Question Name

intro text for the question

1 Ideal Statements

paragraph or list describing the ideal input statements

1.1 statement parameters to utilize

- first param
- second param
- third param

2 TLA Statement problems

paragraph talking about known data issues within current TLA implementation

3 Algorithm

3.1 Summary

1. step 1
2. step 2
3. step 3

3.2 Symbol Definition

Definition of a query to an LRS, updated as needed

```
Agent = ”agent={”account”:
                 {”homePage”: ”https://example.homepage”,
                  ”name”: 123456}}”

Since = ”since=2018-07-20T12:08:47Z”

Until = ”until=2018-07-21T12:08:47Z”

Base = ”https://example.endpoint/statements?”

endpoint = Base + Agent + ”&” + Since + ”&” + Until
```

```
Auth = Hash generated from basic auth

S = curl -X GET -H ”Authorization: Auth”
         -H ”Content-Type: application/json”
         -H ”X-Experience-API-Version: 1.0.3”
         Endpoint
```

3.3 Z Specifications

Outline of Z, includes templates and an example of a system used to check staff members in and out of a building

3.3.1 xAPI Statement(s) Schema

[Statement] [Actor] [Verb] [Object] [Result] [Context] [Timestamp]

Statement

$$s : Statement$$

$$\begin{aligned} s = {} & \{Actor, Verb, Object, Timestamp\} \vee \\ & \{Actor, Verb, Object, Timestamp, Context\} \vee \\ & \{Actor, Verb, Object, Timestamp, Result\} \vee \\ & \{Actor, Verb, Object, Timestamp, Result, Context\} \end{aligned}$$

- The variable s is of type Statement and consists of an Actor, Verb, Object, Timestamp and optionally Context and Result

Statements

$$S : Statements$$

$$S = \{\, s : Statement \mid S \neg \emptyset \}$$

- The variable S is of type Statements and is a set of objects s, each of type Statement
- The variable S is a non empty set

3.3.2 Introduce Basic Types

Template [Name of variable(s) of type set]

Example [X]

3.3.3 Example Schema

Basic unit of specification, defines state variables, system state, operations, etc.

Template

$$\begin{array}{|l}\hline \mathit{SchemaName} \\ \mathit{VariableDeclarations} \\ \hline \mathit{Predicate/Invariants} \\ \hline \end{array}$$

Example

$$\begin{array}{|l}\hline \mathit{Counter} \\ \mathit{ctx} : \mathbb{N} \\ \hline 0 \leq \mathit{ctr} \leq \mathit{max} \\ \hline \end{array}$$

Variables

$$\begin{array}{|l}\hline \mathit{Counter} \\ \mathit{ctx} : \mathbb{N} \\ \hline \\ \hline \end{array}$$

- the variable ctx is a natural number

Predicates

$$\begin{array}{|l}\hline \mathit{Counter} \\ \hline 0 \leq \mathit{ctr} \leq \mathit{max} \\ \hline \end{array}$$

- ctr is greater than or equal to 0
- ctr is less than or equal to max

3.3.4 Initialisation

The starting conditions

Template

$$\begin{array}{|l}\hline \mathit{Init}[\mathit{VarName}] \\ \mathit{NameOfExistingSchema} \\ \hline \mathit{InitStateOfVarsWithinRefSchema} \\ \hline \end{array}$$

Example

$$\begin{array}{|l}
\multicolumn{1}{l}{\mathit{InitCounter}} \\ \hline
\mathit{Counter} \\ \hline
\mathit{ctr} = 0 \\ \hline
\end{array}$$

- the value of the counter starts at 0

3.3.5 Operations

an operation is specified in Z with a predicate relating the state before and after the invocation of that operation

Template

$$\begin{array}{|l}
\multicolumn{1}{l}{\mathit{OperationName}} \\ \hline
\Delta \mathit{SchemaName} \\
\mathit{inputParam}? : \mathit{SomeType} \\
\mathit{outputParam}! : \mathit{SomeType} \\ \hline
\mathit{InvariantPredicate} \\
NewValForVar' = \mathit{OperationOnInput}/\mathit{OutputParams} \\ \hline
\end{array}$$

Example

$$\begin{array}{|l}
\multicolumn{1}{l}{\mathit{Increment}} \\ \hline
\Delta \mathit{Counter} \\ \hline
\mathit{ctr} < \mathit{max} \\
\mathit{ctr}' = \mathit{crt} + 1 \\ \hline
\end{array}$$

- There is an implicit conjunction (logical-and) between successive lines of the predicate

$$\begin{array}{|l}
\multicolumn{1}{l}{\mathit{Decrement}} \\ \hline
\Delta \mathit{Counter} \\
d? : \mathbb{N} \\ \hline
\mathit{ctr} \geq d? \\
\mathit{ctr}' = \mathit{ctr} - d? \\ \hline
\end{array}$$

- input params suffixed with ?

Display

$\Xi Counter$

$c! : \mathbb{N}$

$c! = ctr$

- output params suffixed with !
- the greek symbol means that the operation cannot change the state of Counter

3.4 Pseudocode

Algorithm 1: How to write algorithms

```
Input: this text
Result: how to write algorithm with LaTeX2e
initialization;
while not at end of this document do
    read current;
    if understand then
        go to next section;
        current section becomes this one;
    else
        go back to the beginning of current section;
    end
end
```

3.5 Result JSON Schema

3.6 Visualization Description

description of the associated visualization in english

3.7 VEGA example

This section will be updated to include a VEGA JSON blob for prototype viz