

Toward a Proper Semantics for the Logic of Occurrence

Satoshi Tojo
 Japan Advanced Institute of
 Science and Technology
 tojo@jaist.ac.jp

Ken Kaneiwa
 National Institute of Informatics
 kaneiwa@nii.ac.jp

1 Property and Occurrence

A predicate of predicate logic generally represents a static, atemporal *property* of its arguments. For example, if we write:

$$\forall x[\text{beef}(x) \rightarrow \text{meat}(x)], \quad (1)$$

then “if x is *beef* then it is *meat*.” However,

$$\forall x[\text{roast}(x) \rightarrow \text{cook}(x)], \quad (2)$$

is somewhat a misleading formula. We imagine that x refers an object to be cooked, or in some cases x might be an agent who cooks; anyway, (2) represents that “for any individual x if it has the property of *roast*-ing then it also has that of *cook*-ing.” In order to state that there is an occurrence of an event of *cook*-ing, we need to mention a token e_1 of such a *situation* as $e_1: \ll \text{cook} \gg$; e_1 is an identifier of an event that happened once and for all at some location l and some time t . The formula:

$$e_1: \ll \text{roast} \gg \Rightarrow e_1: \ll \text{cook} \gg, \quad (3)$$

means “if e_1 is an event of *roast*-ing then the same situation can be called *cook*-ing.” While the variable x in (2) is an individual in a universe, the token e_1 in (3) is not.

If we embed the implication of (1) into (3), the expected syntax would be:

$$\begin{aligned} \forall x, e_1: \ll \text{roast}, x: \text{beef} \gg \\ \Rightarrow e_1: \ll \text{cook}, x: \text{beef} \gg \\ \Rightarrow e_1: \ll \text{cook}, x: \text{meat} \gg . \end{aligned}$$

Our objective in this paper is to give the semantics of the logic of occurrence, to formalize temporal locations, negations, and other natural language phenomena properly.

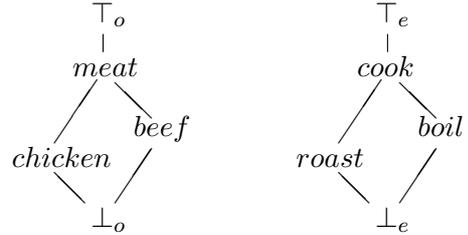


Figure 1: Two sort hierarchies

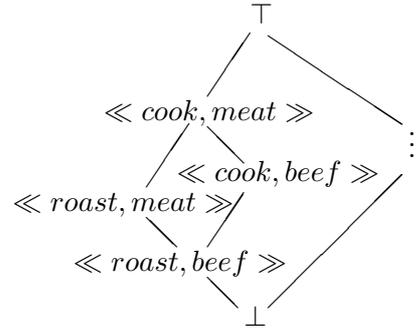


Figure 2: Combined diagram

2 Dual Sort Hierarchy and Its Semantics

Sort hierarchy Sorts and subsumptions are useful concepts to classify objects and to represent relations in them. Generally, we write ‘ $\text{beef} \sqsubseteq_o \text{meat}$ ’ in short to represent the implication of (1). These **subsort declarations** organize the **object sort hierarchy**, as in the left-hand side of Figure 1.

Here, we propose to define subsumption relations also in the meanings of events; the declaration ‘ $\text{roast} \sqsubseteq_e \text{cook}$ ’ represents the *information flow* of (3), and organize the **event sort hierarchy**, as in the right-hand side of Figure 1. Thus, we strictly distinguish ‘ \sqsubseteq_e ’ and ‘ \sqsubseteq_o .’

The event sort hierarchy is quite feasible for inference when it is combined with the object sort hierarchy [1], and its *Hasse* diagram greatly improves the visibility of conceptual relations as in Figure 2.

The reason we divide sorts into two kinds is twofold. Firstly, each of hierarchies may have different mathematical features as to whether it is a lattice, distributive, Boolean, and so on. Secondly, we would like to define additional operations between tokens for further applications.

In this paper, we often write concisely

$$e_1: \ll \text{roast}, o_1: \text{beef} \gg \quad (4)$$

to represent that e_1 belongs to the sort of $\ll \text{roast} \gg$ and o_1 belongs to the sort of beef . However, the same formula can be written also as follows:

$$\begin{aligned} (4) &= \ll e_1: \text{roast}, o_1: \text{beef} \gg \\ &= (e_1, o_1): \ll \text{roast}, \text{beef} \gg \\ &= \ll \text{roast}, \text{beef} \gg (e_1, o_1), \end{aligned}$$

for we treat the two sort hierarchies equivalently, considering that both of *roast* and *beef* work as predicates of first-order logic.

Naturally, we introduce token variables and can quantify them.

Signature We use the following symbols:

$e_1, e_2, \dots \in E$	a set of tokens
$o_1, o_2, \dots \in O$	a set of objects
$t_1, t_2, \dots \in T$	a set of token variables
$x_1, x_2, \dots \in X$	a set of object variables
$p, q, \dots \in S_e$	event sorts
$a, b, \dots \in S_o$	object sorts.

The signature of the language is the tuple $\langle E, O, T, X, :, \sqsubseteq_o, \sqsubseteq_e \rangle$, where ‘:’ is the sort membership, \sqsubseteq_e and \sqsubseteq_o are sets of subsort declarations.

For any pair of sorts a and b , we define the greatest lower bound $a \sqcap b$ (*meet*) and the least upper bound $a \sqcup b$ (*join*). If all the meets and joins exist uniquely, the hierarchy is called a *lattice*. \top (*top*) and \perp (*bottom*) are the join and the meet of all the sorts, respectively. For any sort a , such \bar{a} that $\bar{a} \sqcap a = \perp$ is called the

complementary sort; if it is unique and $\bar{a} \sqcup a = \top$ then the lattice is called *Boolean*.

Provided that each hierarchy is Boolean,

$$\begin{aligned} \top_e &= \sqcup S_e, \top_o = \sqcup S_o, \perp_e = \sqcap S_e, \perp_o = \sqcap S_o, \\ a \sqcup \bar{a} &= \top_o, a \sqcap \bar{a} = \perp_o, \\ p \sqcup \bar{p} &= \top_e, p \sqcap \bar{p} = \perp_e. \end{aligned}$$

Semantics As we cannot actually enumerate all the individuals in the universe, the set of *occurrences*¹ of events for the model of tokens is just a matter of mathematical abstraction.

Given a set of occurrences U_e for tokens, and a set of individuals U_o for objects, we interpret (4) as follows: $\ll e_1 \gg \in \ll \text{roast} \gg \subseteq U_e$, $\ll o_1 \gg \in \ll \text{beef} \gg \subseteq U_o$, and

$$\begin{aligned} \ll (e_1, o_1) \gg &= \ll e_1 \gg \otimes \ll o_1 \gg \\ &\in \ll \text{roast} \gg \times \ll \text{beef} \gg \subseteq U_e \times U_o, \\ &= \ll \ll \text{roast}, \text{beef} \gg \gg \end{aligned}$$

where ‘ \times ’ is the Cartesian product. Thus, our logic is interpreted by the triplet $\langle U_e, U_o, \ll \gg \rangle$ instead of conventional $\langle U_o, \ll \gg \rangle$. Then,

$$\ll \ll \text{roast}, \text{beef} \gg \gg \subseteq \ll \ll \text{cook}, \text{meat} \gg \gg$$

if and only if $\ll \text{roast} \gg \subseteq \ll \text{meat} \gg$ and $\ll \text{beef} \gg \subseteq \ll \text{meat} \gg$, that is, $\text{roast} \sqsubseteq_e \text{cook}$ and $\text{beef} \sqsubseteq_o \text{meat}$,² as in Figure 2.

In case \sqsubseteq_o is a Boolean lattice,

$$\begin{aligned} [a \sqcap b] &= [a] \cap [b], [a \sqcup b] = [a] \cup [b], \\ [\top_o] &= U_o, [\perp_o] = \emptyset, [\bar{a}] = U_o - [a]. \end{aligned}$$

Ditto as to \sqsubseteq_e over U_e ; however, because such an event sort $p \sqcap q$ rarely exists, $\ll p \sqcap q \gg$ often comes to \emptyset , as $\ll \text{roast} \sqcap \text{boil} \gg = \emptyset$.

We can extend the above discussion to multiple arguments for an event. For example, $\ll \text{roast}, a: \text{human}, o: \text{beef} \gg$ can be interpreted as a member of

$$\ll \text{roast} \gg \times \ll \text{human} \gg \times \ll \text{beef} \gg \subseteq U_e \times U_o^2.$$

In which case, a missing argument can be regarded as ‘ \top_o ’ of the universe U_o .

¹The name ‘situation’ for a member of the set is rather confusing because it represents again a set of facts; the adequateness for the name requires further discussion.

²Giving a strict syntax for the logic of occurrence, we can avoid the distinction between $\ll \gg_e$ and $\ll \gg_o$. For example, only the first element of $\ll \gg$ should be interpreted in U_e , and other elements in U_o ; thus $\ll \ll \text{chicken} \gg \gg$ would be null.

3 Temporal Relation, Negation, and Referent

The dual sort logic is applied to natural language semantics in various ways, by (i) token variables (as that in first-order logic), and by (ii) sort hierarchies.

Temporal relation If we assume a relation R between tokens, we would like to imply $e': \ll \dots \gg$ from $e: \ll \dots \gg$ and eRe' . We give an example of temporal logic, defining *inclusion* ' \trianglelefteq ' and *precedence* ' \preceq ', as follows:

$$e: \mathbf{P}\phi \Leftrightarrow \forall t \preceq e[t: \phi], \quad e: \mathbf{F}\phi \Leftrightarrow \forall t \succeq e[t: \phi].$$

where \mathbf{P} and \mathbf{F} are modal operators of 'all the past' and 'all the future,' respectively, and ϕ is an abbreviation of $\ll \dots \gg$. Furthermore, we can extend this notion to aspects.

$$e: \mathbf{Pg} \phi \Leftrightarrow \forall t \trianglelefteq e[t: \phi]$$

where \mathbf{Pg} simulates the *progressive* aspect with *downward heredity*³ [2].

In addition to the set-theoretical interpretation $\llbracket e \rrbracket$ for a token e , we propose the way to retrieve the spatio-temporal location $\|e\| \subseteq l \times t$. The difficulty in sizing a location in general has long been discussed since $S \mathcal{E} A$; therefore, we consider ' $\| \cdot \|$ ' again as a mathematical abstraction. Assuming some class of e 's can fix their locations, we define $\|e\|$ as a closed continuous region in the time-space. When we restrict $\|e\|$ only to the temporal location, we attach the suffix as $\|e\|_t \subseteq t$. As for the inclusion relation, $e_1 \trianglelefteq e_2$ if and only if $\|e_1\|_t \subseteq \|e_2\|_t$.

Scope of negation The goodness of sort hierarchy is that we can easily specify the scope of the negation, indicating a complementary sort [3]. We use ' \neg ' to negate the membership of the sort as: $\neg e: \ll p \gg = \ll \neg e: p \gg$ if and only if $\llbracket e \rrbracket \notin \llbracket p \rrbracket$. Because the conventional polarity '0' can be regarded as the syntactic negation, $\ll p, a; 0 \gg$ comes equal to the disjunction: $\ll \neg e: p, o: a \gg \vee \ll e: p, \neg o: a \gg$.

³The temporal feature is inherited to its internal temporal extents.

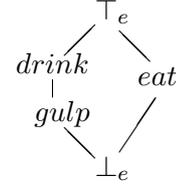


Figure 3: Non-Boolean lattice

While ' \neg ' is the classical (weak) negation, $e: \ll \bar{p} \gg$ states that e is in the sort of \bar{p} positively (strong negation). For example, in case of a non-Boolean lattice in Figure 3, because $gulp \sqcap eat = \perp_e$, $e: \ll \overline{gulp} \gg$ implies $e: \ll eat \gg$, therefore $\neg e: \ll drink \gg$ as well as $\neg e: \ll gulp \gg$ even though $drink \sqsupset_e gulp$.

Furthermore, assuming the lattice is Heyting algebra,⁴ we can construct an intuitionistic logic, that is, $\neg\neg e: \ll p \gg$ does not necessarily imply $e: \ll p \gg$.

Discussion on referent We consider the spatio-temporal location of an object in the similar way, that would be a line in $l \times t$ space. If the proposition $e: \ll roast, o: beef \gg$ is true the object 'o' must exist in $\|e\|$, while this is not the case for $e: \ll dream_of, o: beef \gg$, in which 'o' may refer to an object that is outside of $\|e\|$, or there may not be the referent. Thus, whether $\|o\|$ is in $\|e\|$ concerns the *de re/de dicto* distinction as other linguistic theories.

References

- [1] K. Kaneiwa and S. Tojo. Event, Property, and Hierarchy in Order-Sorted Logic. In D. de Schreye (ed.) Proc. of ICLP 1999, pp.94-108, The MIT Press 1999.
- [2] S. Tojo. Aspect Analysis in Arrow Logic. In Information-Oriented Approaches to Language, Logic, and Computation, Vol. 3, CSLI Publications, 2000.
- [3] K. Kaneiwa and S. Tojo. An Order-Sorted Resolution with Implicitly Negative Sorts. In P. Codgnet (ed.) Proc. of ICLP 2001, pp.300-314, LNCS2237 Springer, 2001.

⁴For a pair of sorts a and b , ' $a \supset b$ ' is the relative pseudo-complementary sort.