Multisorted languages and structures

Uri Abraham *

March 26, 2014

Abstract

We continue with multisorted first-order logics and Tarskian structures.

In the previous lectures we covered first-order signatures, languages, and their interpretations. These were *single-sort* structures in which all members of the universe have the same status. In most applications however objects of different characters are considered, and for a more natural modeling, multi-sorted structures have a certain advantage. It is obvious that everything that can be done with multi-sorted structures can also be done with single-sorted ones: predicates can be used to distinguish between different kinds of members. Because of its simplicity and elegance, the usage of single-sorted structures and logic is more common in mathematical logic textbooks. Yet, for applications we prefer here multi-sorted structures and find them useful and natural. So we assume not a single and uniform universe of discourse, but several domains, called sorts, in a single, multi-sorted structure. I give an example for this in the following subsection, but we proceed with the formal definition first. As before, we define multi-sorted signatures, then the resulting language, and finally the interpretations.

Definition 0.1 (Multi-sorted signature) *A multi-sorted signature is a sequence of the form*

$$L = \langle S_1, \ldots, S_n; \overline{P}, \overline{F}, \overline{c}, \mathit{arity}, \mathit{sort} \rangle$$

*Models for Concurrency 2014, Ben-Gurion University.

where (as before) $\overline{P}$ is a sequence of predicates, $\overline{F}$ a sequence of function names, and $\overline{c}$ a sequence of constants. And arity is a natural-number valued function defined over the set of predicates and function symbols. In a multi-sorted signature we have in addition a list $S_1, \ldots, S_n$ of symbols called sorts, and a function sort that associates with each predicate, function, or constant its sort as follows.

If P is a predicate of arity k, then sort(P) is a k-tuple of sorts. That is sort$(P) = \langle X_1, \ldots, X_k \rangle$ where each X_i is some S_j (repetitions are allowed). The intention is that the question of whether $P(x_1, \ldots, x_k)$ holds or not can be asked only if each x_i is of sort X_i.

Similarly, with each k-ary function symbol F, sort$(F) = \langle X_1, \ldots, X_k; X_{k+1} \rangle$ is a $k+1$ tuple of sorts defining the sorts both of the domain and the range of F: X_i for $1 \leq i \leq k$ is the sort of the i-th entry of F, and X_{k+1} is the sort of its return value.

Finally sort(c) for a constant c gives the sort of c.

Another feature of multi-sorted signature which is often very useful is the assignment of variables to specific sorts. That is, a list $V_1, \ldots, V_n$ where each V_i is a set of variables that correspond to sort S_i. The intention is to allow any variable $x \in V_i$ to vary only over domain S_i. In addition to these sort-specific variables, we may have variables whose range of possible values is all of the structure universe.

Definition 0.2 (Multi-sorted interpretation) *An interpretation $\mathcal{M}$ for a multi-sorted signature L consists of the following.*

1. *A non-empty set $|\mathcal{M}|$ which is called the universe of $\mathcal{M}$, and for every sort S_i we have $S_i^{\mathcal{M}} \subseteq \mathcal{M}$ which is said to be the set of members of $\mathcal{M}$ of sort S_i. The universe of $\mathcal{M}$ is the union of its sorts: $|\mathcal{M}| = S_1^{\mathcal{M}} \cup S_2^{\mathcal{M}} \cdots \cup S_n^{\mathcal{M}}$ (not necessarily a disjoint union and in some cases it may be useful to specify that some sorts are subsets of some other sorts.)*

2. *The structure interprets predicates, function symbols and constants in accordance with their sorts. In details, this means the following.*

   (a) *If P is a k-place relation symbol and sort$(P) = \langle S_1, \ldots, S_k \rangle$, then $P^{\mathcal{M}} \subseteq S_1^{\mathcal{M}} \times \cdots \times S_k^{\mathcal{M}}$.*

(b) *If G is a function symbol with arity k and $sort(G) = \langle S_1, \ldots, S_k; S\rangle$, then $G^{\mathcal{M}} : S_1^{\mathcal{M}} \times \cdots \times S_k^{\mathcal{M}} \to S^{\mathcal{M}}$.*

Definition 0.3 (Isomorphism) *Let L be a signature with sorts $S_1, \ldots, S_n$, and let $\mathcal{M}_1$, $\mathcal{M}_2$ be two interpretations of L. We say that $f : |\mathcal{M}_1| \to |\mathcal{M}_2|$ is an* isomorphism *of $\mathcal{M}_1$ and $\mathcal{M}_2$ if f is one-to-one from the universe of $\mathcal{M}_1$ and onto the universe of $\mathcal{M}_2$ such that:*

1. *For every sort S_i, $m \in S_i^{\mathcal{M}_1}$ iff $f(m) \in S_i^{\mathcal{M}_2}$. (*iff *stands for "if and only if".)*

2. *For every n-ary predicate symbol P and n-tuple $m_1, \ldots, m_n \in |\mathcal{M}_1|$,*

$$\langle m_1, \ldots, m_n\rangle \in P^{\mathcal{M}_1} \text{ iff } \langle f(m_1), \ldots, f(m_n)\rangle \in P^{\mathcal{M}_2}.$$

3. *For every n-place function symbol F,*

$$f(F^{\mathcal{M}_1}(m_1, \ldots, m_n)) = F^{\mathcal{M}_2}(f(m_1), \ldots f(m_n)).$$

Similarly, if c is any constant then $f(c^{\mathcal{M}_1}) = c^{\mathcal{M}_2}$.

Since we shall identify two isomorphic structures, this concept reflects our understanding that the inner composition of the elements of the universe of a structure is not relevant. These elements are just abstract "points" devoid of any inherent meanings, were it not for the sorts, functions, and predicates defined on them.

As before, an assignment for M is a function σ from the set of variables into the universe $|M|$. But now an assignment has to respect the sort of the variables. That is, if $x \in V_i$ is a variable of sort S_i, then $\sigma(x) \in S_i^M$ is required. On variables that are not tied to any specific sort there is no such restriction.

Given a multi-sorted signature, the language is defined as before but now with some attention to the issue of sorts. To see the problem suppose for example that F is a function symbol of arity n and $sort(F) = \langle S_1, \ldots, S_n; T\rangle$ That is, F is specified to take arguments from sort S_i at its ith coordinate and to return arguments in sort T. Now, what shall we do with a term $F(x_1, \ldots, x_n)$ when one of the variables x_i is not of sort S_i? The simplest solution is to allow such "meaningless terms" in the language and to have a special object (denoted $\perp$) in each structure to be their denotation. We repeat the definition of the Val_M function, but allowing this $\perp$ value. For every term τ and assignment σ we define $Val_M(\tau, \sigma) \in |M|$ with the possibility that $Val_M(\tau, \sigma) = \perp$.

1. For every variable x, $Val_M(x,\sigma) = \sigma(x)$. For every constant c in the signature $Val_M(c,\sigma) = c^M$.

2. If τ has the form $F(\tau_1,\ldots,\tau_k)$ and if $a_i = Val_M(\tau_i,\sigma)$ for every i, if the sort of F is $(X_1,\ldots,X_k,T)$ (where each X_i and T is some sort) and $a_i \in X_i^M$ for every i then $Val_M(\tau,\sigma) = F^M(a_1,\ldots,a_k)$. In case some a_i is not of sort X_i (or is $\perp$) then we define $Val_M(\tau,\sigma) = \perp$.

For the definition of the truth value $Val_M(\alpha,\sigma)$ for formulas α (and assignment σ into the structure M) we must decide if we are ready to have the undefined value in addition to the two truth values true and false. Suppose for example that P is a unary predicate of sort S. How should we define $Val_M(P(x),\sigma)$ when $\sigma(x) \notin S$? It seems that there are two options: to define this value as $false$ or as the undefined value $\perp$. We prefer the first one which seems more natural to us. So for every formula α we shall define $Val_M(\alpha,\sigma) \in \{true, false\}$ for every assignment σ.

We begin with atomic formulas $P(\tau_1,\ldots,\tau_k)$ and we assume that P has sort $(X_1,\ldots,X_k)$. Say $a_i = Val(\tau_i,\sigma)$ for every $i \leq k$. In case every a_i is of sort X_i and $\langle a_1,\ldots,a_k\rangle \in P^M$, then we define $Val_M(P(\tau_1,\ldots,\tau_k),\sigma) = true$. In any other case we define $Val_M(P(\tau_1,\ldots,\tau_k),\sigma) = false$. Thus in case some a_i is $\perp$ or is of the wrong sort (i.e. not of sort X_i) then $Val_M(P(\tau_1,\ldots,\tau_k),\sigma) = false$.

In particular, for an equality formula $\tau_1 = \tau_2$ we let $Val_M(\tau_1 = \tau_2,\sigma) = true$ if and only if $Val_M(\tau_1) = Val_M(\tau_2)$. So if $Val_M(\tau_1,\sigma) = Val_M(\tau_2,\sigma) = \perp$ then the equality formula gets the value $true$.

If φ is any formula (in the given signature) we define $Val_M(\varphi,\sigma)$ (for every assignment σ) by induction on the structure of φ as before.

1. If φ is an atomic formula then $Val_M(\varphi,\sigma)$ was defined above.

2. Suppose α and β are formulas and both of $Val_M(\alpha,\sigma)$ and $Val_M(\beta,\sigma)$ were defined. Then $Val_M((\alpha\to\beta),\sigma) = false$ if and only if $Val_M(\alpha,\sigma) = true$ and $Val_M(\beta,\sigma) = false$. The truth value definitions for the other logical connectives are similarly defined.

3. Suppose a formula φ for which we have defined $Val_M(\varphi,\sigma')$ for every assignment σ'. Then we define $Val_M(\exists x\varphi,\sigma) = true$ iff there exists an assignment σ' so that

   (a) For every variable y that is different from x $\sigma'(y) = \sigma(y)$, and

(b) $Val_M(\varphi, \sigma') = true$.

We can write $M \models \varphi[\sigma]$ for $Val_M(\varphi, \sigma) = true$ (we say then that M *satisfies* φ under the assignment σ).

Definition 0.4 (Model) *We say that a* collection *of sentences T in some language is satisfied by a structure $\mathcal{M}$ if every sentence in T is satisfied by $\mathcal{M}$; we then say that $\mathcal{M}$ is a* model *of T.*

We write this as $\mathcal{M} \models T$. A collection of sentences that has a model is called a theory *.*

0.1 Examples of multi-sorted structures

In the first example we model the situation in which several measurements of light intensity are made by some light-meter. A measurement evaluates the intensity of light during the opening of the meter. We want to be able to express in our language the fact that distinct measurements may be made over different exposure intervals and may show different values. For this, every structure contains a set of "events" that represent measurements, and a value is assigned to each event to represent the result of the measurement. Hence two sorts of individuals are needed: measurement events and real numbers (if we want the results to be real numbers). Moreover, to evaluate the measurement error, we would like to have the "true" light intensity function in the structure, that is the function that gives the real intensity at each instant. We shall denote this function by I and let $I(t)$ denotes the light intensity at moment t.

In our example the measurement events are one sort and the real numbers are another. An advantage of this two-sorted approach is that functions and relations may be specific to certain sorts. The addition operation, for example, is defined on the real numbers, and it is meaningless to ask for the addition of events.

So we define first a signature K that contains two sorts: E and R (the elements of E are called events and those of R numbers). In addition, K contains the following.

1. A binary predicate $<$ and binary function symbols $+$, $-$, $\times$, $/$ over sort R. (For typographical clarity, we no longer use the asterisks.) When we say that $+$, for example, is over R we mean that $sort(+) =$

$\langle R, R; R\rangle$; that is to say that + is interpreted as a function taking pairs of individuals of sort R into individuals of sort R.

2. For each rational number q, a constant $\boldsymbol{q}$.

3. Two function symbols, *Left_End* and *Right_End*, are defined on E and give values in R. (The intention is to use them to give for every event e its temporal interval $(\mathit{Left\_End}(e), \mathit{Right\_End}(e)))$.

4. A unary function symbol *Value* is of sort $(E; R)$. That is, the function is defined on E and returns values in R. (The intention is to use it for the values of the measurement events.)

5. A unary function symbol I is of sort $(R; R)$. The intension is to have $I(t)$ for the light intensity at time t.

A standard real number interpretation $\mathcal{M}$ of K is a structure containing the real numbers $\Re$ (interpreting the sort R) and a set $E^{\mathcal{M}}$ of "events". The interpretation of the arithmetical operations is the standard interpretation on $\Re$ (for example, $<^{\mathcal{M}}$ is the natural ordering on $\Re$). For each individual e in $E^{\mathcal{M}}$, $\mathit{Left\_End}^{\mathcal{M}}(e)$, $\mathit{Right\_End}^{\mathcal{M}}(e)$, $\mathit{Value}^{\mathcal{M}}(e)$ are real numbers with $\mathit{Left\_End}^{\mathcal{M}}(e) <^{\mathcal{M}} \mathit{Right\_End}^{\mathcal{M}}(e)$. The interpretation of I is an arbitrary real-valued function $I^{\mathcal{M}}$.

Of course there is nothing in the symbols to force this particular interpretation, and many other interpretations are possible. This example is brought here just to give some idea of the diversity of situations where multi-sorted structures can be used, and it will not be used later on.

The duration of any event e is the number $\mathit{Right\_End}(e) - \mathit{Left\_End}(e)$ which is the length of the interval of e.

Exercise 0.5 *1. Write a sentence that says that for every measurement of a positive duration, the value returned is the I value at some inner point in the interval.*

2. Write a sentence that says that item 1 holds provided that the measurement is of length more than 0.23.

We return to the light intensity measurement from above. Suppose that we want to say about a light-meter that it guarantees that every measurement is accurate to within 0.1% of the intensity at some point in its interval,

provided its exposure interval is at least 20 seconds but not more than 35. I will write this statement three times, in increased degrees of formality.

For every measurement event e, if the duration of e is between 20 seconds and 35 seconds, then the value returned by e is between $0.999 \times r$ and $1.001 \times r$, where r is the intensity at some point of this period.

This is the easiest and most understandable formulation. But it leaves some doubts, as for example what do we mean by "between". A more formal writing clarifies that we mean "between in the strict sense".

$$\forall e \in E\ [(20 \leq \text{duration}(e) \leq 35) \rightarrow 0.999 \times r < \mathit{Value}(e) < 1.001 \times r]$$

where r is the intensity at some point in the temporal interval of e

The problem with this formulation is that there are no facilities in our language for using expressions such as "where r is etc." So we can use existential quantifiers to express this property.

$$\forall e \in E\ 20 \leq (\mathit{Right\_End}(e) - \mathit{Left\_End}(e)) \leq 35 \rightarrow$$
$$(\exists t \in R)(\mathit{Left\_End}(e) < t < \mathit{Right\_End}(e) \wedge 0.999 \times I(t) < \mathit{Value}(e) < 1.001 \times I(t)).$$

The reader can appreciate now why we seldom write the formal sentences in full, but prefer an informal discourse. Completely formal writing is very difficult to read. Yet we must exercise and get used to read and write such formal statements in order to get a feeling for what can be done formally and what cannot.

The expression $\forall e \in E\ \phi$ used above as a formalization of "for all events e ϕ" can be written simply as $\forall e\ \phi$ if we agree that variable e varies only over events. In general we adopt this convention of using special variables for each sort of objects, but the qualified quantification $\forall x \in E\ \phi$ is still used here and there. (Many authors use expressions such as $\forall e : E$, which is very reasonable because the relationship between a variable and its sort is not exactly the membership relation.)

0.1.1 Pairs

Two-sorted structures are useful to handle pairs and finite sequences, as the second example shows. This will be significant for later development and

specifications. Usually we write pairs with angled brackets $\langle a, b\rangle$, but for typographical simplicity we often use round brackets (a, b).

In set theory one learns how to form pairs as sets. A pair $\langle a, b\rangle$ can be defined as a set $\{\{a\}, \{a, b\}\}$. We do not use this (or any other) particular representation here, as only the abstract properties of pairs and finite sequences are needed. We shall define now the language in which these abstract properties are expressed.

We first define the pair signature. There are two sorts, U (for the set of individuals from which pairs are formed) and P (for the pairs). There is a binary function symbol *make_pair* defined on U and returning values in P, and there are two unary function symbols *first* and *second* defined on P and returning values in U.

Two axioms determine the properties of these pairing and unpairing functions. Let us agree that variables x and y vary through U and variables p and p' vary through P.

$$\forall x, y\ [\mathit{first}(\mathit{make\_pair}(x, y)) = x \wedge \mathit{second}(\mathit{make\_pair}(x, y)) = y] \qquad (1)$$

$$\forall p, p'\ [\mathit{first}(p) = \mathit{first}(p') \wedge \mathit{second}(p) = \mathit{second}(p') \longrightarrow p = p'] \qquad (2)$$

Observe that we did not have to say in these axioms that $\mathit{make\_pair}(x, y)$ is in P, as this is a consequence of the signature requirement which says that the sort of *make_pair* is $\langle U, U, P\rangle$, namely that *make_pair* is a binary function from U to P. An advantage of using a multi-sorted language is that each variable is connected with a certain fixed sort, and this brings some economy in writing formulas. For example, a longer form of the first axiom would be $\forall x, y \in U \ldots$, but here there was no need to qualify the range of x and y which were assumed to be U-variables. If the traditional notation $\langle x, y\rangle$ for the expression "$\mathit{make\_pair}(x, y)$" is used, then the first axiom becomes $\forall x, y(\mathit{first}(\langle x, y\rangle) = x \ldots$.

We did not require that sorts U and P are disjoint. In many cases it is natural to add $\forall x \in U(x \notin P)$, but there are occasions when we want $P \subset U$, which enables forming pairs $\langle x, y\rangle$ when x or y (or both) are themselves pairs.

If $\mathcal{M}$ is an interpretation of the signature defined above that satisfies the two axioms, then $U^{\mathcal{M}}$ and $P^{\mathcal{M}}$ are just sets, and the fact that we shall call members of $U^{\mathcal{M}}$ "elements" and members of $P^{\mathcal{M}}$ "pairs" does not say anything about their structure. In fact, the individuals of $\mathcal{M}$ are devoid of any structure and only the functions and predicates defined over them can convey a meaning.

Let $\mathcal{M}$ be an interpretation for this pairing/unpairing signature, and assume that $P^{\mathcal{M}} \subset U^{\mathcal{M}}$. Then a notation $\langle a_0, \ldots, a_{n-1} \rangle$ can be introduced for $n \geq 2$ and for a_i's in $U^{\mathcal{M}}$ as follows: If $n > 2$ then

$$\langle a_0, \ldots, a_{n-1} \rangle = \mathit{make\_pair}(\langle a_0, \ldots, a_{n-2} \rangle, a_{n-1}).$$

There are other possibilities to model tuples, and I do not want to suggest that this is the best way: the reader can keep his favorite definition if he wants to, and I will avoid referring to a particular representation by adhering to the following convention. Whenever a structure $\mathcal{N}$ is mentioned and $U^{\mathcal{N}}$ is a specific sort in $\mathcal{N}$, if I say that $\mathcal{N}$ is *equiped with pairs*, I mean that there is in fact another sort in $\mathcal{N}$, a sort of pairs of members of $U^{\mathcal{N}}$, even though this sort and the related functions were not specified beforehand. Since pairs are ubiquitous, it is reasonable to assume that all our structures are equiped with pairs.

Exercise 0.6 *In this exercise we want to specify finite sequences similarly to the way that we specified pairs. The language is defined below, and your exercise is to write axioms in this language that express properties of finite sequences.*

There are three sorts: U for the set of individual of which the finite sequences are formed, S for the sort of all finite sequences, and N for the sort of natural numbers. We have the following functions in our signature.

$\mathit{make\_seq}(e)$: forms the one element sequence $\langle e \rangle$.

$\mathit{concatenate}(s, t)$: forms the sequence $s^\frown t$.

$\mathit{length}(s)$: gives the length of the sequence s.

$\mathit{At}(s, i)$: returns the i-th entry in s (when $i < \mathit{length}(s)$).

Then we have the arithmetic functions on N (such as $+$) and the ordering relation $<$.