

Mystery1 Module Interface

Uses

None

Syntax

Exported Types

Mystery1 = ?

Exported Access Routine

Routine Name	In	Out	Exceptions
new Mystery1		Mystery1	None
f1		\mathbb{N}	None
f2	String		None
f3	String, String		None
f4	String, String	\mathbb{B}	None

Semantic

Local Types

String = char[]

X = tuple(x:String, y:String)

State Variables

S_1 : String{ }

S_2 : X{ }

Assumptions

- All inputs are of the proper type.
- The notation $X(s_1, s_2)$ is shorthand for saying (s_1, s_2) is a tuple of type X, where $x = s_1$ and $y = s_2$.

Access Routine Semantics

Mystery1():

- transition: $S_1, S_2 = \{ \}, \{ \}$
- output: out := this

f1():

- output: out := $|S_1|$

f2(s):

- transition: $S_1 := S_1 \cup \{s\}$

f3(s_1, s_2):

- transition: $S_2 := S_2 \cup \{X(s_1, s_2)\}$

f4(s_1, s_2):

- output: $\text{out} := s_2 \in \cup(s : \text{String} \mid s \in g(s_1) : g(s)) \wedge s_2 \notin g(s_1)$

Local Functions

$g(s) : \text{String} \rightarrow \text{String}\{ \}$
 $g(s) = \{s' : X(s, s') \in S_2\}$

Mystery2(T) Generic Module

Uses

None

Syntax

Exported Types

Mystery2 = ?

Exported Access Routine

Routine Name	In	Out	Exceptions
new Mystery2		Mystery2	None
f1	T, T		Exception
f2		T	Exception

Semantic

Local Types

$X = \text{tuple}(x_1:T, x_2:T \cup \{k\}, x_3:\mathbb{N})$

Local Constants

$k = -1$

State Variables

$S = X\{ \}$

Assumptions

- All inputs are of the proper type.
- $k \notin T$
- The notation $X(a, b, c)$ is shorthand for saying (a, b, c) is a tuple of type X, where $x_1 = a$, $x_2 = b$ and $x_3 = c$.

Access Routine Semantics

Mystery2():

- transition: $S = \{ \}$
- output: $\text{out} := \text{this}$

f1(t_1, t_2):

- transition: $S := S \cup \{X(t_1, t, n)\}$, where the following all hold:

1. $\forall(x : X | x \in S : x.x_3 < n)$
2. $\exists(x : X | x \in S : x.x_1 = t_2) \Rightarrow t = t_2$
3. $\neg \exists(x : X | x \in S : x.x_1 = t_2) \Rightarrow t = k$

- exception: $\exists(x : X | x \in S : x.x_2 = t_1) \Rightarrow \text{Exception}$

f2():

- transition: $S := g_2(x.x_1) \cup g_3(x.x_1) - \{x\}$, where: $x \in g_1() \wedge \forall(x' : X | x' \in g_1() : x'.x_3 \leq x.x_3)$
- output: $\text{out} := x.x_1$ where, $x \in g_1() \wedge \forall(x' : X | x' \in g_1() : x'.x_3 \leq x.x_3)$
- exception: $S = \{\} \Rightarrow \text{Exception}$

Local Functions

$g_1() : \text{None} \rightarrow X\{\}$
 $g_1() = \{x : x \in S \wedge x.x_2 = k\}$

$g_2(t) : T \rightarrow X\{\}$
 $g_2(t) = \{X(x_1, k, x_3) : X(x_1, t, x_3) \in S\}$

$g_3(t) : T \rightarrow X\{\}$
 $g_3(t) = \{X(x_1, x_2, x_3) : X(x_1, x_2, x_3) \in S \wedge x_2 \neq t\}$

Considerations

It is assumed that $t_1 = t_2$, where $t_1, t_2 : T$, would be implemented using the `.equals` method in Java. That is, $t_1 = t_2$ would be equivalent to `t1.equals(t2)`.