───────────────────────────── MODULE *Assumes* ─────────────────────────────

EXTENDS *Integers*, *Sequences*

You can run this as a model using "No behavior spec" mode
Single line comment

ASSUME
 ∧ TRUE = TRUE
 ∧ ¬FALSE = TRUE

$Jason \triangleq$ "jason"
ASSUME
 $Jason =$ "jason"

$record \triangleq [name \mapsto$ "jason"$, age \mapsto 2]$
ASSUME
 ∧ $record.name =$ "jason"
 ∧ $record.name \neq$ "foo"

ASSUME
 $\forall F \in \{\text{TRUE}\} : F = F$

ASSUME
 $\forall F \in \{\text{FALSE}\} : F = F$

ASSUME ⇒ means "implies", as in $A \Rightarrow B$ is "(*not* A) OR B"
 FALSE ⇒ TRUE = TRUE

ASSUME ⇒ means "implies", as in $A \Rightarrow B$ is "(*not* A) OR B"
 FALSE ⇒ FALSE = TRUE

ASSUME ⇒ means "implies", as in $A \Rightarrow B$ is "(*not* A) OR B"
 TRUE ⇒ TRUE = TRUE

ASSUME ⇒ means "implies", as in $A \Rightarrow B$ is "(*not* A) OR B"
 TRUE ⇒ FALSE = FALSE

ASSUME
 TRUE ≡ TRUE

ASSUME
 FALSE ≡ FALSE

ASSUME
 $\forall F, G \in \{\text{TRUE}, \text{FALSE}\} : (F \Rightarrow G) \equiv \neg F \vee G$

Sets

1

ASSUME
$\{1,\,2,\,2,\,2,\,3\} = \{1,\,2,\,3\}$

ASSUME
$\{1,\,2,\,3,\,3,\,4,\,4\} \setminus \{4\} = \{1,\,2,\,3\}$

ASSUME
$\{1,\,2,\,3\} \cup \{4,\,5,\,6\} = \{1,\,2,\,3,\,4,\,5,\,6\}$

ASSUME
$\exists\, x \in \{3,\,4,\,5\} : x = 5$

ASSUME
$\{1,\,3\} \subseteq \{3,\,2,\,1\}$

$IsPrime(x) \;\triangleq\; x > 1 \wedge \neg \exists\, d \in 2\,..\,(x-1) : x\%d = 0$

For all y in S such that y is not prime or y is less than or equal to x

$LargestPrime(S) \;\triangleq\;$ CHOOSE $x \in S :$
$\qquad\qquad \wedge\, IsPrime(x)$
$\qquad\qquad \wedge\, \forall\, y \in S :$
$\qquad\qquad\qquad IsPrime(y) \Rightarrow y \leq x$
$\qquad\qquad\qquad$ or $y > x \Rightarrow \neg IsPrime(y)$

ASSUME
$LargestPrime(\{1,\,2,\,3,\,4,\,5,\,6,\,7,\,8,\,9,\,10\}) = 7$

$IsEven(x) \;\triangleq\; x\%2 = 0$

$LargetEven(S) \;\triangleq\;$ CHOOSE $x \in S :$
$\qquad\qquad \wedge\, IsEven(x)$
$\qquad\qquad \wedge\, \forall\, y \in S :$
$\qquad\qquad\qquad IsEven(y) \Rightarrow y \leq x$

ASSUME
$LargetEven(\{1,\,2,\,3,\,4,\,5,\,5,\,5\}) = 4$

ASSUME
$\forall\, x \in \{\} :$ FALSE

ASSUME
$\forall\, x \in \{\} :$ TRUE

ASSUME

$$\forall\, x \in \{\} : 7$$

ASSUME
  $$\forall\, x \in \{\text{FALSE}\} : \text{TRUE}$$

ASSUME
  $$\forall\, x \in \{\text{TRUE}\} : \text{TRUE}$$

ASSUME
  $$(\forall\, x \in \{\text{FALSE}\} : \text{FALSE}) = \text{FALSE}$$

$$IsCommutative(Op(\_,\ \_),\ S) \;\triangleq\; \forall\, x \in S :$$
$$\forall\, y \in S : Op(x,\ y) = Op(y,\ x)$$

$$Add(x,\ y) \;\triangleq\; x + y$$
$$Divide(x,\ y) \;\triangleq\; x \div y$$

ASSUME
  $$IsCommutative(Add,\ \{1,\ 2,\ 3\})$$

ASSUME
  $$IsCommutative(Divide,\ \{1,\ 2,\ 3\}) = \text{FALSE}$$

ASSUME
  $$IsCommutative(Divide,\ \{1,\ 2,\ 3\}) \Rightarrow \text{FALSE}$$

ASSUME
  $$IsCommutative(Divide,\ \{1,\ 2,\ 3\}) \Rightarrow \text{TRUE}$$

ASSUME
  $$\neg IsCommutative(Divide,\ \{1,\ 2,\ 3\})$$

ASSUME
  $$\neg \exists\, x \in \{1,\ 3,\ 5\} : IsEven(x)$$

$$Pick(S) \;\triangleq\; \text{CHOOSE } s \in S : \text{TRUE}$$
RECURSIVE $SetReduce(\_,\ \_,\ \_)$
$$SetReduce(Op(\_,\ \_),\ S,\ value) \;\triangleq\; \text{IF } S = \{\} \text{ THEN } value$$
$$\text{ELSE LET } s \;\triangleq\; Pick(S)$$
$$\text{IN } SetReduce(Op,\ S \setminus \{s\},\ Op(s,\ value))$$

$$Sum(S) \;\triangleq\; \text{LET } \_op(a,\ b) \;\triangleq\; a + b$$
$$\text{IN } SetReduce(\_op,\ S,\ 0)$$

ASSUME
  $$Sum(\{1,\ 2,\ 3\}) = 6$$

$$Min(S) \;\triangleq\; \text{CHOOSE } x \in S : \forall\, y \in S : x \leq y$$

ASSUME
  $Min(\{5,\,3,\,7,\,10,\,2,\,9\}) = 2$

$Max(S) \triangleq \text{CHOOSE } x \in S : \forall\, y \in S : x \geq y$

ASSUME
  $Max(\{4,\,6,\,1,\,2,\,9,\,3,\,5\}) = 9$

ASSUME
  $\langle 1,\,2,\,3 \rangle \in Seq(\{1,\,2,\,3\})$

ASSUME
  $\langle 4 \rangle \notin Seq(\{1,\,2,\,3\})$

ASSUME
  $\langle 1,\,2,\,3,\,4 \rangle \notin Seq(\{1,\,2,\,3\})$

Sets of tuples.

$chessboard\_squares \triangleq \{\text{``a''},\text{``b''},\text{``c''},\text{``d''},\text{``e''},\text{``f''},\text{``g''},\text{``h''}\} \times (1\mathbin{..}8)$

ASSUME
  $\wedge \langle \text{``a''},\,1 \rangle \in chessboard\_squares$
  $\wedge \langle \text{``a''},\,2 \rangle \in chessboard\_squares$
  $\wedge \langle \text{``a''},\,3 \rangle \in chessboard\_squares$
  $\wedge \langle \text{``a''},\,4 \rangle \in chessboard\_squares$

$jason \triangleq (1\mathbin{..}2) \times \{\text{``Jason''},\text{``DeBolt''}\}$

ASSUME
  $\wedge \langle 1,\,\text{``Jason''} \rangle \in jason$
  $\wedge \langle 2,\,\text{``Jason''} \rangle \in jason$
  $\wedge \langle 1,\,\text{``DeBolt''} \rangle \in jason$
  $\wedge \langle 2,\,\text{``DeBolt''} \rangle \in jason$

$digits \triangleq \{\text{``one''},\text{``three''}\} \times \{\text{``two''},\text{``four''}\}$

ASSUME
  $\wedge \langle \text{``one''},\,\text{``two''} \rangle \in digits$
  $\wedge \langle \text{``three''},\,\text{``four''} \rangle \in digits$

$A \triangleq \{1\}$
$B \triangleq \{2\}$
$C \triangleq \{3\}$

ASSUME
    $\wedge \langle 1,\, 2,\, 3 \rangle \in A \times B \times C$
    $\wedge \langle 1,\, \langle 2,\, 3 \rangle \rangle \in A \times (B \times C)$
    $\wedge \langle \langle 1,\, 2 \rangle,\, 3 \rangle \in (A \times B) \times C$

Structures.

Structures are hashes. They have keys and values. You specify them as [key $\mapsto$ *value*] and query them with either ["key"] or *.key*. Both are legal and valid.

$SomeHash \triangleq [x \mapsto 1,\, y \mapsto \{2,\, 3\}]$

ASSUME
    $\wedge SomeHash.x = 1$
    $\wedge SomeHash[\text{"x"}] = 1$
    $\wedge SomeHash.y = \{2,\, 3\}$
    $\wedge SomeHash[\text{"y"}] = \{2,\, 3\}$
    $\wedge$ DOMAIN $SomeHash = \{\text{"x"},\, \text{"y"}\}$

Aside from that, there's one extra trick structures have. Instead of key $\mapsto$ *value*, you can do key : set. In that case, instead of a structure you get the set of all structures which have, for each given key, a value in the set.

$SetOfStructures \triangleq [x : \{1\},\, y : \{2,\, 3,\, 4\}]$

If you use : syntax and any of the values are not sets, then the entire construct is invalid. In other words, while [a: {1}, b: {2, 3}] is the above set, [a: 1, b: {2, 3}] will throw an error if you try to use it.

ASSUME
    $\wedge [x \mapsto 1,\, y \mapsto 2] \in SetOfStructures$
    $\wedge [x \mapsto 1,\, y \mapsto 3] \in SetOfStructures$
    $\wedge [x \mapsto 1,\, y \mapsto 4] \in SetOfStructures$

Type Composition

Any type can be squeezed inside any other type.

$crazy \triangleq [a \mapsto \{\langle \rangle,\, \langle 1,\, 2,\, 3 \rangle,\, \langle 3,\, 2,\, 1 \rangle\},\, b \mapsto \langle [a \mapsto 0] \rangle]$

A function of keys mapping to sets of tuples or of keys mapping to tuples of functions.

ASSUME
  $crazy.b[1].a = 0$ Remember that tuples are 1 indexed.

$blah \triangleq [name \mapsto \text{"jason"},\, hobbies \mapsto [outdoor \mapsto \langle \text{"cycling"},\, \text{"hiking"} \rangle,\, indoor \mapsto \langle \text{"reading"},\, \text{"watching tv"} \rangle]]$

ASSUME
    $\wedge blah.name = \text{"jason"}$

$\land\ blah.hobbies.outdoor[1] =$ "cycling"

\ * Modification History
\ * Last modified Sun Apr 21 17:24:01 PDT 2019 by $jasondebolt$
\ * Created Sat Apr 20 20:01:34 PDT 2019 by $jasondebolt$