — then continue making matches as "close as possible" to mutual 1st choice

- Add rankings — in considering a pair, add the rank of opposite person + minimize over all pairs.

What about (3,4) + (4,3)? How do we know this will stable?

- Pair arbitrarily & then shuffle the pairs to improve "stability"
• Start w/ an arbitrary man M:

  look at his first choice W.

  let W consider all men above M in her rank,

  the first one that ranks W at least as high as W ranks him is chosen

  ? what if chosen man prefers someone else?

• Start w/ most "popular" — let them pick, t

  continue w/ next popular, etc.

• choose one gender to take precedence & use

  the other as a tie breaker.

• start with 2 people — give them first choice—

  check for stability & keep expanding

  → REARRANGE
Start w/ an arbitrary man M:

let M make a proposal to his top choice W

if W accepts (if M's rank is better than her current situation)
engage M & W

(if W was already engaged, she dumps her former fiancee)

Will this terminate?
If so, will the result be stable?
Maybe some people will be left out?
let eligible be a list of eligible men (not engaged) (initially, all men)

while (there are still eligible men)
let M be first man in eligible
M makes a proposal to 1st person on his list
if (M is in W's rankings) //she accepts
   W removes M + everyone "worse" from her rankings
   put W's old fiance back into eligible list
M removes W from his list

Start w/ an arbitrary man M:
let M make a proposal to his top choice W
if W accepts (if M's rank is better than her current situation)
   engage M + W
(if W was already engaged, she dumps her former fiance'.)
Correctness = safety + liveness

if it does something, eventually it does something
it will do the right thing

In our case:

liveness: alg. terminates

safety: output is a stable arrangement
including all the people

<table>
<thead>
<tr>
<th>ABST</th>
<th>YES</th>
<th>NO</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>20</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>16</td>
<td>9</td>
</tr>
</tbody>
</table>
Variant function — maps the state to some well-founded set

- \# in eligible list
- \# in decreasing
- length of ranking lists

At each step of the alg,

- Some man in eligible list makes a proposal to a woman \( W \) in his ranking list and then removes \( W \) from that list.
- When the variant function is 0, the alg must terminate!
- Every man must have proposed to every woman once a woman gets at least one proposal, she remains engaged
What about safety? Is the arrangement stable?

**Proof by contradiction:**

Suppose the output is not stable.

\[ \therefore \text{There exists } M \text{ and } W \text{ that prefer each other to their own fiancés.}\]

Since M prefers W over his own fiancé, M must have proposed to W.

Two cases for that proposal:

1. **W accepted M:** She must have dumped him later \( \Rightarrow \) she improved her position.

   So her current fiancé must be “better” than M.

2. **W rejected M:** She already had someone better.

   By same argument (women only improve the position)

   so she couldn’t have been engaged to M.

In both cases, have a contra. \( \Rightarrow \text{Supposition is false.} \)
Who is favored?

<table>
<thead>
<tr>
<th></th>
<th>Men</th>
<th>Women</th>
<th>Even</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>6</td>
<td>36</td>
<td>6</td>
</tr>
</tbody>
</table>

If all men have different 1st choices, they get them!