CNF as Semantic Metalanguage

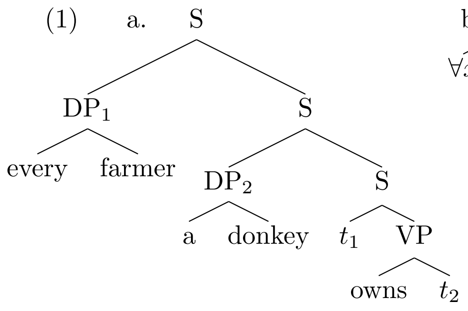
Steven Abney & Ezra Keshet, University of Michigan

1 Introduction

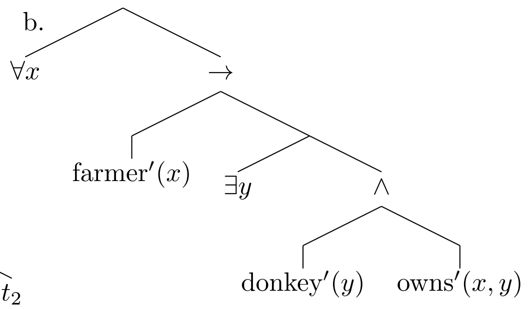
First Order Predicate Calculus (FOPC) is often used as a semantic metalanguage, but its syntax differs from that of natural language in several ways:

- "Normal" Quantifiers:

(1) a.



b.



- Extended-Scope Quantifiers:

(2) a. A man entered. He was flustered.
b. $\exists x$. [man$'(x) \wedge$ entered$'(x) \wedge$ flustered$'(x)$]

(3) (But: No/Every man entered. #He was flustered.)

(4) a. Every player choses a token. It goes on square one.
b. $\forall p \exists t \ldots$ [chooses$'(p, t) \wedge$ goes-on-square-one$'(t)$

- Extended-Scope Quantifiers sometimes "invert":

(5) a. Every farmer who owns a donkey beats it.
b. $\forall f \forall d \ldots$ beats$'(f, d)$

(6) a. Either John doesn't own a donkey, or he keeps it very quiet.
b. $\forall d \ldots$ [$\neg$owns$'(j, d) \vee$ keeps-quiet$'(j, d)$

1.1 Introducing CNF

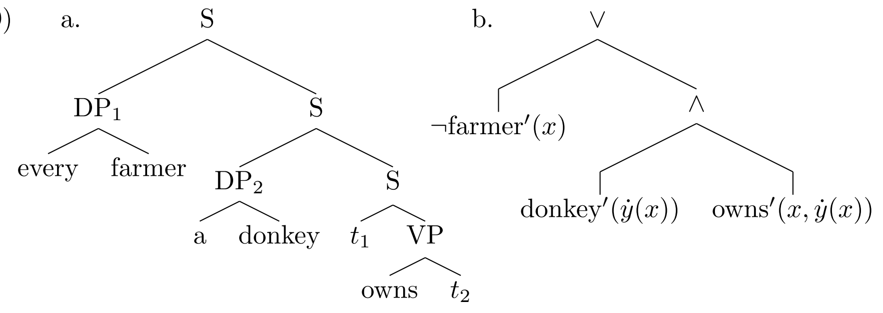
CNF is a quantifier-free subset of predicate calculus. It is the standard representation used for automated reasoning systems, and it is inferentially equivalent to FOPC. The following diagram labels the parts of the CNF expression $[A(c) \vee \neg B(f(x))] \wedge C(x)$ with predicates A, B, C, variable x, constant c, and function f:

(7) $$[\overbrace{\overbrace{\underbrace{\underbrace{A}_{\text{predicate}}\ (\underbrace{c}_{\text{term}})}_{\text{atomic formula}}}^{\text{literal}} \vee \overbrace{\neg \underbrace{\underbrace{B}_{\text{predicate}}\ (\underbrace{f(x)}_{\text{term}})}_{\text{atomic formula}}}^{\text{literal}}}^{\text{clause}}\ ] \wedge \overbrace{\overbrace{\underbrace{\underbrace{C}_{\text{predicate}}\ (\underbrace{x}_{\text{term}})}_{\text{atomic formula}}}^{\text{literal}}}^{\text{clause}}$$

For any CNF expression ϕ, let $x_1, \ldots, x_n$ be the variables that occur in ϕ, and let $\dot{y}_1, \ldots, \dot{y}_m$ be the Skolem functions that occur. Then ϕ is interpreted identically to the (second-order) predicate calculus formula in (8):

(8) $$\exists \dot{y}_1 \ldots \exists \dot{y}_m \, \forall x_1 \ldots \forall x_n \, \phi.$$

2 CNF as Semantic Metalanguage

(9)


The main intuition behind our semantic interpretation of LF trees such as (9a) is that they are homomorphic to (the parse tree of) a CNF expression. Certain LF terminal nodes denote literals, which are combined at higher node levels via either conjunction or disjunction in a predictable way. The variables and Skolem functions are introduced using an indexing system that slightly expands on the numerical indexing shown here.

2.1 Semantic types

An extended type has three parts: a *basic type* $\tau \in \{0, C, \forall, \exists, \wedge, \vee\}$, governing how nodes combine; an *index* i, governing predicate-argument relationships; and a *polarity* $p \in \{+1, -1\}$. (We generally notate negate types with a line over them: $\overline{\tau}$.)

Basic types. The basic types correspond to the mode of combination used when interpreting the node:

Node	**Basic Type**
Literals: nouns, verbs, adjectives, prepositions	L
Uninterpreted nodes: all other terminal nodes	0
Copy nodes (share a denotation with one child)	C
Universally quantified DPs	$\forall$
Existentially quantified DPs	$\exists$
Disjunction	$\vee$
Conjunction	$\wedge$

Polarity. Matrix sentences have positive polarity, and polarity is inherited downward with two exceptions:

- In a determiner phrase $[_{\text{DP}}$ D NP$]$ in which the determiner is a universal (*every, all, each, . . .*), the polarity of the NP is opposite to that of the DP.
- In a negation $[_{\text{S}}$ *not* S$]$, the two S nodes have opposite polarity.

Indices. An *index term* may be a variable, a Skolem term consisting of a Skolem function and its arguments, or a name (individual constant). An *index* is a tuple of index terms. For example:

- $\langle\,\rangle$ – an empty index. Used for uninterpreted nodes and complete clauses or sentences.
- $\langle u\rangle$ – a single index term. Used for one-place predicates, such as simple nouns and intransitive verbs.
- $\langle u, v\rangle$ – used for two-place predicates, such as transitive verbs.
- $\langle u, v, w\rangle$ – used for three-place predicates, such as ditransitive verbs.

Tuples of any length are allowed. We often omit the angle brackets when writing indices. For example, the extended type $\overline{C}_x$ has negative polarity and index $\langle x\rangle$.

2.2 The interpretation function defined

The semantic types and interpretations of nodes are determined by four sets of rules: a *semantic lexicon,* a set of *root constraints,* the *interpretation rules* of section 2.3, and the *indexing constraints* of section 2.4.

The semantic lexicon determines the semantic type of each word. We do not give an explicit listing, but the general principles are very simple. Content words (nouns, non-copular verbs, adjectives, and prepositions) have type L, and all other words have type 0. The lexicon also determines a CNF predicate ω' corresponding to each content word ω.

The root constraints are also simply stated: the root node of a matrix sentence must have positive polarity and its index must be $\langle\,\rangle$.

2.3 Rules of interpretation

In the rules (and elsewhere), the notation $\alpha : \tau_i$ represents a node α whose semantic type is τ_i. We use variables u, v, w for individual index terms, and we use i and j for indices.

2.3.1 Terminal nodes

Literals. In the following rules, ω must be a terminal node.

(10) $[\![\omega : L_{u_1,\ldots,u_n}]\!] = \omega'(u_1, \ldots, u_n)$ $\qquad [\![\omega : \overline{L}_{u_1,\ldots,u_n}]\!] = \neg\omega'(u_1, \ldots, u_n)$

Thus, a content word ω with an index i comprising n index terms denotes a CNF literal whose predicate is the value ω' in the lexicon for ω and whose argument list is identical to i. The cases $n = 1$ and $n = 0$ are permitted.

Other terminal nodes. For any terminal node α whose type is not L, the interpretation $[\![\alpha]\!]$ is undefined. Such nodes may, however, contribute to the interpretation less directly.

2.3.2 Nonterminal nodes

The rules for nonterminals are written so that, in any local configuration, only one rule is applicable, and the choice of rule is determined by the nature of the children. The linear order of a node's daughters is not important and may be reversed in the rules below. Unless otherwise specified, the variables σ and τ stand for any basic type *except* 0.

Copy. The following rules apply to non-branching nodes.

(11) $[\![\, [_{C_i}\ \alpha : \tau_i\,]\,]\!] = [\![\alpha]\!]$ $\qquad [\![\, [_{\overline{C}_i}\ \alpha : \overline{\tau}_i\,]\,]\!] = [\![\alpha]\!]$

The index of the child is copied to the parent. Child and parent also share polarity and denotation.

In the following rules, α is an uninterpreted word, with no index, such as a copula (discounting any tense interpretion on the copula for the time being).

(12) $[\![\, [_{C_i}\ \alpha : 0\ \beta : \tau_i\,]\,]\!] = [\![\beta]\!]$ $\qquad [\![\, [_{\overline{C}_i}\ \alpha : \overline{0}\ \beta : \overline{\tau}_i\,]\,]\!] = [\![\beta]\!]$

Again, the parent shares index, polarity, and denotation with one child – the one not of type 0.

Negation. In these rules, the first child must be negation, represented here as *not.* The position of negation can be the result of operator raising at LF.

(13) $[\![\, [_{C_i}\ \text{not}\ \alpha : \overline{\tau}_i\,]\,]\!] = [\![\alpha]\!]$ $\qquad [\![\, [_{\overline{C}_i}\ \text{not}\ \alpha : \tau_i\,]\,]\!] = [\![\alpha]\!]$

The only effect of *not* is to flip the polarity of its complement.

Application. These rules correspond roughly to Functional Application. In these rules, β must be a referential DP—a proper noun, a pronoun, or a trace. We use $\langle i, v\rangle$ as a shorthand for $\langle u_1, \ldots, u_n, v\rangle$ where $i = \langle u_1, \ldots, u_n\rangle$.

(14) $$\llbracket\, [_{C_i}\ \alpha\!:\!\sigma_{i,v}\ \beta\!:\!0_v\,]\,\rrbracket = \llbracket\alpha\rrbracket \qquad \llbracket\, [_{\overline{C}_i}\ \alpha\!:\!\overline{\sigma}_{i,v}\ \beta\!:\!\overline{0}_v\,]\,\rrbracket = \llbracket\alpha\rrbracket$$

In words, a predicate α with a non-empty index may "discharge" its last index term by combining with an uninterpreted node β having a matching index term as its entire index. The resulting node has the same interpretation and polarity as the predicate, but its index list is one term shorter, missing this last item.

Abstraction. Next, we have rules akin to Predicate Abstraction. Here, α must be a relative pronoun.

(15) $$\llbracket\, [_{C_u}\ \alpha\!:\!0_u\ \beta\!:\!\tau\,]\,\rrbracket = \llbracket\beta\rrbracket \qquad \llbracket\, [_{\overline{C}_u}\ \alpha\!:\!\overline{0}_u\ \beta\!:\!\overline{\tau}\,]\,\rrbracket = \llbracket\beta\rrbracket$$

In words, a relative pronoun (of basic type 0) may combine with a node having an empty index to form a node having a singleton index. (Note that the lack of index on τ is significant.) The relative pronoun α and its parent—which is usually a relative clause—share the same index. The other node β and the parent share a denotation.

Modification. The next rules correspond to Predicate Modification. In these rules, the children may not be quantified DPs. That is, in these rules σ and τ may be any basic type except 0, $\forall$, or $\exists$.

(16) $$\llbracket\, [_{\wedge_i}\ \alpha\!:\!\sigma_i\ \beta\!:\!\tau_i]\,\rrbracket = \llbracket\alpha\rrbracket \wedge \llbracket\beta\rrbracket \qquad \llbracket\, [_{\overline{\wedge}_i}\ \alpha\!:\!\overline{\sigma}_i\ \beta\!:\!\overline{\tau}_i]\,\rrbracket = \llbracket\alpha\rrbracket \vee \llbracket\beta\rrbracket$$

In words, two nodes with matching polarity and indices may combine to form a node of type $\wedge$ with the same polarity and index as the children. The interpretation of this mother node is the conjunction of the interpretations of the its children if is positive and the disjunction if it is negative.

Quantifiers. A set of syncategorematic rules are required to capture quantificational DPs. In the following, α must be a universal quantificational determiner such as *every, all, each,* etc., and ϵ must be an existential quantificational determiner such as *a(n), some,* etc.

(17) $$\begin{array}{ll} \llbracket\, [_{\exists_u}\ \epsilon\!:\!0_u\ \gamma\!:\!\tau_u\,]\,\rrbracket = \llbracket\gamma\rrbracket & \llbracket\, [_{\overline{\exists}_u}\ \epsilon\!:\!\overline{0}_u\ \gamma\!:\!\overline{\tau}_u\,]\,\rrbracket = \llbracket\gamma\rrbracket \\ \llbracket\, [_{\forall_u}\ \alpha\!:\!0_u\ \gamma\!:\!\overline{\tau}_u\,]\,\rrbracket = \llbracket\gamma\rrbracket & \llbracket\, [_{\overline{\forall}_u}\ \alpha\!:\!\overline{0}_u\ \gamma\!:\!\tau_u\,]\,\rrbracket = \llbracket\gamma\rrbracket \end{array}$$

Existentially quantified DPs carry the basic type $\exists$ while universally quantified DPs carry $\forall$. With existentials, the parent and both children share the same polarity; but universal quantificational determiners flip the polarity of their complements. In all cases, the parent and both children share a common index, which must consist of a single index term. The parent always shares a denotation with the NP complement, γ.

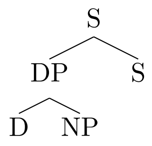
Quantification. Next we need rules to combine quantificational DPs with their nuclear scope.

(18)
$$\llbracket\, [_{\wedge}\, \beta\!:\!\exists_i\ \gamma\!:\!\tau\,]\, \rrbracket = \llbracket\beta\rrbracket \wedge \llbracket\gamma\rrbracket \qquad \llbracket\, [_{\overline{\wedge}}\, \beta\!:\!\overline{\exists}_i\ \gamma\!:\!\overline{\tau}\,]\, \rrbracket = \llbracket\beta\rrbracket \vee \llbracket\gamma\rrbracket$$
$$\llbracket\, [_{\vee}\, \beta\!:\!\forall_i\ \gamma\!:\!\tau\,]\, \rrbracket = \llbracket\beta\rrbracket \vee \llbracket\gamma\rrbracket \qquad \llbracket\, [_{\overline{\vee}}\, \beta\!:\!\overline{\forall}_i\ \gamma\!:\!\overline{\tau}\,]\, \rrbracket = \llbracket\beta\rrbracket \wedge \llbracket\gamma\rrbracket$$

A DP of basic type $\exists$ has a type-$\wedge$ parent (a conjunction) while a DP of basic type $\forall$ has a type-$\vee$ parent (a disjunction). This basic type, plus the polarity, determines whether the DP combines with with its complement via logical disjunction or logical conjunction. Note that the parent and the second child must have an empty index.

2.4 Indexing constraints

We define the **scope of a quantificational DP** to be the set of nodes dominated by its parent. Thus the scope of DP in (19) includes both NP and S:

(19)


We assume a fixed enumeration of variables $x_1, x_2, \ldots$ and a fixed enumeration of Skolem functions $\dot{y}_1, \dot{y}_2, \ldots$. We also enumerate the quantificational DPs in a sentence in preorder as $\mathrm{DP}_1, \ldots, \mathrm{DP}_n$.

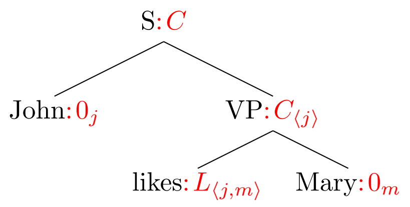
The following indexing constraints determine the indices for DP nodes, and those indices are propagated through the tree by the interpretation rules above:

- The index of a proper noun ω is $\langle\omega'\rangle$. ω' is an individual constant, determined by the semantic lexicon.
- The index of a trace or pronoun must be the same as the index of its antecedent.
- If DP_t has signed type $\forall$ or $\overline{\exists}$, then its index is $\langle x_t\rangle$ – i.e., a singleton variable.
- Otherwise the index of DP_t is of form $\langle \dot{x}_t(\ldots)\rangle$ – i.e., a Skolem function applied to an argument list –, and the arguments "..." consist of the indices of all DPs of type $\forall/\overline{\exists}$ that take scope over DP_t.

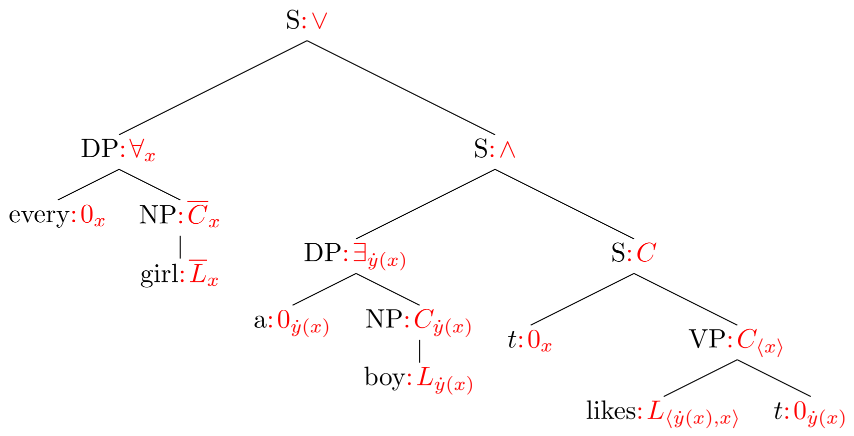
We leave open whether these constraints are enforced via syntactic or semantic means, or some combination of the two.

2.5 Examples

(20)



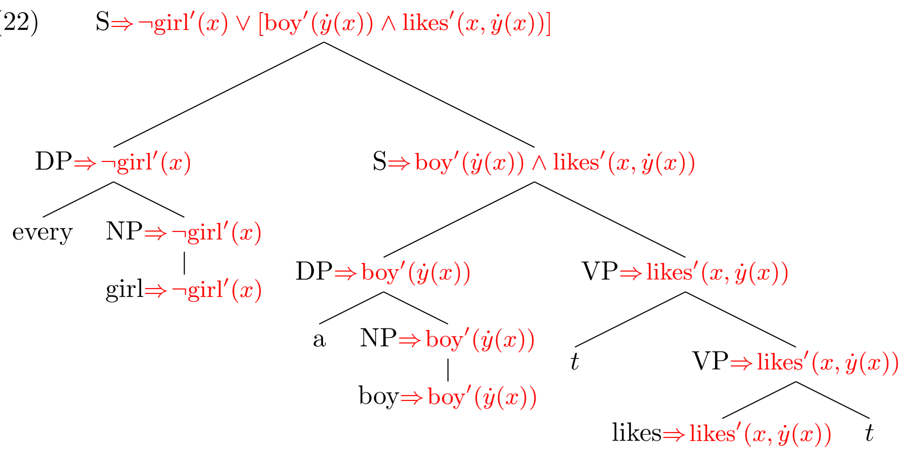
(21)



A few items to note:

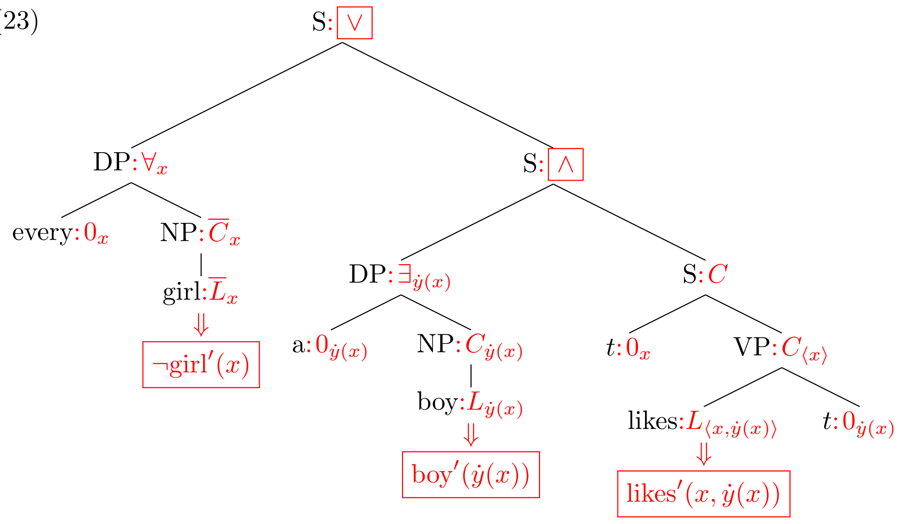
- The top node has a type of $\vee$, meaning that its two daughters are combined via disjunction.
- The second highest S has type $\wedge$, meaning that its daughters are combined via conjunction.
- The N *girl* and its parent NP have a negative polarity, due to the NP combining with a universal quantifier.
- The index on *boy* and its surrounding nodes is $\dot{y}(x)$: it is a Skolem function due to the existential quantifier and its argument list comprises the only outscoping variable, x.
- The index on each of the two traces matches its antecedent. Thus, the original sentence for this tree was "every girl likes a boy" and not "A boy likes every girl."

(22)


S⇒ $\neg\mathrm{girl}'(x) \vee [\mathrm{boy}'(\dot{y}(x)) \wedge \mathrm{likes}'(x, \dot{y}(x))]$
DP⇒ $\neg\mathrm{girl}'(x)$
every
NP⇒ $\neg\mathrm{girl}'(x)$
girl⇒ $\neg\mathrm{girl}'(x)$
S⇒ $\mathrm{boy}'(\dot{y}(x)) \wedge \mathrm{likes}'(x, \dot{y}(x))$
DP⇒ $\mathrm{boy}'(\dot{y}(x))$
a
NP⇒ $\mathrm{boy}'(\dot{y}(x))$
boy⇒ $\mathrm{boy}'(\dot{y}(x))$
VP⇒ $\mathrm{likes}'(x, \dot{y}(x))$
t
VP⇒ $\mathrm{likes}'(x, \dot{y}(x))$
likes⇒ $\mathrm{likes}'(x, \dot{y}(x))$
t

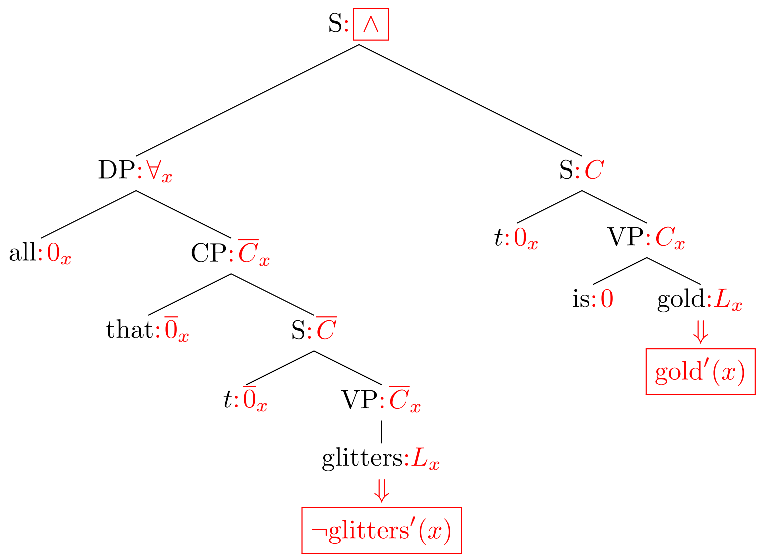
Such an annotated tree is quite repetitious. In fact, it is enough to interpret the L nodes and to connect them by translating the nodes with types $\wedge$, $\vee$, $\overline{\wedge}$, $\overline{\vee}$ to connectives ($\wedge$, $\vee$, $\vee$, $\wedge$, respectively):

(23)



One can then read off the interpretation simply by reading the boxed elements: $\neg\mathrm{girl}'(x) \vee [\mathrm{boy}'(\dot{y}(x)) \wedge \mathrm{likes}'(x, \dot{y}(x))]$

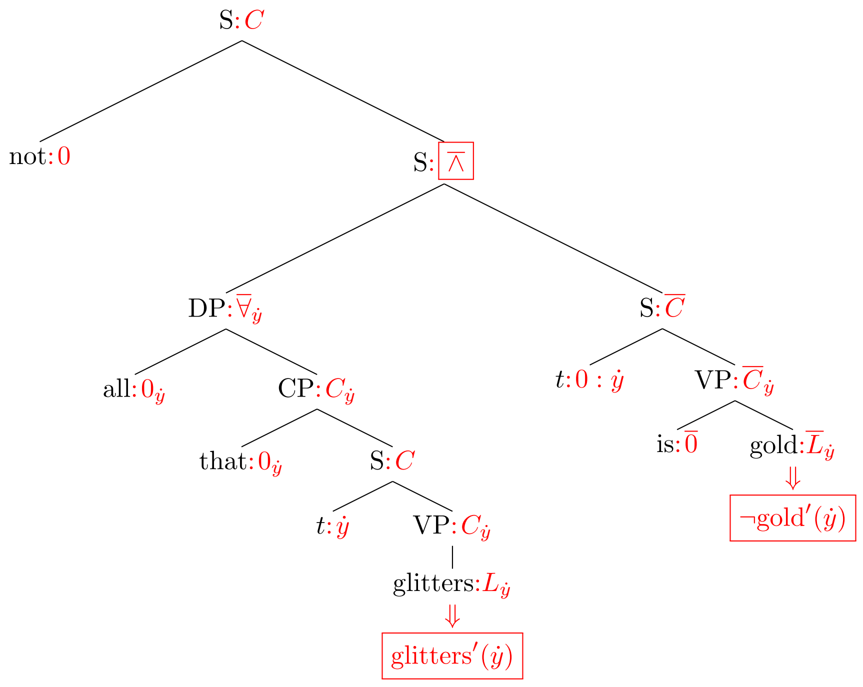
(24)



The translation, ¬ glitters$'(x)\vee$ gold$'(x)$ roughly means that for every thing x, either x does not glitter or x is gold.

Now let us consider what happens when we negate the sentence: “all that glitters is not gold.” This of course has two readings; the following LF assigns wide scope to the negation.

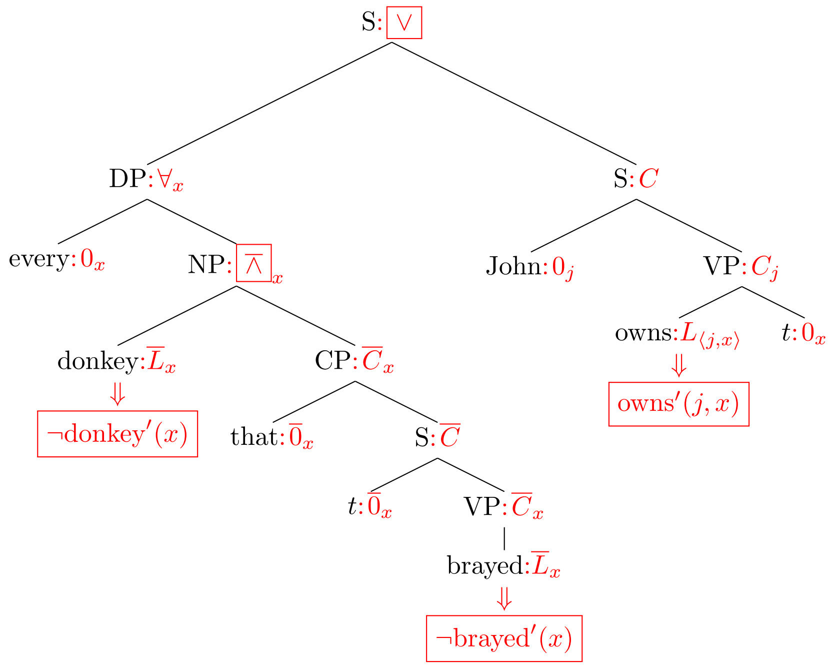
(25)


S: C
not: 0
S: $\boxed{\overline{\wedge}}$
DP: $\overline{\forall}_{\dot{y}}$
all: $0_{\dot{y}}$
CP: $C_{\dot{y}}$
that: $0_{\dot{y}}$
S: C
t: $\dot{y}$
VP: $C_{\dot{y}}$
glitters: $L_{\dot{y}}$
$\boxed{\text{glitters}'(\dot{y})}$
S: $\overline{C}$
t: $0 : \dot{y}$
VP: $\overline{C}_{\dot{y}}$
is: $\overline{0}$
gold: $\overline{L}_{\dot{y}}$
$\boxed{\neg\text{gold}'(\dot{y})}$

There are a few points to note.

- The basic type of a node combining a universally quantified DP with its sister is $\vee$, but it is inverted because the node is negated.
- The negation spreads through the whole tree, except that it is canceled by a second negation on the sibling of "all."
- The semantic translation is $\text{glitters}'(\dot{y}) \wedge \neg\text{gold}'(\dot{y})$ which can be read roughly as "(there is) a thing $\dot{y}$ such that $\dot{y}$ glitters, but $\dot{y}$ is not gold."

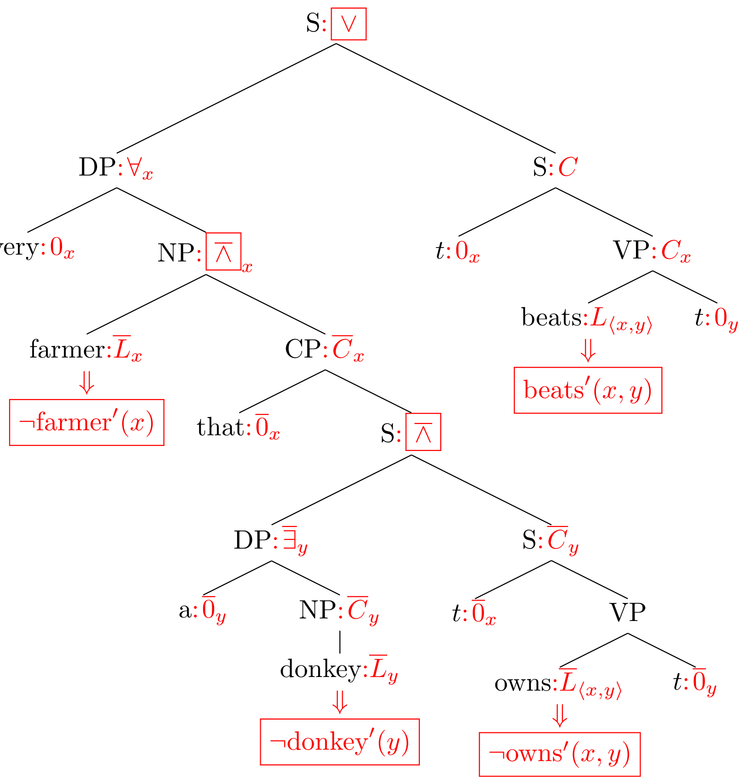
(26)



Points to note:

- The basic type for the NP is $\wedge$. Here, it is inverted by the negation.
- The relative pronoun has moved from the lower subject position, leaving a co-indexed trace.
- The CP is a predicate, and it shares an argument with the N *donkey*, through a process similar to predicate modification.
- The transitive verb *owns* here is treated as a two-place predicate, taking x from t_x and a constant such as j from *John* .
- The final CNF is $\neg\text{donkey}'(x) \vee \neg\text{brayed}'(x) \vee \text{owns}'(j,x)$.

(27)



The highest S type is $\vee$ because it involves quantification by a universal. The *basic* type of the S in the RC is $\wedge$ because it involves quantification by an existential. However, the S is negated, yielding a disjunction.

Otherwise, the only thing that is new is the pronoun, which has been marked (via co-indexation) as having the "donkey" noun phrase as antecedent. This—rather magically—gives the correct interpretation:

(28) $\neg\text{farmer}'(x) \vee \neg\text{owns}'(x,y) \vee \text{beats}'(x,y)$

2.6 Shorthand

Consider "every dog that chases every cat is happy." Switching to LF word order and adding some annotations, this becomes:

(29) $\text{every}_x[\overline{\text{dog}\ {}^{\wedge}\text{that [every}_{\dot{y}}[\ \overline{\text{cat}}\ ]^{\vee}\ \text{chases]]}}^{\vee}\ \text{is happy}$

Now we can read off the interpretation by applying the predicates to the correct arguments, canceling double negations, and flipping the connectives $\overline{\wedge}$ and $\overline{\vee}$. Grouping follows the LF tree structure:

(30) $\neg\text{dog}'(x) \vee (\text{cat}'(\dot{y}) \wedge \neg\text{chases}'(x,\dot{y})) \vee \text{happy}'(x).$

Here is an example with wide-scope "not."

(31) a. every $\underline{\text{dog that barks does not bite}}$

b. not [every$_{\dot{x}}$ [$\overline{\text{dog }^\wedge\text{that [barks]}}$]$^\vee$ bites]

c. $\text{dog}'(\dot{x}) \wedge \text{barks}'(\dot{x}) \wedge \neg\text{bites}'(\dot{x})$

3 Anaphora: CNF vs. DRT/DPL

One of the main motivations for Discourse Representation Theory (Kamp and Reyle 1993), and its compositional cousin Dynamic Predicate Logic (Groenendijk and Stokhof 1991), is the treatment of pronouns – in particular, cross-sentential anaphora and intra-sentential donkey anaphora.

3.1 Basic intra-sentential anaphora

Two co-indexed nodes will have the same referent in this system, whether this referent is derived via a constant, or via a universal or Skolem variable:

(32) a. every$_x$[$\overline{\text{dog}}$]$^\vee$ loves himself$_x$ $\to \neg\text{dog}'(x) \vee \text{loves}'(x,x)$

b. some$_{\dot{x}}$[dog]$^\wedge$ loves himself$_{\dot{x}}$ $\to \text{dog}'(x) \wedge \text{loves}'(x,x)$

c. John$_j$ loves himself$_j$ $\to \text{loves}'(j,j)$

Even the distinction between bound and referential readings of pronouns can be captured via the abstraction rule, if names are given predicate meanings:

(33) $\langle$some$_{\dot{x}}\rangle$ [John]$^\wedge$ *pro*$_x$ [loves himself$_x$] $\to \text{John}'(\dot{x}) \wedge \text{loves}'(x,x)$

3.2 Discourse (inter-sentential) anaphora

CNF databases are designed to be added to incrementally. Thus, the short discourse "a donkey brayed; it was hungry" would yield the following CNF formulas, assuming the following indexing:

(34) a. a$_{\dot{x}}$[donkey]$^\wedge$ brayed $\to \text{donkey}'(\dot{x}) \wedge \text{brayed}'(\dot{x})$

b. it$_{\dot{x}}$ was hungry $\to \text{hungry}'(\dot{x})$

Sentences are implicitly conjoined, and so the final CNF formula is $\text{donkey}'(\dot{x}) \wedge \text{brayed}'(\dot{x}) \wedge \text{hungry}(\dot{x})$. This formula is interpreted by existentially quantifying over the zero-arity Skolem function $\dot{x}$, yielding the meaning "there was a donkey that brayed and was hungry."

DPL explicitly rules out sequences such as the following:

(35) Every donkey$_i$ brayed. *It$_i$ was hungry.

At first glance, it may seem as the the CNF system incorrectly allows such sequences. For instance, the sentence in (35) might come out as follows in CNF:

(36) a. every$_x$[$\overline{\text{donkey}}$]$^\vee$ brayed $\to \neg\text{donkey}'(x) \vee \text{brayed}'(x)$

b. it$_x$ was hungry $\to \text{hungry}'(x)$

This yields a very odd meaning for the discourse:

(37) $(\neg\text{donkey}'(x) \vee \text{brayed}'(x)) \wedge \text{hungry}'(x)$

Basically: "every donkey brayed, and everything was hungry." Notice, though, that this is equivalent to:

(38) $(\neg\text{donkey}'(x) \vee \text{brayed}'(x)) \wedge \text{hungry}'(y)$

In other words, there is no connection conveyed between being a (braying) donkey and being hungry; the only contribution that the universal quantifier *every donkey* makes towards the interpretation of the pronoun *it* is to make it a universal, rather than a Skolem, variable.

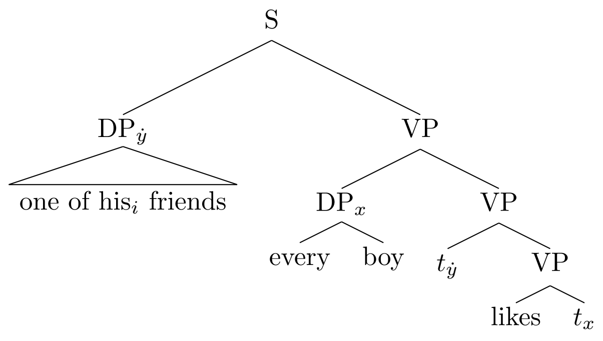
We therefore propose a rule to prohibit such cases (cf. Prohibition Against Coreference (PACO) due to Büring (2005) and other conditions):

(39) **Prohibition against Vacuous Binding**: A pronoun α in a discourse ϕ may not have as its index a variable σ if there is another variable σ' such that the discourse ϕ' only differing from ϕ in that α_σ is replaced by $\alpha_{\sigma'}$ is such that $[\![\phi']\!] = [\![\phi]\!]$.

This prohibition also rules out certain illicit cases where the quantifier and pronoun appear in the same sentence:

(40) One of his$_1$ friends likes every$_1$ boy.

(41)



This sequence is not ruled out by any syntactic binding condition, since the coindexed *his* and *every boy* do not c-command one another. Furthermore, it does not run afoul of any prohibition against crossover, since the two items are in their surface order. However, such an indexing is undesirable, since the meaning derived would be something like the following:

(42) $\text{friend-of}'(\dot{y}, x) \wedge (\neg\text{boy}'(x) \vee \text{likes}'(\dot{y}, x))$

In other words, there is a someone $\dot{y}$ who is friends with everyone, and $\dot{y}$ likes every boy. (42) is prohibited by (39), though, since the variable x in the first clause could be replaced by another variable without changing the meaning of the whole expression.

3.3 Donkey anaphora

(43) a. every farmer who owns a donkey beats it
b. $\text{every}_x\overline{[\text{farmer}\ {}^\wedge\text{who}\ [\text{a}_y\ [\text{donkey}]^\wedge\ \text{owns}]]}^\vee$ beats it_y
c. $\neg\text{farmer}'(x) \vee \neg\text{donkey}(y) \vee \neg\text{owns}'(x,y) \vee \text{beats}'(x,y)$

Problem? It has been proposed that donkey sentences have two different readings: a strong reading wherein, e.g., every farmer beats every donkey s/he owns and a weak reading wherein, e.g., each farmer need only beat one of his/her donkeys in order for the sentence to be true.

3.4 Negation, Disjunction, and Universal Quantification

The DPL definitions for negation, disjunction, and universally quantified sentences were designed to block existentials from scoping beyond a certain point:

(44) a. John doesn't own a car$_i$. *It$_i$'s in his driveway.
b. Every farmer who owns a donkey$_i$ beats it$_i$. *It$_i$ is very stubborn.
c. *John owns a car$_i$ or it$_i$'s in his driveway.

CNF correctly rules out these first two cases via the Prohibition against Vacuous Binding defined in (39) above. The analysis for (44a) goes as follows:

(45) a. $\overline{[\text{a}_{\dot{x}}\ [\text{car}]^\wedge[\text{John}_j\ \text{owns}\ t_{\dot{x}}]]}\ {}^\wedge[\text{It}_{\dot{x}}\ \text{is in his driveway}]$
b. $(\neg\text{car}'(x) \vee \neg\text{owns}'(j,x)) \wedge \text{in-driveway}'(x)$

This final formula runs afoul of the Prohibition against Vacuous Binding, since the last clause in-driveway$'(x)$ could well have used any universal variable, as in in-driveway$'(y)$ instead of in-driveway$'(x)$. As for (44b):

(46) a. $[\text{every}_x\ \overline{[\text{farmer}\ {}^\wedge\text{who}\ [\text{a}_y\ [\text{donkey}]^\wedge\ \text{owns}]]}^\vee\ \text{beats it}_y]$
${}^\wedge[\text{it}_y\ \text{is stubborn}]$
b. $(\neg\text{farmer}'(x)\vee\neg\text{donkey}'(y)\vee\neg\text{owns}'(x,y)\vee\text{beats}'(x,y))\wedge\text{stubborn}'(y)$

Once again, the last clause of the formula violates the Prohibition against Vacuous Binding, since stubborn$'(y)$ might well have been written stubborn$'(z)$ without altering the truth conditions.

The case in (44c) is not as clearly infelicitous as it seems at first glance, though. For instance, consider the following disjunctions, which sound fine despite exhibiting a scoping that is illegal under the rules of DPL:

(47) a. John bought a car$_i$, or perhaps he stole it$_i$
b. John bought a car$_i$, or Mary bought it$_i$
c. A dog$_i$ ate our dinner, or at least it$_i$ ate most of our dinner.

And, in fact, the original example (44c) improves (slightly) in the following scenario:

(48) Imagine that John always parks his car in his garage. However, he has been trying to sell the car, and you know that he will leave it out in the driveway for the seller to pick it up just in case he has successfully sold it. Thus, either John (still) owns a car, or he has left it in his driveway.

3.5 The Limits of DPL

The strict rules of DPL quickly run into problems in cases quite similar to those given above, as pointed out by Groenendijk and Stokhof (1991) themselves:

(49) a. Either there isn't a bathroom$_i$ here or it$_i$'s in a funny place.
b. Every player chooses a token$_i$. It$_i$ goes on square one.

The simplest formulation of DPL predicts these sentences to sound odd – since they involve scoping out of negation, disjunction, and/or a universally quantified sentence – but they are actually entirely acceptable. They actually fall out beautifully from the CNF system without any further definitions. For (49a):

(50) a. $[\text{There isn't } \overline{[\text{a}_x[\text{bathroom}]^{\wedge}\text{here}]}] \text{ or}^{\vee}[\text{it}_x \text{ is in a funny place}]$
b. $\neg\text{bathroom}'(x) \vee \neg\text{here}'(x) \vee \text{in-funny-place}'(x)$

Since the existential *a bathroom* is negated, it denotes a universal variable. And, a negated universal variable in one clause of a disjunction is equivalent to the interpretation of a universal quantifier (cf. *every/any bathroom here is in a funny place.*).

And for (49b):

(51) a. $\text{every}_x\overline{[\text{player}]}^{\vee}\text{a}_{\dot{y}(x)}[\text{token}]^{\wedge}\ t_x \text{ chooses } t_{\dot{y}(x)}$
$^{\wedge}[\text{it}_{\dot{y}(x)} \text{ goes on square one}]$
b. $[\ \neg\text{player}'(x) \vee [\text{token}'(\dot{y}(x)) \wedge \text{chooses}'(x, \dot{y}(x))]\ ] \wedge$
$\text{goes-on-sq-one}'(\dot{y}(x))$

$\text{goes-on-sq-one}'(\dot{y}(x))$ means that for all x, $\dot{y}(x)$ goes on square one, where it has already been established for all players p that $\dot{y}(p)$ is a token.

4 Quantification in a quantifier-free language

4.1 Definite determiners and presuppositions

Under a definition due to Russell (1905), "The red dog barked" comes out as:

(52) $(\neg\text{red}'(x) \vee \neg\text{dog}'(x) \vee x = \dot{y}) \wedge \text{barked}'(\dot{y})$

In other words, there is a unique red dog, $\dot{y}$ is that dog, and $\dot{y}$ barked. Such an interpretation could be achieved via the rules in (53). Note that *the* has a complex index comprising one variable and one Skolem function application.

(53) $[\![\ [_{\exists_v} \text{the}\!:\!0_{\langle u,v\rangle}\ \gamma\!:\!\overline{\tau}_u\ ]\ ]\!] = ([\![\gamma]\!] \vee u = v)$
$[\![\ [_{\overline{\exists}_u} \text{the}\!:\!\overline{0}_{\langle u,v\rangle}\ \gamma\!:\!\tau_v\ ]\ ]\!] = ([\![\gamma]\!] \wedge u \neq v)$

This analysis makes *the* into a special kind of existential determiner that flips the polarity of its complement. Negative sentence "the red dog didn't bark" comes out as:

(54) $(\text{red}'(\dot{x}) \wedge \text{dog}'(\dot{x}) \wedge \dot{x} \neq y) \vee \neg\text{barked}'(y)$

In other words, if there is a unique red dog y, then y didn't bark.

However, most modern semantic theories hold that the English sentence actually *presupposes* it. But what is a presupposition but a query against prior knowledge? Such queries are simple to do in a CNF question-answering system, which is designed to give answers to questions posed as CNF formulas including a special predicate, such as **Ans**:

(55) "Who is the red dog?" $= (\neg\text{red}'(x) \vee \neg\text{dog}'(x) \vee x = \dot{y}) \wedge \mathit{Ans}(\dot{y})$

A definition of *the* based on this concept could be designed assuming a discourse database $\mathcal{D}$ and a question-answering operator $\vdash$. The following are special combination rules where the parent node is still type 0 and hence technically uninterpreted:

(56) $[_{0_v}\ \text{the}\!:\!0_{u,v}\ \gamma\!:\!\overline{\tau}_u\,]$ and $[_{\overline{0}_v}\ \text{the}\!:\!0_{u,v}\ \gamma\!:\!\overline{\tau}_u\,]$ are allowed
if $\mathcal{D} \vdash ([\![\gamma]\!] \vee u = v)$

Potential Problems.

- "The farmer who owns a donkey fed it." If "a donkey" is only evaluated in the presupposition, then is it available for future reference? Also, since the NP is negated in the presupposition, "it" will come out as universal: "the farmer who owns a donkey fed all of his donkeys." This might be OK, but it is a bit odd. Maybe [the NP] should reiterate the material in the presupposition, for these reasons – but should the reiteration be positive [$\text{NP}_{\dot{x}} \wedge \text{VP}_{\dot{y}}$] or negative [$\neg\text{NP}_y \vee \text{VP}_y$]?
- What about plural definites [the NPs]? Do they need to be ALL and ONLY the satisifiers of NPs?

4.2 Plurals

In order to capture plurals in the CNF system, we will have to axiomatize set theory, and allow sets to be individuals in our domain. Many of our definitions might not need to change. For instance, *some dogs* could come out as $\text{dogs}'(\dot{x})$, but this time $\dot{x}$ will be an individual representing a set of dogs. However, something more is needed to handle other simple numeric indefinite determiners. For a first pass, we consider determiners such as *two* predicates over sets, and revise our existential quantification rules accordingly:

(57) $[\![\ [_{\exists_u}\ \epsilon\!:\!L_u\ \gamma\!:\!\tau_u\,]\ ]\!] = [\![\epsilon]\!] \wedge [\![\gamma]\!] \qquad [\![\ [_{\overline{\exists}_u}\ \epsilon\!:\!\overline{L}_u\ \gamma\!:\!\overline{\tau}_u\,]\ ]\!] = [\![\epsilon]\!] \vee [\![\gamma]\!]$

Note that these are now identical to the Modification rules – the only difference comes in the indexing constraints on a quantificational determiner. Some sample definitions for determiners are given below:[1]

(58) a. $\text{a}' = \text{one}' = \lambda X.|X = 1|$
b. $\text{two}' = \lambda X.|X = 2|$
c. $\text{more-than-three}' = \lambda X.|X > 3|$

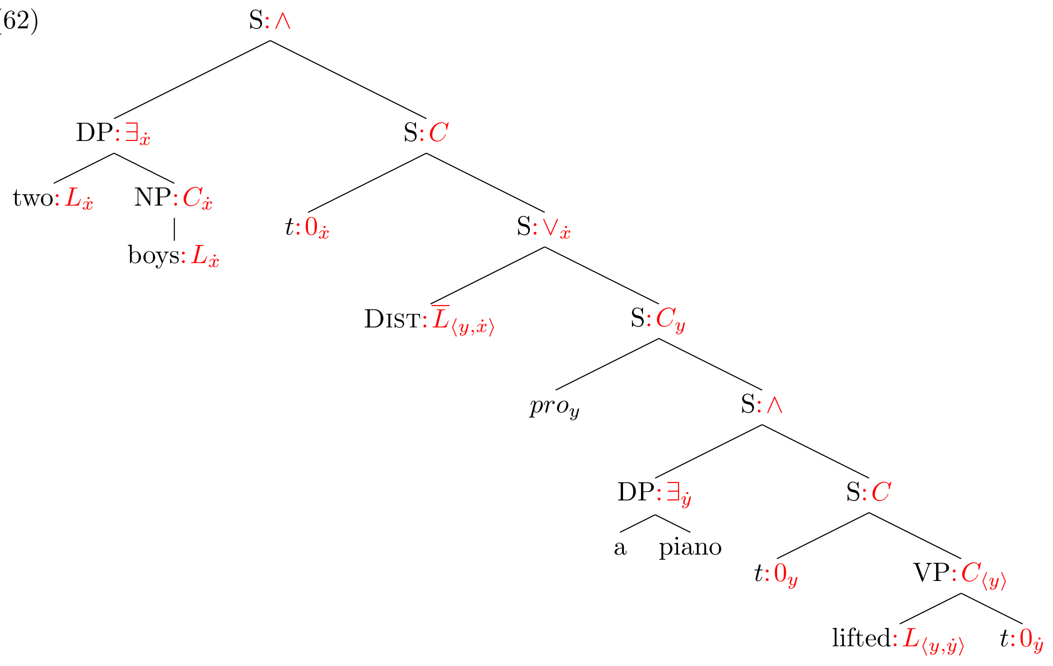
(59) a. Two dogs fought.
b. $\text{two}^{\wedge}_{\dot{x}}$ $[\text{dogs}]^{\wedge}$ fought
c. $|\dot{x} = 2| \wedge \text{dogs}'(x) \wedge \text{fought}'(\dot{x})$

(60) a. More than three dogs fought.
b. $[\text{more than three}]^{\wedge}_{\dot{x}}$ $[\text{dogs}]^{\wedge}$ fought
c. $|\dot{x} > 3| \wedge \text{dogs}'(\dot{x}) \wedge \text{fought}'(\dot{x})$

Distributive readings require a distributive operator that acts like a universal quantifier over the members of a set:

(61)
$$[\![\ [_{\wedge_u}\ \textsc{Dist}\!:\!\overline{L}_{\langle u,v\rangle}\ \gamma\!:\!\tau_v\ ]\ ]\!] = u \notin v \vee [\![\gamma]\!]$$
$$[\![\ [_{\overline{\wedge}_v}\ \textsc{Dist}\!:\!L_{\langle v,v'\rangle}\ \gamma\!:\!\overline{\tau}'_v\ ]\ ]\!] = v \in v' \wedge [\![\gamma]\!]$$

It receives its first index term via an index constraint and its second matches its parent. Next, we need a structure like the following:

(62)


[1] A more nuanced analysis might give a "≥ 2" meaning to *two* and derive the "exactly two" reading via implicature.

Then, the two different reading of the sentence "two boys lifted a piano" are given as follows:[2]

(63) a. Two boys lifted a piano (together).
b. two$^{\wedge}_{\dot{x}}$ [boys]$^{\wedge}$ a$^{\wedge}_{\dot{y}}$ [piano]$^{\wedge}$ $t_{\dot{x}}$ lifted $t_{\dot{y}}$
c. $|\dot{x}| = 2 \wedge \text{boys}'(\dot{x}) \wedge |\dot{y}| = 1 \wedge \text{piano}'(\dot{y}) \wedge \text{lifted}'(\dot{x}, \dot{y})$

(64) a. Two boys (each) lifted a piano
b. two$^{\wedge}_{\dot{x}}$ [boys]]$^{\wedge}$ $t_{\dot{x}}$[DIST$_{\langle y,\dot{x}\rangle}$]$^{\vee}a^{\wedge}_{\dot{z}}$ [piano]$^{\wedge}$ t_y lifted $t_{\dot{z}}$
c. $|\dot{x}| = 2 \wedge \text{boys}'(\dot{x}) \wedge y \notin \dot{x} \vee (|\dot{z}| = 1 \wedge \text{piano}'(\dot{z}) \wedge \text{lifted}'(y, \dot{z}))$

Problem: What if there is a Skolem in the subject DP:

(65) 10 boys who made 2 sandwiches (each) left them on the counter.

4.3 Generalized Quantifiers

There is no meaning for *most* that can make (66) capture the correct truth conditions for "most dogs bark":

(66) $\text{most}'(\sigma, \sigma') \wedge/\vee \text{dogs}'(\sigma) \wedge/\vee \text{bark}'(\sigma')$

4.4 Every as Set-denoting

Unembedded *every*-DPs always allow a plural pronoun to refer back to them:

(67) Every student failed. They didn't study enough.

5 Remaining Issues

5.1 Telescoping

Roberts (1987, 1989):

(68) Each candidate$_i$ approached the stage (one by one). He$_i$ shook the dean's hand and returned to his$_i$ seat.

(69) Every runner$_i$ took his$_i$ mark (at the same time). He$_i$ put his$_i$ hand to the track and got ready to race.

These cases seem to involve the illegal co-indexation of a (positive-context) universal over two sentences. The closest we could come to capturing these sentences with pure quantification over individuals is as follows (where ⟦candidate⟧ = C, ⟦approached the stage⟧ = A, ⟦shook the dean's hand⟧ = H, ⟦returned to one's seat]] = S, ⟦runner⟧ = R, ⟦took one's mark⟧ = M, ⟦put one's hand to the track⟧ = P, and ⟦got ready to race⟧ = G):

$$(\overline{Cx} \vee Ax) \wedge Hx \wedge Sx$$

[2]Technically, there is also a third reading, where the two boys each lift the same piano on different occasions.

$$(\overline{Rx} \vee Mx) \wedge Px \wedge Gx$$

Both of these formulas violate the Prohibition against Vacuous Binding. In addition, they do not actually capture the meaning of the discourses (68) and (69), since they assert (among other things) that every individual shook the dean's hand, and every individual got ready to race.

5.2 Every-DPs that cannot be antecedents

Quantifiers whose determiner is *every* represent universal variables in positive contexts and Skolem variables in negative contexts in the CNF system. This suggests that *every*-DPs in negative contexts ought to be suitable antecedents for pronouns. However, this does not seem to be the case. Consider:

(70) a. Every girl that answered every$_1$ question was rewarded.
b. *It$_1$ wasn't even the trick question, usually.

The CNF for (70a) is:

$$\overline{Gx} \vee (Q\dot{y} \wedge \overline{Ax\dot{y}}) \vee Rx$$

(71) a. Every player who is missing a suit$_1$ must wait until he gets it$_1$.
b. ??Every player that does not hold every suit$_1$ must wait until he gets it$_1$.

The CNF for these two sentences is:

(72) a. $\overline{Px} \vee \overline{Sy} \vee \overline{Mxy} \vee Wxy$
b. $\overline{Px} \vee \overline{Sy} \vee Hxy \vee Wxy$

Despite these two issues, the CNF system does not fall prey to the many the issues raised against E-type analyses , illustrated by the following minimal pair:

(73) a. If one of the ten marbles is missing, it is probably under the table
b. *If nine of the ten marbles are accounted for, it is probably under the table

There is an antecedent for the pronoun *it* in (73a), but not in (73b), and this fact is enough to account for the difference in acceptability.

References

Büring, D.: 2005, *Binding theory*, Cambridge Univ Pr.

Groenendijk, J. and Stokhof, M.: 1991, Dynamic predicate logic, *Linguistics and philosophy* **14**(1), 39–100.

Kamp, H. and Reyle, U.: 1993, *From discourse to logic: Introduction to modeltheoretic semantics of natural language, formal logic and discourse representation theory*, Vol. 42, Kluwer Academic Dordrecht,, The Netherlands.

Roberts, C.: 1987, *Modal subordination, anaphora, and distributivity*, PhD thesis, University of Massachusetts at Amherst.

Roberts, C.: 1989, Modal subordination and pronominal anaphora in discourse, *Linguistics and philosophy* **12**(6), 683–721.

Russell, B.: 1905, On denoting, *Mind* **14**(4), 479–493.