# The IFF Metashell

Robert E. Kent

February 6, 2006

$$
\begin{tikzcd}
 & \mathsf{function} & \mathsf{set} & \mathsf{relation} & \\
\mathsf{generic} & \mathsf{ftn}_\infty \arrow[r, shift left, "\partial_0^\infty"] \arrow[r, shift right, "\partial_1^\infty"'] \arrow[dr, phantom, "\llcorner"] & \mathsf{set}_\infty \arrow[dr, phantom, "\lrcorner"] & \mathsf{rel}_\infty \arrow[l, shift right, "\sigma_0^\infty"'] \arrow[l, shift left, "\sigma_1^\infty"] & \infty = \mathtt{ur} \\
 & \vdots \arrow[u, hook] & \vdots \arrow[u, hook] & \vdots \arrow[u, hook] & \vdots \\
\mathsf{very\ large} & \mathsf{ftn}_3 \arrow[u, hook] \arrow[r, shift left, "\partial_0^3"] \arrow[r, shift right, "\partial_1^3"'] \arrow[dr, phantom, "\llcorner"] & \mathsf{set}_3 \arrow[u, hook] \arrow[dr, phantom, "\lrcorner"] & \mathsf{rel}_3 \arrow[u, hook] \arrow[l, shift right, "\sigma_0^3"'] \arrow[l, shift left, "\sigma_1^3"] & 3 = \mathtt{vlrg} \\
\mathsf{large} & \mathsf{ftn}_2 \arrow[u, hook] \arrow[r, shift left, "\partial_0^2"] \arrow[r, shift right, "\partial_1^2"'] \arrow[dr, phantom, "\llcorner"] & \mathsf{set}_2 \arrow[u, hook] \arrow[dr, phantom, "\lrcorner"] & \mathsf{rel}_2 \arrow[u, hook] \arrow[l, shift right, "\sigma_0^2"'] \arrow[l, shift left, "\sigma_1^2"] & 2 = \mathtt{lrg} \\
\mathsf{small} & \mathsf{ftn}_1 \arrow[u, hook] \arrow[r, shift left, "\partial_0^1"] \arrow[r, shift right, "\partial_1^1"'] & \mathsf{set}_1 \arrow[u, hook] & \mathsf{rel}_1 \arrow[u, hook] \arrow[l, shift right, "\sigma_0^1"'] \arrow[l, shift left, "\sigma_1^1"] & 1 = \mathtt{sml}
\end{tikzcd}
$$

Figure 1: Metastack Structural Framework

# 1 Introduction

The metashell is a simple lisp-like syntactic framework that allows logical expression. It consists of a standard set of logical symbols (subsection 1.1) plus a minimum set of mathematical symbols (sections 2, 3, 4 and 5).

## 1.1 Logical Symbols

The metashell uses parentheses for general delimiters.

`( ... )`

Tuples of terms are formed by using a tuple notation (special delimiters) inherited from KIF.

`[`$\tau_0$ ... $\tau_n$`]`

Equational formulas are constructed with the usual equality symbol = "between" pairs of terms.

`(=` $\tau_0$ $\tau_1$`)`

For propositional expression, the logical symbols contain the connectives for conjunction $\wedge$, disjunction $\vee$, negation $\neg$, implication $\Rightarrow$ and equivalence $\equiv$. These operate on formulas (called here expressions). There are special abbreviations for implication and equivalence.

`(and` $\varphi_0$ ... $\varphi_n$`)`
`(or` $\varphi_0$ ... $\varphi_n$`)`
`(not` $\varphi$`)`
`(implies` $\varphi_0$ $\varphi_1$`)` `(=>` $\varphi_0$ $\varphi_1$`)`
`(equivalent` $\varphi_0$ $\varphi_1$`)` `(<=>` $\varphi_0$ $\varphi_1$`)`

For predicative expression, the logical symbols contain the two quantification symbols $\forall$ and $\exists$. The standard notation is restricted quantification (thus enforcing restricted or limited comprehension): all quantified variables should be sorted (restricted to particular sets).

```
(forall (?x (A ?x) ?y (B ?y)) ...)
(exists (?x (A ?x) ?y (B ?y) ?z (C ?z)) ...)
```

For convenience of expression, the IFF metashell allows for guard exspressions at the end of the list of bound variables. The following universally quantified expressions are equivalent

(forall (?x1 (A1 ?x1) ... ?xm (Am ?xm) $\varphi_1$ ... $\varphi_n$) $\varphi$)
(forall (?x1 (A1 ?x1) ... ?xm (Am ?xm)) (implies (and $\varphi_1$ ... $\varphi_n$) $\varphi$)),

and the following existentially quantified expressions are equivalent

(exists (?x1 (A1 ?x1) ... ?xm (Am ?xm) $\varphi_1$ ... $\varphi_n$) $\varphi$)
(exists (?x1 (A1 ?x1) ... ?xm (Am ?xm)) (and $\varphi_1$ ... $\varphi_n$ $\varphi$)).

For convenience of expression, the IFF metashell allows for definite descriptions using the "the" operator. These are usually used in definitions. Definite description syntactically resembles quantification. A definite description represents the unique thing that satisfies some condition. This thing may be a set, a morphism, a relation or something else. The following notation states that '`a`' is the unique element of '`A`' that satisfies some expression $\varphi$.

(= a (the (?x (A ?x)) $\varphi$))

In general, definite descriptions can be unpacked as follows. Let '`r`' be some relation symbol, let $\tau_1$, ..., $\tau_m$ be some terms, and let $\varphi_1$, ..., $\varphi_n$ and $\varphi$ be some expressions (typically containing free occurrences of the variable '`?x`'). Then the expression

(r $\tau_1 \cdots$ (the (?x $\varphi_1 \cdots \varphi_n$) $\varphi$) ... $\tau_m$)

shall be regarded as an abbreviation for the expression

(exists (?y)
   (and (forall (?x)
            (<=> (and $\varphi_1 \cdots \varphi_n$ $\varphi$)
                 (= ?x ?y)))
        (r $\tau_1 \cdots$ ?y ... $\tau_m$)))

## 1.2 Mathematical Symbols

The structural framework (kernel) of the IFF metastack is illustrated in Figure 1. This lattice-like structure consists of sets, (unary) functions and (binary) relations, their connecting component maps, and the inclusion maps for the subset relationships between metalevels

$$\mathsf{set}_1 \subset \mathsf{set}_2 \subset \cdots \subset \mathsf{set}_n \subset \mathsf{set}_\infty,$$
$$\mathsf{ftn}_1 \subset \mathsf{ftn}_2 \subset \cdots \subset \mathsf{ftn}_n \subset \mathsf{ftn}_\infty, \text{ and}$$
$$\mathsf{rel}_1 \subset \mathsf{rel}_2 \subset \cdots \subset \mathsf{rel}_n \subset \mathsf{rel}_\infty.$$

The IFF metashell represents and axiomatizes this structural framework. The minimal collection of mathematical symbols in the metashell is split into set, function and relation symbols. The four relations of subset, restriction, optimal restriction and abridgment are used to structure the metastack.

$$\mathsf{set}_\infty \subset \mathsf{set}_{\mathrm{meta}} \subset \mathsf{set}_{\mathrm{type}} \subset \mathsf{set},$$
$$\mathsf{ftn}_\infty \subset \mathsf{ftn}_{\mathrm{meta}} \subset \mathsf{ftn}_{\mathrm{type}} \subset \mathsf{ftn}, \text{ and}$$
$$\mathsf{rel}_\infty \subset \mathsf{rel}_{\mathrm{meta}} \subset \mathsf{rel}_{\mathrm{type}} \subset \mathsf{rel}.$$

The terminology and axiomatization for the IFF metashell is partitioned into three levels (iff, type, meta) with one namespace on each metalevel. The metashell is closely related to the grammar, which specifies the correct form for IFF expressions.

# References


[1] F. William Lawvere and Robert Rosebrugh. *Sets for Mathematics*. Cambridge University Press, 2003.

[2] Saunders Mac Lane. *Categories for the Working Mathematician*. Springer-Verlag, 1971.