A Basic Logic for Textual Davidsonian Inference

D. Bobrow, C. Condoravdi, R. Crouch, R. Kaplan, L. Karttunen, T. King, V. de Paiva, A. Zaenen
Palo Alto Research Center
3333 Coyote Hill Road
Palo Alto 94304

Introduction

This note describes the basics of a logical system, based on concepts and contexts, whose aim is to serve as a representation language for meanings of natural language sentences in the system that we are developing as part of the Aquaint framework. Some higher level discussion of our system and the rationale behind it can be found in (Crouch 2005; Condoravdi *et al.* 2003; Crouch *et al.* 2002). As explained in (Crouch 2005) our system maps textual sentences to f-structures and these to logical formulae, using the "Glue language" approach for the construction of the meanings of the sentences. The output of glue semantics is then flattened to produce sets of clauses of the logic we discuss in this paper.

In this note we simply describe the logical system, as it can be distilled from the system's output, with the caveat that as logicians we are able to simplify matters as much as we deem necessary. This logic is based on the idea of using events in a Neo-Davidsonian style. We will not discuss how pre-theoretical notions of event (especially in natural language semantics) relate to the events here, which are just representations, formal strings of symbols, describing predications.

We start by describing the language of the proposed logic, to be called TIL for textual inference logic. The system TIL is a simplified logic of concepts and contexts, that is, a kind of "contexted" description logic (Baader *et al.* 2003), but one that does not make use of instances or individuals. Our contexts are similar to McCarthy's contexts (McCarthy 1993). We give ourselves a collection of contexts, which we write as $c_0, c_1, c_2, \ldots$, the whole collection is written as Context. We also have a collection of *concepts*, for which, to avoid difficulties with names, we use as variables $t_1, t_2, \ldots$, thinking of types. The whole collection we write as Concept.

Our concepts are of two very different kinds. Some are a priori given, sitting in a hierarchy, based on the hierarchy underlying the CYC knowlege base (Lenat 1995). These concepts we write, following the CYC notation, using a typescript font, like `MovementTranslationEvent` or `City`. Other concepts are built from the ones just mentioned via a series of operations which we describe in the next section. They also sit in the same hierarchy, whose most underspecified concept is called `Thing`. These concepts, which we write without the typescript font, are dynamically created by the system, in the process of interpreting the input text. So they come with names with arbitrary numbers, which correspond to skolem variables. For example, for a sentence like *Ed arrived in the city*, we will create an arrival event concept, written as $arriveEv1$, a concept for the person Ed, written as $Ed2$, and a concept for city, $city3$ related in the predicatable manner, specified in the next section.

We have several relations between concepts, between contexts, and between concepts and contexts. First, we discuss relations between concepts, then relations between concepts and contexts, and finally relations between contexts. Then we discuss some of the inferences that our logic should do. Finally we discuss some of things that our logic will have to do eventually, but does not do at the moment.

Relations Between Concepts

We start by describing the basic relations between concepts in Concept. We have two main kinds of relations, first the *subconcept* relation between two concepts and second, the notion of a *role* between two concepts.

Subconcepts

Our concepts are organized into a hierarchy and the subconcept relation just expresses this a priori given order. We could write this as $t_1 \sqsubseteq t_2$, but we prefer to write $subconcept(t_1, t_2)$. For example, if we consider a specific class of arrival events which we write as $arriveEv1$, then we place this in the a priori hierarchy by declaring that $arriveEv1$ is a subconcept of `ArrivingAtAPlace`, a CYC concept in the hierarchy. This we write as

$$subconcept(arriveEv1, \texttt{ArrivingAtAPlace})$$

The notion of subconcept serves to locate the new concepts of our logical system with respect to the concepts given by the CYC hierarchy, but the relation is general and given two concepts like $arriveEv1$ and $arriveEv100$ it makes sense to ask whether one is a subconcept of the other. We assume that concepts inherit from one or more concepts and that there are no circularities or inconsistencies in the given hierarchy.

Roles

We also take as given a collection of relations between concepts that we think of as *roles*, relating concepts. For example if we consider the previous concept of an arrival event $arriveEv1$ and the concept of a person called Ed, which we write as a concept $Ed2$, then an example of a role relation between these two concepts is that the concept $Ed2$ might play the role of `objectMoving` for the arrival event $arriveEv1$. This we will write as

$$\texttt{objectMoving}(arriveEv1, Ed2)$$
$$subconcept(Ed2, \texttt{Person})$$
$$subconcept(arriveEv1, \texttt{ArrivingAtAPlace})$$

Note that the roles are always binary relations and are written in a infix notation, i.e. $role(t_1, t_2)$ and are generally (but not always) *actor slots* in CYC. In general we write these relations between concepts as $role_1, role_2, \ldots$ and the whole collection is written as Role.

Syntax so far

The logical system we are describing looks like a description logic. We have concepts Concept with their own partial order (written as $subconcept(t_1, t_2)$) and roles Role, which are binary relations on Concept. We write *clauses* that either relate concepts via subconcept relations or relate roles to pairs of concepts, like $\texttt{objectMoving}(arriveEv1, Ed2)$ in the example above. Note that these roles are obtained in the mapping from language to logic from the grammatical relations that relate verbs to their arguments.

We operate on these collections of subconcept and role clauses by adding some more of the same. Adding to the example above, we could say that the same arrival event $arriveEv1$ was connected via a `toLocation` role to the concept of a city, say $city3$, which we would write as

$$\texttt{objectMoving}(arriveEv1, Ed2)$$
$$\texttt{toLocation}(arriveEv1, city3)$$
$$subconcept(arriveEv1, \texttt{ArrivingAtAPlace})$$
$$subconcept(Ed2, \texttt{Person})$$
$$subconcept(city3, \texttt{City})$$

This intuitively says that there was an arrival event, whose object moving is Ed and whose location of arrival is a specific city. Thus it corresponds, roughly, to a sentence like *Ed arrived in the city*.

More generally given, say, concepts t_1, t_2, t_3, t_4, together with clauses like $subconcept(t_1, t_3)$, $subconcept(t_1, t_4)$ and $role_1(t_1, t_2)$, $role_2(t_1, t_3)$ we can add new concepts t_7, t_8, t_9, new roles $role_3$, $role_4$ and new subconcept relations, and so far, this is all we can do.

Contexts and Concepts

However, our simple logic has contexts Context, as well as concepts. This means all of the relations above should hold within a given context. In particular, there is a first initial context (written as top or c_0) that corresponds roughly to what we take the world to be like. More precisely, in an interpretation of a sentence, the top context corresponds to what the author of the sentence is committed to, about the world she is describing. Since this circumlocution is awkward, we will usually talk about this top level context as the "true" context.

Apart from the true context, what other contexts are there in our logic? Contexts in our logic support making statements about the existence and non-existence of entities that satisfy the intensional descriptions specified by our concepts. We first discuss the class of propositional attitude contexts.

Propositional Attitudes

Traditional *propositional attitudes* relate contexts and concepts. Thus a concept like 'knowing' or 'believing' or 'saying' introduces a context that represents the proposition that is known, believed or said.

For example, if we want to represent the sentence *Ed believes that the diplomat arrived*, we will need concepts for the arrival event, for the believing event, for the diplomat and for Ed. Assuming all of these to be subconcepts of reasonable concepts, we end up with roles that describe how they relate. Thus there might be a role `believer` of the believing event and this should relate the believing event to the concept associated with Ed, $\texttt{believer}(believeEv1, Ed2)$. There is a role `whatsBelieved` which relates a concept, the concept `believeEv1` to (the contents of) a context c_1, giving us $\texttt{whatsBelieved}(believeEv1, c_1)$. Also the contents of what is believed in the believing event is the proposition that *The diplomat arrived*.

We represent the concepts (intensional descriptions) used in this sentence as:

$$\texttt{believer}(believeEv1, Ed2)$$
$$\texttt{whatsBelieved}(believeEv1, c_1)$$
$$subconcept(Ed2, \texttt{Person})$$
$$subconcept(believeEv1, \texttt{CognitionEvent})$$
$$\texttt{objectMoving}(arriveEv3, diplomat4)$$
$$subconcept(diplomat4, \texttt{Diplomat})$$
$$subconcept(arriveEv3, \texttt{ArrivingAtAPlace})$$

In this set of clauses, the first four lines represent the believing event, and the last three ones represent the arrival event.

The reason we introduced contexts in our logic is to be able to localize reasoning. While the existence of the believing concept and of Ed are supposed to be true in the real world, the existence of the arrival of the diplomat is only supposed to be true in those worlds compatible with what Ed believes. Thus we introduce two new classes of clauses, *instantiable* and `contextRelation` clauses. Intuitively the *instantiable* predicate takes a context and a concept and specifies that the concept is instantiable in that context. The `contextRelation` clauses indicate the relationship between two contexts induced by the meaning of the lexical item: $\texttt{contextRelation}(believe, top, c_1)$ says that there is a relation induced by the believing concept between the contexts top and c_1, where c_1 is the context of what is believed by Ed.

The following clauses augment/use the representation above to make those statements:

$$
\begin{array}{ll}
top: & \texttt{contextRelation}(believe, top, c_1) \\
 & instantiable(believeEv1) \\
 & instantiable(Ed2) \\
c_1: & instantiable(diplomat4) \\
 & instantiable(arriveEv3)
\end{array}
$$

This representation is not faithful to our intuitions, as the existence of the diplomat is not restricted to the context c_1. It is the arrival of the diplomat that needs to exist only in the world of the things believed by Ed. Thus we push the clause $instantiable(diplomat4)$ to the top context, but we keep the instantiability of the arrival event $arriveEv3$ restricted to the context c_1.

$$
\begin{array}{ll}
top: & \texttt{contextRelation}(believe, top, c_1) \\
 & instantiable(Ed2) \\
 & instantiable(Ev1) \\
 & instantiable(diplomat4) \\
c_1: & instantiable(arriveEv3)
\end{array}
$$

Consider the similar sentence *Ed said that the diplomat arrived*. This also sets up a context containing the contents of the information transferred by Ed, hence we get the new concepts:

$$
\begin{array}{l}
\texttt{senderOfInfo}(sayEv1, Ed2) \\
\texttt{informationTransferred}(sayEv1, c_1) \\
subconcept(Ed2, \texttt{Person}) \\
subconcept(sayEv1, \texttt{InformCommunicationAct})
\end{array}
$$

and the statements about instantiability

$$
\begin{array}{ll}
top: & \texttt{contextRelation}(infTransfer, top, c_1) \\
 & instantiable(Ed2) \\
 & instantiable(sayEv1) \\
 & instantiable(diplomat4) \\
c_1: & instantiable(arriveEv3)
\end{array}
$$

As before the claim about the existence of the diplomat is not restricted to the context c_1. The top context tells us that Ed said that the diplomat arrived; things Ed says may well be false, even if Ed is a very reliable source. So while it is perfectly fine to say that the arrival of the diplomat is instantiable in the context, c_1, of the things said by Ed, we certainly do not want to push this instantiability up to the top context.

The examples lead us to a new kind of relation between contexts. We say that the context c_1 introduced by the believing concept, like the one introduced by the saying event, is *averidical* with respect to the initial context top. In the next sections we consider also *veridicality* and *antiveridicality* between contexts.

Negation Contexts

Consider the sentence *The diplomat did not arrive*. One way of representing the information conveyed by this sentence is to consider all the worlds where the specific diplomat we are thinking of did arrive (at the appropriate place) and then 'subtract'[1] the collection of worlds where this event takes place from the collection of all possible worlds. In this case we have the same concepts of diplomat and of an arriving event that we had before, but we have two contexts, the top (or true) one where there was no arrival by the diplomat and the negated context, where the arrival of the diplomat did happen.

For the sentence *The diplomat did not arrive*, we know that the concept $arriveEv1$ is instantiable in the negated context, but not in the top or true one. We end up with a representation like

$$
\begin{array}{ll}
 & \texttt{objectMoving}(arriveEv1, diplomat2) \\
 & subconcept(diplomat2, \texttt{Diplomat}) \\
 & subconcept(arriveEv1, \texttt{ArrivingAtAPlace}) \\
top: & \texttt{contextRelation}(not, top, c_1) \\
 & \texttt{antiveridical}(c_1, top) \\
 & instantiable(diplomat2) \\
c_1: & instantiable(arriveEv1)
\end{array}
$$

As with the saying event, we expect that the concept $diplomat2$ can be instantiable in the true context, but the arrival event should not. Moreover, in this case, since the negated context is `antiveridical`, instead of simply saying that the $arriveEv1$ is instantiable in the negated context, we want to say that it is *not* instantiable in the top context. Thus we simplify the instantiability clauses above to

$$
\begin{array}{ll}
top: & \texttt{contextRelation}(not, top, c_1) \\
 & \texttt{antiveridical}(c_1, top) \\
 & instantiable(diplomat2) \\
 & uninstantiable(arriveEv1)
\end{array}
$$

As motivation for making the uninstantiable statement in the top context, we can look at the interpretation of contexts as sets of possible worlds. Using the uninstantiability statement in the top context is analogous to interpretating the negated context as subtracting out a set of worlds in which the arrival is instantiated.

Having introduced relations between two contexts, we have expanded the expressive power of our language of representations considerably. So far, we have a fairly general (nestable) mechanism that can represent some positive and some negative information, mostly in a 'conjunctive' form. In the next section we discuss possible ways of representing disjunctive and implicative information.

Other Logical Relations Between Contexts

To deal with disjunctive information like "Either the diplomat arrived or the car broke" we introduce a four-place version of the `contextRelation` predicate with two subcontexts that are 'arms' of the disjunction – in the example below, orA and orB, are arms of the disjunction introduced in the dontext top.

[1] This view of negation is constructively appealing, as it transforms negation into some sort of implication into falsum. (Think of implication in possible worlds as assuming the antecedent, checking where the worlds that satisfy the antecedent can take us and verifying that the consequent is satisfied there.)

$$
\begin{array}{ll}
 & \texttt{objectMoving}(arriveEv1, diplomat4) \\
 & \texttt{objectActedOn}(breakEv2, car6) \\
 & subconcept(arriveEv1, \texttt{ArrivingAtAPlace}) \\
 & subconcept(diplomat4, \texttt{Diplomat}) \\
 & subconcept(breakEv2, \texttt{BreakingEvent}) \\
 & subconcept(car6, \texttt{Automobile}) \\
top: & \texttt{contextRelation}(or, top, orA3, orB4) \\
 & instantiable(diplomat4) \\
 & instantiable(car6) \\
orA3: & instantiable(arriveEv1) \\
orA4: & instantiable(breakEv2)
\end{array}
$$

We can have a disjunction of events sharing one argument. For example, if we have *"The diplomat watched television or slept"*, our representation will construct a new event, here `group-ev1`, of which the events of watching television and sleeping are sub-events. The representation will be something like:

$$
\begin{array}{ll}
 & \texttt{evmember}(ev4, groupEv1) \\
 & \texttt{evmember}(ev7, groupEv1) \\
 & \texttt{performedBy}(ev4, diplomat6) \\
 & \texttt{perceivedThings}(ev4, television5) \\
 & \texttt{bodilyDoer}(ev7, diplomat6) \\
 & subconcept(ev4, \texttt{WatchingSomething}) \\
 & subconcept(diplomat6, \texttt{Diplomat}) \\
 & subconcept(ev7, \texttt{Sleeping}) \\
top: & \texttt{contextRelation}(or, top, orA2, orB3) \\
 & instantiable(diplomat6) \\
 & instantiable(television5) \\
orA2: & instantiable(ev7) \\
orB3: & instantiable(ev4)
\end{array}
$$

In this case the disjunctive event `groupEv1` has two subevents `ev4` and `ev7`, respectively the watching and the sleeping events, but these happen in *separate* contexts, `orA2` and `orB3`, respectively.

Similarly, logical implications introduce new contexts, one antecedent context and one consequent context, which have to be connected to the relevant whole context. In the example "If the reporter arrived the diplomat will leave" we have

$$
\begin{array}{ll}
 & \texttt{objectMoving}(arriveEv2, reporter5) \\
 & \texttt{objectMoving}(leaveEv1, diplomat6) \\
 & subconcept(arriveEv2, \texttt{ArrivingAtAPlace}) \\
 & subconcept(leaveEv1, \texttt{LeavingAPlace}) \\
 & subconcept(reporter5, \texttt{Journalist}) \\
 & subconcept(diplomat6, \texttt{Diplomat}) \\
top: & \texttt{contextRelation}(if, top, ante4, consq3) \\
 & \texttt{precondForEvents}(arriveEv2, leaveEv1) \\
 & instantiable(diplomat6) \\
 & instantiable(reporter5) \\
ante4: & instantiable(arriveEv2) \\
consq3: & instantiable(leaveEv1)
\end{array}
$$

Inferences

Our aim in this section is to show the kinds of inferences that this simple logic of concepts and contexts already allows.

The reason for introducing events (here treated as concepts) in neo-Davidsonian semantics was the fact that it makes it very easy to make inferences that can be complicated in other semantic traditions. For example it should be obvious how to obtain *"Ed arrived in the city"* from the sentence *"Ed arrived in the city by bus"*. This is clearly so in our logic, where this inference corresponds simply to clause (conjunction) dropping.

The sentence *"Ed arrived in the city by bus"* corresponds to the set of clauses

$$
\begin{array}{ll}
top: & \texttt{objectMoving}(arriveEv1, Ed4) \\
 & \texttt{toLocation}(arriveEv1, city3) \\
 & \texttt{vehicle}(arriveEv1, bus2) \\
 & subconcept(arriveEv1, \texttt{ArrivingAtAPlace}) \\
 & subconcept(Ed4, \texttt{Person}) \\
 & subconcept(city3, \texttt{City}) \\
 & subconcept(bus2, \texttt{BusRoadVehicle})
\end{array}
$$

From these clauses we can infer

$$
\begin{array}{ll}
top: & \texttt{objectMoving}(arriveEv1, Ed4) \\
 & \texttt{toLocation}(arriveEv1, city3) \\
 & subconcept(arriveEv1, \texttt{ArrivingAtAPlace}) \\
 & subconcept(Ed4, \texttt{Person}) \\
 & subconcept(city3, \texttt{City})
\end{array}
$$

which corresponds to *"Ed arrived in the city"* omitting the two clauses:

$$
\begin{array}{ll}
top: & \texttt{vehicle}(arriveEv1, bus2) \\
 & subconcept(bus2, \texttt{BusRoadVehicle})
\end{array}
$$

Note that a limited amount of 'going up and down the taxonomy' can also be accounted for in this framework, at least in principle. So *"Ed arrived in the city"* does entail that *"A person arrived in the city"*, since `Ed4` is a subconcept of `Person`. Finally since "arriving" is a kind of movement event, *"Ed arrived in the city"* can be 'generalized' to *"Ed went to the city"*, but care is required when relating events by subsumption.

Note also that the clauses we construct satisfy the usual monotonicity patterns, both in positive and in negative form. Thus *"Ed arrived in the city by bus"* entails that *"Ed arrived in the city"*. But *"Ed did not arrive in the city"* entails that *"Ed did not arrive in the city by bus"*, while *"Ed did not arrive in the city by bus"* does *not* entail that *"Ed did not arrive in the city"*.

Inferences and Contexts

The main reason for introducing contexts in our logic is to create boundaries corresponding loosely to the idea that contexts create certain opaque boxes that purely extensional reasoning should respect. Thus the top context corresponds to what the author of the sentence takes the described world to be like and when confronted with intensional notions like 'knowledge' or 'believe' or 'desire', we create one of these boxes and segregate what is known, believed or wanted into that box. The idea of segregating the contents of a propositional atittude predicate into its own context corresponds to the semantic notion that propositional atittudes introduce alternative worlds into the discourse.

We want to keep the restrictions on what we know about these boundaries as underspecified as possible, minimizing the number of conditions (or axioms) on these boxes. We therefore assume that these boxing operators satisfy simply the usual K axiom, which can be read as saying that this notion of context distributes over conjunction (or if one prefers, as saying that this very weak notion of context respects modus ponens).

While it seems important to keep our system as general and underspecified as possible, it would be better for reasoning with our representations if we had more information about different kinds of contexts.

As a simple example of the kind of inferences that the interaction between contexts brings about, consider the sentence *Ed knows that the diplomat arrived*. As before we will need concepts for the arrival event, for the knowing event, for the diplomat and for Ed. Assuming all of these concepts and roles, as before, we represent this sentence as:

$$
\begin{array}{ll}
 & \texttt{knower}(knowEv1, Ed2) \\
 & \texttt{whatsKnown}(knowEv1, c_1) \\
 & subconcept(Ed2, \texttt{Person}) \\
 & subconcept(knowEv1, \texttt{CognitionEvent}) \\
 & \texttt{objectMoving}(arriveEv3, diplomat4) \\
 & subconcept(diplomat4, \texttt{Diplomat}) \\
 & subconcept(arriveEv3, \texttt{ArrivingAtAPlace}) \\
top & \texttt{contextRelation}(know, top, c_1) \\
 & instantiable(knowEv1) \\
 & instantiable(Ed2) \\
 & instantiable(diplomat4) \\
c_1: & instantiable(arriveEv3)
\end{array}
$$

In principle the arrival of the diplomat is only supposed to be true in these worlds compatible with what Ed knows. However, we can make the simplifying assumption that everything true in a *know* context is also true in the containing context. This notion of veridicality of the contained context, c_1 with respect to its container top is represented by $\texttt{veridical}(c_1, top)$. This veridicality assumption allows us to move $instantiable(arriveEv3)$ to the top context.

More formally, if $\texttt{contextRelation}(know, c_1, c_2)$ holds then we may want to say that all instantiability statements in c_2 can be moved up to context c_1. We plan to build on the distinctions beween the contexts introduced by "know" and "say" to discuss a large class of 'factive' verbs. The kind of reasoning mechanism at play here seems to be of the following form. If c is a veridical context with respect to the context top and if t is a concept instantiable in c then t is instantiable in the more general context top too, or:

$$\frac{c : instantiable(t) \qquad \texttt{veridical}(c, top)}{top : instantiable(t)}$$

More generally:

$$\frac{c_i : instantiable(t) \qquad \texttt{veridical}(c_i, c_j)}{c_j : instantiable(t)}$$

We previously used:

$$\frac{c_i : instantiable(t) \qquad \texttt{antiveridical}(c_i, c_j)}{c_j : uninstantiable(t)}$$

More work is required to check whether all inferences we want can be accounted for using such simple mechanisms.

Further Work

This short note only starts the discussion of the kinds of inferences that we expect to be able to make using a simple logic of concepts and contexts. Amongst the issues we have not discussed are the following: how to deal with temporal modifiers and temporal interpretation in general; how to deal with presupposition and conventional implicatures in general; what to say about cardinality and plurality as well as about adjectival and adverbial modification. We hope to discuss these as the project unfolds.

References

Baader, F.; Calvanese, D.; McGuiness, D.; Nardi, D.; and Patel-Schneider, P. 2003. *The Description Logic Handbook*. Cambridge University Press.

Condoravdi, C.; Crouch, D.; Stolle, R.; de Paiva, V.; and Bobrow, D. 2003. Entailment, intensionality and text understanding. In *Proceedings Human Language Technology Conference, Workshop on Text Meaning, Edmonton, Canada*.

Crouch, D.; Condoravdi, C.; Stolle, R.; King, T.; de Paiva, V.; O.Everett, J.; and Bobrow, D. 2002. Scalability of redundancy detection in focused document collections. In *Proceedings First International Workshop on Scalable Natural Language Understanding (ScaNaLU-2002), Heidelberg, Germany*.

Crouch, R. 2005. Packed rewriting for mapping semantics to kr. In *Proceedings Sixth International Workshop on Computational Semantics, Tilburg, The Netherlands*.

Lenat, D. 1995. Cyc: A large-scale investment in knowledege infrastructure. *Communications of the ACM, 38(11):33–38*.

McCarthy, J. 1993. Notes on formalizing context. In *Proc. of the 13th Joint Conference on Artificial Intelligence (IJCAI-93)*.