Circus Cookie Machine - typechecking

Leo Freitas

March 2008

section $circus\_cookies$**parents** $circus\_toolkit$

$$cookieValue, cookieQuantity, MAX\_QUANTITY : \mathbb{N}$$

$$COOKIE ::= ok \mid notok$$

channel $in, change : \mathbb{N}$
channel $out : COOKIE$

process $CookieMachine \mathrel{\widehat=}$ **begin**

$State$

$$money, quantity : \mathbb{N}$$

$$quantity \leq MAX\_QUANTITY$$

state $State$

$OutputCookieOk$

$$\begin{array}{l} \Delta State \\ o! : COOKIE \end{array}$$

$$\begin{array}{l} money \geq cookieValue \\ quantity > 0 \\ money' = money - cookieValue \\ quantity' = quantity - 1 \\ o! = ok \end{array}$$

$OutputCookieNotOk$

$$\begin{array}{l} \Delta State \\ o! : COOKIE \end{array}$$

One can also add comments within formal material - for Z

$$\begin{array}{l} money \geq cookieValue \\ quantity = 0 \\ money' = money \\ o! = notok \end{array}$$

It is not a total operation because there might not be enough money.

$$OutputCookie \;==\; OutputCookieOk \lor OutputCookieNotOk$$

Schema expressions as actions.

$$InitState \;\widehat{=}\; ([\, State' \mid money' = 0 \land quantity' = cookieQuantity \,])$$

Note this will generate type error for *InputMoney* because $x?$ is not into scope.

The next line is not being parsed...

$$InputMoney \;\widehat{=}\; ([\, \Delta State;\; x? : \mathbb{N} \mid money \leq cookieValue \land money' = money + x? \,])$$

$$InputMoney \;==\; [\, \Delta State;\; x? : \mathbb{N} \mid money \leq cookieValue \land money' = money + x? \,]$$

One can also add comments within formal material - or Circus

$$Input \;\widehat{=}\; \langle money \leq cookieValue \rangle \,\&\; in?x \longrightarrow (InputMoney)$$

The parser also admits some special commands that are tokenised as hard spaces, such as `\circblockbegin`, `\circblockend`, *etc.*

$$\begin{aligned} Output \;\widehat{=}\; & \langle money \geq cookieValue \rangle \,\& \\ & (\mathbf{var}\; o : COOKIE \bullet (OutputCookie) \,;\, (out!o \longrightarrow change!money \longrightarrow Skip)) \end{aligned}$$

$$\bullet\; InitState \,;\, (\mu\; X \bullet (Input \;\Box\; Output) \,;\, X)$$

end