The Welkin Standard

Oscar Bender-Stone

Abstract

Welkin is an information language. Welkin stores three independent structures: a tree of nodes, a hypergraph between nodes, and a tree of node labels. An Information Graph has a unique encoding. Using this encoding, the original Information Graph may be recovered. This document "bootstraps" Welkin to provide a finitistic basis for all information.

Syntax

Terminals

- Logic
- Symbols (1): $0, 1$
- Concatenation $\cdot$.
- Implication $\Rightarrow$
- Table of US-ASCII:
- A **word** is recursively defined.
  - Base case (5):
    - 0 is a word.
    - 1 is a word.
  - Recursion (6): let w be a word.
    - $w \cdot 0$ is a word.
    - $w \cdot 1$ is a word.

Atoms

- Strings are words with delimiters: d_1.w.d_2, where $d_1 \not\subset w$ and $d_2 \not\subset w$.
- Identifiers are strings without white space.
- Numbers are a subset of strings with an injective function $q : \text{NUMBER} \to Q$.
  - Q is set of strings

$$\frac{p}{q}$$

  where p, q are in scientific notation.

Grammar

- LALR
  - Not ambiguous
- Welkin Grammar:

Semantics

Equality on Terms

- Two strings are equal if they contain the same strings, in order.
- Two numbers are equal if $q(a) = q(b)$.

Valid Strings

- No relative members at toplevel (with length 2).
- No duplicate members, graphs, or connections.

Welkin Information Graphs

A **Welkin Information Graph (WIG)** is a structure $G = (T, H, L)$ with:

- A tree T,
- A hypergraph H,
- A tree L isomorphic to T called the **label tree.**

AST ()

- Units:
- Members are words of units
- Connections are WIGs with
- Graphs are WIGs with
  - Derived terms as children
  - Ordered triples are arcs.

Encoding

The **encoding** $E(G)$ of the WIG G is the unique string where

- All nodes are listed in breadth-first order
- Leaves are terms ending with "#"
- Edges are enumerated, starting from 0. They are included in nodes:
  - s means source,
  - c means connector,
  - t means target.

Bootstrap

Theorem. The Bootstrap File (Appendix A) has the encoding

.

We prove this in the following calculations:

$$(1) 0, 1 \Rightarrow \{0, 1\}$$

$$(3)\ \text{start} - \{0, 1\} \to \text{word} \Rightarrow (\text{start}, \{0, 1\}, \text{word})$$

Appendix A: Boostrap File