```
\title{Type Theoretic Foundations for Context, Part 1:\verb|\\|
Contexts as Complex Type-Theoretic Objects}
```


```
\author{Richmond H.~Thomason\inst{1}}

\institute{
Computer Science Department,
University of Michigan,
Ann Arbor, MI 48109-2110, USA
}

\email{rich@thomason.org}
```


```
\maketitle
```


```
 \begin{abstract}
  This paper presents Contextual Intensional Logic, a
type-theoretic logic intended as a general foundation for
reasoning about context.  I motivate and illustrate
the logical framework, and conclude by indicating extensions
that may be desirable.
 \end{abstract}
```


## 1 Introduction

In several previous works, [15, 16], I proposed and explored (very briefly), the idea of fitting the logic into a version of type theory that is designed to deal with the phenomena that are discussed in the literature of context. This paper will improve, refine and extend these type-thoretic ideas.

There are a number of logical advantages to type theory.

(1) The underlying logical architecture, which goes back to [4], is beautifully simple and has been thoroughly investigated by logicians.
(2) The framework of types provides a rich, highly structured ontology that is potentially useful in formalization.
(3) The theory is a straightforward extension of Richard Montague's Intensional Logic, [11, 6, 1], which has been the dominant formalism for the logical interpretation of natural language. Using it provides direct connections to an extensive body of work in natural language semantics. So the type theoretic approach can facilitate linguistic applications of the theory of context.
(4) The use of types provides conceptual clarity.

In this paper, I will introduce and explain the basic ideas of the formalism, using these to illustrate and support point (4), above. I will also discuss some ways in which the formalism can (and should) be extended.

## 2 Brief Introduction to Intensional Logic

Any version of type theory will involve not only a domain of individuals, with variables ranging over this domain, but domains corresponding to higher-order types: sets of individuals, sets of sets of individuals, etc. Formalizations of type theory based on [4] use functional abstraction to organize these domains; in general, where $D_1$ and $D_2$ are domains of the type theory, the set $D_2^{D_1}$ of functions from $D_1$ to $D_2$ is also a domain of the theory.

This leads to the following recursive definition of types, in which there are primitive types for individuals and truth values, and all other types are functional.

(2.1) $e$ is a type.
(2.2) $t$ is a type.
(2.3) If $\sigma$ and $\tau$ are types, so is $\langle\sigma,\tau\rangle$.

Here, $e$ is the type of individuals (entities), $t$ is the type of truth-values, and $\langle\sigma,\tau\rangle$ stands for the type of functions from objects of type $\sigma$ to objects of type $\tau$.

The language of type theory has an infinite set of variables of each type.[1] The language has only three primitive syntactic constructions.

| | |
|---|---|
| *Identity*: | If $\alpha$ and $\beta$ are expressions of type $\tau$, so is $\alpha = \beta$. |
| *Functional application*: | If $\zeta$ is an expression of type $\langle\sigma,\tau\rangle$ and $\alpha$ is an expression of type $\sigma$, then $\zeta(\alpha)$ is an expression of type $\tau$. |
| *Lambda abstraction*: | If $\zeta$ is an expression of type $\tau$, then $\lambda x_\sigma \zeta$ is an expression of type $\langle\sigma,\tau\rangle$. |

With these resources, the full set of boolean operations can be defined, as well as universal and existential quantification over domains of any type. The model theory of the logic is straightforward; arbitrary domains are assigned to primitive types, the domain of $\langle\sigma,\tau\rangle$ is the set of functions from the domain of type $\sigma$ to the domain of type $\tau$; $=$ is interpreted as identity, $()$ is interpreted as functional application, $\lambda$ is interpreted as functional abstraction. See [6] for details on these matters.

A number of ontological policies come along with this approach to types: sets are represented by the corresponding characteristic functions (i.e., a set of objects of type $\tau$ is represented as a function of type $\langle\tau,t\rangle$), and $n$-place functions from types $\langle\sigma_1,\sigma_2,\ldots,\sigma_n\rangle$ to type $\tau$ are represented as a nested type $\langle\sigma_1,\langle\sigma_2,\ldots\langle\sigma_n,\tau\rangle\ldots\rangle\rangle$ made up of 1-place functions. According to these policies, for instance, a set of 2-place relations between individuals and sets of individuals

[1] In writing formulas of intensional logic, I will label the first occurrence of a variable with its type unless the type is $e$, in which case the label may be omitted. Later occurrences will not be marked for type; no confusion can arise, as long as all independent uses of bound variables involve distinct variables.

would have type $\langle\langle e\rangle, \langle\langle e, t\rangle, t\rangle\rangle$. An object of this type would be a function that inputs a first-order object and outputs a function that inputs a set of first-order objects (which itself is a function from first-order objects to truth-values) and outputs a truth-value.

This formalism uses truth values to represent sentences. Obviously, a domain containing only two values will be unable to represent sentence meanings adequately, an inadequacy that is reflected in the inability of the logic to deal with *propositional attitudes* such as belief. Montague's intensional logic remedies this problem by introducing a third primitive type $w$, the type of possible worlds. This makes available a type $\tau\text{-Prop} = \langle w, t\rangle$ of propositions.

Treating propositions as sets of possible worlds is, of course, problematic. (See, for instance, [12, 7].) But it is an approach that has been pursued with some success in philosophical logic, computer science, economics, and natural language semantics. It is certainly possible to generalize the possible worlds approach to intensionality to obtain a less restrictive account of propositions. But it seems to me that such generalizations are premature. Without constraints, hyperintensional theories are uninformative. Appropriate constraints seem to require better general models of the reasoning agent than we have at present. Until such models are developed, it seems better to me to use the possible worlds formalisms, which in any case have many features that would need to be preserved in any more general approach.

Richard Montague showed not only that the framework of intensional logic provides not only a type for propositions, but that many other higher-order types are ontologically natural and useful in the interpretation of natural language semantics. I will now explain why this framework also provides an appropriate starting point for categorizing contexts.

## 3 Contexts as Modalities

The integration of knowledge sources seems at present to be the origin of the most detailed and illuminating examples of the ways in which the theory can be used.[2] If we look at a context as a knowledge source, the most simple way to model a context in the ontology of type theory would be to identify a context with the set of propositions that it delivers. This would locate contexts in the type $\langle\langle w, t\rangle, t\rangle$: this is the type that I assigned to contexts in [16].

This type-theoretic account of contexts would suffice for applications in which an agent (which is not itself a context) is simultaneously accessing information from many contexts. Each context would then deliver its set of propositions to the collecting agent. But if we want to allow *contexts* to access information from other contexts, we need an enriched representation of contexts.

The problem is this. The type $\langle\langle w, t\rangle, t\rangle$ makes contexts have propositions as inputs. This intensionality of contexts is mandated by the desired applications: we certainly do not want a context to support *every* true sentence if it supports

[2] See, especially, the example in [10, Section 6].

*any* true sentence. Since the output of a context $c$ is only a truth value, although we can access a proposition $p$ that holds in $c$ there will be no way to construct from $c$ and $p$ the proposition that says that $p$ holds in $c$. Therefore, we can't pass the output of applying a context to a proposition to another context.

To solve this problem, we need to assign contexts a type with an enriched output. Iteration of contexts can be managed in several ways within intensional logic; the following formalization corresponds directly to multi-agent epistemic logics, of the sort discussed in [5], in which epistemic agents $a$ are associated with a binary relation $R_a$ over worlds.

This standard relational semantics for modal operators specifies that $\Box_a A$ is true in $w$ if and only if $A$ is true in all worlds $w'$ such that $wR_aw'$. The idea can be captured in Intensional Logic by (i) locating contexts in the type $\tau_1\text{-Mod} = \langle\langle w,t\rangle,\langle w,t\rangle\rangle = \langle\tau\text{-Prop},\tau\text{-Prop}\rangle$, (ii) introducing a function *Rel* of type $\langle\tau_1\text{-Mod},\langle w,w\rangle\rangle$ that associates a relation over worlds with each contexts, and (iii) adding the following axiom, which guaratees that the behavior of the context is determined according to the standard satisfaction condition for modalities.

$$(3.1)\ \ \forall x_{\tau_1\text{-Mod}}\forall p_{\tau\text{-Prop}}\forall y_w[x(p)\leftrightarrow\forall z_w[\mathit{Rel}(y,z)\rightarrow p(z)]$$

Within this framework we can provide a definition of McCarthy's *ist* relation, reconstructed here as a relation between contexts and the *propositions* (not the sentences) that hold in these contexts.

$$(3.2)\ \ \mathit{ist}_1\ =\ \lambda x_{\tau_1\text{-Mod}}\lambda y_{\tau\text{-Prop}}\lambda z_w\forall z'_w[x(z)\rightarrow y(z')]$$

This definition gives *ist* the type $\langle\tau_1\text{-Mod},\langle\tau\text{-Prop},\tau\text{-Prop}\rangle\rangle$ which inputs a modality and a proposition, and outputs a proposition.

This approach to context has several shortcomings, some of them substantive and some of them a matter of public relations. I will address the second of these first, leaving the substantive issues for the remaining sections.

If we treat contexts as modal operators, it is hard to see what is new about the logic of context. Modal logic is a well developed area of logic that has received a great deal of attention over the last forty years. Regarding contextual logic as a branch of modal logic seems to leave relatively little work for us to do.

While it is true that this conservative approach may rule out a more logically creative program, it still leaves room for some innovations, because (as I will argue in the next section), the logic of context can't in fact be identified with modal logic. But treating it as a generalization which preserves the main features of modal logic has many advantages. First, it enables us to import results and applications from modal logic. It is in fact, very useful to regard contexts as simple epistemic agents, agents which know information about other agents and can communicate with other, similar agents. This enables us to import ideas concerning protocol design and knowledge-based programming into the logic of context.[3] We may also be able to import the techniques that have been developed for modal theorem proving. (See [14].)

[3] See [5] for discussion of these matters, and for further references.

## 4 Contextual Intensional Logic

There is a serious limitation to this approach to context; it will not deal with cases in which the meanings of terms can differ from context to context. The computational literature has regarded variation in meaning as an essential application of the logic of context. Modal logic can't represent the reasoning that deals with ambiguous expressions. Take the simplest case: two "personal assistant" databases with two users, $a$ and $b$. Both databases record information about their users, using an internal constant @USER to refer to their users. To merge this information coherently, we have to assign different propositions to expressions like

(4.1) BIRTHDATE(@USER, $\langle 4, 4, 1969\rangle$).

But modal logic has no natural way to represent the reasoning that produces these different assignments.[4]

Translating ideas from Kaplan [9] into the type-theoretic framework proposed above, I want to address this problem by introducing a fourth primitive type: the type $i$ of indices. Kaplan thinks of indices as contextual interpretations of "indexical expressions" such as 'I', 'here', and 'now'.

I want to extend this notion, by thinking of an index as a simultaneous disambiguation of the relevant *contextualizations*: the lexical indexicalities and ambiguities that can arise in an application. If the only contextualizations arise from 'I' and 'here', we could identify an index with a pair consisting of a person and a place. If the only contextualizations arise in the ambiguity of ten lexical items that we have identified as each having two possible meanings, then we will need indices corresponding to the $2^{10}$ possible disambiguations. Introduce a primitive type $i$ for indices. I will call the extension of Intensional Logic that is obtained by adding the primitive type $i$ *Contextual Intensional Logic* (CIL).

Following Kaplan, I want to think of the evaluation of contextualized expressions as taking place in two phases: (1) a disambiguation phase, where the expression is assigned an intension and (2) an evaluation phase, where the intension is evaluated in a world. Thus, to interpret an expression like

(4.2) 'I'm over 21 years old'

we first need to identify the speaker, $s$; this yields the proposition that is true in a world if and only if $s$ is over 21 years old in that world. We can think of this as the two-stage evaluation of an abstract representation of the sentence's meaning called its *character*, a representation which captures the potentiality of a sentence meaning to yield different propositions in different "contexts" by treating the character as a function from indices to propositions. In general, the

[4] We could, of course, formalize the reasoning as syntactic, i.e. we could treat it as reasoning about expressions. I will not explore this alternative here; despite its apparent naturalness, it is much less satisfactory in the long run, I believe, than the intensional approach that I assume here. The main formal problem that a syntactic approach raises, of course, is that it reintroduces the semantic paradoxes.

character is first evaluated at an index to yield a *content*, which then may be evaluated in a world to produce an extension. In CIL, it is natural to assign sentences a character of type $\tau$-Char-Prop $= \langle i, \tau\text{-Prop}\rangle$; this type will input an index and output a proposition.

Now, a context of the sort envisaged by McCarthy will perform two different functions: it will serve as a source of disambiguation and as a local knowledge source. From the standpoint of CIL, the first of these functions is represented by an index, or object of type $i$; the second, as before, is represented by a modality, or object of type $\tau_2$-Mod. Therefore, a context is a pair consisting of an index and a modality. This yields no very natural type for such contexts in CIL, which provides no clean encoding for ordered pairs of objects of different type. However, if we use one of the (unnatural) encodings in CIL of the cross product $\sigma \times \tau$ of $\sigma$ with $\tau$, or (better) if we add a cross-product operation to the underlying type definition, we can situate *indexical contexts* in the type

$$\tau_2\text{-Con} = i \times \tau_1\text{-Con}.$$

But we can avoid the need to provide a type for indexical contexts by making indices and modalities separate inputs to *ist*. If we want this *ist* to iterate, it should output a propositional character. Then *ist* will have the type $\langle i, \langle \tau_2\text{-Mod}, \langle \tau\text{-Char-Prop}, \tau\text{-Char-Prop}\rangle\rangle\rangle$.

These ideas are contrary in spirit to remarks in which McCarthy suggests that contexts should be formally treated as primitives. I take these remarks to mean that, although in applications we can axiomatize general knowledge about contexts, it is pointless to attempt to define contexts. Actually, I agree with McCarthy that relatively little of the work that needs to be done to explicate contexts can be done with definitions. But I do think it is enlightening and helpful to separate contexts into two components, one of them (the index) dealing with indexicality and ambiguity, while the other (the modality) deals with knowledge.

Before giving a definition of *ist* in CIL, it will be helpful to present an example. There are two personal databases, $\textsc{Db}(\text{Ann})$ and $\textsc{Db}(\text{Bob})$; both databases have a first-person pronoun, as well as constants referring to Ann and to Bob. $\textsc{Db}(\text{Ann})$ is Ann's database, and in it, 'I' refers to Ann. $\textsc{Db}(\text{Bob})$ is Bob's database, and in it, 'I' refers to Bob. Indices, then, can be identified with people. Assume that each world contains information about meetings, in the form of a set of triples whose whose first and second members are people, and whose third member is a time. They also contain information about databases, in the form of a set of quadruples whose first member is a database, whose second and third members are people, and whose fourth member is a time.

Ann's database induces a relation between worlds that holds between $w$ and $w'$ if and only if the following hold in $w'$, where 'I' refers to Ann:

(ADB.1) $\textsc{Meet}(\text{I}, \text{Charlie}, 9)$
(ADB.2) $\textsc{Meet}(\text{I}, \text{Bob}, 10)$
(ADB.3) $\textsc{Meet}(\textsc{Db}(\text{Ann}), \text{I}, \text{Charlie}, 9)$
(ADB.4) $\textsc{Meet}(\textsc{Db}(\text{Ann}), \text{I}, \text{Bob}, 9)$
(ADB.5) $\Box \forall u, x, y, z[\textsc{Meet}(\textsc{Db}(u), x, y, z) \rightarrow \textsc{Meet}(\textsc{Db}(u), y, x, z)]$

(ADB.6) $\Box\forall x, y, z[\textsc{Meet}(\textsc{Db}(x), x, y, z) \to \textsc{Meet}(\textsc{Db}(y), x, y, z)]$

Bob's database induces a relation between worlds that holds between $w$ and $w'$ if and only if the following hold in $w'$, where 'I' refers to Bob:

(BDB.1) $\textsc{Meet}(\mathrm{I}, \mathrm{Ann}, 10)$
(BDB.2) $\textsc{Meet}(\mathrm{I}, \mathrm{Charlie}, 11)$
(BDB.1) $\textsc{Meet}(\textsc{Db}(\mathrm{Bob}), \mathrm{I}, \mathrm{Ann}, 10)$
(BDB.2) $\textsc{Meet}(\textsc{Db}(\mathrm{Bob}), \mathrm{I}, \mathrm{Charlie}, 11)$
(BDB.3) $\Box\forall u, x, y, z[\textsc{Meet}(\textsc{Db}(u), x, y, z) \to \textsc{Meet}(y, x, z)]$
(BDB.4) $\Box\forall x, y, z[\textsc{Meet}(\textsc{Db}(x), x, y, z) \to \textsc{Meet}(\textsc{Db}(y), x, y, z)]$

The first and second conditions represent data; the third entry represents general knowledge about meetings. The fourth entry represents knowledge about knowledge of meetings; it ensures that, for instance, it is necessary (and hence, is known by all the databases) that if Ann's database knows that Ann is to meet with Bob at 9, then Bob's database knows that Ann is to meet with Bob at 9. In this case, we can show informally that

$$\mathit{ist}(\mathrm{Ann}, \textsc{Db}(\mathrm{Ann}), \mathit{ist}(\mathrm{Bob}, \textsc{Db}(\mathrm{Bob})\, \mathit{Ch}(\textsc{Meet}(\mathrm{I}, \mathrm{Ann}, 10))),$$

where $\mathit{Ch}(\textsc{Meet}(\mathrm{I}, \mathrm{Ann}, 10))$ is the character that holds at an index $a$ and world $w$ if and only if $\textsc{Meet}(a, \mathrm{Ann}, 10)$ holds in $w$. The argument goes as follows.

(i) Ann's database knows that Ann's database knows that Ann has a meeting at 10 with Bob.
(ii) So Ann's database knows that Ann's database knows that Bob has a meeting at 10 with Ann.
(iii) So Ann's database knows that Bob's database knows that Bob has a meeting at 10 with Ann.
(iv) The character expressed by 'I have a meeting at 10 with Ann' expresses in Bob's database the proposition that, according to (8.3), Ann's database knows that Bob's database knows.
(v) So there is a character expressing in Bob's database the same proposition that is expressed in Ann's database by the character 'Bob has a meeting at 10 with me', and Ann's database knows that Bob's database knows this proposition.
(vi) Restating (v),
$\mathit{ist}(\mathrm{Ann}, \textsc{Db}(\mathrm{Ann}), \mathit{ist}(\mathrm{Bob}, \textsc{Db}(\mathrm{Bob}), \mathit{Ch}(\textsc{Meet}(\mathrm{I}, \mathrm{Ann}, 10))))$

This leads to the following definition of *ist* in CIL.

(4.3) $\mathit{ist} = \lambda u_i \lambda x_{\tau_2\text{-Mod}} \lambda y_{\tau\text{-Char-Prop}} \lambda v_i \lambda z_w \exists y'_{\tau\text{-Char-Prop}}[y'(u) = y(v) \wedge \forall z'_w[x(z)(z') \to y'(u)(z')]$

Recall that in this example, we are identifying indices with people; and we can represent each of the two databases with an appropriate modality. Let $\mathrm{M}\textsc{Db}(\mathrm{Ann})$ and $\mathrm{M}\textsc{Db}(\mathrm{Bob})$ be the modaliites corresponding to Ann's and Bob's databases; also recall that these modalities are functions from propositions to propositions. Then $ist(\mathrm{Ann}, \textsc{Db}(\mathrm{Bob}), \textsc{Meet}(\mathrm{Bob}, \mathrm{I}, 10))$, for instance, says that the proposition expressed by 'Bob has a meeting with me at 10' true according to Ann's interpretation of 'I' in Bob's database.

For example, it follows from this definition that

$$\text{(4.4)}\quad ist(\mathrm{Ann}, \textsc{Db}(\mathrm{Ann}), ist(\mathrm{Bob}, \textsc{Db}(\mathrm{Bob}), \textsc{Meet}(\mathrm{I}, \mathrm{Ann}, 10)))$$

returns a propositional character that expresses (at any index) a proposition that holds in $w$ if and only if there is a propositional character $cp$ that expresses for Ann a proposition that is the same as the one expressed for Bob by $Ch(\textsc{Meet}(\mathrm{I}, \mathrm{Ann}, 10))$, and this proposition is known in $w$ by $\textsc{Db}(\mathrm{Ann})$.

In the version of the theory presented here, where all functions are total functions and quantifiers are unrestricted, it will be trivially true for any proposition and any index that there is a character expressing the proposition at this index. Therefore, (4.3) is equivalent to the following simplified definition.

$$\text{(4.5)}\quad ist = \lambda u_i \lambda x_{\tau_2\text{-Mod}} \lambda y_{\tau\text{-Char-Prop}} \lambda v_i \lambda z_w \forall z'_w [x(z)(z') \rightarrow y(v)(z')]$$

Since the variable $u$ does not appear on the right side of this equation, it follows that $ist(\iota, \mu, \xi)$ will be a constant character, one whose value is the same for all values of $x$. Although *ist* statements may vary from context to context, this variation can only be due to what the context knows about other contexts—it cannot be due to how the context disambiguates characters.

In a partial version of CIL, (4.3) would not be equivalent to (4.5), so that indices could play a significant role in iterations of *ist*. The equivalence would fail in cases where there are propositions expressible at one index that are not expressible than the other. In cases where all indices are equally expressive, however, the index-independence of *ist* is not implausible. At any rate, I have been unable to construct a plausible case of index-dependence of *ist* where all indices are equally expressive and knowledge of other contexts is not a factor.

## 5 Conclusion

There are several dimensions in which the logical framework that I have presented needs to be generalized in order to obtain adequate coverage:

(5.1) The logic needs to be made partial, to account for expressions which simply lack a value in some contexts.
(5.2) The logic needs dynamic operators of the sort described in McCarthy's papers; e.g., an operator which chooses a context and *enters* it.
(5.3) To account for default lifting rules, we need a nonmonotonic logic of context.

We have a general sense of what is involved in making a total logic partial, in making a static logic dynamic, and in making a monotonic logic nonmonotonic. For these reasons, I have adopted the strategy of concentrating on how to formulate an appropriate base logic to which these extensions can be made.

There are a number of approaches to the formalization of partial logics; indeed, the main problem with the logic of partiality, it seems to me, is that there are so many alternatives, and it is hard to select between them. Three-valued logic has been used in connection with the logic of context; see [3]. However, a four-valued logic is more symmetrical, and plausible arguments, starting with [2], have been given for its computational usefulness. Most important for the project at hand, [13] provides an extended study of how to modify Intensional Logic using this approach to partiality. It is relatively straightforward to adopt Muskens' work to CIL.

There is, however, a much more ambitious application of partiality, according to which indices are regarded not as full, but as partial disambiguations of expressions. This program, which would require a more radical rethinking of the theory, may be needed to deal with applications of context to natural language interpretation, though perhaps it is unnecessary in cases in which indices correspond to carefully constructed knowledge sources. See, for instance, [17] for information on partial disambiguation.

For dynamics, I favor an approach along the lines of [8]. This involves relativizing satisfaction not to just one index, but to a pair of indices, an input index and an output index. The resulting logic would be dynamic with respect only to indices, not to modalities. A more general dynamic contextual logic could be devised, but I'm not sure whether we really need such a logic.

As for nonmonotonicity, it is relatively straightforward to add a theory of circumscription to Intensional Logic, and to its extension to CIL. (Circumscription is usually formulated in second-order extensional logic, but the generalization to intensional logic of arbitrary order is straightforward.)

None of these logical developments is entirely trivial, and in fact there is material here for many years of work. I hope to report on developments in these directions in future work.

## References

1. C. Anthony Anderson. General intensional logic. In Dov Gabbay and Franz Guenther, editors, *Handbook of Philosophical Logic, Volume II: Extensions of Classical Logic*, pages 355–385. D. Reidel Publishing Co., Dordrecht, 1984.
2. Nuel D. Belnap, Jr. A useful four-valued logic. In J.M. Dunn and G. Epstein, editors, *Modern Uses of Multiple-Valued Logic*. D. reidel Publishing Co., Dordrecht, 1977.
3. Saša Buvač and Ian Mason. Propositional logic of context. In Richard Fikes and Wendy Lehnert, editors, *Proceedings of the Eleventh National Conference on Artificial Intelligence*, pages 412–419, Menlo Park, California, 1993. American Association for Artificial Intelligence, AAAI Press.
4. Alonzo Church. A formulation of the simple theory of types. *Journal of Symbolic Logic*, 5:56–68, 1940.

5. Ronald Fagin, Joseph Y. Halpern, Yoram Moses, and Moshe Y. Vardi. *Reasoning About Knowledge*. The MIT Press, Cambridge, Massachusetts, 1995.
6. Daniel Gallin. *Intensional and Higher-Order Logic*. North-Holland Publishing Company, Amsterdam, 1975.
7. Fausto Giunchiglia, Luciano Serafini, Enrico Giunchiglia, and Marcello Frixione. Non-omniscient belief as context-based reasoning. In Ruzena Bajcsy, editor, *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*, pages 548–554, San Mateo, California, 1993. Morgan Kaufmann.
8. Jeroen Groenendijk and Martin Stokhof. Dynamic predicate logic. *Linguistics and Philosophy*, 14:39–100, 1991.
9. David Kaplan. On the logic of demonstratives. *Journal of Philosophical Logic*, 8:81–98, 1978.
10. John McCarthy and Saša Buvač. Formalizing context (expanded notes). Available from http://www-formal.stanford.edu/buvac., 1995.
11. Richard Montague. Pragmatics and intensional logic. *Synthèse*, 22:68–94, 1970. Reprinted in *Formal Philosophy*, by Richard Montague, Yale University Press, New Haven, CT, 1974, pp. 119–147.
12. Yorav Moses. Resource-bounded knowledge. In Moshe Y. Vardi, editor, *Theoretical Aspects of Reasoning About Knowledge: Proceedings of the Second Conference on Theoretical Aspects of Reasoning About Knowledge*, pages 261–276, San Francisco, 1988. Morgan Kaufmann.
13. Reinhard Muskens. *Meaning and Partiality*. Cambridge University Press, Cambridge, England, 1996.
14. Matthew Stone. *Modality in Dialogue: Planning, Pragmatics and Computation*. Ph.d. dissertation, Computer Science Department, University of Pennsylvania, Philadelphia, Pennsylvania, 1998.
15. Richmond H. Thomason. Type theoretic foundations for context. In Sasa Buvač and Łucia Iwańska, editors, *Working Papers of the AAAI Fall Symposium on Context in Knowledge Representation and Natural Language*, pages 173–175, Menlo Park, California, 1997. American Association for Artificial Intelligence, American Association for Artificial Intelligence.
16. Richmond H. Thomason. Representing and reasoning with context. In Jacques Calmet and Jan Plaza, editors, *Artificial Intelligence and Symbolic Computation; Proceedings of the International Conference AISC'98, Plattsburgh, New York*, pages 29–41, Berlin, 1998. Springer-Verlag.
17. Kees van Deemter and Stanley Peters, editors. *Semantic Ambiguity and Underspecification*. Cambridge University Press, Cambridge, England, 1996.