The Semantics of Logical Disjunction

Lecture 19

March 28, 2024

Announcements:

This lecture is supplemented by the following readings:

- Chierchia & McConnell-Ginet: Ch.4 (187–203)

Your sixth homework assignment is available and is due on Thursday, April 4th.

1 Introduction

We currently have a semantic system for interpreting sentences that relies on the following set of rules:

(1) **Functional Application (FA)**
If X is a node that has two daughters, Y and Z, and if $[\![\text{Y}]\!]$ is a function whose domain contains $[\![\text{Z}]\!]$, then $[\![\text{X}]\!] = [\![\text{Y}]\!]([\![\text{Z}]\!])$.

(2) **Predicate Modification (PM)**
If X is a node that has two daughters, Y and Z, and if $[\![\text{Y}]\!]$ and $[\![\text{Z}]\!]$ are in $D_{\langle e,t\rangle}$, then $[\![\text{X}]\!] = [\lambda x : x \in D_e \,.\, [\![Y]\!](x) = T \text{ and } [\![Z]\!](x) = T\ ]$

(3) **Non-Branching Nodes (NN) Rule**
If X is a non-branching node that has Y as its daughter, then $[\![\text{X}]\!] = [\![\text{Y}]\!]$

(4) **Terminal Nodes (TN) Rule**
If X is a terminal node, then $[\![\text{X}]\!]$ is specified in the lexicon.

These rules are capable of composing and interpreting structures that consist of a range of different kinds of expressions. More specifically they allow us to compute the **extension** of a sentence and derive its **truth conditions**.

(5) $[\![\text{S}]\!] = T$ *iff* p

Our semantic system also has a means for deriving the presuppositions that are introduced by an expression. These can be represented as a **domain restriction** on the extension of some expression.

(6) $[\lambda x : \boldsymbol{x \in D} \textbf{ and } \boldsymbol{D} \textbf{ is} \ldots \ . \ \ldots x \ldots\ ]$

Thus we have a system that is capable of computing two of the three types of informational content that we identified at the beginning of the semester.

(7) a. **Assertion**
Information explicitly contributed by an expression

b. **Presupposition**
Information that is taken for granted to be true by an expression

c. **Implicature**
Information that is implied/inferred from an expression

Our current goal is to develop a theory of implicatures like the one from the familiar exchange in (8).

(8) A: How was Fred's doctor appointment?
B: Well, he stopped smoking.
$\rightsquigarrow$ Fred's doctor appointment didn't go well.

That is, we are working on developing a set of rules/principles that allow us to compute the implicatures associated with any given expression.

Following the Gricean Theory of implicatures, we proposed the following overarching principle that governs rational, cooperative participation in a conversation:

(9) **The Cooperative Principle**
Make your conversational contribution such as is required, at the stage at which it occurs, by the accepted purpose or direction of the talk exchange in which you are engaged.

This made it possible to understand implicatures to represent deductive inferences that:

(i) arise from the fact that an expression *S* was asserted in a particular context, and

(ii) are validated by the assumption that the speaker is observe the Maxims of Conversation.

The Maxims of Converation can be stated succinctly as follows:

(10) **Maxims of Conversation**
a. *Maxim of Relevance*: Be relevant.
b. *Maxim of Quality*: Don't say false/unjustified things.
c. *Maxim of Quantity*: Don't say less/more than necessary.
d. *Maxim of Manner*: Be brief/orderly.

This theory of implicatures correctly predicts that implicatures show the following key properties:

(11) **Cancelability**
"*S* and/but not *p*" is consistent.

(12) **Reinforceability**
"*S* and/but *p*" is not redundant

2 The Puzzle of *either, but not both*

With the Maxims of Conversation at our disposal as a means for explaining the association of utterances with implicatures, we have the tools necessary to account for the puzzle of the logical connective *or*.

2.1 Recalling the Puzzle of *Some*

With respect to a sentence like (13), it was unclear whether a sentence containing the expression *S1 or S2* is *True* or *False* in a context where a parallel sentence containing *S1 and S2* is *True*.

(13) Mike dances or Mike golfs.

We observed that in a context like the one provided in (14), there is an intuition that we should draw the inference that it's not possible to have both potatoes and beans.

(14) *Waiter* : With your meal **you can have potatoes or you can have beans**.
$\rightsquigarrow$ You cannot have both potatoes and beans.

The question we asked at the time was what kind of informational content this inference represents.

As a piece of non-asserted content that is both cancelable and reinforceable, it seems that the inference in (14) is an implicature. This is an intuition that comes out relatively clearly when compared to the behavior of the asserted content.

(15) **Cancelability**
- a. You can have potatoes or you can have beans and you can also have both. (consistent)
- b. #You can have potatoes or you can have beans but you cannot have either. (inconsistent)

(16) **Reinforceability**
- a. You can have potatoes or you can have beans but you cannot have both. (non-redundant)
- b. #You can have potatoes or you can have beans and you can have either. (redundant)

Thus, the inference in (14) is an implicature. This means:

- When someone uses an expression of the form *S1 or S2*, it may give rise to the associated implicature that the parallel sentence of the form *S1 or S2* is False.
- The actual asserted content of an expression of the form *S1 or S2* does not include the information that *S1 and S2* is False.
- Instead, the asserted content of an expression of the form *S1 or S2* asserts only that *S1 or S2* is True and is entirely consistent with *S1 and S2* being True.

The question now becomes exactly how an utterance like (14) comes to have its associated implicature.

2.2 Deriving the Implicature of *either, but not both*

Our Gricean Theory of implicatures, which rely on the adopted Maxims of Conversation should generate this implicatures as a deductive inference.

(17) With your meal **you can have potatoes or you can have beans**.
$\rightsquigarrow$ You cannot have both potatoes and beans.

According to the Cooperative Principle, the Maxims of Conversation validate the following line of reasoning by the other conversational participants:

- The waiter has said only I can have the potatoes or I can have the beans.
- The waiter is following the **Maxim of Quantity**. Therefore, their statement was as informative as possible without breaking the other maxims.
- If the waiter had said instead that I can have the potatoes and the beans, they would have made a **more informative statement.** This is because saying that I can have the potatoes and the beans asymmetrically entails that I can have the potatoes or the beans.
- Since the waiter didn't say that I can have the potatoes and the beans, it follows that such an statement would have broken some other maxim.
- Saying that I can have the potatoes and the beans would have been relevant **(Relevance)** and it would have been brief and orderly **(Manner)**. Thus, it must violate the **Maxim of Quality**.
- Thus, it must be the case that the waiter believes that such a statement is false of they don't have enough evidence to assert it.
- But, **the waiter should know whether or not I can have the potatoes and the beans**. Thus, the waiter must know that it is false.
- **Therefore, I can have the potatoes or the beans, but I cannot have both.**

Informativity. Observe that the calculation of this implicature relies on our Informativity Metric.

(18) **Informativity Metric**
A sentence S1 is more informative than a sentence S2 if (S1 $\Rightarrow$ S2):
i. S1 entails S2 and
ii. S2 does not entail S1.

In the case at hand, *S1 and S2* entails *S1 or S2*.

(19) S1 : You can have the potatoes **and** you can have the beans. $\Rightarrow$
S2 : You can have the potatoes **or** you can have the beans.

Thus, S1 would have been more informative than the waiter's actual utterance.

Speaker Knowledge. The calculation of this implicature also crucially relies on a specific assumption regarding the speaker's knowledge.

- Namely, it is assumed that the waiter does or should know whether or not it is true that I can have both the potatoes and the beans.
- This licenses the inference that the waiter didn't say this because they know it's false, meaning that we should believe its false.
- Therefore, our theory predicts that, if this assumption regarding the speaker's knowledge can't or doesn't hold, then the implicature will not arise.

Reasons for thinking this is correct come from contexts like the one below:

(20) A: How did Louis usually get to work?
B: I'm not sure, but he doesn't have a car, so **he rode his bike or he took the bus**.
$\not\rightsquigarrow$ He doesn't both ride his bike and take the bus.

In this context, B explicitly denies knowledge of how Louis got to work. This correlates with the observation that the implicature does not arise.

2.3 Section Summary

Faced with the puzzle of whether a sentence containing the expression *S1 or S2* is *True* or *False* in a context where a parallel sentence containing *S1 and S2* is *True*, we now have an answer:

- A sentence containing an expression of the form *S1 or S1* **is True** in a context where a parallel sentence containing an expression of the form *S1 and S2* is also True.

  (21) *Context : Mike both dances and golfs*
  $[\![$ Mike dances or Mike golfs $]\!] = T$

- A sentence containing an expression of the form *S1 or S2* sometimes seems to be incompatible with the truth of the parallel sentence containing an expression of the form *S1 and S2*.

  (22) *Context : Mike both dances and golfs*
  #Mike dances or Mike golfs.

- But this intuition is the result of an **Scalar Imlicature** that arises due to the assumption that the speaker is following the **Maxim of Quantity** and knows whether *S1 and S2* is True or not.

  (23) $\langle$ and, or $\rangle$

3 Generalized and Particularized Implicatures

There is another interesting difference, which was observed in the original work by Grice, between the kinds of implicatures that we have been uncovering.

Put simply, implicatures differ on the basis of whether they should always be expected to arise or not. This is the difference between being a **generalized** implicature or a **particularized** implicature.

Particularized Implicatures. The appearance of certain implicatures crucially relies on the **particulars** of the context in which an expression S is uttered.

(24) **Particularized Implicature**
An implicature whose association with some sentence S depends on the particulars of the context in which S appears.

This means that, in most imaginable contexts, an expression S should not be expected to give rise to the implicature in question.

Consider the familiar exchange below:

(25) A: Are you going to the party tonight?
B: I have homework.
⇝ I'm not going to the party tonight.

In most other contexts, B's utterance does not give rise to the implicature that B is not going to the party. Of course, it may be that we calculate some other implicature on the basis of the particulars of the context.

(26) A: What are you doing?
B: I have homework.
$\not\rightsquigarrow$ I'm not going to the party.

Generalized Implicatures. The appearance of certain implicatures rely on very general features of the context in which S is uttered alongside our world knowledge.

(27) **Generalized Implicature**
An implicature that is typically associated with some sentence S with reliance only on general properties of the context in addition to world knowledge.

This means that, in most imaginable contexts, an expression S typically gives rise to the implicature in question.

Consider the familiar exchange below:

(28) Cindy has some of the markers.
⇝ Cindy does not have all of the markers.

In most contexts, an utterance containing an expression of the form *some of the NPs* give rise to the implicature that *all of the NPs* is false.

(29) Some of the chairs are dirty.
$\rightsquigarrow$ Not all of the chairs are dirty.

Not all generalized implicatures are Scalar Implicatures. But it seems that all Scalar implicatures are generalized implicatures.

It is, at least in part, due to the fact that generalized implicatures are so frequently associated with the usage of some expression, that it may at first appear that the implicature is part of the asserted meaning of an expression.

(30) **Scalar Implicatures as Generalized Implicatures**

a. *some of the NPs* $\rightsquigarrow$ *not all of the NPs*

b. *n NPs* $\rightsquigarrow$ *not more than* n *NPs*

c. *S1 or S2* $\rightsquigarrow$ *not S1 and S2*

4 The Semantics of Sentential Disjunction

Among the things that we have accomplished over the past few lectures is an understanding of the meaning of the logical connective *or*.

More specifically, we have determined that the asserted meaning of *or* simply says that, given two sentence S1 and S2, S1 is *True* **or** S2 is *True*, and this is entirely compatible with the possibility that S1 is *True* **and** S2 is *True*.

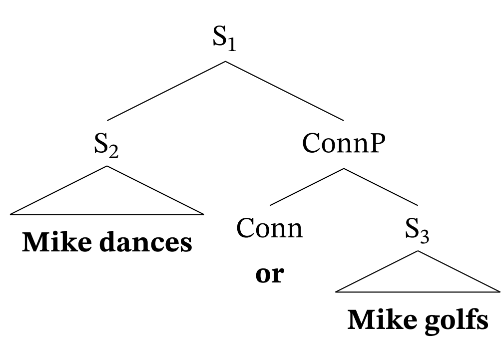
In other words we can provide truth conditional statements regarding the asserted content of sentences containing disjunctive *or* like the following:

(31) *Mike dances or Mike golfs* is *T iff* Mike dances or Mike golfs

This makes it possible for us to provide a lexical entry for the logical connective *or* and compute the meaning of sentences containing this expression.

Toward this end, let's assume that the syntax of disjunctive statements is as in (32):

(32)



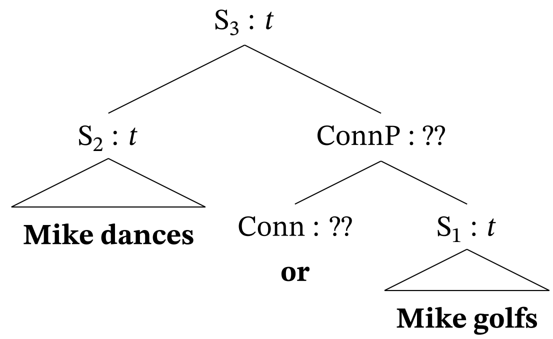
4.1 The Semantic Type of *Or*

1. Known Semantic Types. We start by listing out the semantic types we know and annotating the syntactic representation with this information.

(33) **Known Semantic Types in (34)**

a. $[\![S_1]\!] \in D_t$
b. $[\![S_2]\!] \in D_t$
b. $[\![S_3]\!] \in D_t$

(34)



2. Reasoning out the Semantic Type of a ConnP. We can appeal to the known semantic types and the rules of composition, to determine that the ConnP must be of type $\langle t, t \rangle$.

- The S_3 node fits the structural description for Functional Application. So, the extension of the S_2 must combine with the extension of the ConnP to return the extension of the S_3.

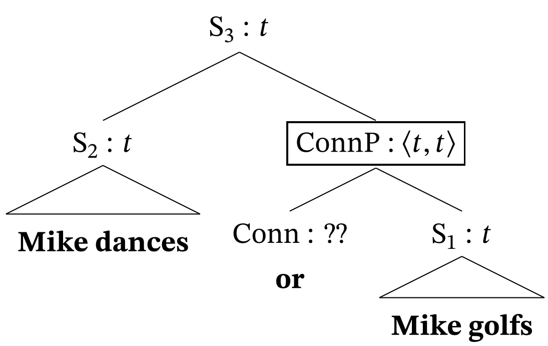
(35) $[\![S_2]\!] + [\![\text{ConnP}]\!] = [\![S_3]\!]$

- Because $[\![S_2]\!]$ is an expression of type t and $[\![S_3]\!]$ is an expression of type t, the extension of the ConnP must take as its argument the extension of the S_2 to return the type t extension of the S_3 as its value.

(36) $[\![\text{ConnP}]\!]([\![S_2]\!]) = [\![S_3]\!]$

- Thus, a ConnP must be a function of type $\langle t, t \rangle$.

(37)



3. Reasoning out the Semantic Type of *Or*. We can appeal to the known semantic types and the rules of composition, to determine that the logical connective *or* must be of type $\langle t, \langle t, t \rangle \rangle$.

- The ConnP node fits the structural description for Functional Application. So, the extension of the *or* must combine with the extension of the S_1 to return the extension of the ConnP.

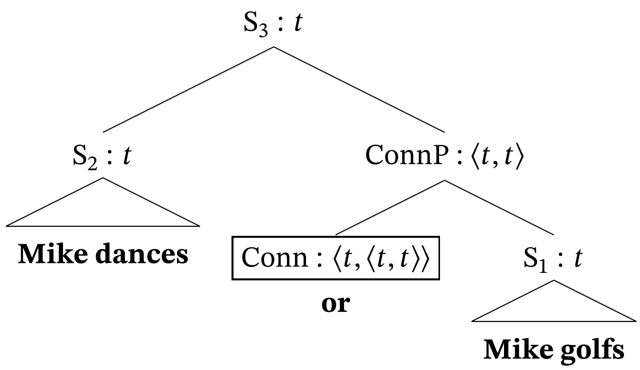
(38) $[\![\text{or}]\!] + [\![S_1]\!] = [\![\text{ConnP}]\!]$

- Because ⟦ S_1 ⟧ is an expression of type t and ⟦ ConnP ⟧ is an expression of type $\langle t, t\rangle$, the extension of *or* must take as its argument the type t extension of the S_1 to return the type $\langle t, t\rangle$ extension of the ConnP as its value.

(39) $[\![\text{ or }]\!]([\![\text{ S}_1\text{ }]\!]) = [\![\text{ ConnP }]\!]$

- Thus, the logical connective *or* must be a function of type $\langle t, \langle t, t\rangle\rangle$.

(40)



4.2 The Extension of *Or*

We now know what kind of expression the logical connective *or* must be. It is a type $\langle t, \langle t, t\rangle\rangle$ function, meaning it maps **truth values** to **function** from truth values to truth values.

We therefore need to develop a lexical entry for *or* that does the following:

- assigns as its extension a function of type $\langle t, \langle t, t\rangle\rangle$ and
- allows our system to derive an appropriate truth-conditional statement for sentences with the logical connective *or*.

(41) *Mike dances or Mike golfs* is T *iff* Mike dances or Mike golfs

Reasoning out the Extension of *ConnP*. We can start by reasoning out a type $\langle t, t\rangle$ extension for a ConnP.

- Let's start by considering the truth conditions of other sentences containing the ConnP *or Mike golf*.

(42) a. "Kuniko sings or Mike golfs" is T *iff* **Kuniko sings** or Mike golfs
b. "Sam dances or Mike golfs" is T *iff* **Sam dances** or Mike golfs
c. "Jason likes Taylor or Mike golfs" is T *iff* **Jason likes Taylor** or Mike golfs

- The key generalization to be appreciated here is that the ConnP *or Mike golfs* systematically combines with a sentence **S** to generate sentences with meanings of the type below:

(43) $[\![\ [_{S}\ \mathbf{S}\ [_{\text{ConnP}} \text{ or Mike golfs }]]\ ]\!] = T$ *iff* **S** or Mike golfs

- Given our notation, the meaning of some S, like *Kuniko sings*, in our metalanguage is essentially equivalent to the extension of that sentence having the value T.

(44) a. Kuniko sings $\approx$ $[\![$ Kuniko sings $]\!] = T$
b. Sam dances $\approx$ $[\![$ Sam dances $]\!] = T$
c. Jason likes Taylor $\approx$ $[\![$ Jason likes Taylor $]\!] = T$

- This means that, more generally, that the ConnP *or Mike golfs* systematically combines with a sentence S to generate sentences with meanings that are equivalently represented as:

(45) $[\![\ [_S\ \text{S}\ [_{\text{ConnP}}\ \text{or Mike golfs}\ ]]\ ]\!] = T \quad \textit{iff} \quad [\![\ \text{S}\ ]\!] = T$ or Mike golfs

- Now, we determined in section 4.1 that $[\![\ [_{\text{ConnP}}\ \text{or Mike golfs}\ ]\ ]\!]$ is an expression of type $\langle t, t\rangle$. Consequently, our rule of FA entails that the following equivalency holds:

(46) $[\![\ [_S\ \text{S}\ [_{\text{ConnP}}\ \text{or Mike golfs}\ ]]\ ]\!] \quad = \quad [\![\ [_{\text{ConnP}}\ \text{or Mike golfs}\ ]\ ]\!]([\![\ \text{S}\ ]\!])$

- It follows from the previous two points that:

(47) $[\![\ [_{\text{ConnP}}\ \text{or Mike golfs}\ ]\ ]\!]([\![\ \text{S}\ ]\!]) = T \quad \textit{iff} \quad [\![\ \text{S}\ ]\!] = T$ or Mike golfs

- Therefore, $[\![$ ConnP $]\!]$ is a function which takes some sentence p as its argument and yields T *iff* either $p = T$ or Mike golfs.

(48) $[\![\ [_{\text{ConnP}}\ \text{or Mike golfs}\ ]\ ]\!] = [\lambda p\ :\ p \in D_t\ .\ p = T$ or Mike golfs$]$

Reasoning out the Extension of *Or*. We can now use the results above to reason out a type $\langle t, \langle t, t\rangle\rangle$ extension for the logical connective *or*.

- We can start again by considering the truth conditions of other sentences with different ConnPs containing the logical connective *or*.

(49) a. "Mike dances or Kuniko sings" is T *iff* Mike dances or **Kuniko sings**
b. "Mike dances or Sam golfs" is T *iff* Sam dances or **Sam golfs**
c. "Mike dances or Jason likes Taylor" is T *iff* Mike dances or **Jason likes Taylor**

- With the results above still in mind, these data reveal that that *or* combines with a sentence S to generate each of the ConnP meanings below:

(50) a. $[\![\ [_{\text{ConnP}}\ \text{or Kuniko sings}\ ]\ ]\!] = [\lambda p\ :\ p \in D_t\ .\ p = T$ or **Kuniko sings**$]$
b. $[\![\ [_{\text{ConnP}}\ \text{or Sam golfs}\ ]\ ]\!] = [\lambda p\ :\ p \in D_t\ .\ p = T$ or **Sam golfs**$]$
c. $[\![\ [_{\text{ConnP}}\ \text{or Jason likes Taylor}\ ]\ ]\!] = [\lambda p\ :\ p \in D_t\ .\ p = T$ or **Jason likes Taylor**$]$

- The key generalization to be appreciated here is that *or* systematically combines with a sentence **S** to generate ConnPs with meanings of the type below:

(51) $[\![\ [_{\text{ConnP}}\ \text{or}\ \mathbf{S}\ ]\ ]\!] \quad = \quad [\lambda p\ :\ p \in D_t\ .\ p = T\ \text{or}\ \mathbf{S}\ ]$

- Given our notation, the meaning of some S, like *Kuniko sings*, in our metalanguage is essentially equivalent to the extension of that sentence having the value T.

(52) a. Kuniko sings $\approx$ $[\![$ Kuniko sings $]\!] = T$
b. Sam dances $\approx$ $[\![$ Sam dances $]\!] = T$
c. Jason likes Taylor $\approx$ $[\![$ Jason likes Taylor $]\!] = T$

- This means that, more generally, that the logical connective *or* systematically combines with a sentence S to generate ConnPs with meanings that are equivalently represented as:

(53) $[\![\ [_{\text{ConnP}} \text{ or S }]\ ]\!] \quad = \quad [\lambda p \in D_t \,.\, p = T \text{ or } [\![\text{ S }]\!] = T]$

- Now, we determined in section 4.1 that $[\![$ or $]\!]$ is an expression of type $\langle t, \langle t, t\rangle\rangle$. Consequently, our rule of FA entails that the following equivalency holds:

(54) $[\![\ [_{\text{ConnP}} \text{ or S }]\ ]\!] \quad = \quad [\![\text{ or }]\!]([\![\text{ S }]\!])$

- It follows from the previous two points that:

(55) $[\![\text{ or }]\!]([\![\text{ S }]\!]) \quad = \quad [\lambda p \in D_t \,.\, p = T \text{ or } [\![\text{ S }]\!] = T]$

- Therefore, $[\![$ or $]\!]$ is a function which takes some sentence q as its argument and yields the following function as its value:

(56) $[\lambda p : p \in D_t \,.\, p = T \text{ or } q = T]$

- Putting everything together, we end up with the following lexical entry for the logical connective *or*:

(57) $[\lambda q : q \in D_t \,.\, [\lambda p : p \in D_t \,.\, p = T \text{ or } q = T]]$

4.3 A Proof of the Truth Conditions

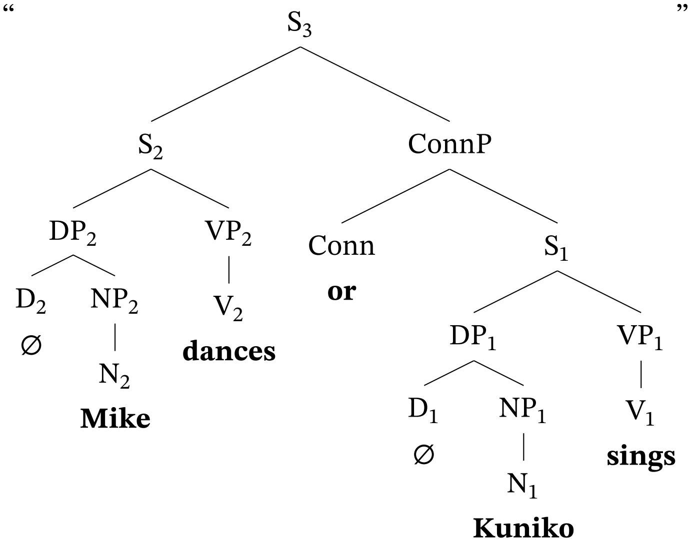
We can now see how the predicted truth-conditional statement in (58) can be computed with subproofs of the constituents of the utterance.

(58) **Truth-Conditional Statement of *Mike dances or Kuniko sings***
Mike dances or Kuniko sings is T *iff* Mike dances or Kuniko sings

(59) **Lexical Entries for *Mike dances or Kuniko sings***
a. $[\![D_{\varnothing}]\!] = [\lambda x \in D_e \,.\, x\ ]$
b. $[\![$ Mike $]\!] =$ Mike
c. $[\![$ Kuniko $]\!] =$ Kuniko
d. $[\![$ dances $]\!] = [\lambda x \in D_e \,.\, x \text{ dances }]$
e. $[\![$ sings $]\!] = [\lambda x \in D_e \,.\, x \text{ sings }]$
f. $[\![$ or $]\!] = [\lambda q \in D_t \,.\, [\lambda p \in D_t \,.\, p = T \text{ or } q = T]]$

(60) **Calculation of the Truth Conditions of *Mike dances or Kuniko sings***

i. "Mike dances or Kuniko sings" is T *iff* (by syntax)

ii. " 

" is T *iff* (by notation)

iii. ⟦ S_3 ⟧ $= T$

iv. *Calculation of* ⟦ NP_2 ⟧

   a. ⟦ NP_2 ⟧ = (by NN, TN)

   b. Mike

v. *Calculation of* ⟦ DP_2 ⟧

   a. ⟦ DP_2 ⟧ = (by FA)

   b. ⟦ $\varnothing_D$ ⟧(⟦ NP_2 ⟧) = (by TN)

   c. $[\lambda x \in D_e\ .\ x]$(⟦ NP_2 ⟧) = (by iv.)

   d. $[\lambda x \in D_e\ .\ x]$(Mike) = (by LC)

   e. Mike

vi. *Calculation of* ⟦ VP_2 ⟧

   a. ⟦ VP_2 ⟧ = (by NN, TN)

   b. $[\lambda x \in D_e\ .\ x$ dances $]$

vii. *Calculation of* ⟦ S_2 ⟧

   a. ⟦ S_2 ⟧ = (by FA)

   b. ⟦ VP_2 ⟧(⟦ DP_2 ⟧) = (by v., vi.)

   c. $[\lambda x \in D_e\ .\ x$ dances $]$(Mike)

viii. *Calculation of* $[\![\text{NP}_1]\!]$

a. $[\![\text{NP}_1]\!]$ = (by NN, TN)

b. Kuniko

ix. *Calculation of* $[\![\text{DP}_1]\!]$

a. $[\![\text{DP}_1]\!]$ = (by FA)

b. $[\![\varnothing_\text{D}]\!]([\![\text{NP}_1]\!])$ = (by TN)

c. $[\lambda x \in D_e \,.\, x]([\![\text{NP}_1]\!])$ = (by viii.)

d. $[\lambda x \in D_e \,.\, x]$(Kuniko) = (by LC)

e. Kuniko

x. *Calculation of* $[\![\text{VP}_1]\!]$

a. $[\![\text{VP}_1]\!]$ = (by NN, TN)

b. $[\lambda x \in D_e \,.\, x$ sings $]$

xi. *Calculation of* $[\![\text{S}_1]\!]$

a. $[\![\text{S}_1]\!]$ = (by FA)

b. $[\![\text{VP}_1]\!]([\![\text{DP}_1]\!])$ = (by ix., x.)

c. $[\lambda x \in D_e \,.\, x$ sings $]$(Kuniko)

xii. *Calculation of* $[\![\text{ConnP}]\!]$

a. $[\![\text{ConnP}]\!]$ = (by FA)

b. $[\![\text{or}]\!]([\![\text{S}_1]\!])$ = (by TN)

c. $[\lambda q \in D_t \,.\, [\lambda p \in D_t \,.\, p = T \text{ or } q = T]]([\![\text{S}_1]\!])$ = (by LC)

d. $[\lambda p \in D_t \,.\, p = T \text{ or } [\![\text{S}_1]\!] = T]$ = (by xi.)

e. $[\lambda p \in D_t \,.\, p = T$ or $[\lambda x \in D_e \,.\, x$ sings $]$(Kuniko) $= T]$ = (by LC, def..)

f. $[\lambda p \in D_t \,.\, p = T$ or Kuniko sings $]$

xiii. *Calculation of* $[\![\text{S}_3]\!]$

a. $[\![\text{S}_3]\!]$ = (by FA)

b. $[\![\text{ConnP}]\!]([\![\text{S}_2]\!])$ = (by xii.)

c. $[\lambda p \in D_t \,.\, p = T$ or Kuniko sings $]([\![\text{S}_2]\!])$ = (by LC)

d. $[\![\text{S}_2]\!] = T$ or Kuniko sings = (by vii.)

e. $[\lambda x \in D_e \,.\, x$ dances $]$(Mike) $= T$ or Kuniko sings = (by LC, def..)

f. Mike dances or Kuniko sings

Thus, "Mike dances or Kuniko sings" is T *iff* Mike dances or Kuniko sings.

With these truth conditions, we expect the sentence at hand to have the following truth values in the corresponding contexts:

(61) ⟦ **Mike dances or Kuniko sings** ⟧ =

a. T if $p = T$ and $q = F$
Mike dances but Kuniko doesn't sing.

b. T if $p = F$ and $q = T$
Mike doesn't dance but Kuniko sings.

c. F if $p = F$ and $q = F$
Mike doesn't dance and Kuniko doesn't sing

d. $\boldsymbol{T}$ if $p = T$ and $q = T$
Mike dances and Kuniko sings.

As noted, the intuition that *S1 or S2* is incompatible with the truth of *S1 and S2* in (61d) is the result of a Scalar Implicature.

5 Practice

Exercise. Consider again the lexical entry that we provided for the logical connective *or*:

(62) $[\![\text{ or }]\!] = [\lambda q : q \in D_t . [\lambda p : p \in D_t . p = T \text{ or } q = T]]$

What lexical entry could we provide for the logical connective *and* that would deliver truth-conditional statements like in (63):

(63) *Mike dances and Kuniko sings* is T *iff* Mike dances and Kuniko sings

(64) ⟦ and ⟧ =