Modelling Reactive Systems

- **Plant Model**
  - Model of the system to be controlled
  - Possible behaviour

- **Specification**
  - Model of the control program
  - Desired behaviour

Controlling Reactive Systems

- **Commands**
- **Responses**

Supervisory Control

- **Uncontrollable Events**: Cannot be prevented by supervisor
- **Controllable Events**: Can be disabled by supervisor

Controlling Small Factory

- **Commands**
  - start1, ...
- **Responses**
  - finish1, ...

Desired Behaviour

- We want small factory together with its controller to behave exactly like the Buffer automaton.

  - *Is this possible?*
**Controllability**

**Definition**

Let $P$ and $S$ be two automata. $S$ is called **controllable** with respect to $P$ if, for every state $(q_P, q_S)$ reachable in $P \parallel S$, every uncontrollable event which is enabled in $q_P$ also is enabled in $q_S$.

---

**Small Factory is Not Controllable**

To check whether plants $P_1, \ldots, P_n$ are controllable with respect to specifications $S_1, \ldots, S_m$:

1. Add initial state $q_0 = (q_{01}, \ldots, q_{0n}, q_{01}, \ldots, q_{0m})$ to state set $Q$.
2. While there are unvisited states $q \in Q$ do:
   1. For each event $e$ enabled by all plants $P_i$ in state $q$ do:
      1. If $e$ is uncontrollable and there exists a specification $S_j$ that cannot execute $e$ in state $q$ then return "The system is not controllable."
      2. If $e$ can be executed by all specifications then compute successor state $r$ such that $q \xrightarrow{e} r$.
      3. Add $r$ to state set $Q$ if not yet present.

---

**Making Small Factory Controllable**

Observations:

- We cannot disable the uncontrollable event $\text{finish1}$.
- If the system ever enters state $\text{WIF}$, we have a problem.
- We can avoid entering this state by disabling the controllable event $\text{start1}$ in state $\text{IF}$.

---

**Controllability Check Algorithm**

**Checking for Controllability**

**Reading**

**Texts on Supervisory Control**