Calculus of Communicating Systems

( CCS )

Ali Ahmadzadeh asl – 92204914
Verification of Reactive Systems
March 2014
Introduction

- **CCS**: Calculus of Communicating Systems
- Introduced by Robin Milner, 1980.
- Modeling of communication and concurrency.
- Description of process networks with static topologies.
• CCS process as a black box
Language :: Action Prefixing

• The behavior of a process is described by giving a ‘CCS program’

• The most basic process of all is the process 0 (read ‘nil’). This is the most boring process imaginable, as it performs no action whatsoever.

• The most basic process constructor in CCS is action prefixing: coin.tea.0

• TM \[\text{def} \] coin.tea.0

• TM \[\text{def} \] coin.tea.TM or Clock \[\text{def} \] tick.Clock
• To allow for the description of processes whose behavior may follow different patterns of interaction with their environment, CCS offers the choice operator, which is written ‘+’.

• A vending machine offering either tea or coffee may be described thus:

\[ VM \overset{\text{def}}{=} \text{coin.}(\text{tea}.VM + \text{coffee}.VM) \]
Language :: Operations :: Parallel composition

• To describe systems consisting of two or more processes running in parallel, and possibly interacting with each other, CCS offers the parallel composition operation, which is written ‘|’.

• In general, given two CCS expressions P and Q, the process P | Q describes a system in which:

  1- P and Q may proceed independently or
  2- may communicate via complementary ports
Example (1)

\[\text{STU} \overset{\text{def}}{=} \text{paper.coin.tea.STU}\]

\[\text{TM} \overset{\text{def}}{=} \text{coin.tea.TM}\]
Example (2)

\[ \text{paper} \quad \text{STU} \quad \text{tea} \quad \text{TM} \quad \text{coin} \]

\[ \text{STU} \quad \text{TM} \]
Example (3)

```
STU1 | STU2 | TM
```
The language CCS offers an operation called restriction, whose aim is to delimit the scope of channel names.

CCS offers the restriction operation, which is written ‘\’.

We can hide the coin and coffee ports from the environment of the processes TM and STU1:

\[ \text{Uni} \overset{\text{def}}{=} (\text{STU1} \mid \text{TM}) \backslash \text{coin} \backslash \text{tea} \]
Example (4)

```
uni | STU2
```

```
<table>
<thead>
<tr>
<th>paper</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>tea</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>paper</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>coin</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td></td>
</tr>
</tbody>
</table>
Language :: Operations :: Relabeling

• We may define the processes:

\[ \text{TM} \overset{\text{def}}{=} \text{coin.tea.TM} \]
\[ \text{WM} \overset{\text{def}}{=} \text{coin.water.WM} \]
\[ \text{CM} \overset{\text{def}}{=} \text{coin.caffee.CM} \]

• All of these machines follow a common behavioral pattern, and may be seen as specific instances of a generic vending machine:

\[ \text{VM} \overset{\text{def}}{=} \text{coin.item.VM} \]
• CCS offers the Relabeling (renaming) operation, which is written ‘[ x / y ]’ that x is a new label (name) for y.

• The aforementioned machines may be obtained as appropriate renaming of VM:

\[
\begin{align*}
\text{TM} & \equiv \text{VM}[\text{tea} / \text{item}] \\
\text{WM} & \equiv \text{VM}[\text{water} / \text{item}] \\
\text{CM} & \equiv \text{VM}[\text{caffee} / \text{item}] \\
\end{align*}
\]
LTS of CCS processes

\[
\text{STU} \overset{\text{def}}{=} \text{paper.coin.tea.STU}
\]

\[
\text{STU} \overset{\text{def}}{=} \text{paper.STU}_1 \\
\text{STU}_1 \overset{\text{def}}{=} \text{coin.STU}_2 \\
\text{STU}_2 \overset{\text{def}}{=} \text{tea.STU}
\]

\[
\text{TM} \overset{\text{def}}{=} \text{coin.tea.TM}
\]

\[
\text{TM} \overset{\text{def}}{=} \text{coin.TM}_1 \\
\text{TM}_1 \overset{\text{def}}{=} \text{tea.TM}
\]
LTS of CCS processes (cont.)

\[
\begin{align*}
\text{Uni} & \overset{\text{def}}{=} (\text{STU} \mid \text{TM}) \setminus \text{coin} \setminus \text{tea} \\
\text{STU} & \overset{\text{def}}{=} \text{paper.STU}_1 \\
\text{STU}_1 & \overset{\text{def}}{=} \text{coin.STU}_2 \\
\text{STU}_2 & \overset{\text{def}}{=} \text{tea.STU} \\
\text{TM} & \overset{\text{def}}{=} \text{coin.TM}_1 \\
\text{TM}_1 & \overset{\text{def}}{=} \text{tea.TM}
\end{align*}
\]
The Transition Rules

- **Act** \( \alpha.E \xrightarrow{\alpha} E \)
- **Sum\(_j\)** \[ \frac{E_j \xrightarrow{\alpha} E'_j}{\sum E_i \xrightarrow{\alpha} E'_j} \]
- **Com\(_1\)** \[ \frac{E \xrightarrow{\alpha} E'}{E|F \xrightarrow{\alpha} E'|F} \]
- **Com\(_2\)** \[ \frac{F \xrightarrow{\alpha} F'}{E|F \xrightarrow{\alpha} E|F'} \]
- **Com\(_3\)** \[ \frac{E \xrightarrow{\ell} E' \quad F \xrightarrow{\ell} F'}{E|F \xrightarrow{t} E'|F'} \]
- **Res** \[ \frac{E \xrightarrow{\alpha} E'}{E\setminus L \xrightarrow{\alpha} E'\setminus L} \] (\(\alpha, \bar{\alpha}\) not in \(L\))
- **Rel** \[ \frac{E \xrightarrow{\alpha} E'}{E[f] \xrightarrow{f(\alpha)} E'[f]} \]
Other capabilities

• Value passing:

\[ B \overset{\text{def}}{=} \text{in}(x).B(x) \]
\[ B(x) \overset{\text{def}}{=} \text{out}(x + 1).B \]

• Condition:

\[ \text{Pred} \overset{\text{def}}{=} \text{in}(x).\text{Pred}(x) \]
\[ \text{Pred}(x) \overset{\text{def}}{=} \text{if } x=0 \text{ then } \text{out}(0).\text{Pred} \text{ else } \text{out}(x-1).\text{Pred} \]

• Time:

\[ \text{in}(x)@t.P(t) : t \text{ is the delay before the message is available} \]
\[ \text{out}(x)@t.P(t) : t \text{ is the delay before the message is delivered} \]
Thank you for your attention