Note: Other solutions to problems 1-3 are possible.

1. Create role frames for the following sentences. (With these examples, represent a trajectory with a phrase, e.g. “to the theatre”, rather than with a trajectory frame.)
   a. John fried some eggs for Mary in a frying pan.
   b. Robbie went to the theatre with Suzie.
   c. The plants were grown with fertilizer by Pat.

   ![Frame System Solutions](attachment:images/frame_system_solutions.png)

2. Create trajectory frames for sentences b and c above. Why is it difficult to use a trajectory frame to represent sentence a? (Use the definition of trajectory frame described in class and in the recitation notes. You might also work out a trajectory frame representation that uses the more structured frame system used in Problem Set 3. It uses the terms Go, Path, etc.)

   ![Trajectory Frames](attachment:images/traj_frames.png)

   Note: There’s no good way to represent Suzie in b. The best representation would be a role frame with a trajectory frame as the value of the trajectory slot. Also, the result frame could be a transition frame. What might it look like?

   It’s difficult to represent sentence a using a trajectory frame because the sentence is describing the roles being played, rather than a trajectory. There’s no good way to represent “for Mary” in a trajectory frame.
3. Alyssa P. Hacker was working on the description of a car crashing into a wall using a transition frame representation. After some thought, she settled on a description using two variables: car speed, and distance between car and wall. Unfortunately, she had to leave before filling out the frame information. Please finish her work.

After you finish with Alyssa’s work, Ben Bitdiddle walks in and notices, “Wait! I can describe a kiss the same way – as lips crashing into another person.” Revise Alyssa’s original description to differentiate the two (very obviously) different situations and show Ben a counter example to his claim.

### Car crashing into wall

<table>
<thead>
<tr>
<th></th>
<th>Speed</th>
<th>Distance</th>
<th>Integrity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speed</td>
<td>Not Δ</td>
<td>D</td>
<td>Not A</td>
</tr>
<tr>
<td>Distance</td>
<td>D</td>
<td>Not A</td>
<td></td>
</tr>
</tbody>
</table>

### Car crashing into wall (revised)

<table>
<thead>
<tr>
<th></th>
<th>Speed</th>
<th>Distance</th>
<th>Integrity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speed</td>
<td>Not Δ</td>
<td>D</td>
<td>Not A</td>
</tr>
<tr>
<td>Distance</td>
<td>D</td>
<td>Not A</td>
<td></td>
</tr>
<tr>
<td>Integrity</td>
<td>Not Δ</td>
<td>Not Δ</td>
<td>Δ</td>
</tr>
</tbody>
</table>

### Kissing

<table>
<thead>
<tr>
<th></th>
<th>Speed</th>
<th>Distance</th>
<th>Integrity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Speed</td>
<td>↓</td>
<td>D</td>
<td>A</td>
</tr>
<tr>
<td>Distance</td>
<td>↓</td>
<td>A</td>
<td></td>
</tr>
<tr>
<td>Integrity</td>
<td>Not Δ</td>
<td>Not Δ</td>
<td>Not Δ</td>
</tr>
</tbody>
</table>

Note: In order to support inference about cause and effect, an effect must be explicitly represented (e.g. integrity), and events must not occur in the same time slot. If distance to wall disappeared in one time slot, and speed disappeared in the next, for example, a program could infer that the speed disappeared because the car crashed into the wall.

4. Use topological sorting to provide a linearization of the inheritance hierarchies shown below. If there is no such linearization, explain why. Use alphabetical order to resolve ambiguities.

#### a.

```
A B C X L X Y M G Y F E
```

Order: (A B C X L M G Y F E)

#### b.

```
A B C X L X Y M G Y F E
```

Order: (A B C X L Y F M G E)

#### c.

```
A B C X L X Y M G Y F E
```

Order: (A B C X L M G Y F E)

#### d.

```
A B C X L X Y M G Y F E
```

Order: (A B C X L Y F M G E)