Overview

We introduce a version of dynamic predicate logic (DPL, see Groenendijk&Stokhof (1991)) as a framework to model in a compositional way the dynamics of definite descriptions put forward in Lewis (1979). The resulting system, dynamic predicate logic with descriptions (DPLD) borrows the concept of referent system from an upgraded version of DPL introduced in Groenendijk&Stokhof&Veltman (1996). It is an interesting feature of the system that not only formulas but also individual terms are capable of updating discourse information.

The dynamics of definite descriptions

It is well-known that standard first-order logic (PL) cannot deal with definite descriptions in a compositional way. It is less well-known that dynamic predicate logic solves this problem. This is an obvious (but so far apparently unnoticed) consequence of its refined manner of handling anaphoric relations.

(1) The present king of France is bald. He shaves himself.

\[ (1_{PL}) \exists x (\forall y (\text{present\_king\_of\_France}(y) \leftrightarrow y = x) \land \text{bald}(y) \land \text{shave}(x, x)) \]

\[ (1_{DPL}) \exists x (\forall y (\text{present\_king\_of\_France}(y) \leftrightarrow y = x) \land \text{bald}(y) \land \text{shave}(x, x)) \]

While ‘the present king of France’ is a component in (1), its first-order translation, \( \exists x (\forall y (\text{present\_king\_of\_France}(y) \leftrightarrow y = x)) \) is not a component in (1_{PL}). Nevertheless, it is present in the formula (1_{DPL}). Similarly, the translation of the first sentence is a component of (1_{DPL}), but not of (1_{PL}).

However, not even DPL can handle the dynamics of definite descriptions put forward in David Lewis’ seminal paper Lewis (1979). According to Lewis, examples like “The dog got in a fight with another dog” show that “[i]t is not true that a definite description ‘the F’ denotes x if and only if x is the one and only F in existence. Neither is it true that ‘the F’ denotes x if and only if x is the one and only F in some contextually determined domain of discourse.” (p348) Instead, “[t]he proper treatment of descriptions must be more like this: ‘the F’ denotes x if and only if x is the most salient F in the domain of discourse, according to some contextually determined salience ranking.” (ibid.)

Salience may be determined by the context of utterance or antecedent expressions. We focus on the second case, especially on occurrences of definite descriptions that anaphorically refer to discourse referents introduced previously, as in the following discourse:

(2) A man walks in the park. He meets a woman. The man hugs her. A man watches from a distance. He walks a dog. The dog sniffs. The man is jealous.

Here, every occurrence of a definite description is bounded by an occurrence of an indefinite expression in a previous sentence. For example, “the man” in the third sentence is anaphorically linked to “a man” in the first one, while the same expression in the seventh sentence is bounded by “a man” in the fourth one.

Lewis’ reasoning shows that the use of definite descriptions like the ones in the above example excludes Russell’s analysis of descriptions as incomplete quantified sentences. But it does not exclude Frege’s old description operator \( \iota \) as a means of analysis. This leads to the following translation to the language of dynamic predicate logic enriched with a Fregean description operator (DPLD):

\[ (2_{DPLD}) \exists x (\text{man}(x) \land \text{walk\_in\_the\_park}(x) \land \exists y (\text{woman}(y) \land \text{meets}(x, y) \land \text{hug}(\iota w \text{man}(w), y) \land \exists x (\text{watch\_from\_a\_distance}(x) \land \exists u (\text{dog}(v) \land \text{walk}(u, v) \land \text{sniff}(\iota w \text{dog}(w)) \land \text{jealous}(\iota w \text{man}(w))))\]
Salience ranking in this discourse is quite obvious: every definite description with a predicate $P$ is bound by the nearest preceding indefinite expression with the same predicate. However, there are less obvious cases. Consider the following examples.

(3) The cat is black.

$$3_{\text{DPLD}} \quad \textbf{black}(\iota x \text{cat}(x))$$

(4) John has a cat. It is white. Mary has a cat, too. It is black. The white one is beautiful.

$$4_{\text{DPLD}} \quad \exists x \, \text{cat}(x) \land \text{have}(\text{John}, x) \land \text{white}(x) \land \exists y \, \text{cat}(y) \land \text{have}(\text{Mary}, y) \land \text{black}(y) \land \text{beautiful}(\iota z \, \text{white}(x))$$

(5) John has a daughter and a son. The boy is asleep.

$$5_{\text{DPLD}} \quad \exists x \, \text{daughter}(\text{John}, x) \land \exists y \, \text{son}(\text{John}, y) \land \text{asleep}(\iota z \, \text{boy}(z))$$

There is no antecedent to the definite description in (3). Since we don’t attempt to model saliances that are due to the context of utterance, it is quite intuitive that the description itself introduces a discourse referent, and thus at the given point of the discourse $3_{\text{DPLD}}$ is equivalent with

$$3'_{\text{DPLD}} \quad \exists x \text{cat}(x) \land \text{black}(x)$$

Of course, the equivalence does not generally hold.

As for example (4), the definite description in the third sentence is bounded by the indefinite expression in the first, despite the fact that the predicate in the description is not the same as the one in the indefinite expression. The link between the two is the occurrence of the intermediate predicate ‘white’ in the second sentence. With the predicate ‘white’ the second sentence refreshes the salience information attached to the discourse referent introduced in the first sentence.

Example (5) is a problematic one. It is by no means obvious that synonymy or meaning inclusion itself is enough to establish an anaphoric connection. We choose not to deal with this problem. Thus, in our framework salience information connects ($n$-tuples of) discourse referents to predicates, not to their intensions.

**DPLD: a system of dynamic predicate logic with descriptions**

The syntax of DPLD is the same as that of DPL, with the addition of a clause that governs the use of atomic descriptions:

- If $P$ is an $n$-argument predicate, $x$ a variable, $t_1, \ldots, t_n$ are terms, then $\iota x \, P(t_1, \ldots, t_n)$ is a term.

We restrict to atomic formulas as the scope of the $\iota$ operator. This is not necessary, but most natural language examples of dynamic descriptions are atomic.

To provide this clause with sufficient semantics is far from obvious. In DPL, terms are evaluated in a static way. However, a definite description, besides having a reference (if any), is also capable of updating salience information. This is most explicit when there is no antecedent expression and the description itself introduces a discourse referent. As we will see, in DPLD even a single variable occurrence may update discourse information. Thus, the updating process in DPLD is more elaborate than usual.

Like in DPL, the basis of the evaluation process is an ordinary first-order model $\langle D, \varrho \rangle$, consisting of a domain and an interpretation function. These are treated statically. Besides the model, at any stage of the evaluation process, we need three pieces of discourse information:
i a referent system that connects the variables introduced so far in the discourse to discourse referents;

ii a salience ranking that connects predicates to n-tuples of;

iii a set of assignment functions that map the set of discourse referents into the domain of discourse.

These components are updated both by occurrences of terms and occurrences of sentences that contain them. Let us discuss them one at a time.

Discourse referents are intermediate objects that differ both from the terms that refer to them and from the elements of \( D \) they denote. We follow Groenendijk & Stokhof & Veltman (1996) both in terminology and in technical implementation, with slight differences. We call the technical equivalents of discourse referents pegs, and we identify them with natural numbers. Pegs are connected to terms by a function \( r \) called referent system. \( r \) is a partial function from \( \text{Var} \), the set of variables, to a natural number.\(^1\) Different variables may belong to the same peg.

A salience ranking \( s \) is a tuple of tuples. Its \( i \)th member is an \( n+1 \)-tuple \( \langle P,p_1,\ldots,p_n \rangle \), where \( P \) is an \( n \)-argument predicate and \( p_1,\ldots,p_n \) are pegs, i.e., natural numbers.\(^2\)

An assignment function \( g \) maps the set of pegs into \( D \). At any stage of the evaluation, there is a set \( G \) of assignments that satisfy all the information gathered so far. This information is also updated at every stage of the evaluation process. The assignment of a variable \( x \) is \( g(r(x)) \), if the variable is connected to a peg. Otherwise \( x \) does not have a value. The assignment of variables is partial.

By discourse information we mean a triple \( \langle r,s,G \rangle \) of the above values. It is updated functionally; for any \( \langle r,s,G \rangle \) and any well-formed expression \( \varphi \), there is exactly one update \( \langle r,s,G \rangle(\varphi) \).

Now we can turn to the evaluation of well-formed expressions. For the sake of simplicity, in this abstract we restrict to monadic predicates and atomic sentences. More complex expressions raise difficult but surmountable problems the detailed discussion of which would require more space.

Some simple definitions are in place. \( R(t) \) is the referent and \( F(t) \) is the extension of an individual term \( t \). \( r[x : k] \) is a modification of a referent system \( r : v \to n \) in which a variable \( x \) is connected to a peg \( k \); and \( G[k] \) is the expansion of a set of assignments with all possible values of peg \( k \). \( G[t : P] \) is a restriction to those assignments in which the value of \( t \) is in the extension of \( P \). \( S[P,k] \) and \( S[P,t] \) are expansions of a salience ranking \( s \) with a new pair. \( s(P) \) is the discourse referent in the rightmost pair of the list in which \( P \) appears.

\[ R(t) = \begin{cases} r(t) \text{ if } t \text{ is a variable,} \\ r(x) \text{ if } t \text{ is of the form } \lambda x P(x) \end{cases} \]

\[ F(t) = g(R(t)) \]

\[ r[x : k] = (r \cup \{ \langle x,k \rangle \}) \setminus \{ \langle x,i \rangle : i < n \} \]

\[ G[x] = \{ g : \text{there is a } g' \in G \text{ and a } d \in D \text{ such that } g = g' \cup \{ \langle r(x),d \rangle \} \} \]

\[ G[t : P] = \{ g : g \in G \land g(r(x)) \in g(P) \} \]

\[ s[P,k] = (s,\langle P,k \rangle) \]

\[ s(P) = t, \text{ if } \langle P,t \rangle \in s \text{ and for all } t', \text{ if } \langle P,t' \rangle \in s \text{ then } \langle P,t' \rangle \leq_s \langle P,t \rangle \]

---

\(^1\)We use Von Neumann’s set-theoretic definition of numbers. \( 0 = \emptyset, \ n+1 = n \cup \{ n \} \): that is, \( n+1 = \{ 0,1,\ldots,n \} \).

\(^2\)We define \( n \)-tuples in the usual way: \( \langle a \rangle = a, \langle a_1,\ldots,a_n,a_{n+1} \rangle = \langle a_1,\ldots,a_n,a_{n+1} \rangle \). We call \( a_1,\ldots,a_n \) the members of the tuple. The somewhat sloppy notation \( b \in \langle a_1,\ldots,a_n \rangle \) means that for some \( 1 \leq i \leq n, b = a_i \). We say that \( b \leq \langle a_1,\ldots,a_n \rangle b' \) iff for some \( 1 \leq i \leq n, b = a_i \) and \( b' = a_j \).
Now let us see how an occurrence of an individual term updates \( \langle r, s, G \rangle \), where \( r : v \rightarrow n \):

- \( \langle r, s, G \rangle(x) = df \begin{cases} \langle r[x : n], s, G[n] \rangle & \text{if } x \notin \text{dom}(r); \\ \langle r, s, G \rangle & \text{otherwise}. \end{cases} \)

- \( \langle r, s, G \rangle(\iota x P(x)) = df \begin{cases} \langle r[x : s(P)], s, G[x : P] \rangle & \text{if for some } k < n, \ (P, k) \in s; \\ \langle r[x : n], s[P, n], G[x : P] \rangle & \text{otherwise}. \end{cases} \)

Finally, the evaluation of a monadic atomic formula is pretty straightforward:

- \( \langle r, s, G \rangle(P(t)) = df \langle r', s'[P, R'(t)], G'[t : P] \rangle \), ahol \( \langle r', s', G' \rangle = \langle r, s, G \rangle(t) \).

We leave the definition of semantics hereby incomplete. Like it was mentioned before, poliadic predicates and embedded descriptions raise some technical problems, but most of the ideas put to use here work in a poliadic context, too. As for complex formulas, rules governing the inner and outer dynamics of the connectives can almost be copied from DPL. The DPL definitions of dynamic inference and truth can be applied to the present context, too. This would complete the construction of a logical system.

However, there is one considerable problem with the construction. Note that if there is no individual that satisfies the predicate in a description, it does not imply that atomic formulas that contain this description do not have a truth value. In fact, every such atomic formula is false. This violates the standard rule that the \( \iota \) operator transmits semantic value gap.

**Appendix: proper names in the DPLD framework**

Let us close the present abstract with a possible extension of DPLD, which models the anaphoric links between pronouns and proper names in a compositional way without second-order means. We can introduce a proper name-operator \(!\), and treat \( a!x \) as an individual term that bounds the variable \( x \) to the individual constant \( a \). It has the same effect as the atomic formula \( \exists x x = a \) in DPL, except that it is an individual term. Without going into the details of its semantics, we give one example of how it works:

(6) John loves Mary. He wants to marry her.

\( \text{(6}_D\text{PLD})\) love(John!x, Mary!y) \& want_to_marry(x, y)

**References**

