APPENDIX

PROOF MECHANIZATION

Mechanization of the proof is done in HOL4 theorem prover [21] using the context-free grammar model of [29]. HOL4 is a LCF-style [30] proof assistant for Hight-Order-Logic built on a minimal proof kernel which implements the axioms and basic inference rules. HOL4 uses the ML type system to force all proofs to pass its logical kernel. This prevents proving false statements, thus lending high trustworthiness to the verification exercise conducted in HOL4.

Using the context-free grammar model of [29], a grammar is defined using the following HOL4 type definitions:

```
('nts, 'ts) symbol = NTS of 'nts | TS of 'ts
('nts, 'ts) rule = rule of 'nts → ('nts,'ts) symbol list
('nts, 'ts) grammar = G of ('nts,'ts) rule list →' nts
```

Where "$'nts$" and "$'ts$" represent nonterminal and terminal symbols respectively, and $\rightarrow$ denotes curried arguments to an algebraic type constructor. Therefor, $rule$ is a curried function which takes a nonterminal as the head of the rule and a list of symbols which the head is mapped to, and it returns a $('nts,\ 'ts)$ `rule`. Similarly, a grammar is composed of a list of rules and a nonterminal which represents the grammar start symbol.

Definition (*derives*) A list of symbols (or sentential form) $\boldsymbol{s}$ of the form $\alpha A \gamma$ derives $\boldsymbol{t}$ of the form $\alpha\beta\gamma$ in a single step, if $A \mapsto \beta$ is one of the rules in the grammar:

```
derives g lsl rls :=
  ∃ s1 s2 rhs lhs.
    s1 ++ [NTS lhs] ++ s2 = lsl) ∧
    (s1 ++ rhs ++ s2 = rsl) ∧
    rule lhs rhs ∈ rules g
```

In the definition above $+\!\!+$ denotes list concatenation and it is used to represent concatenation of symbols, e.g., $\alpha A \gamma \equiv \alpha +\!\!+ A +\!\!+ \gamma$. The *reflexive transitive closure* of the relation `derive` is denoted by `(derive g)* sf1 sf2` and it means that $\boldsymbol{sf_2}$ is derivable from $\boldsymbol{sf_1}$ in zero or more steps.

Say $\boldsymbol{x}$ is a fault and `containAtLeastOne` is the function which for a given string (i.e. list of terminals) checks if $\boldsymbol{x}$ is contained in the string. Then the following Lemma shows that all string in the language of the grammar $\boldsymbol{g}$ contain at least one fault ($\boldsymbol{x}$).

Lemma (*At Least One Fault*) Say $\boldsymbol{g}$ is a context-free grammar and $\boldsymbol{s_1 + s_3 + s_2}$ is derivable from the start symbol of $\boldsymbol{g}$, then if all strings derivable from $\boldsymbol{s_3}$ is guaranteed to always contain at least one fault $\boldsymbol{x}$ then all strings in the language of $\boldsymbol{g}$ contain at least one fault. That is:

```
∀ g s1 s2 s3 x .
  (derives g)* [NTS (startSym g)] (s1 ++ s3 ++ s2) ⟹
  (∀w3. (derives g)*s3 w3 ⟹
        containAtLeastOne w3 x)      ⟹
  (∀w. w ∈ language g                 ⟹
        (derives g)* (s1 ++ s3 ++ s2) w ⟹
        containAtLeastOne w x )
```

Proof of the Lemma is Straight forward. We refer the interested readers to check our Git repository (https://github.com/anonymous-ewok/anonymous-ewok.github.io) for the complete list of theorems which we have proved.