1 Introduction

• The same word, and, can be used both intersectively and collectively.

(1) a. John lies and cheats. \((\text{intersective})\)
b. That liar and cheat can not be trusted. \((\text{intersective})\)

(2) a. John and Mary met in the park last night. \((\text{collective})\)
b. A man and woman met in the park last night. \((\text{collective})\)

• Analytical options:

1. Posit lexical ambiguity (e.g., [Link 1984]).
2. Unify based on “non-boolean” set/sum formation [Heycock and Zamparelli 2005].
3. This talk: Unify based on (boolean) intersection [Winter 2001].

• Analyzing the intersective behavior is easy:

(3) \([\text{liar and cheat}] = \lambda x.\text{liar}(x) \land \text{cheat}(x)\)

• What about the collective behavior?

• In a nutshell: and interacts with independently motivated type shifters.

\([\text{\[\text{N}\] man and woman}} = \text{[[DP a man and a woman]]}\]

*I thank Chris Barker, Dylan Bumford, Simon Charlow, Anna Szabolcsi, and Yoad Winter.
2 Analysis: Boolean And plus Type Shifters

2.1 The Boolean Assumption in Winter (2001)

- Basic meaning of and (e.g., Gazdar 1980; Winter 2001) is intersection

\[
\text{and}_{\text{bool}} = \cap_{(\tau,\tau')} = \begin{cases} \lambda X_\tau \lambda Y_\tau \lambda Z_{\sigma_1}. X(Z) \cap_{(\sigma_2,\sigma_2_{\tau'})} Y(Z) & \text{if } \tau = \sigma_1 \sigma_2 \\ \end{cases}
\]

- Winter’s idea: “distill” generalized quantifiers into (collective) individuals:

\[
\text{min} = \lambda Q_\tau t \lambda A_\tau. \ A \in Q \land \forall B \in Q. [B \subseteq A \rightarrow B = A]
\]

- For example:

\[
\begin{align*}
(6) & \text{a. } [\text{John}] = \{P \mid j \in P\} \\
& \text{b. } [\text{Mary}] = \{P \mid m \in P\} \\
& \text{c. } [\text{John and}_{\text{bool}} \text{Mary}] = \{P \mid j \in P\} \cap \{P \mid m \in P\} \\
& \text{d. } [\text{min(John and}_{\text{bool}} \text{Mary})] = \{\{j, m\}\}
\end{align*}
\]

- Collective predicates are predicates of sets of individuals:

\[
[\text{met (in the park)}] = \{P_{et} \mid P \in \text{met}\}
\]

- Existential closure acts in this case as a silent determiner:

\[
(7) \begin{align*}
& \text{a. } \text{Existential closure: } E = \lambda P_{\tau t} \lambda Q_\tau t. P \cap Q \neq \emptyset \\
& \text{b. } [E(\text{min}(\lambda P.P(\text{john}) \cap \lambda P.P(\text{mary})))] = \lambda C_{(et,t)}. \{j, m\} \in C
\end{align*}
\]

2.2 Application to Noun Coordination

- Basic idea: same operators as before, but in different order.

- Assume that E may apply to nouns and nominals (N’) without affecting their syntactic category.

\[
(8) \begin{align*}
& [[N’. \ E(\text{man}) \ \text{and}_{\text{bool}} \ E(\text{woman})]] \\
& = [[DP_a \ \text{a man and}_{\text{bool}} \ \text{a woman}]] \\
& = \lambda P_{et} \exists x \exists y. \ \text{man}(x) \land \text{woman}(y) \land P(x) \land P(y)
\end{align*}
\]
We need to “distill” this before further use:

\[(9) \quad \left[ \text{min}(E(\text{man}) \land \text{bool } E(\text{woman})) \right] = \lambda P_{et} \exists x \exists y. \text{man}(x) \land \text{woman}(y) \land P = \{x, y\}\]

Abbreviate this predicate as \textbf{mw-pair}.

This is of type \langle et, t \rangle. Determiners expect arguments of type \langle et \rangle.

With Winter, I assume this mismatch is repaired by determiner fitting \textit{dfit}:

\begin{equation}
(10) \quad \text{dfit} =_{\text{def}} \lambda D_{\langle et, (et, t) \rangle} \lambda A_{\langle et, t \rangle} \lambda B_{\langle et, t \rangle}. D(\bigcup A)(\bigcup (A \cap B))
\end{equation}

Winter motivates this operator by sentences like (11):

\begin{equation}
(11) \quad \text{No students met.}
\end{equation}

Assume that \textit{students} denotes the set of all nonempty sets of students.

The \textit{dfit} operator allows us to combine the GQ with the collective predicate:

\begin{equation}
(12) \quad \left[ \text{dfit}(\text{a})(\text{min}(E(\text{man}) \land \text{bool } E(\text{woman}))(\text{meet in the park})) \right] = \exists (\bigcup \text{mw-pair})(\bigcup (\text{mw-pair} \cap \text{meet in the park}))
\end{equation}

\begin{equation}
= \exists x. \exists y. \text{man}(x) \land \text{woman}(y) \land \text{meet in the park}(\{x, y\})
\end{equation}

“At least one man-woman pair met in the park.”

My LF for sentence (2b) is:

\begin{equation}
(13) \quad \left[ \text{dfit}(\text{a})(\text{min}(E(\text{man}) \land \text{bool } E(\text{woman}))(\text{meet in the park})) \right] = \exists (\bigcup \text{mw-pair})(\bigcup (\text{mw-pair} \cap \text{meet in the park}))
\end{equation}

\begin{equation}
= \exists x. \exists y. \text{man}(x) \land \text{woman}(y) \land \text{meet in the park}(\{x, y\})
\end{equation}

“At least one man-woman pair met in the park.”

Bonus: using \textit{dfit} makes sure the two people attend the same meeting.

3 Comparison to Previous Work

Improvement on Link (1984): no lexical ambiguity, hence no redundancy.
• Generalizes to S, VP, DP coordination since our conjunction is boolean.

3.1 Heycock and Zamparelli (2005)

Heycock and Zamparelli (2005): only one entry for and, but non-boolean

(14) \[ \text{[and, coll]} = \lambda Q_{(rt,t)} \lambda Q'_{(rt,t)} \lambda P_{rt} \exists A_{rt} \exists B_{rt}. Q(A) \land Q'(B) \land P = A \cup B \]

• Essentially, this combines two sets of sets by computing their cross-product.
• But instead of combining two elements into a pair, it forms their union.
• Nouns and VPs denote sets of singletons. Proper nouns denote singletons or their Montague lifts.

(15) \[ \text{[man and, coll woman]} = \lambda P_{et} \exists A_{et} \exists B_{et}. |A| = 1 \land A \subseteq \text{man} \land |B| = 1 \land B \subseteq \text{woman} \land P = A \cup B \]

\[ = \lambda P_{et} \exists x \exists y. \text{man}(x) \land \text{woman}(y) \land P = \{x, y\} \]

• This is equivalent to my denotation, but uses non-boolean and.
• Non-boolean and has problems with non-upward-entailing quantifiers:

(16) No man and no woman smiled.

• Suppose a man (John) and a woman (Mary) smile, and nobody else smiles.
• Here we judge (16) to be false but it turns out to be predicted true:

(17) a. \[ \text{[no man]} = \lambda Q_{(et,t)}. \neg \exists X_{(et)}. [\text{man}](X) \land Q(X) \]

b. \[ \text{[no woman]} = \lambda Q_{(et,t)}. \neg \exists X_{(et)}. [\text{woman}](X) \land Q(X) \]

• No man in (17a) holds of the set A = \{\{m\}\}, since A contains no man.
• No woman in (17b) holds of the set B = \{\{j\}\}, since B contains no woman.
• So no man and no woman holds of A \cup B, that is, \{\{j\}, \{m\}\}
• But this set is precisely the denotation of smiled. So (16) comes out as true!
• Heycock and Zamparelli suggest scope-splitting negative-concord analyses:

(18)  \( \text{no man} \approx \text{[not]} + \ldots + \text{[some man]} \)

(Ladusaw 1992)

• First problem: this doesn’t work well with coordination.

• Suppose that John and only John smiled.

(19)  
  a. Mary and nobody else smiled. (false)
  b. It’s not the case that Mary and someone else smiled. (true)

• Second problem: Standard English doesn’t have negative concord.


(20)  \[
\text{[man and pair woman]} = \langle \lambda x. \text{man}(x), \lambda x. \text{woman}(x) \rangle
\]

• This ordered pair is propagated upwards, a bit like in alternative semantics.

• At any point, \( \sqcap \) can be applied. So, \textit{and} can take variable scope.

• This correctly predicts the two readings of (21):

(21)  Every linguist and philosopher knows the Gödel Theorem.
  a. Everyone who is both a linguist and a philosopher knows the
      Gödel Theorem.
  b. Every linguist knows the Gödel Theorem, and every philoso-
      pher knows the Gödel Theorem.

• But it overgenerates:

(22)  \[ \text{[No girl sang and danced]} \neq \text{[No girl sang and no girl danced]} \]

(23)  
  a. \[ \text{[no girl]} = \lambda P. \neg \exists x[\text{girl}(x) \land P(x)] \]
  b. \[ \text{[sang and pair danced]} = \langle \lambda x. \text{sing}(x), \lambda x. \text{dance}(x) \rangle \]
  c. \[ \langle 23a \rangle \langle 23b \rangle = \langle \neg \exists x[\text{girl}(x) \land \text{sing}(x)], \neg \exists x[\text{girl}(x) \land \text{dance}(x)] \rangle \]
  d. Application of \( \sqcap \): \[ \neg \exists x[\text{girl}(x) \land \text{sing}(x)] \land \neg \exists x[\text{girl}(x) \land \text{dance}(x)] \]
  e. \[ = \text{[No girl sang and pair no girl danced]} \]
• The problem arises because boolean intersection can be delayed.

• To prevent this, Winter (1998) would have to make VP an “island” for $\sqcap$.

• As for me, I have to prevent $E$ from applying to verbs.

• The latter is natural since $E$ is a choice-function operator (Winter, 2001).

4 And vs. Or

• I assume Gazdar (1980)’s entry for or:

\[
\begin{align*}
[or_{\text{bool}}] = \sqcup(\tau,\tau) & \overset{\text{def}}{=} \begin{cases}
\lor(t,t) & \text{if } \tau = t \\
\lambda x \lambda y \lambda z \_ \_ X (Z) \sqcup (\sigma_1, \sigma_2, \sigma_2) & \text{if } \tau = \sigma_1 \sigma_2
\end{cases}
\end{align*}
\]

• Bergmann (1982): Why is (25a) equivalent but not (25b)?

(25) a. Every cat and dog is licensed. $\iff$ Every cat or dog is licensed.
   b. A cat and dog came running in. $\not\iff$ A cat or dog came running in.

• My answer: Two equivalent LFs for the sentences in (25a):

(26) a. $\text{dfit(every)}(\text{min}(E(\text{cat}) \text{ and}_{\text{bool}} E(\text{dog})))$($\text{dist(be\_licensed)}$)
   b. $\bigcup\{(x, y)|\text{cat}(x) \land \text{dog}(y)\} \subseteq \bigcup\{(x, y)|\text{cat}(x) \land \text{dog}(y) \land \{x, y\} \subseteq \text{be\_licensed}\}$

(27) a. $\text{every(}\text{cat or}_{\text{bool}} \text{ dog})(\text{be\_licensed})$
   b. $\text{cat} \cup \text{dog} \subseteq \text{be\_licensed}$

• My account generates only nonequivalent LFs for the sentences in (25b):

(28) a. $\text{dfit(a)}(\text{min}(E(\text{cat}) \text{ and}_{\text{bool}} E(\text{dog})))$($\text{dist(come\_running\_in)}$)
   b. $\exists x \exists y. \text{cat}(x) \land \text{dog}(y) \land \{x, y\} \subseteq \text{come\_running\_in}$

(29) a. $a(\text{cat or}_{\text{bool}} \text{ dog})(\text{come\_running\_in})$
   b. $\exists x. (\text{cat}(x) \lor \text{dog}(x)) \land \text{come\_running\_in}(x)$
5 The Non-Ambiguity of Or

- Unlike and, or never has boolean/collective “ambiguity” \cite{Payne1985}.

- Why is this so? The answer for noun coordination:

- \( N1 \text{ and } N2 \) can be “\( N1 \text{ and bool } N2 \)” or “\( \min(E(N1) \text{ and bool } E(N2)) \)”.
  These structures lead to different readings, as we have seen.

- \( N1 \text{ or } N2 \) can be “\( N1 \text{ or bool } N2 \)” or “\( \min(E(N1) \text{ or bool } E(N2)) \)”.
  These two structures evaluate to almost the same thing, and determiner fitting obliterates the remaining difference:

\[
(30) \quad \text{Every cat or dog was licensed.}
\]

\[
(31) \quad \text{dfit(every)} \left( \min(E(\text{cat})) \text{ or bool } \min(E(\text{dog})) \right) \text{ (dist(\text{be licensed}))} \\
= \text{dfit(every)} \left( \min\{\{P | P \cap \text{cat} \neq \emptyset \lor P \cap \text{dog} \neq \emptyset\}ight) \left(\{P | P \neq \emptyset \land P \subseteq \text{be licensed}\}\right) \\
= \bigcup\{\{x \mid x \in (\text{cat} \cup \text{dog})\} \subseteq \bigcup\{\{x \mid x \in (\text{cat} \cup \text{dog})\} \cap \{P | P \neq \emptyset \land P \subseteq \text{be licensed}\}\} \\
= (\text{cat} \cup \text{dog}) \subseteq ((\text{cat} \cup \text{dog}) \cap \text{be licensed}) \\
= (\text{cat} \cup \text{dog}) \subseteq \text{be licensed} \\
= \text{dfit(every)} \left( \min(\text{cat or bool } \text{dog}) \right) \text{ (dist(\text{be licensed}))}
\]

6 Summary and Outlook

- The boolean option is arguably the only unproblematic one outside the DP.

- I have shown that it is also preferrable within the DP.

- Next steps:
  - plural nouns (e.g. Ten men and women got married today) \cite{Link1984}.
  - “hydras” (e.g. every man and woman who met) \cite{Link1984}.
  - agreement across languages \cite{HeycockZamparelli2005}. 
References


