Notes on Dekker's 2004 Predicate Logic with Anaphora

Chris Barker, March 17, 2021

This is a re-presentation of Dekker's PLA (Predicate Logic with Anaphora)—more specifically, the elegant static version from his 2004 'Grounding dynamic semantics', in Anne Bezuidenhout and Marga Reimer (eds.), *Descriptions and Beyond*, OUP. I've suppressed mention of the model M that gives values to predicates and constants, and I've relativized denotations to sequences of individuals rather than framing the semantics in terms of support. So expressions are evaluated with respect to an assignment function g and a sequence of individuals σ.

Variables get their values from the assignment function:

$$[\![x]\!]^{g,\sigma} = g(x)$$

Pronouns get their values from the sequence of individuals:

$$[\![p_i]\!]^{g,\sigma} = \sigma_i$$

where σ_i is the ith element of σ.

Predicates. Predicates are applied to terms in the usual way:

$$[\![R(t_1, ..., t_n)]\!]^{g,\sigma} = [\![R]\!]^{g,\sigma}([\![t_1]\!]^{g,\sigma}, ..., [\![t_n]\!]^{g,\sigma})$$

Existentials. Let $a \cdot \sigma$ be the sequence formed by adding the individual a as the new first element of the sequence σ. So the length of $a \cdot \sigma$ is one greater than the length of σ.

$$[\![\exists x\phi]\!]^{g,a\cdot\sigma} = [\![\phi]\!]^{g[x\mapsto a],\sigma}$$

Note that there is no quantification! (But consider the net effect of this definition in combination with the definition of truth given below.) This rule simply checks whether ϕ is true when all free occurrences of x are assigned a as a value. It may help to read this equation from right to left: if setting x to a makes ϕ true, then we add the witness a to σ in order to underwrite subsequent anaphora. This is a strange and beautiful rule, worth contemplating.

Negation. Let $|\phi|$ be the number of indefinites in ϕ.

$$[\![\neg\phi]\!]^{g,\sigma} = \neg\exists a_1, ..., a_{|\phi|} : [\![\phi]\!]^{g,a_1\cdot...\cdot a_{|\phi|}\cdot\sigma}$$

A negated formula $\neg\phi$ is true with respect to a sequence σ just in case there is no way to extend σ that makes ϕ true. How many witnesses a_i should we try adding? Answer: one for each indefinite in ϕ. Small inefficiency: double negation uses $2 * |\phi|$ witnesses.

Conjunction.

$$[\![\phi \wedge \psi]\!]^{g,\tau\sigma} = [\![\phi]\!]^{g,\sigma} \wedge [\![\psi]\!]^{g,\tau\sigma}$$

where $\tau\sigma$ is the sequence consisting of τ followed by σ, and $\tau = \tau_1 \cdot ... \cdot \tau_{|\psi|}$. That is, we have to be able to make ϕ true without the benefit of the witnesses contributed by evaluating ψ.

Truth. A sentence ϕ is true with respect to an assignment g and a sequence σ iff $\exists a_1, ..., a_{|\phi|} : [\![\phi]\!]^{g,a_1 \cdot ... \cdot a_{|\phi|} \cdot \sigma}$, that is, just in case we can find suitable witnesses for the indefinites in ϕ. It is the existential in this definition that provides the engine for the existential quantification triggered by indefinites. Note that the negation rule in effect checks for the truth of the prejacent.

1 Examples

(1) A^x woman nominated herself$_x$ and she$_1$ made a^y speech.

$$[\![(\exists x.\textbf{nominated}(x,x)) \wedge (\exists y.\textbf{made}(p_1,y))]\!]^{g,s\cdot a} \quad (2)$$

$$= [\![\exists x.\textbf{nominated}(x,x)]\!]^{g,a} \wedge [\![\exists y.\textbf{made}(p_1,y)]\!]^{g,s\cdot a} \quad (3)$$

$$= [\![\textbf{nominated}(x,x))]\!]^{g[x\mapsto a],[]} \wedge [\![\textbf{made}(p_1,y)]\!]^{g[y\mapsto s],a} \quad (4)$$

$$= \textbf{nominated}([\![x]\!]^{g[x\mapsto a],[]}, [\![x]\!]^{g[x\mapsto a],[]}) \wedge \textbf{made}([\![p_1]\!]^{g[y\mapsto s],a}, [\![y]\!]^{g[y\mapsto s],a}) \quad (5)$$

$$= \textbf{nominated}(a,a) \wedge \textbf{made}(a,s) \quad (6)$$

There's a lot going on here. In step (3), the conjunction rule evaluates the left conjunct with respect to a sequence with the second conjunct's one witness lopped off. In step (4), each existential uses the first individual in its evaluation sequence as the value for its variable, then evalautes the prejacent against a sequence with that witness removed. I'm using [] to represent the empty sequence. Note that *herself* and *she* end up indicating the individual a via different mechanisms: within the scope of the first conjunct, a will be the value of the variable x; within the scope of the second conjunct, pronouns can access a from the list of witnesses established by the earlier conjunct.

The net prediction is that (1) will be true just in case we can find a woman a and a speech s such that a nominated herself and made s.

(7) If a^x farmer owns a^y donkey, he$_x$ beats it$_y$.

Dekker approximates the conditional (as do other discussions of dynamic semantics, e.g., Heim 1983) with material implication, expressed here via conjunction and negation: $A \to B \equiv \neg(A \wedge \neg B)$.

$$[\![\neg((\exists x \exists y.\textbf{owns}(x,y)) \wedge \neg(\textbf{beats}(p_1,p_2)))]\!]^{g,[]} \tag{8}$$

$$= \neg\exists f,d : [\![(\exists x \exists y.\textbf{owns}(x,y)) \wedge \neg(\textbf{beats}(p_1,p_2))]\!]^{g,f\cdot d} \tag{9}$$

$$= \forall f,d : \neg[\![(\exists x \exists y.\textbf{owns}(x,y)) \wedge \neg(\textbf{beats}(p_1,p_2))]\!]^{g,f\cdot d} \tag{10}$$

$$= \forall f,d : \neg([\![\exists x \exists y.\textbf{owns}(x,y)]\!]^{g,f\cdot d} \wedge [\![\neg(\textbf{beats}(p_1,p_2))]\!]^{g,f\cdot d}) \tag{11}$$

$$= \forall f,d : \neg([\![\textbf{owns}(x,y)]\!]^{g[x\mapsto f][y\mapsto d],[]} \wedge \neg[\![\textbf{beats}(p_1,p_2)]\!]^{g,f\cdot d}) \tag{12}$$

$$= \forall f,d : \neg(\textbf{owns}(f,d) \wedge \neg\textbf{beats}(f,d)) \tag{13}$$

$$= \forall f,d : \neg\textbf{owns}(f,d) \vee \textbf{beats}(f,d) \tag{14}$$

The net truth conditions say that (7) is true just in case there is no way of choosing a farmer and a donkey such that the farmer owns the donkey but doesn't beat it. The universal force comes from the outer negation in the translation of the material implication interacting with the existential introduced by the semantics of negation. The indefinites in the antecedent in effect control the value of the pronouns in the consequent by linking the variables associated with the indefinites to the same individuals that serve as the witnesses indexed by the pronouns.

Here is an implementation in Haskell:

```
var i g s = g!!(i-1)
pn i g s = s!!(i-1)
dpred rel subj obj g s = rel (subj g s) (obj g s)
dex i phi g (a:s) = phi (replaceAt i a g) s
dneg n phi g s = not (or [phi g (as ++ s) | as <- gen n]) -- n = |phi|
dand n phi psi g s = and [phi g (drop n s), psi g s] -- n = |psi|

replaceAt i x xs = take (i-1) xs ++ [x] ++ drop i xs
gen n = if n == 0 then [[]] else [a:s | s <- gen (n - 1), a <- [1..5]]

nominated x y = and [x == 3, y == 3]
made x y = and [x == 3, y == 2]
owns x y = x > y && x > 3 && y < 4 -- both farmers own all three donkeys
beats x y = x > y && x > 4 && y < 4 -- farmer 4 doesn't beat her donkeys

-- A^x woman nominated herself_x and she_1 made a^y speech
s1 = dand 1 (dex 1 (dpred nominated (var 1)(var 1)))
            (dex 2 (dpred made (pn 1) (var 2)))
            [1,1]
            [2,3]

-- If a^x farmer owns a^y donkey, he_x beats it_y
s7 = dneg 2 (dand 0 (dex 1 (dex 2 (dpred owns (var 1)(var 2))))
                    (dneg 0 (dpred beats (pn 1) (pn 2))))
            []
            []
```