1 Introduction

```
%-------------------------------------------------------------------------------
%-- A REACTIVE BUFFER    - Case Study                                           -
%-- Action Refinement: controller and ring partitions                          -
%-- Example extracted from the paper "A Refinement Strategy for Circus"        -
%-------------------------------------------------------------------------------
```

You must always include the circus_toolkit, and also check inside it to see the LaTeX commands the parser recognises (or to create your own commands)

section *buffer_refinement_singleenv* **parents** *circus_toolkit*

$maxbuff, maxring : \mathbb{N}$

channel $input, output : \mathbb{N}$

channel $write, read : (1 \,..\, maxring) \times \mathbb{N}$

channel $read\_1 : (1 \,..\, maxring)$

channel $read\_2 : \mathbb{N}$

Unfortunately boxed processes (those spread across multiple begin/end Circus) are not yet available. You need to define your processes within one environment only. The only real problem is for axiomatic definitions (that I am looking into now). Schemas can be given horizontally.

I know it doesn't typeset nicely. This boxed-process feature will be available soon. The bottom line is that apart from

process $Buffer \mathrel{\widehat=}$ **begin**
$ControllerState \;==\; [\, cache : \mathbb{N}$
$size : 0 \,..\, maxbuff$
$ringsize : 0 \,..\, maxring$
$top, bot : 1 \,..\, maxring \mid (ringsize \bmod maxring) = ((top - bot) \bmod maxring)$
$ringsize = size - 1\,]$
$RingState \;==\; [\, ring : \mathrm{seq}\,\mathbb{N} \mid \#\, ring = maxring\,]$
$CBufferState \;==\; (ControllerState \lor RingState)$
state $CBufferState$
$ControllerInit \;==\; [\, ControllerState'; \; RingState' \mid (size' = 0) \land (bot' = 1) \land (top' = 1)\,]$
$CacheInput \;==\; [\Delta ControllerState;$
$\Xi RingState$
$x? : \mathbb{N} \mid (size = 0) \land (size' = 1)$
$(cache' = x?) \land (bot' = bot) \land (top' = top)\,]$
$StoreInput \;==\; [\Delta CBufferState$
$x? : \mathbb{N} \mid (0 < size) \land (size < maxbuff)$
$(size' = size + 1) \land (cache' = cache)$
$(bot' = bot) \land (top' = (top \bmod maxring) + 1)$
$ring' = ring \oplus \{top \mapsto x?\}\,]$
$StoreInputController \;==\; [\Delta ControllerState$
$\Xi RingState \mid (0 < size) \land (size < maxbuff)$
$(size' = size + 1) \land (cache' = cache)$
$(bot' = bot) \land (top' = (top \bmod maxring) + 1)\,]$
$InputController \;\mathrel{\widehat=}$
$\quad (size < maxbuff)\;\&\; input\,?x \longrightarrow ((size = 0)\;\&\; CacheInput \,\Box\, (size > 0)\;\&\; write.top\,!x \longrightarrow StoreInpu$
$CInput \mathrel{\widehat=}$
$\quad (size < maxbuff)\;\&\; input\,?x \longrightarrow ((size = 0)\;\&\; CacheInput) \,\Box\, ((size > 0)\;\&\; StoreInput)$
$NoNewCache \;==\; [\Delta ControllerState$
$\Xi RingState \mid size = 1$
$size' = 0 \land cache' = cache$
$bot' = bot \land top' = top\,]$
$StoreNewCache \;==\; [\Delta CBufferState \mid size > 1$
$size' = size - 1 \land cache' = ring\; bot$
$bot' = (bot \bmod maxring) + 1 \land top' = top$
$ring' = ring\,]$
$StoreNewCacheController \;==\; [\Delta ControllerState$
$\Xi RingState$
$x? : \mathbb{N} \mid size > 1$
$size' = size - 1 \land cache' = x?$
$bot' = (bot \bmod maxring) + 1 \land top' = top\,]$
$OutputController \;\mathrel{\widehat=}\; (size > 0)\;\&\; output\,!cache \longrightarrow ((size > 1)\;\&\; read.bot\,?x \longrightarrow StoreNewCacheC$
$COutput \;\mathrel{\widehat=}\; (size > 0)\;\&\; output\,!cache \longrightarrow ((size > 1)\;\&\; StoreNewCache) \,\Box\, ((size = 1)\;\&\; NoNewC$
$ControllerAction \mathrel{\widehat=} ControllerInit \;;\; (\mu\, X \bullet ((InputController \,\Box\, OutputController) \;;\; X))$
$StoreRingCmd == [\Xi ControllerState$
$\Delta RingState$
$i? : 1 \,..\, maxring$
$x? : \mathbb{N} \mid ring' = ring \oplus \{i? \mapsto x?\}\,]$
$StoreRing \mathrel{\widehat=} write\,?i\,?x \longrightarrow StoreRingCmd$
$NewCacheRing \mathrel{\widehat=} read\,?i\,!(ring\; i) \longrightarrow \mathbf{Skip}$
$RingAction \mathrel{\widehat=} \mu\, X \bullet ((StoreRing \,\Box\, NewCacheRing) \;;\; X)$
$\quad \bullet\; (ControllerAction \;[\![\, \{size, ringsize, cache, top, bot\} \mid \{\!|\, write, read \,|\!\} \mid \{ring\} \,]\!]\; RingAction) \setminus \{\!|\, write$
end

Declarations	**This Section**	**Globally**
Unboxed items	57	57
Axiomatic definitions	1	1
Generic axiomatic defs.	0	0
Schemas	0	0
Generic schemas	0	0
Total	**58**	**58**

Table 1: Summary of *Circus* declarations for Section 1.