THE PUBLIC IS MORE FAMILIAR WITH BAD DESIGN THAN GOOD DESIGN. IT IS, IN EFFECT, CONDITIONED TO PREFER BAD DESIGN, BECAUSE THAT IS WHAT IT LIVES WITH. THE NEW BECOMES THREATENING, THE OLD REASSURING.

PAUL RAND

A DESIGNER KNOWS THAT HE HAS ACHIEVED PERFECTION NOT WHEN THERE IS NOTHING LEFT TO ADD, BUT WHEN THERE IS NOTHING LEFT TO TAKE AWAY.

ANTOINE DE SAINT-EXUPÉRY

. . . THE DESIGNER OF A NEW SYSTEM MUST NOT ONLY BE THE IMPLEMENTOR AND THE FIRST LARGE-SCALE USER; THE DESIGNER SHOULD ALSO WRITE THE FIRST USER MANUAL. . . IF I HAD NOT PARTICIPATED FULLY IN ALL THESE ACTIVITIES, LITERALLY HUNDREDS OF IMPROVEMENTS WOULD NEVER HAVE BEEN MADE, BECAUSE I WOULD NEVER HAVE THOUGHT OF THEM OR PERCEIVED WHY THEY WERE IMPORTANT.

DONALD E. KNUTH

HENGFENG WEI

NOTES ON TLA+

ANT

PUBLISHED BY ANT

TUFTE-LATEX.GOOGLECODE.COM

First printing, August 2018

Contents

List of Figures

List of Tables

Dedicated to those who appreciate LaTeX

and the work of Edward R. Tufte and Donald E. Knuth.

Introduction

Notes on TLA+.

Experiences

1. Use “ASSUME” for “Constants”
2. Decompose a large specification into several modules
3. Test-driven

Debug

`Print`

Description: Print Twice.

Effect: The Print may be executed twice (unexpectedly).

Cause: The reason is that "TLC prints its output once during the generation of the state space and then again when reconstructing the counter-example trace that leads to the violation."

Solution:

Reference: Unexpected `Print` Output @ tlaplus-googlegroup.

Parse

Model Checking

Description: Model is Stuck in the 'modelchecking' State.

Effect: I run the model, it gets stuck in a "modelchecking" state. I can't stop or restart the model checker. I have to restart the entire toolbox to reset the model's state.

Cause: The model is wrong. For example, use 'a'.

Solution: Use "a".

Reference: Model is stuck in 'modelchecking' state @ tlaplus-googlegroup.

Effect: I run the model, it gets stuck in a "modelchecking" state. I can't stop or restart the model checker. I have to restart the entire toolbox to reset the model's state. *A .tla source file which is included by "EXTENDS" is missing in the Model folder.*

Cause: The project structure is broken.

Solution:

TLA+

UNCHANGED

var1 == «a, b»
var2 == «c, d»
var == var1 øvar2
Error: “a” is either undefined or not an operator.
Solution: Use var == «var1, var2».

Model

Question from hwayne on May 28, 2017:

Lamport defines a memory cache with the operator $NoVal \triangleq CHOOSE\ v : v \notin Val$. He explicitly calls this best practice, because adding a ‘NOVAL‘ constant would be making the model more complex. Except unbounded CHOOSE is not checkable by TLC! While ‘NoVal‘ is nice for abstract specifications, it breaks things in the 99% of cases where you’re actually using TLA+.

Answer from pron on May 29, 2017:

You definitely should write $NoVal \triangleq CHOOSE\ v : v \notin Val$ in your spec (as that clarifies the intent) and override it as a "model value" (i.e., a constant) in TLC. This is best practice. Even $\exists x \in Nat$ is not checkable by TLC, yet best practice is to leave it like that (as that is the intent), and override ‘Nat‘ in TLC. In either case, the more general specification is verifiable in TLAPS.

Bibliography