Expanding a State

For each unvisited state we have to ...
- check controllability condition;
- compute successor states.

Assumptions:
- source state tuple in array \( q[1], \ldots, q[N] \)
- target state to be put in \( q'[1], \ldots, q'[N] \)

Controllability Check Algorithm

To check whether plants \( A_1, \ldots, A_p \) are controllable with respect to specifications \( A_{p+1}, \ldots, A_N \):

1. Add initial state \( q_0 = (q_{0,1}, \ldots, q_{0,N}) \) to state set \( Q \)
2. While there are unvisited states \( q \in Q \) do
   - For each event \( e \) enabled by all plants in state \( q \) do
     - If \( e \) is uncontrollable and there exists a specification that cannot execute \( e \) in state \( q \) then
       - return "The system is not controllable."
     - If \( e \) can be executed by all specifications then
       - Compute successor state \( r \) such that \( q \rightarrow r \)
6. Add \( r \) to state set \( Q \) if not yet present

Listing Automata

<table>
<thead>
<tr>
<th>Automata:</th>
<th>( A_1 )</th>
<th>( \ldots )</th>
<th>( A_p )</th>
<th>( A_{p+1} )</th>
<th>( \ldots )</th>
<th>( A_N )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Plants</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Specifications</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>State Tuples:</td>
<td>( q_1 )</td>
<td>( \ldots )</td>
<td>( q_p )</td>
<td>( q_{p+1} )</td>
<td>( \ldots )</td>
<td>( q_N )</td>
</tr>
</tbody>
</table>

Data Structure for State Set \( Q \)

Requirements:
- new items to be added
- need to find whether items are contained
- need to get next unvisited state
- can get very large!

Available in Java:
- java.util.LinkedList
- java.util.ArrayList
- java.util.HashMap

Expanding a State continued

```java
for each event \( e \)
  for \( i := 1 \) to \( N \)
    if \( e \) is not in the Alphabet of \( A_i \), then
      \( q'[i] := q[i] \)
    else if \( q[i] \in A_j \) and \( j > p \), then
      \( q'[i] := q' \)
    else \( e \) is uncontrollable and \( i > p \), then
      return "The system is not controllable."
  end if
end for

store successor state tuple \( (q'[1], \ldots, q'[N]) \)
end for
```