Parameters Module

Template Module

param

Uses

N/A

Syntax

Exported Constants

Constant name	Type	Value
E	$\mathbb{R}$	7.17×10^{7}
TD	$\mathbb{R}$	3.0
M	$\mathbb{R}$	7.0
MK	$\mathbb{R}$	2.86×10^{-53}
LSF	$\mathbb{R}$	1.0

Exported Types

Type name	Type
Param	?

Exported Access Programs

Routine name	In	Out	Exceptions
Param		Param	

Semantics

State Variables

Variable name	Type
a	$\mathbb{R}$
b	$\mathbb{R}$
t	$\mathbb{R}$
w	$\mathbb{R}$
tnt	$\mathbb{R}$
pbtol	$\mathbb{R}$
asprat	$\mathbb{R}$
sd	$\mathbb{R}$
h	$\mathbb{R}$
gtf	$\mathbb{R}$
ldf	$\mathbb{R}$
wtnt	$\mathbb{R}$
sdvect	sequence [3] of $\mathbb{R}$
gt	String

Environment Variables

N/A

Access Routine Semantics

Param():

- transition:

$$
\begin{aligned}
a &:= 0.0 \\
b &:= 0.0 \\
t &:= 0.0 \\
w &:= 0.0 \\
tnt &:= 0.0 \\
pbtol &:= 0.0 \\
asprat &:= 0.0 \\
sd &:= 0.0 \\
h &:= 0.0 \\
gtf &:= 0.0 \\
ldf &:= 0.0 \\
wtnt &:= 0.0 \\
sdvect &:= \langle 0.0, 0.0, 0.0 \rangle
\end{aligned}
$$

gt := “”

- output:

  out := self

- exception:

  N/A

Input Format Module

Module

inputFormat

Uses

param

Syntax

Exported Constants

N/A

Exported Types

N/A

Exported Access Programs

Routine name	In	Out	Exceptions
get_input	string	Param	

Semantics

State Variables

N/A

Environment Variables

Variable name	Type
filesys	FileSystem Read

Access Routine Semantics

get_input(filename, p):

- transition:

$$
\begin{aligned}
filesys &:= filename \\
p.a &:= filesys.readline \\
p.b &:= filesys.readline \\
p.t &:= filesys.readline \\
p.gt &:= filesys.readline
\end{aligned}
$$

$$
\begin{aligned}
p.w &:= filesys.readline \\
p.tnt &:= filesys.readline \\
p.sdvect[0] &:= filesys.readline \\
p.sdvect[1] &:= filesys.readline \\
p.sdvect[2] &:= filesys.readline \\
pbtol &:= filesys.readline
\end{aligned}
$$

- output:

  N/A

- exception:

  N/A

Input Constraints Module

Module

derivedValues

Uses

param

Syntax

Exported Constants

N/A

Exported Types

N/A

Exported Access Programs

Routine name	In	Out	Exceptions
derived_params	Param	Param	

Semantics

State Variables

N/A

Environment Variables

N/A

Access Routine Semantics

derived_params(p):

- transition:

$$
\begin{aligned}
p.asprat &:= \frac{p.a}{p.b} \\
p.sd &:= \sqrt{p.sdvect[0]^2 + p.sdvect[1]^2 + p.sdvect[2]^2} \\
p.ldf &:= \frac{p.td}{60.0}\frac{p.m}{16.0} \\
p.wtnt &:= p.w \times p.tnt
\end{aligned}
$$

$$p.h := \begin{cases} p.t = 2.50 \implies 2.16 \\ p.t = 2.70 \implies 2.59 \\ p.t = 3.0 \implies 2.92 \\ p.t = 4.0 \implies 3.78 \\ p.t = 5.0 \implies 4.57 \\ p.t = 6.0 \implies 5.56 \\ p.t = 8.0 \implies 7.42 \\ p.t = 10.0 \implies 9.02 \\ p.t = 12.0 \implies 11.91 \\ p.t = 16.0 \implies 15.09 \\ p.t = 19.0 \implies 18.26 \\ p.t = 22.0 \implies 21.44 \\ \text{True} \implies 0.0 \end{cases}$$

$$p.gtf := \begin{cases} p.gt = \text{“AN”} \implies 1.0 \\ p.gt = \text{“HS”} \implies 2.0 \\ p.gt = \text{“FT”} \implies 3.0 \\ \text{True} \implies 0.0 \end{cases}$$

- output:

  $out := p$

- exception:

  N/A

Input Constraints Module

Module

checkConstraints

Uses

param

Syntax

Exported Constants

N/A

Exported Types

N/A

Exported Access Programs

Routine name	In	Out	Exceptions
check_constraints	Param		INPUTERROR

Semantics

State Variables

N/A

Environment Variables

N/A

Access Routine Semantics

derived_params(p):

- transition:

  N/A

- output:

  N/A

- exception:

$$
exc := \begin{cases}
p.a \leq 0.0 \wedge p.b \leq 0.0 & \Longrightarrow \text{INPUTERROR} \\
\neg(1.0 \leq p.asprat \leq 5.0) & \Longrightarrow \text{INPUTERROR} \\
p.t \notin \left\{ \begin{matrix} 2.50,\ 2.70,\ 3.0,\ \ 4.0, \\ 5.0,\ \ 6.0,\ \ 8.0,\ \ 10.0, \\ 12.0,\ 16.0,\ 19.0,\ 22.0, \end{matrix} \right\} & \Longrightarrow \text{INPUTERROR} \\
p.qt \notin \left\{ \begin{matrix} \text{“AN”},\ \text{“HS”},\ \text{“FT”}, \\ \text{“an”},\ \ \text{“hs”},\ \ \text{“ft”}, \end{matrix} \right\} & \Longrightarrow \text{INPUTERROR} \\
p.tnt \leq 0.0 & \Longrightarrow \text{INPUTERROR} \\
\neg(4.5 \leq p.wtnt \leq 910.0) & \Longrightarrow \text{INPUTERROR} \\
\neg(6.0 \leq p.sd \leq 130.0) & \Longrightarrow \text{INPUTERROR}
\end{cases}
$$

Table Input Module

Module

readTable

Uses

None

Syntax

Exported Access Programs

Routine name	In	Out	Exceptions
read_table	string	sequence [2, ?, ?] of real	FILEERROR

Semantics

State Variables

contents : sequence [?, ?] of string

Environment Variables

$filesys$: FileSystem Read

Assumptions

None

Access Routine Semantics

read_table($filename$):

- transition:

  $contents := map\ splitOn(`,')\ filesys.readall(filename)$

- output:

  $out \quad := map\ (\lambda x \rightarrow x[1: \ : 2])\ contents\ ||\ map\ (\lambda x \rightarrow x[2: \ : 2])\ contents$

- exception:

  $exc := \neg filesys.exists(filename) \implies$ FILEERROR