Path Semantics in Concept Grammar Language

by Sven Nilsen, 2016

Abstract:
In this paper I formalize some aspects of deriving normal programming from path semantics using the Concept Grammar Language[1].

Path semantics is a formal language which concept grammar is constructed from symbols `F` and `X` according to an axiom of equality:

$$\frac{F_0 X_0, F_1 X_1, F_0 = F_1, F_0 > X_0}{X_0 = X_1}$$

- `F=F`, `X=X` are weak equalities and `F>X` is a weak partial order
- "weak" means encoded and interpreted in some external language[1]
- Indices are local and chosen for a single sentence
- Matching is decided against some syntactic structure

Using the advanced index notation of Concept Grammar Language the axiom looks like this:

$$\{ \{F_i\ X_i,\ F_j\ X_j,\ F_i = F_j,\ F_i > X_i\} \rightarrow \{X_i = X_j\} \}^{ij}$$

Normal programming languages do have path semantics in the strict sense, but not directly according to the axiom of equality. When somebody says "using path semantics" they refer to extending a language with "paths":

$[F]\ X_1 \rightarrow (F(X_0) \rightarrow X_1)$ (path)
$F_0([F_1]\ X) \rightarrow [F_2]\ X$ (path between functions)

References

[1] Concept Grammar Language (Sven Nilsen, 2016)