Foundations of Programming

– Concurrency –

Session 12 – April 25, 2002

Uwe Nestmann

EPFL-LAMP

Booleans

$$\mathsf{True}(\,l\,) \stackrel{\text{def}}{=} l(t,f).\bar{t}\langle\rangle$$

$$\mathsf{False}(\,l\,) \stackrel{\text{def}}{=} l(t,f).\boxed{}$$

$$\mathsf{If}(\,l, \mathsf{foo}, \mathsf{bar}\,) \stackrel{\text{def}}{=} (\boldsymbol{\nu} t f)\,\bar{l}\langle t,f\rangle.(\,\boxed{} + \boxed{}\,)$$

Check that for all P, Q:

$$(\boldsymbol{\nu} l)\,\big(\mathsf{True}\langle\,l\,\rangle \mid \mathsf{If}\langle\,l, \mathsf{foo}, \mathsf{bar}\,\rangle\big) \approx \boxed{}$$

$$(\boldsymbol{\nu} l)\,\big(\mathsf{False}\langle\,l\,\rangle \mid \mathsf{If}\langle\,l, \mathsf{foo}, \mathsf{bar}\,\rangle\big) \approx \boxed{}$$

Elastic Buffers: Setup

$$
\begin{aligned}
B(\,i,l,o,r\,) &\stackrel{\text{def}}{=} i().C\langle\, i,l,o,r\,\rangle \\
&\quad + \ \ldots \\
C(\,i,l,o,r\,) &\stackrel{\text{def}}{=} \overline{o}\langle\rangle.B\langle\, i,l,o,r\,\rangle \\
&\quad + \ i().(\ C\langle\, i,l,o,r\,\rangle \frown C\langle\, i,l,o,r\,\rangle\ ) \\
&\quad + \ \ldots
\end{aligned}
$$

where

$$
X\langle\, i,l,o,r\,\rangle \frown Y\langle\, i,l,o,r\,\rangle \stackrel{\text{def}}{=} \ldots
$$

Elastic Buffers: cut-when-left

$$
\begin{aligned}
B &\stackrel{\text{def}}{=} (i,l,o,r). \\
&\qquad i().C\langle\, i,l,o,r\,\rangle \\
&\quad + \ \ldots \\
C &\stackrel{\text{def}}{=} (i,l,o,r). \\
&\qquad \overline{o}\langle\rangle.B\langle\, i,l,o,r\,\rangle \\
&\quad + \ i().\big(\ (C \frown C)\langle\, i,l,o,r\,\rangle\ \big) \\
&\quad + \ \ldots
\end{aligned}
$$

Elastic Buffers: cut-when-right

$$
\begin{aligned}
B \stackrel{\text{def}}{=}\ & (i,l,o,r). \\
& \quad i().C\langle\, i,l,o,r\,\rangle \\
& + \ \ldots \\
C \stackrel{\text{def}}{=}\ & (i,l,o,r). \\
& \quad \overline{o}\langle\rangle.B\langle\, i,l,o,r\,\rangle \\
& + \ i().\big(\, (C \frown C)\langle\, i,l,o,r\,\rangle\ \big) \\
& + \ \ldots
\end{aligned}
$$

Elastic Buffers: Guideline

- cut-when left $\stackrel{?}{\approx}$ cut-when-right !
- pencil & paper (PP) vs MWB :
  check state spaces !

- omit growing part of the state-space: consequences for $\stackrel{?}{\approx}$
- PP vs MWB :
  check the non-growing state spaces

The Mobility Workbench

- tool for manipulating and analyzing mobile concurrent systems described in the pi-calculus
- powerful equivalence-checking (and model-checking)
- written (like the CWB) in SML/NJ
- not very convenient command-line interface
- outdated (written in '95, slightly updated later on)
- soon to be rewritten at EPFL . . .

`help`, `quit`, `input`, `env`, . . . , `t`, . . . , `size`, `weqd`, . . .