1 Z Notation Introduction

The following subsections provide a high level overview of select properties of Z Notation based on "The Z Notation: A Reference Manual" by J. M. Spivey. A copy of this reference manual can be found at dave/docs/z/Z-notation reference manual.pdf. In many cases, definitions will be pulled directly from the reference manual and when this occurs, the relevant page number(s) will be included. For a proper introduction with tutorial examples, see chapter 1, "Tutorial Introduction" from pages 1 to 23. For the $LaTeX$ symbols used to write Z, see the reference document found at dave/docs/z/zed-csp-documentation.pdf.

1.1 Decorations

The following decorations are used through this document and are taken directly from the reference manual. For a complete summary of the Syntax of Z, see chapter 6, Syntax Summary, startingon page 142.

$'$	[indicates final state of an operation]
$?$	[indicates input to an operation]
$!$	[indicates output of an operation]
Δ	[indicates the schema results in a change to the state space]
Ξ	[indicates the schema does not result in a change to the state space]
$\gg$	[indicates output of the left schema is input to the right schema]

1.2 Types

Objects have a type which characterizes them and distinguish them from other kinds of objects.

- Basic types are sets of objects which have no internal structure of interest meaning the concrete definition of the members is not relevant, only their shared type.
- Free types are used to describe (potentially nested and/or recursive) sets of objects. In the most simple case, a free type can be an enumeration of constants.

Within the xAPI Formal Specification, both of these types are used to describe the Inverse Functional Identifier property.

- Introduction of the basic types $MBOX$, $MBOX\_SHA1SUM$, $OPENID$ and $ACCOUNT$ allows the specification to talk about these constraints within the xAPI specification without defining their exact structure
- The free type IFI is defined as one of the above basic types meaning an object of type IFI is of type $MBOX$ or $MBOX\_SHA1SUM$ or $OPENID$ or $ACCOUNT$.

Types can be composed together to form composite types and thus complex objects.

$$[MBOX, MBOX\_SHA1SUM, OPENID, ACCOUNT]$$
$$IFI ::= MBOX \mid MBOX\_SHA1SUM \mid OPENID \mid ACCOUNT$$

Within the xAPI Formal Specification, IFI is used within the definition of an *agent* as presented in the schema *Agent*.

$$\begin{array}{|l}
\textit{Agent} \\
\hline
agent : AGENT \\
objectType : OBJECTTYPE \\
name : \mathbb{F}_1 \# 1 \\
ifi : IFI \\
\hline
objectType = Agent \\
agent = \{ifi\} \cup \mathbb{P}\{name, objectType\} \\
\hline
\end{array}$$

See section 2.2, pages 28 to 34, and chapter 3, pages 42 to 85, for more information about Schemas and the Z Language.

1.3 Sets

A collection of elements that all share a type. A set is characterized solely by which objects are members and which are not. Both the order and repetition of objects are ignored. Sets are written in one of two ways:

- listing their elements
- by a property which is characteristic of the elements of the set.

such that the following law from page 55 holds for some object y

$$y \in \{x_1, ..., x_n\} \iff y = x_1 \lor ... \lor y = x_n$$

1.4 Ordered Pairs

Two objects (x, y) where x is paired with y. An n-tuple is the pairing of n objects together such that equality between two n-tuple pairs is given by the law from page 55

$$(x_1, ..., x_n) = (y_1, ..., y_n) \iff x_1 = y_1 \land ... \land x_n = y_n$$

When ordered pairs are used with respect to application (as seen on page 60)

$$fx \Rightarrow f(x) \iff (x, y) \in f$$

which states that $f(x)$ is defined if and only if there is a unique value y which result from fx Additionally, application associates to the left

$$fxy \Rightarrow (fx)y \Rightarrow (f(x), y)$$

meaning $f(x)$ results in a function which is then applied to y.

1.5 Sequences

A collection of elements where their ordering matters such that

$$\langle a_1, ..., a_n \rangle \Rightarrow \{1 \mapsto a_1, ..., n \mapsto a_n\}$$

as seen on page 115. Additionally, iseq is used to describe a sequence whose members are distinct.

1.6 Bags

A collection of elements where the number of times an element appears in the collection is meaningful.

$$[\![a_1, ..., a_n]\!] \Rightarrow \{a_1 \mapsto c_1, ..., a_n \mapsto c_n\}$$

As described on page 124 (replacing c with k), each element a_i appears c_i times in the list $a_1, ..., a_n$ such that the number of occurances of a_i within bag A is returned by

$$count\, A\, a_i \ \equiv A \,\#\, a_i$$

1.7 Maps

This document introduces a named subcategory of sets, *map* of the free type KV, which are akin to sequences and bags. To enumerate the members of a *map*, $\langle\!\langle ... \rangle\!\rangle$ is used but should not be confused with $d_i \langle\!\langle E_i[T] \rangle\!\rangle$ within a Free Type definition. The distinction between the two usages is context dependent but in general, if $\langle\!\langle ... \rangle\!\rangle$ is used outside of a constructor declaration within a Free Type definition, it should be assumed to represent a *map*.

$$KV ::= base \mid associate \langle\!\langle KV \times X \times Y \rangle\!\rangle$$

where

base	[is a constant which is the empty $KV \Rightarrow \langle\!\langle\rangle\!\rangle$]
associate	[is a constructor and is infered to be an injection]

The full enumeration of all properties, constraints and functions specific to a *map* with type KV will be defined elsewhere but *associate* can be understood to (in the most basic case) operate as follows.

$$associate(base, x_i, y_i) = \langle\!\langle (x_i, y_i) \rangle\!\rangle \Rightarrow \langle\!\langle x_i \mapsto y_i \rangle\!\rangle$$

The enumeration of a *map* was chosen to be $\langle\!\langle ... \rangle\!\rangle$ as a *map* is a collection of injections such that if M is the result of $associate(base, x_i, y_i)$ from above then

$$atKey(M, x_i) = y_i \iff x_i \mapsto y_i \wedge (x_i, y_i) \in M$$

1.8 Select Operations

The follow are defined in Chpater 4 (The Mathematical Tool-kit) within the reference manual and are used extensively throughout this document. In many cases, the functions listed here will serve as Operations in the context of Primitives and Algorithms.

1.8.1 Functions

$\nrightarrow$	[relate each x $\in$ X to at most one y $\in$ Y, page 105]
$\rightarrow$	[relate each x $\in$ X to exactly one y $\in$ Y, page 105]
$⤔$	[map different elements of x to different y, page 105]
$\rightarrowtail$	[$⤔$ that are also $\rightarrow$, page 105]
$⤀$	[$X \nrightarrow Y$ where whole of Y is the range, page 105]
$\twoheadrightarrow$	[$X \nrightarrow Y$ whole of X as domain and whole of Y as range, page 105]
$⤖$	[map x $\in$ X one-to-one with y $\in$ Y, page 105]

$$
\begin{array}{l}
X \nrightarrow Y == \{\, f : X \leftrightarrow Y \mid (\forall x : X; y1, y2 : Y \bullet \\
\qquad\qquad\qquad\qquad (x \mapsto y_1 \in f \ \land \ (x \mapsto y_2) \in f \Rightarrow y_1 = y_2))\} \\
X \rightarrow Y == \{\, f : X \nrightarrow Y \mid \operatorname{dom} f = X \} \\
X ⤔ Y == \{\, f : X \nrightarrow Y \mid (\forall x_1, x_2 : \operatorname{dom} f \bullet f(x_1) = f(x_2) \Rightarrow x_1 = x_2)\} \\
X \rightarrowtail Y == (X ⤔ Y) \cap (X \rightarrow Y) \\
X ⤀ Y == \{\, f : X ⤔ Y \mid \operatorname{ran} f = Y \} \\
X \twoheadrightarrow Y == (X ⤀ Y) \cap (X \rightarrow Y) \\
X ⤖ Y == (X \twoheadrightarrow Y) \cap (X \rightarrowtail Y)
\end{array}
$$

1.8.2 Ordered Pairs, Maplet and Composition of Relations

first	[returns the first element of an ordered pair, page 93]
second	[returns the second element of an ordered pair, page 93]
$\mapsto$	[maplet is a graphic way of expressing an ordered pair, page 95]
dom	[set of all x $\in$ X related to atleast one y $\in$ Y by R, page 96]
ran	[set of all y $\in$ Y related to atleast one x $\in$ X by R, page 96]
$⨾$	[The composition of two relationships, page 97]
$\circ$	[The backward composition of two relationships, page 97]

$$
\begin{array}{|l}
[X, Y] \\
\hline
first : X \times Y \rightarrow X \\
second : X \times Y \rightarrow Y \\
\hline
\forall x : X; y : Y \bullet \\
\qquad first(x, y) = x \land \\
\qquad second(x, y) = y \\
\hline
\end{array}
$$

$$
\begin{array}{l}
[X, Y] \\
\hline
\_ \mapsto \_ : X \times Y \rightarrow X \times Y \\
\hline
\forall x : X; y : Y \bullet \\
x \mapsto y = (x, y)
\end{array}
$$

$$
\begin{array}{l}
[X, Y] \\
\hline
\mathrm{dom} : (X \leftrightarrow Y) \rightarrow \mathbb{P}\, X \\
\mathrm{ran} : (X \leftrightarrow Y) \rightarrow \mathbb{P}\, Y \\
\hline
\forall R : X \leftrightarrow Y \bullet \\
\qquad \mathrm{dom}\, R = \{\, x : X;\ y : Y \mid x \underline{R} y \bullet x \} \wedge \\
\qquad \mathrm{ran}\, R = \{\, x : X;\ y : Y \mid x \underline{R} y \bullet y \}
\end{array}
$$

$$
\begin{array}{l}
[X, Y, Z] \\
\hline
\_ \mathbin{⨾} \_ : (X \leftrightarrow Y) \times (Y \leftrightarrow Z) \rightarrow (X \leftrightarrow Z) \\
\_ \circ \_ : (Y \leftrightarrow X) \times (X \leftrightarrow Y) \rightarrow (X \leftrightarrow X) \\
\hline
\forall\, Q : X \leftrightarrow Y; R : Y \leftrightarrow Z \bullet \\
\qquad Q \mathbin{⨾} R = R \circ Q = \{\, x : X; y : Y; z : Z \mid \\
\qquad\qquad\qquad\qquad x\, \underline{Q}\, y \ \wedge\ y\, \underline{R}\, z \bullet x \mapsto z \}
\end{array}
$$

1.8.3 Numeric

succ	[the next natural number, page 109]
..	[set of integers within a range, page 109]
#	[number of members of a set, page 111]
min	[smallest number in a set of numbers, page 113]
max	[largest number in a set of numbers, page 113]

$$
\begin{array}{l}
succ : \mathbb{N} \rightarrow \mathbb{N} \\
\_ \mathinner{..} \_ : \mathbb{Z} \times \mathbb{Z} \rightarrow \mathbb{P}\, \mathbb{Z} \\
\hline
\forall n : \mathbb{N} \bullet succ(n) = n + 1 \\
forall a, b : \mathbb{Z} \bullet \\
\qquad\qquad a \mathinner{..} b = \{\, k : \mathbb{Z} \mid a \leq k \leq b \}
\end{array}
$$

$$
\begin{array}{l}
[X] \\
\hline
\# : \mathbb{F}\, X \rightarrow \mathbb{N} \\
\hline
\forall S : \mathbb{F}\, X \bullet \\
\qquad \# S = (\mu\, n : \mathbb{N} \mid (\exists f : 1 \mathinner{..} n \rightarrowtail X \bullet \mathrm{ran}\, f = S))
\end{array}
$$

$$
\begin{array}{|l}
\min : \mathbb{P}_1 \mathbb{Z} \nrightarrow \mathbb{Z} \\
\max : \mathbb{P}_1 \mathbb{Z} \nrightarrow \mathbb{Z} \\
\hline
\min = \{ S : \mathbb{P}_1 \mathbb{Z}; m : \mathbb{Z} \mid \\
\qquad m \in S \land (\forall n : S \bullet m \leq n) \bullet S \mapsto m \} \\
\max = \{ S : \mathbb{P}_1 \mathbb{Z}; m : \mathbb{Z} \mid \\
\qquad m \in S \land (\forall n : S \bullet m \geq n) \bullet S \mapsto m \}
\end{array}
$$

1.8.4 Sequences

$\frown$	[concatenation of two sequences, page 116]
rev	[reverse a sequence, page 116]
head	[first element of a sequence, page 117]
last	[last element of a sequence, page 117]
tail	[all elements of a sequence except for the first, page 117]
front	[all elements of a sequence except for the last, page 117]
$\upharpoonleft$	[sub seq based on provided indices, order maintained, page 118]
$\upharpoonright$	[sub seq based on provided condition, order maintained, page 118]
squash	[compacts a fn of positive integers into a sequence, page 118]
$\frown/$	[flatten seq of seqs into single seq, page 121]
disjoint	[pairs of sets in family have empty intersection, page 122]
partition	[union of all pairs of sets = the family set, page 122]

$$
\begin{array}{|l}
[X] \\
\hline
\_ \frown \_ : \mathrm{seq}\, X \times \mathrm{seq}\, X \rightarrow \mathrm{seq}\, X \\
rev : \mathrm{seq}\, X \rightarrow \mathrm{seq}\, X \\
\hline
\forall s, t : \mathrm{seq}\, X \bullet \\
\qquad s \frown t = s \cup \{ n : \mathrm{dom}\, t \bullet n + \# s \mapsto t(n) \} \\
\forall s : \mathrm{seq}\, X \bullet \\
\qquad rev s = (\lambda\, n : \mathrm{dom}\, s \bullet s(\# s - n + 1)) \\
\hline
\end{array}
$$

$$
\begin{array}{|l}
[X] \\
\hline
head, last : \mathrm{seq}_1\, X \rightarrow X \\
tail, front : \mathrm{seq}_1\, X \rightarrow \mathrm{seq}\, X \\
\hline
\forall s : \mathrm{seq}_1\, X \bullet \\
\qquad head\, s = s(1) \land \\
\qquad last\, s = s(\# s) \land \\
\qquad tail\, s = (\lambda\, n : 1 \mathrel{..} \# s - 1 \bullet s(n + 1)) \land \\
\qquad front\, s = (1 \mathrel{..} \# s - 1) \lhd s \\
\hline
\end{array}
$$

$$
\begin{array}{|l}
[X] \\
\hline
\_ \upharpoonleft \_ : \mathbb{P}\,\mathbb{N}_1 \times \operatorname{seq} X \rightarrow \operatorname{seq} X \\
\_ \restriction \_ : \operatorname{seq} X \times \mathbb{P}\,X \rightarrow \operatorname{seq} X \\
squash : (\mathbb{N}_1 \rightarrowtail X) \rightarrow \operatorname{seq} X \\
\hline
\forall U : \mathbb{P}\,\mathbb{N}_1; s : \operatorname{seq} X \bullet \\
\quad U \upharpoonleft s = squash(U \lhd s) \\
\forall s : \operatorname{seq} X; V : \mathbb{P}\,X \bullet \\
\quad s \restriction V = squash(s \rhd V) \\
\forall f : \mathbb{N}_1 \rightarrowtail X \bullet \\
\quad squash f = f \circ (\mu\, p : 1 \mathrel{..} \# f \rightarrowtail\!\!\!\rightarrow \operatorname{dom} f \mid p \circ succ \circ p^{\sim} \subseteq (\_ < \_)) \\
\hline
\end{array}
$$

$$
\begin{array}{|l}
[X] \\
\hline
{}^\frown/ : \operatorname{seq}(\operatorname{seq} X) \rightarrow \operatorname{seq} X \\
\hline
{}^\frown/\langle\rangle = \langle\rangle \\
\forall s : \operatorname{seq} X \bullet {}^\frown/\langle s \rangle = s \\
\forall q, r : \operatorname{seq}(\operatorname{seq} X) \bullet \\
\quad {}^\frown/(q \frown r) = ({}^\frown/\, q) \frown ({}^\frown/\, r) \\
\hline
\end{array}
$$

$$
\begin{array}{|l}
[I, X] \\
\hline
\mathsf{disjoint}\ \_ : \mathbb{P}(I \rightarrowtail \mathbb{P}\,X) \\
\_\ \mathsf{partition}\ \_ : (I \rightarrowtail \mathbb{P}\,X) \leftrightarrow \mathbb{P}\,X \\
\hline
\forall S : I \rightarrowtail \mathbb{P}\,X; T : \mathbb{P}\,X \bullet \\
\quad (\mathsf{disjoint}\ S \iff \\
\qquad (\forall i, j : \operatorname{dom} S \mid i \neq j \bullet S(i) \cap S(j) = \emptyset)) \wedge \\
\quad (S\ \mathsf{partition}\ T \iff \\
\qquad \mathsf{disjoint}\ S \wedge \bigcup\{\, i : \operatorname{dom} S \bullet S(i)\} = T) \\
\hline
\end{array}
$$

1.8.5 Bags

$count, \#$	[the number of times something appears in a bag, page 124]
$\otimes$	[scaling across a bag, page 124]
$\uplus$	[union of two bags, sum of occurances, page 126]
$\mathbin{\cup\!\!\!\!-}$	[bag difference, subtract occurances or zero if negative, page 126]
$items$	[converstion from seq to bag, page 127]

$$
\begin{array}{l}
[X] \\
\hline
count : \mathrm{bag}\, X \rightarrowtail (X \rightarrow \mathbb{N}) \\
\_ \# \_ : \mathrm{bag}\, X \times X \rightarrow \mathbb{N} \\
\_ \otimes \_ : \mathbb{N} \times \mathrm{bag}\, X \rightarrow \mathrm{bag}\, X \\
\hline
\forall B : \mathrm{bag}\, X \bullet \\
\qquad countB = (\lambda x : X \bullet 0) \oplus B \\
\forall x : X; B : \mathrm{bag}\, x \bullet \\
\qquad B \# x = count\, B\, x \\
\forall n : \mathbb{N}; B : \mathrm{bag}\, X; x : X \bullet \\
\qquad (n \otimes B) \# x = n * (B \# x)
\end{array}
$$

$$
\begin{array}{l}
[X] \\
\hline
\_ \uplus \_, \_ \cup\!\!\!- \_ : \mathrm{bag}\, X \times \mathrm{bag}\, X \rightarrow \mathrm{bag}\, X \\
\hline
\forall B, C : \mathrm{bag}\, X; x : X \bullet \\
\qquad (B \uplus C) \# x = B \# x + C \# x \land \\
\qquad (B \cup\!\!\!- C) \# x = \max\{B \# x - C \# x, 0\}
\end{array}
$$

$$
\begin{array}{l}
[X] \\
\hline
items : \mathrm{seq}\, X \rightarrow \mathrm{bag}\, X \\
\hline
\forall s : \mathrm{seq}\, X; x : X \bullet \\
\qquad (items\, s) \# x = \#\{ i : \mathrm{dom}\, s \mid s(i) = x \}
\end{array}
$$