Case Study: Mother and Sweets

Think of Mother as a simple resource controller.

- She has a bag of sweets.
- Each child wants a sweet, and can ask for one.
- Mother will only give out one sweet at a time.

**Mother:**

\[
\begin{align*}
\text{asked(jill)} & \Rightarrow \Diamond \text{give(jill)} \\
\Box \text{asked(jack)} & \Rightarrow \Diamond \text{give(jack)} \\
\Box \text{true} & \Rightarrow (\neg \text{give(jill)} \lor \neg \text{give(jack)}) \\
\end{align*}
\]

Intuitive meanings of these formulae:

1. if a child asked for a sweet then it will eventually receive one;
2. only one child can receive a sweet at any moment.

Now, to implement communication, we define the formula

\[
\text{Comms(mother, jill)} = \Box (\text{req(jill)} \Rightarrow \Diamond \text{asked(jill)}) \\
\Box (\text{give(jill)} \Rightarrow \Diamond \text{got(jill)})
\]

‘Jack’

Jack will have similar behaviour to Jill, but will make a request every other moment in time.

**Jack:**

\[
\begin{align*}
\text{start} & \Rightarrow \text{req(jack)} \\
\Box \text{req(jack)} & \Rightarrow \Box \text{wait} \\
\Box \text{wait} & \Rightarrow \Box \text{req(jack)} \\
\end{align*}
\]

Again, to implement communication, we define the formula

\[
\text{Comms(mother, jack)} = \Box (\text{req(jack)} \Rightarrow \Diamond \text{asked(jack)}) \\
\Box (\text{give(jack)} \Rightarrow \Diamond \text{got(jack)})
\]
Properties of the System

Let us consider the specifications of Mother, Jack and Jill, together with the communications formulae.

1. Jill will request infinitely often — $\square \neg req(jill)$.

2. Jack asked infinitely often — $\square \Diamond asked(jack)$.

3. Mother has a *liveness* constraint saying that, if a child has asked, some allocation will be given.

4. Mother has a *safety* constraint saying that allocations to *jack* and *jill* must never happen at the same time.

Can prove many properties of system, e.g. $\square \Diamond got(jack)$

Extension: Jealousy

Imagine we now change the behaviour of Jack so that this element only asks for a sweet when it sees Jill being given one.

\[
\text{JACK: } \square \left[ \neg jealous \Rightarrow \Diamond req(jack) \right]
\]

Now, to implement this, we must have a form of broadcast communication.

Extension: Broadcast

So, $\text{give}(jill)$ in Mother’s specification is linked to more than one other proposition, in more than one other element, for example

\[
\text{Comms(}mother, [jill, jack] \text{)} = \square (\neg req(jill) \Rightarrow \Diamond asked(jill)) \\
\neg (\neg give(jill) \Rightarrow \Diamond got(jill)) \\
\neg (\neg give(jill) \Rightarrow \Diamond jealous)
\]
Many Varieties of Communication

We have seen different communication models:

\[
\begin{align*}
\text{send} & \Rightarrow \text{receive} \\
\text{send} & \Rightarrow \bigcirc \text{receive} \\
\ldots & \ldots \ldots \\
\text{send} & \Rightarrow \Diamond \text{receive}
\end{align*}
\]

as well as multicast/broadcast:

\[
\begin{align*}
\text{send} & \Rightarrow \Diamond \text{receive}_1 \\
\text{send} & \Rightarrow \Diamond \text{receive}_2 \\
\text{send} & \Rightarrow \Diamond \text{receive}_3
\end{align*}
\]

But, notice the difference between the above and

\[
\text{send} \Rightarrow \Diamond (\text{receive}_1 \land \text{receive}_2 \land \text{receive}_3)
\]