# Extending the TPTP Language to Higher-Order Logic with Automated Parser Generation

## System Description


Allen Van Gelder
Computer Science Department
University of California
Santa Cruz, CA, USA

avg@cs.ucsc.edu

Geoff Sutcliffe
Computer Science Department
University of Miami
Miami, FL, USA

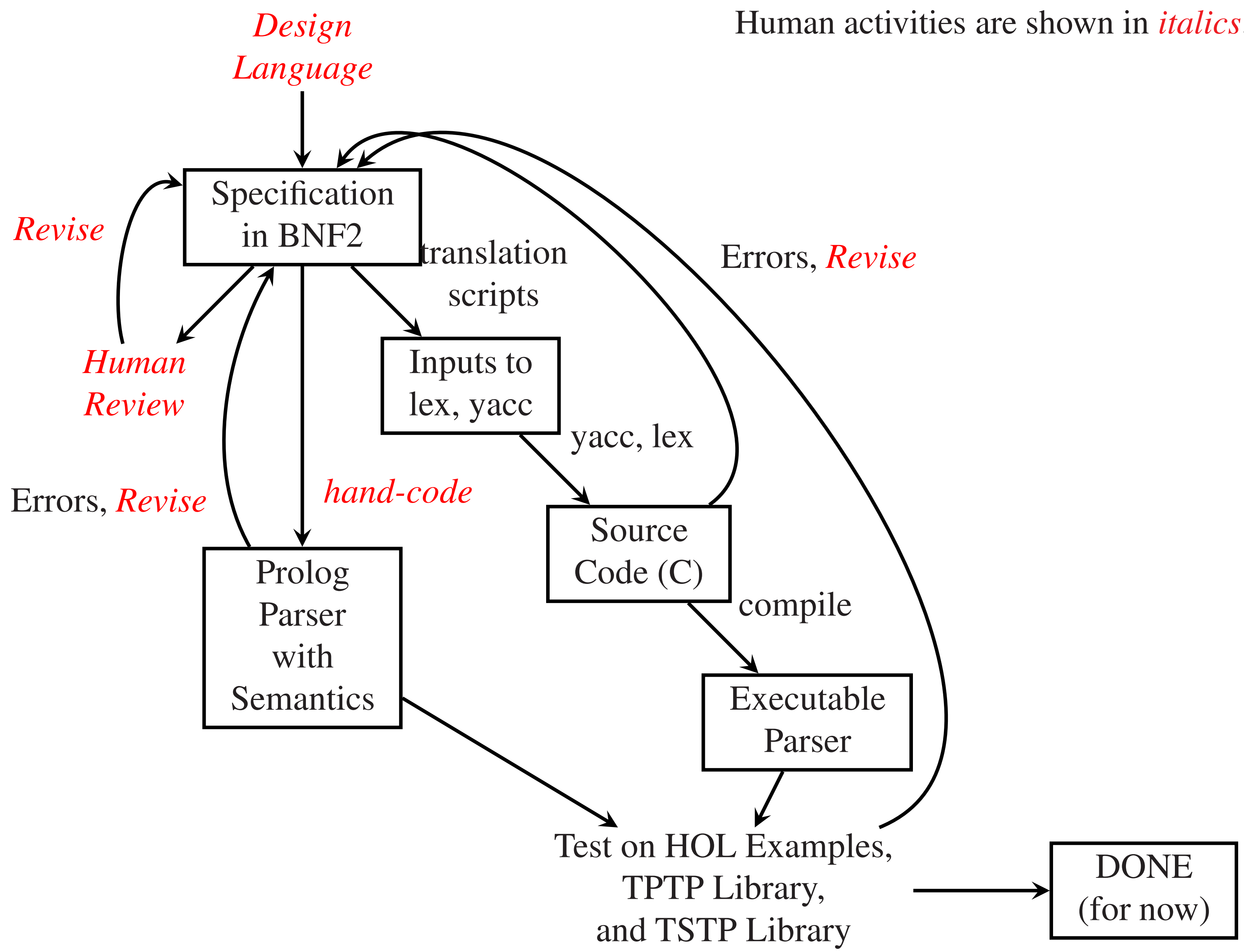
geoff@cs.miami.edu


http://www.cse.ucsc.edu/∼avg/TPTPparser/

http://www.cs.miami.edu/∼tptp/

# Overview of the System to Evaluate a Proposed HOTPTP Language

Human activities are shown in *italics*.



## BNF2 – A Two-Level Specification Language

Traditional BNF (Backus-Naur Form) has one syntax for all grammar symbols.

Modern practice uses two levels of symbols

- *tokens* are defined by regular expressions, and
  are treated as terminal symbols in the context-free part of the grammar.
- *grammar symbols* are the nonterminal symbols of the context-free grammar.

<table>
<tr><th>Meta-<br>Symbol</th><th>Rule<br>Type</th><th>Examples (some are simplified from the TPTP language)</th><th></th><th></th></tr>
<tr><td>::=</td><td>Grammar</td><td><TPTP input><br><nonassoc op><br><formula role></td><td>::=<br>::=<br>::=</td><td><annotated formula> | <comment><br><=> | => | <= | <~><br><lower word></td></tr>
<tr><td>:==</td><td>Semantic</td><td><formula role></td><td>:==</td><td>axiom | conjecture | lemma |<br>theorem | negated_conjecture</td></tr>
<tr><td>::-</td><td>Token</td><td><lower word></td><td>::-</td><td><lower><alphanum>*</td></tr>
<tr><td>::=</td><td>Macro</td><td><lower><br><alphanum></td><td>:::<br>:::</td><td>[a-z]<br>[A-Za-z0-9_]</td></tr>
</table>

## Proposed HOTPTP Syntax, Extending TPTP to Higher-Order Logic

New operators continue TPTP tradition that operators are logical symbols:

- Lambda abstraction: `^` (synonym `lambda`)
- Function application: `@`
- "Maps to" for types: `->` (synonym `>`)
- "Defined as": `:=` (Hudak [1989] uses $\equiv$ which looks reflexive.)

Colon operator (`:`) has several new meanings.

```
hof(1, definition,
    set_union := lambda [A: $type]: lambda [D: ((A-> $o)-> $o), X: A]:
                 ? [S: (A-> $o)]: ( (D @ S) & (S @ X) ) ).
```

The lambda expression shown is:

$\lambda A{:}\tau.\ \lambda D{:}((A \rightarrow o) \rightarrow o).\ \lambda X{:}A.\ \exists S{:}(A \rightarrow o).\ ((D\ @\ S) \wedge (S\ @\ X)).$

More details and examples at http://www.cse.ucsc.edu/∼avg/TPTPparser/.

To be discussed at TPTP Tea Party.