MODULE *Assumes*

EXTENDS *Integers*, *Sequences*, *FiniteSets*, *Naturals*

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

ASSUME
$\land$ TRUE = TRUE
$\land$ $\neg$FALSE = TRUE

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

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

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

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

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

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

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

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

ASSUME
TRUE $\equiv$ TRUE

ASSUME
FALSE $\equiv$ FALSE

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

Sets

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

ASSUME
$(\forall\, i \in \{2, 4, 8\} : i\%2 = 0) = \text{TRUE}$

ASSUME
$(\{1, 2\} \in \text{SUBSET } \{1, 2, 3\}) = \text{TRUE}$

ASSUME
$(\{1, 2\} \in \text{SUBSET } (\{1, 3\} \cup \{4, 2\})) = \text{TRUE}$

ASSUME
$\land \{\text{“one”}, \text{“two”}\} \neq \{\}$
$\land \{\text{“one”}, \text{“two”}\} \neq \{\}$
$\land \{\text{“one”}, \text{“two”}\} \setminus \{\text{“one”}\} = \{\text{“two”}\}$

ASSUME
$\land\, \mathit{IsFiniteSet}(\{1, 2, 3\})$
$\land \neg \mathit{IsFiniteSet}(\mathit{Nat})$

ASSUME
$\land\, \mathit{Cardinality}(\{3, 4, 1\}) = 3$
$\land\, \mathit{Cardinality}(\{\}) = 0$

ASSUME
$\{x \in 1 \,..\, 8 : x\%2 = 1\} = \{1, 3, 5, 7\}$

ASSUME
$\{x \in 1 \,..\, 8 : x\%2 = 1 \land \neg(x\%5 = 0)\} = \{1, 3, 7\}$

ASSUME
$\{x \in \{[name \mapsto \text{“jason”}], [name \mapsto \text{“jake”}]\} : x.name = \text{“jason”}\} = \{[name \mapsto \text{“jason”}]\}$

ASSUME
$\{[name \mapsto s, status \mapsto \text{“default”}] : s \in \{\text{“ec2”}, \text{“s3”}\}\} = \{[name \mapsto \text{“ec2”}, status \mapsto \text{“default”}],$
$[name \mapsto \text{“s3”}, status \mapsto \text{“default”}]\}$

ASSUME
$\{\langle x, y\rangle \in \{\text{“a”}, \text{“b”}\} \times \{[name \mapsto \text{“jason”}], [name \mapsto \text{“bob”}]\} : y.name = \text{“jason”}\} = \{\langle \text{“a”}, [name \mapsto \text{“jason”}$

ASSUME
$\{\langle x, y\rangle \in \{\langle 1, 2\rangle, \langle 4, 2\rangle\} : x > y\} = \{\langle 4, 2\rangle\}$

IF STATEMENTS

ASSUME
$\land$ (IF $1 < 3$ THEN 1 ELSE $0) = 1$
$\land$ (IF $1 < 3$ THEN IF $2 > 1$ THEN 6 ELSE 4 ELSE $7) = 6$

LET STATEMENTS

ASSUME
$\land$ LET $x \triangleq 6$IN $x \in \{6, 7\}$

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

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

$LargestPrime(S) \triangleq$ CHOOSE $x \in S$:
$\land\ IsPrime(x)$
$\land \forall\, y \in S$:
$IsPrime(y) \Rightarrow y \leq x$
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 $x \in S$:
$\land\ IsEven(x)$
$\land \forall\, y \in S$:
$IsEven(y) \Rightarrow y \leq x$

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

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

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

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

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

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

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

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

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

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 $SetReduce(\_, \_, \_)$
$SetReduce(Op(\_, \_), S, value) \triangleq \text{IF } S = \{\} \text{ THEN } value$

$$\begin{array}{l} \quad \text{ELSE LET } s \triangleq Pick(S) \\ \quad \text{IN } \; SetReduce(Op,\ S \setminus \{s\},\ Op(s,\ value)) \end{array}$$

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

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

SEQUENCES

$Seq(S)$ is the set of all finite sequences of set S.

ASSUME
$\begin{array}{l} \land \langle\rangle \quad\ \in Seq(\{1,\ 0\}) \\ \land \langle 1\rangle \quad \in Seq(\{1,\ 0\}) \\ \land \langle 0\rangle \quad \in Seq(\{1,\ 0\}) \\ \land \langle 0,\ 0\rangle \in Seq(\{1,\ 0\}) \\ \land \langle 0,\ 1\rangle \in Seq(\{1,\ 0\}) \\ \land \langle 1,\ 0\rangle \in Seq(\{1,\ 0\}) \\ \land \langle 1,\ 1\rangle \in Seq(\{1,\ 0\}) \end{array}$

ASSUME
$\land \{\langle 0\rangle\} \subseteq Seq(\{0,\ 1\})$

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

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

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

ASSUME
$\land \langle 1,\ 2\rangle \circ \langle 3,\ 4\rangle = \langle 1,\ 2,\ 3,\ 4\rangle$

$LessThanThree(x) \triangleq x < 3$

ASSUME
$\land Head(\langle 2, 3, 4\rangle) = 2$
$\land Tail(\langle 2, 3, 4\rangle) = \langle 3, 4\rangle$
$\land Append(\langle 1, 2\rangle, 3) = \langle 1, 2, 3\rangle$
$\land Len(\langle 5, 2, 1\rangle) = 3$
$\land SubSeq(\langle 9, 3, 5, 6\rangle, 1, 3) = \langle 9, 3, 5\rangle$
$\land SelectSeq(\langle 5, 2, 9\rangle, LessThanThree) = \langle 2\rangle$

TUPLES

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

ASSUME
$\land \langle \text{“a”}, 1\rangle \in chessboard\_squares$
$\land \langle \text{“a”}, 2\rangle \in chessboard\_squares$
$\land \langle \text{“a”}, 3\rangle \in chessboard\_squares$
$\land \langle \text{“a”}, 4\rangle \in chessboard\_squares$

$jason \triangleq (1 \,..\, 2) \times \{\text{“Jason”}, \text{“DeBolt”}\}$

ASSUME
$\land \langle 1, \text{“Jason”}\rangle \in jason$
$\land \langle 2, \text{“Jason”}\rangle \in jason$
$\land \langle 1, \text{“DeBolt”}\rangle \in jason$
$\land \langle 2, \text{“DeBolt”}\rangle \in jason$

$digits \triangleq \{\text{“one”}, \text{“three”}\} \times \{\text{“two”}, \text{“four”}\}$

ASSUME
$\land \langle \text{“one”}, \text{“two”}\rangle \in digits$
$\land \langle \text{“three”}, \text{“four”}\rangle \in digits$

ASSUME
$\land \langle \text{“one”}, \text{“two”}\rangle \circ \langle \text{“three”}\rangle = \langle \text{“one”}, \text{“two”}, \text{“three”}\rangle$
$\land \langle \text{“one”}, \text{“two”}\rangle \circ \langle \text{“three”}\rangle = \langle \text{“one”}, \text{“two”}, \text{“three”}\rangle$

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

ASSUME
$\land \langle 1, 2, 3\rangle \in A \times B \times C$

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

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

$SomeHash2 \triangleq [x \mapsto 1, y \mapsto \{2, 3\}]$
$SomeHash3 \triangleq [SomeHash2$ EXCEPT $![\text{"x"}] = 6]$
$SomeHash4 \triangleq [test \mapsto \text{"jason"}, blah \mapsto \text{"debolt"}]$

ASSUME
$\land SomeHash3.x = 6$

$MyHash \triangleq [a : \{\text{"foo"}, \text{"bar"}\}]$
$MyHash2 \triangleq [a : \{\text{"test"}\}, b : \{\text{"two"}, \text{"three"}\}]$
$MyHash3 \triangleq [t : \{1, 2, \text{"x"}, \text{"j"}\}, p : \{\text{"n"}, \text{"z"}\}]$
$BasicHash \triangleq [name \mapsto \text{"jason"}, job \mapsto \text{"Engineer"}]$

ASSUME
$\land [a \mapsto \text{"foo"}] \in MyHash$
$\land [a \mapsto \text{"test"}, b \mapsto \text{"two"}] \in MyHash2$
$\land [t \mapsto 1, p \mapsto \text{"z"}] \in MyHash3$
$\land BasicHash.name = \text{"jason"} \land BasicHash[\text{"job"}] = \text{"Engineer"}$

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
$\land [x \mapsto 1, y \mapsto 2] \in SetOfStructures$

$\land [x \mapsto 1,\ y \mapsto 3] \in SetOfStructures$
$\land [x \mapsto 1,\ y \mapsto 4] \in SetOfStructures$

Functions

$TestAccounts \triangleq [\{\text{“account1”},\ \text{“account2”}\} \to \{\text{“dev”},\ \text{“qa”},\ \text{“prod”}\}]$

ASSUME
$\land [account1 \mapsto \text{“dev”},\ account2 \mapsto \text{“dev”}] \in TestAccounts$
$\land [account1 \mapsto \text{“prod”},\ account2 \mapsto \text{“prod”}] \in TestAccounts$
$\land [account1 \mapsto \text{“dev”},\ account2 \mapsto \text{“prod”}] \in TestAccounts$

$JobStates \triangleq [\{\text{“job1”},\ \text{“job2”},\ \text{“job3”},\ \text{“job4”}\} \to \{\text{“InProgress”},\ \text{“Succeeded”},\ \text{“Failed”}\}]$

ASSUME
$\land [job1 \mapsto \text{“Failed”},\ job2 \mapsto \text{“Failed”},\ job3 \mapsto \text{“Failed”},\ job4 \mapsto \text{“Failed”}] \in JobStates$

$BedroomDoors \triangleq [\{\text{“door1”},\ \text{“door2”}\} \to \{\text{“unlocked”},\ \text{“locked”}\}]$

ASSUME
$\land [door1 \mapsto \text{“unlocked”},\ door2 \mapsto \text{“unlocked”}] \in BedroomDoors$

$BedroomDoors2 \triangleq [\{\text{“door1”},\ \text{“door2”}\} \to [s : \{\text{“open”},\ \text{“closed”}\},\ l : \{\text{“unlocked”},\ \text{“locked”}\}]]$

ASSUME
$\land [door1 \mapsto [s \mapsto \text{“open”},\ l \mapsto \text{“locked”}],\ door2 \mapsto [s \mapsto \text{“open”},\ l \mapsto \text{“locked”}]] \in BedroomDoors2$
$\land [door1 \mapsto [s \mapsto \text{“closed”},\ l \mapsto \text{“unlocked”}],\ door2 \mapsto [s \mapsto \text{“closed”},\ l \mapsto \text{“locked”}]] \in BedroomDoors2$

$Stoplights \triangleq [\{\text{“light1”},\ \text{“light2”}\} \to \{\text{“green”},\ \text{“yellow”},\ \text{“red”}\}]$

ASSUME
$\land [light1 \mapsto \text{“red”},\ light2 \mapsto \text{“green”}] \in Stoplights$

$Stoplights2 \triangleq [light\_name : \{\text{“light1”},\ \text{“light2”}\},\ state : \{\text{“green”},\ \text{“yellow”},\ \text{“red”}\}]$

ASSUME
$\land [light\_name \mapsto \text{“light1”},\ state \mapsto \text{“green”}] \in Stoplights2$

$PullRequestState \triangleq [pr\_status : \{\text{“open”},\ \text{“closed”}\},\ is\_master : \{\text{TRUE, FALSE}\},\ is\_first\_commit : \{\text{TRUE, FAL}$

ASSUME
$\land [pr\_status \mapsto \text{“open”},\ is\_master \mapsto \text{TRUE},\ is\_first\_commit \mapsto \text{TRUE}] \in PullRequestState$

ASSUME

$\land [\{1, 2, 3\} \to \{\text{“done”}\}] = \{\langle \text{“done”}, \text{“done”}, \text{“done”} \rangle\}$ Turns a function into a set of tuples

$\land [\{\text{“a”}, \text{“b”}\} \to \{\text{“done”}\}] = \{[a \mapsto \text{“done”}, b \mapsto \text{“done”}]\}$ Turns a function into a set of structs

$\land [\{\text{“a”}, \text{“b”}\} \to \{\text{“done”}, \text{“pc”}\}] = \{[a \mapsto \text{“done”}, b \mapsto \text{“done”}],$ Turns a function into a set of structs
$[a \mapsto \text{“done”}, b \mapsto \text{“pc”}],$
$[a \mapsto \text{“pc”}, b \mapsto \text{“done”}],$
$[a \mapsto \text{“pc”}, b \mapsto \text{“pc”}]\}$

$\land [\{\text{“p1”}, \text{“p2”}\} \to \{\text{“a”}, \text{“b”}, \text{“c”}, \text{“done”}\}] = \{[p1 \mapsto \text{“a”}, p2 \mapsto \text{“a”}],$
$[p1 \mapsto \text{“a”}, p2 \mapsto \text{“b”}],$
$[p1 \mapsto \text{“a”}, p2 \mapsto \text{“c”}],$
$[p1 \mapsto \text{“a”}, p2 \mapsto \text{“done”}],$
$[p1 \mapsto \text{“b”}, p2 \mapsto \text{“a”}],$
$[p1 \mapsto \text{“b”}, p2 \mapsto \text{“b”}],$
$[p1 \mapsto \text{“b”}, p2 \mapsto \text{“c”}],$
$[p1 \mapsto \text{“b”}, p2 \mapsto \text{“done”}],$
$[p1 \mapsto \text{“c”}, p2 \mapsto \text{“a”}],$
$[p1 \mapsto \text{“c”}, p2 \mapsto \text{“b”}],$
$[p1 \mapsto \text{“c”}, p2 \mapsto \text{“c”}],$
$[p1 \mapsto \text{“c”}, p2 \mapsto \text{“done”}],$
$[p1 \mapsto \text{“done”}, p2 \mapsto \text{“a”}],$
$[p1 \mapsto \text{“done”}, p2 \mapsto \text{“b”}],$
$[p1 \mapsto \text{“done”}, p2 \mapsto \text{“c”}],$
$[p1 \mapsto \text{“done”}, p2 \mapsto \text{“done”}]\}$

$\land [\{\text{“p1”}, \text{“p2”}, \text{“p3”}\} \to \{\text{“a”}, \text{“b”}, \text{“c”}\}] = \{[p1 \mapsto \text{“a”}, p2 \mapsto \text{“a”}, p3 \mapsto \text{“a”}],$
$[p1 \mapsto \text{“a”}, p2 \mapsto \text{“a”}, p3 \mapsto \text{“b”}],$
$[p1 \mapsto \text{“a”}, p2 \mapsto \text{“a”}, p3 \mapsto \text{“c”}],$
$[p1 \mapsto \text{“a”}, p2 \mapsto \text{“b”}, p3 \mapsto \text{“a”}],$
$[p1 \mapsto \text{“a”}, p2 \mapsto \text{“b”}, p3 \mapsto \text{“b”}],$
$[p1 \mapsto \text{“a”}, p2 \mapsto \text{“b”}, p3 \mapsto \text{“c”}],$
$[p1 \mapsto \text{“a”}, p2 \mapsto \text{“c”}, p3 \mapsto \text{“a”}],$
$[p1 \mapsto \text{“a”}, p2 \mapsto \text{“c”}, p3 \mapsto \text{“b”}],$
$[p1 \mapsto \text{“a”}, p2 \mapsto \text{“c”}, p3 \mapsto \text{“c”}],$
$[p1 \mapsto \text{“b”}, p2 \mapsto \text{“a”}, p3 \mapsto \text{“a”}],$
$[p1 \mapsto \text{“b”}, p2 \mapsto \text{“a”}, p3 \mapsto \text{“b”}],$
$[p1 \mapsto \text{“b”}, p2 \mapsto \text{“a”}, p3 \mapsto \text{“c”}],$
$[p1 \mapsto \text{“b”}, p2 \mapsto \text{“b”}, p3 \mapsto \text{“a”}],$
$[p1 \mapsto \text{“b”}, p2 \mapsto \text{“b”}, p3 \mapsto \text{“b”}],$
$[p1 \mapsto \text{“b”}, p2 \mapsto \text{“b”}, p3 \mapsto \text{“c”}],$
$[p1 \mapsto \text{“b”}, p2 \mapsto \text{“c”}, p3 \mapsto \text{“a”}],$
$[p1 \mapsto \text{“b”}, p2 \mapsto \text{“c”}, p3 \mapsto \text{“b”}],$
$[p1 \mapsto \text{“b”}, p2 \mapsto \text{“c”}, p3 \mapsto \text{“c”}],$
$[p1 \mapsto \text{“c”}, p2 \mapsto \text{“a”}, p3 \mapsto \text{“a”}],$
$[p1 \mapsto \text{“c”}, p2 \mapsto \text{“a”}, p3 \mapsto \text{“b”}],$
$[p1 \mapsto \text{“c”}, p2 \mapsto \text{“a”}, p3 \mapsto \text{“c”}],$
$[p1 \mapsto \text{“c”}, p2 \mapsto \text{“b”}, p3 \mapsto \text{“a”}],$

$[p1 \mapsto$ "c", $p2 \mapsto$ "b", $p3 \mapsto$ "b"$]$,
$[p1 \mapsto$ "c", $p2 \mapsto$ "b", $p3 \mapsto$ "c"$]$,
$[p1 \mapsto$ "c", $p2 \mapsto$ "c", $p3 \mapsto$ "a"$]$,
$[p1 \mapsto$ "c", $p2 \mapsto$ "c", $p3 \mapsto$ "b"$]$,
$[p1 \mapsto$ "c", $p2 \mapsto$ "c", $p3 \mapsto$ "c"$]\}$

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.

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

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

$sing \triangleq \langle\langle 4, 5, 6\rangle, \langle\rangle, \langle\rangle\rangle$

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

* Modification History
* Last modified *Tue Sep* 24 23:06:08 *PDT* 2019 by *jasondebolt*
* Created Sat *Apr* 20 20:01:34 *PDT* 2019 by *jasondebolt*