The Semantics of K

Formal Systems Laboratory
University of Illinois

July 25, 2017

Please feel free to contribute to this report in all ways. You could add new contents, remove redundant ones, refactor and organize the texts, and correct typos.

Definition 1 (Matching Logic Theory)**.** A matching logic theory (S, Σ, A) is a triple that contains a nonempty finite set of sorts, a finite set of symbols, and a recursive set of axioms. Two theories are *equal* if they have the same set of sorts and symbols, and they deduce the same set of theorems.

Example 2. We use serif fonts to denote matching logic theories. Some of the commonly used ones are the theory (theories) of definedness DEF, the theory of Presburger arithmetic PA, the theory of sequences of natural numbers SEQ, the theory of memory heaps HEAP, the theory of IMP programs IMP, the theory (theories) of fixed-points FIX, and the theory (theories) of contexts CTXT.

Definition 3 (The Kore Language)**. We haven't come to an agreement on the syntax of the Kore language yet. One could, though, refer to the Kore text representation at the wiki page at kframework repos on Github, whose link I cannot find, which is considered as the first step towards that direction.**
The Kore language is a language to write matching logic theories. The outcomes are called Kore definitions. Kore definitions are mainly served as the interface between a K frontend and a K backend, but a human should be able to read and write Kore definitions of simple theories, too. The Kore language is designed in a way that:

- Every Kore definition defines exactly one matching logic theory;

- Every matching logic theory can be defined as a Kore definition;

- There is no parsing ambiguity.

- The least amount of inferring is needed;

- And more . . .

Definition 4 (Frontend)**.** A K frontend is an artifact that generates Kore definitions.

Definition 5 (Backend). A K backend is an artifact that consumes a Kore definition of a theory T and does some work. Whatever it does can and should be algorithmically reduced to the task of proving $\mathsf{T} \vdash \varphi$ where φ "encodes" that work. A K backend should justify its results by generating formal proofs that can be proof-checked by the oracle matching logic proof checker.

Object-level and meta-level It is an aspect of life in mathematical logics to distinguish the *object-level* and *meta-level* concepts. The basic principle is to use serif fonts letters ($\mathsf{Nat}, \mathsf{x : Nat}, \mathsf{plus(0, x : Nat)}$) for object-level concepts and normal math fonts letters for meta-level concepts ($s, x : s, \sigma(\sigma_0, x : s)$).

Variables and metavariables for variables This paragraph includes some discussions and observations about the usage of variables and metavariables in writing matching logic theories and Kore definitions. For any matching logic theory $\mathsf{T} = (S, \Sigma, A)$, it comes for each sort $s \in S$ a countably infinite set V_s of variables. We use $\mathsf{x : s}, \mathsf{y : s}, \mathsf{z : s}, \ldots$ for variables in V_s, and omit their sorts when that is clear from the contexts. Different sorts have disjoint sets of variables, so $V_s \cap V_{s'} = \emptyset$ if $s \neq s'$.

Proposition 6. *Let A be a set of axioms and $\bar{A} = \forall A$ be the universal quantification closure of A, then for any pattern φ, $A \vdash \varphi$ iff $\bar{A} \vdash \varphi$.*

Remark 7 (Free variables in axioms). The free variables appearing in the axioms of a theory can be regarded as implicitly universal quantified, because a theory and its universal quantification closure are equal.

Example 8.

$$A_1 = \{\mathsf{mult(x, 0) = 0}\}$$
$$A_2 = \{\forall \mathsf{x.mult(x, 0) = 0}\}$$
$$A_3 = \{\forall \mathsf{y.mult(y, 0) = 0}\}$$
$$A_4 = \{\mathsf{mult}(x, 0) = 0\}$$
$$A_5 = \{\forall x.\mathsf{mult}(x, 0) = 0\}$$
$$A_6 = \{\forall y.\mathsf{mult}(y, 0) = 0\}$$
$$A_7 = \{\mathsf{mult(x, 0) = 0, mult(y, 0) = 0, mult(z, 0) = 0}, \ldots\}$$
$$A_8 = \{\forall \mathsf{x.mult(x, 0) = 0}, \forall \mathsf{y.mult(y, 0) = 0}, \forall \mathsf{z.mult(z, 0) = 0}, \ldots\}$$

All the eight theories are equal. Theories A_4, A_5, A_6 are finite representations of theories A_7, A_8, A_8 respectively.

Rewriting logic

Contexts

Fixed points

Lambda Calculus