A Framework for Debugging
Olivier, P.A.

Citation for published version (APA):
A

Syntax of TOOLBUS Scripts

In this appendix we will give the complete definition of the syntax of TOOLBUS scripts using SDF (Syntax Definition Formalism, see [HHKR92]) which is part of the ASF+SDF formalism that has been discussed in the second part of this thesis. SDF should be quite easy to understand for people who are familiar with syntax formalisms like BNF. The main difference is that in SDF the non-terminals (sorts) are placed on the right-hand side instead of on the left.

exports

sorts BOOL NAT INT SIGN EXP UNSIGNED-REAL REAL STRING

ID NAME VNAME BSTR TERM TERM-LIST VAR GEN-VAR TYPE

ATOM ATOMIC-FUN PROC PROC-APPL FORMALS TIMER-FUN

FEATURE-ASG FEATURES TB-CONFIG DEF T-SCRIPT

lexical syntax

[ \t\n] -> LAYOUT

"%" "[\n]* -> LAYOUT

[0-9] + -> NAT

NAT -> INT

SIGN NAT -> INT

[+-] -> SIGN

[eE] NAT -> EXP

[eE] SIGN NAT -> EXP

NAT "." NAT -> UNSIGNED-REAL

NAT "." NAT EXP -> UNSIGNED-REAL

UNSIGNED-REAL -> REAL

SIGN UNSIGNED-REAL -> REAL

[a-zA-Z][A-Za-z0-9-]* -> ID

"\" "\"[\"]* "\"" -> STRING

[A-Z][A-Za-z0-9-]* -> NAME

[A-Z][A-Za-z0-9-]* -> VNAME

[a-zA-Z][a-z\-]* -> ATOMIC-FUN
A / SYNTAX OF TOOLBUS SCRIPTS

delay -> TIMER-FUN
abs-delay -> TIMER-FUN
timeout -> TIMER-FUN
abs-timeout -> TIMER-FUN

collection-free syntax
true -> BOOL
false -> BOOL
BOOL -> TERM
INT -> TERM
REAL -> TERM
STRING -> TERM

TERM -> TYPE

VNAME -> VAR
VNAME ":" TYPE -> VAR

VAR -> GEN-VAR
VAR ":?" -> GEN-VAR
GEN-VAR -> TERM
"<" TERM ">" -> TERM
ID -> TERM
ID "(" TERM-LIST ")" -> TERM
(TERM "," )* -> TERM-LIST
"[" TERM-LIST "]" -> TERM

NAME -> VNAME

ATOMIC-FUN "(" TERM-LIST ")" -> ATOM
delta -> ATOM
tau -> ATOM
create "(" PROC-APPL "," TERM ")" -> ATOM
ATOM TIMER-FUN "(" TERM ")" -> ATOM
VNAME ":=" TERM -> ATOM

ATOM -> PROC
PROC "+" PROC -> PROC {left}
PROC "." PROC -> PROC {right}
PROC "|" PROC -> PROC {right}
PROC "**" PROC -> PROC {left}
"(" PROC ")" -> PROC {bracket}
if TERM then PROC else PROC fi -> PROC
if TERM then PROC fi -> PROC
execute(TERM-LIST) -> PROC
let {VAR ","} in PROC endlet -> PROC
NAME        --> PROC-APPL
NAME "(* TERM-LIST ")" --> PROC-APPL
PROC-APPL   --> PROC

"(" {GEN-VAR ","}* ")" --> FORMALS

process NAME FORMALS is PROC --> DEF
ID ":=" STRING    --> FEATURE-ASG
"{ " { FEATURE-ASG ";"}* "}" --> FEATURES
tool ID FORMALS is FEATURES --> DEF
toolbus "{(PROC-APPL ","+)" --> TB-CONFIG
DEF* TB-CONFIG    --> T-SCRIPT

priorities
PROC "*" PROC --> PROC > PROC "." PROC --> PROC >
PROC "+" PROC --> PROC > PROC "|" PROC --> PROC