

Circus-L^AT_EX style explained Community Z Tools (CZT)

Leo Freitas
leo@cs.york.ac.uk

Department of Computer Science
University of York, YO10 5DD
York, United Kingdom

October 2008

Contents

A Environments	3
A.1 Informal argument	3
B Reference card	4
B.1 Special <i>Circus</i> symbols	4
Refinement	4
B.2 <i>Circus</i> channels and name sets	4
B.3 <i>Circus</i> actions	4
Action definition	4
Basic actions	4
Prefixing action	4
Prefixing action (extra)	5
Unary actions	5
Binary actions	5
Replicated actions	5
Parenthesised actions	6
B.4 <i>Circus</i> command definitions	6
Guarded commands	6
Parameterised commands	6
B.5 <i>Circus</i> processes	7
Process definition	7
Basic process	7
Unary processes	7
Binary processes	7

Parameterised and indexed processes	7
Parenthesised processes	8
Replicated processes	8
B.6 Mathematical toolkits	8
<i>Circus</i> prelude	8
<i>Circus</i> model checking toolkit	8
<i>Circus</i> Spivey's <i>Z</i> bag toolkit	8

List of Tables

* PARSER DOES NOT YET SUPPORT LOGICAL CONSTANTS

A Environments

A.1 Informal argument

To typeset an informal argument, you write in L^AT_EX

```
\begin{argue}
  S \dres (T \dres R) \\
\t1    = \id S \comp \id T \comp R \\
\t1    = \id (S \cap T) \comp R & law about $\id$ \\
\t1 = (S \cap T) \dres R.
\end{argue}
```

which corresponds to

$$\begin{aligned} S \triangleleft (T \triangleleft R) \\ &= \text{id } S \ ; \ \text{id } T \ ; \ R \\ &= \text{id}(S \cap T) \ ; \ R && \text{[law about id]} \\ &= (S \cap T) \triangleleft R. \end{aligned}$$

B Reference card

B.1 Special *Circus* symbols

Refinement

$n \Vdash S = I$	$n \text{ \circircassertref } S = I$
$n \Vdash S \preceq I$	$n \text{ \circircassertref } S \text{ \circircsimulates } I$
$n \Vdash S \sqsubseteq I$	$n \text{ \circircassertref } S \text{ \circircrefines } I$
$n \Vdash S \sqsubseteq Tr I$	$n \text{ \circircassertref } S \text{ \circircrefines } Tr \sim I$
$n \Vdash S \sqsubseteq SFI I$	$n \text{ \circircassertref } S \text{ \circircrefines } SFI \sim I$
$n \Vdash S \sqsubseteq FIDv I$	$n \text{ \circircassertref } S \text{ \circircrefines } FIDv \sim I$

B.2 *Circus* channels and name sets

channel e	$\text{\circircchannel } e$
channel $c : T$	$\text{\circircchannel } c : T$
channel $[X]c : X$	$\text{\circircchannel } [X] c : X$
channelfrom S	$\text{\circircchannelfrom } S$
channelfrom $[X]S[X]$	$\text{\circircchannelfrom } [X] S[X]$
channelset $n == \{c\}$	$\text{\circircchannelset } n == \text{\lchanset } c \text{ \rchanset}$
channelset $n == CSRef$	$\text{\circircchannelset } n == CS$
channelset $[X]n == CSRef$	$\text{\circircchannelset } [X] n == CS$
nameset $n == \{x\}$	$\text{\circircnameset } n == \{\sim x \sim\}$
nameset $n == NSRef$	$\text{\circircnameset } n == NS$

B.3 *Circus* actions

Action definition

$n \hat{=} A$	$n \text{ \circircdef } A$
$n \hat{=} x : T \bullet A$	$n \text{ \circircdef } x : T \text{ \circircspot } A$

Basic actions

Skip	\Skip
Stop	\Stop
Chaos	\Chaos

Prefixing action

$e \longrightarrow A$	$e \text{ \then } A$
$c.0 \longrightarrow A$	$c.0 \text{ \then } A$
$c!v \longrightarrow A$	$c!v \text{ \then } A$
$c?x \longrightarrow A$	$c?x \text{ \then } A$
$c?x!y?z \longrightarrow A$	$c?x!y?z \text{ \then } A$

Prefixing action (extra)

$c?x : (P)!(f x) \longrightarrow A$
`c?x \prefixcolon (P)!(f~x) \then A`
 $c?x : (x > 1)!(f x) \longrightarrow A$
`c?x \prefixcolon (x > 1)!(f~x) \then A`
 $c[\mathbb{N} \times \mathbb{P} \mathbb{N}]?x!(\text{dom } R) \longrightarrow A$
`c[\nat \cross \power~\nat]?x!(\dom~R) \then A`

Unary actions

(S) `\lschexpract S \rschexpract`
 $\mu X \bullet A$ `\circircmu X \circircspot A`
 $A \setminus CS$ `A \circirchide CS`
 $(P) \& A$ `\lcircguard P \rcircguard \circircguard A`

Binary actions

$A \parallel NSa \mid NSb \parallel B$ `A \linter NSa \mid NSb \rinter B`
 $A \parallel B$ `A \interleave B`
 $A \parallel NSa \mid CS \mid NSb \parallel B$ `A \lpar NSa \mid CS \mid NSb \rpar B`
 $A \parallel CS \parallel B$ `A \lpar CS \rpar B`
 $A[NSb \mid CSa \mid CSb \mid NSb]B$ `A [NSb \mid CSa \mid CSb \mid NSb] B`
 $A[CSa \mid CSb]B$ `A [CSa \mid CSb] B`
 $A \sqcap B$ `A \intchoice B`
 $A \square B$ `A \extchoice B`
 $A ; B$ `A \circircseq B`
 $AName$ `AName`
 $AName(x, y)$ `AName(x, y)`
 $AName[\text{new/old}, x/y]$ `AName[new/old, x/y]`

Replicated actions

$\parallel x : T \bullet A$ `\Interleave x: T \circircspot A`
 $\parallel x : T \parallel NS \bullet A$ `\Interleave x: T \linter NS \rinter \circircspot A`
 $\parallel CS \parallel x : T \bullet NS \parallel A$ `\lpar CS \rpar x: T \circircspot \lpar NS \rpar A`
 $\parallel CS \parallel x : T \bullet A$ `\lpar CS \rpar x: T \circircspot A`
 $\sqcap x : T \bullet A$ `\Intchoice x : T \circircspot A`

$\square x : T \bullet A$	<code>\Extchoice x : T \circspot A</code>
$; x : T \bullet A$	<code>\Semi x : T \circspot A</code>

Parenthesised actions

(A)	<code>(A)</code>
$(x : T \bullet A)$	<code>(x : T \circspot A)</code>
$(\mathbf{val} x : T \bullet A)$	<code>(\circval x : T \circspot A)</code>
$(\mathbf{res} x : T \bullet A)$	<code>(\circres x : T \circspot A)</code>
$(\mathbf{vres} x : T \bullet A)$	<code>(\circvres x : T \circspot A)</code>
$(x : T \bullet A)(v)$	<code>(x : T \circspot A)(v)</code>
$(\mathbf{val} x : T \bullet A)(v)$	<code>(\circval x : T \circspot A)(v)</code>
$(\mathbf{res} x : T \bullet A)(v)$	<code>(\circres x : T \circspot A)(v)</code>
$(\mathbf{vres} x : T \bullet A)(v)$	<code>(\circvres x : T \circspot A)(v)</code>
$(\mu X \bullet x : T \bullet A)(v)$	<code>(\circmu X \circspot x : T \circspot A)(v)</code>
$(\mu X \bullet (x : T \bullet A))(v)$	<code>(\circmu X \circspot (x : T \circspot A))(v)</code>
$(\mu X \bullet \mathbf{val} x : T \bullet A)(v)$	<code>(\circmu X \circspot \circval x : T \circspot A)(v)</code>
$(\mu X \bullet \mathbf{res} x : T \bullet A)(v)$	<code>(\circmu X \circspot \circres x : T \circspot A)(v)</code>
$(\mu X \bullet \mathbf{vres} x : T \bullet A)(v)$	<code>(\circmu X \circspot \circvres x : T \circspot A)(v)</code>

B.4 Circus command definitions

Guarded commands

$x, y := v1, v2$	<code>x, y := v1, v2</code>
$x, y : [P, Q]$	<code>x, y \prefixcolon [~ P, Q ~]</code>
$:[P, Q]$	<code>\prefixcolon [~ P, Q ~]</code>
$\{P\}$	<code>\{~ P ~\}</code>
$[P]$	<code>[~ P ~]</code>
$\mathbf{if} P \longrightarrow A \parallel B \mathbf{fi}$	<code>\circif P \circrcthen A \circrclse B \circrcfi</code>
$\mathbf{do} P \longrightarrow A \mathbf{od}$	<code>\circdo P \circrcthen A \circrcod</code>
$\mathbf{con} X \bullet A$	<code>\circcon X \circspot A</code>
$\mathbf{var} x : T \bullet A$	<code>\circvar x : T \circspot A</code>

Parameterised commands

$\mathbf{val} x : T \bullet A$	<code>\circval x : T \circspot A</code>
$\mathbf{res} x : T \bullet A$	<code>\circres x : T \circspot A</code>

$\text{vres } x : T \bullet A$ $\backslash\text{circvres } x : T \backslash\text{circspot } A$

B.5 Circus processes

Process definition

$\text{process } n \hat{=} PD$ $\backslash\text{circprocess } n \backslash\text{circdef } PD$
 $\text{process}[X]n \hat{=} PD$ $\backslash\text{circprocess } [X] n \backslash\text{circdef } PD$

Basic process

$\text{process } n \hat{=} \text{begin} \dots BP \dots \text{end}$
 $\backslash\text{circprocess } n \backslash\text{circdef } \backslash\text{circbegin } \backslash\text{ldots } BP \backslash\text{ldots } \backslash\text{circend}$
 $\text{state } n == S$ $\backslash\text{circstate } n == S$
 $\text{state } S$ $\backslash\text{circstate } S$

Unary processes

$P \backslash CS$ $P \backslash\text{circhide } CS$
 $PName$ $PName$
 $PName[\mathbb{N}]$ $PName[\backslash\text{nat}]$
 $PName(x, y)$ $PName(x, y)$
 $PName[\mathbb{N}](x, y)$ $PName[\backslash\text{nat}](x, y)$
 $PName[x]$ $PName \backslash\text{circindex } x \backslash\text{rcircindex}$
 $PName[\mathbb{N}][x]$ $PName[\backslash\text{nat}] \backslash\text{lcircindex } x \backslash\text{rcircindex}$
 $PName[c, d := e, f]$ $PName \backslash\text{circrename } c, d := e, f \backslash\text{rcircrename}$
 $PName[\mathbb{N}][c, d := e, f]$ $PName[\backslash\text{nat}] \backslash\text{lcircrename } c, d := e, f \backslash\text{rcircrename}$

Binary processes

$P \parallel Q$ $P \backslash\text{interleave } Q$
 $P \llbracket CS \rrbracket Q$ $P \backslash\text{lpar } CS \backslash\text{rpar } Q$
 $P \sqcap Q$ $P \backslash\text{intchoice } Q$
 $P \square Q$ $P \backslash\text{extchoice } Q$
 $P ; Q$ $P \backslash\text{circseq } Q$

Parameterised and indexed processes

$x : T \bullet P$ $x : T \backslash\text{circspot } P$
 $x : T \odot P$ $x : T \backslash\text{circindex } P$

Parenthesised processes

(P)	(P)
$(x : T \bullet P)$	$(x : T \text{ \circircspot } P)$
$(x : T \odot P)$	$(x : T \text{ \circircindex } P)$
$(P)[c := d]$	$(P) \text{ \lcircrename } c := d \text{ \rcircrename}$
$(x : T \bullet P)[c := d]$	$(x : T \text{ \circircspot } P) \text{ \lcircrename } c := d \text{ \rcircrename}$
$(x : T \odot P)[c := d]$	$(x : T \text{ \circircindex } P) \text{ \lcircrename } c := d \text{ \rcircrename}$
$(x : T \bullet P)(v)$	$(x : T \text{ \circircspot } P)(v)$
$(x : T \odot P)(v)$	$(x : T \text{ \circircindex } P)(v)$
$[X](x : X \bullet P)[\mathbb{N}](1)$	$[X](x : X \text{ \circircspot } P)[\text{ \nat }](1)$
$[X](x : X \odot P)[\mathbb{N}](1)$	$[X](x : X \text{ \circircindex } P)[\text{ \nat }](1)$
$(\mu X \bullet x : T \bullet P)(v)$	$(\text{ \circircmu } X \text{ \circircspot } x : T \text{ \circircspot } P)(v)$
$(\mu X \bullet (x : T \bullet P))(v)$	$(\text{ \circircmu } X \text{ \circircspot } (x : T \text{ \circircspot } P))(v)$
$(\mu X \bullet \text{ val } x : T \bullet P)(v)$	$(\text{ \circircmu } X \text{ \circircspot } \text{ \circircval } x : T \text{ \circircspot } P)(v)$

Replicated processes

$\parallel x : T \bullet P$	$\text{ \Interleave } x : T \text{ \circircspot } P$
$\parallel x : T \llbracket CS \rrbracket \bullet P$	$\text{ \Parallel } x : T \text{ \lpar } CS \text{ \rpar } \text{ \circircspot } P$
$\sqcap x : T \bullet P$	$\text{ \Intchoice } x : T \text{ \circircspot } P$
$\sqcup x : T \bullet P$	$\text{ \Extchoice } x : T \text{ \circircspot } P$
$;\ x : T \bullet P$	$\text{ \Semi } x : T \text{ \circircspot } P$

B.6 Mathematical toolkits $SS \diamond S$ $SS \text{ \dcap } S$

Circus prelude

\mathbb{B}	 \boolean
\mathbb{U}	 \universe
True	 \true
False	 \false

Circus model checking toolkit

$SS \text{ \textcircled{ } } TT$	$SS \text{ \gendj } TT$
$\otimes SS$	$\text{ \regions } SS$
$SS \searrow S$	$SS \text{ \dsetminus } S$

Circus Spivey's Z bag toolkit

$\text{ bag } X$	$\text{ \bag } X$
$B \# n$	$B \text{ \bcount } n$
$n \otimes B$	$n \text{ \otimes } B$
$B \uplus C$	$B \text{ \uplus } C$
$B \cup C$	$B \text{ \uminus } C$
$x \in B$	$x \text{ \inbag } B$
$B \sqsubseteq C$	$B \text{ \subbageq } C$
$\llbracket x, y \rrbracket$	$\text{ \lbag } x, y \text{ \rbag}$