Lecture notes and example exercise

Lab class session 1, for assignment 2

Zeqi Zhao

Nov 8, 2023

Basics of Sets

Set membership

Note the difference betwen $\in$ and $\subseteq$.

$x \in$ S means '**x is a member of set S**'

{a, b, c} is the set whose only members are a, b, and c, and nothing else.

$\{a, b, c\} = \{b, c, a\} = \{a, a, a, c, c, b, b\}$

A set can also be a member of another set.

$\{a, b, c\} \in \{\{a, b, c\}, a, b\}$

$a \in \{a\}$

$\{a\} \in \{\{a\}\}$

but $a \neq \{a\} \neq \{\{a\}\}$

Subset relation

$A \subseteq B$ means '**A is a subset of B**'.

How to check: see if all members of A are also members of B

$A = \{a, b, c\}$

$B = \{a, b\}$

$B \subseteq A$

$\varnothing$ is a subet of every set, including itself.

Set Equivalence

When A and B have the same members, we say 'A and B are equal' and write A = B

$A = \{a, b, c\}$

$B = \{a, b, c\}$

$A = B$

Note that when A = B, A and B are each other's subset.

Exercise: Basics of Sets

Is (1) true?

(1)   b $\in \{2, 3, 4\}$

No, the only members in $\{2, 3, 4\}$ are 2, 3, 4; b is not among them.

Exercise: Basics of Sets

Is (2) true?

(2) $\{2, 3\} \in \{2, 3, 4\}$

No, the only members in $\{2, 3, 4\}$ are 2, 3, 4; the set $\{2, 3\}$ is not among them.

Exercise: Basics of Sets

Is (3) true?

(3) $\varnothing = \{\varnothing\}$

No, an empty set is not the same as a set that contains an empty set.

Exercise: Basics of Sets

Is (4) true?

(4) $\{a, c\} \in \{a, b, 3, \{a, c\}\}$

Yes, $\{a, b, 3, \{a, c\}\}$ has the set $\{a, c\}$ as a member.

Exercise: Basics of Sets

Is (5) true?

(5) $\{a, c\} \subseteq \{a, b, 3, \{a, c\}\}$

No, the members in $\{a, c\}$ are a, c;

the members in $\{a, b, 3, \{a, c\}\}$ are a, b, 3, $\{a, c\}$.

Not all the members in $\{a, c\}$ are members of $\{a, b, 3, \{a, c\}\}$.

How to define sets

Two ways to define sets

- Name all the members
put { } around the members. like {1, 2, 3}. But what if there are infinite members?

- Abstraction: We define sets by using **variables** like x, y.

If I want a set that contains all the white cats.

This set would contain
an object such that this object is a white cat.

We write this as:
{x : x is a white cat}

Variables don't refer

The variable don't refer to a particular object, so the names don't matter:

$\{x : x$ is a white cat$\} = \{y : y$ is a white cat$\} = \{z : z$ is a white cat$\}$

All three sets are equal: they are the same set that contains all the white cats.

Variables must be bound

If a variable appears before the : , you go and find it after the : .

{ x : x is a white cat}

Since the part after the : is describing the condition for the variable x, so x must be in this part.

This is wrong because x is not bound.

{x : y is a white cat}

Variables vs. Non-Variables

This is a set of the objects who kissed Eve.

$\{x : x$ kissed Eve$\}$

This is a set of the objects who kissed Mia.

$\{x : x$ kissed Mia$\}$

What about this then?

$\{x : x$ kissed $b\}$

Variables vs. Non-Variables

Since b is not bound, i.e. you don't find b before the : ,
b is not a variable.

$\{x : x \text{ kissed } b\}$

Just like *Eve* and *Mia*, b is a name of an object.

So this is a set of objects who kissed b.

How to read complex sets

When you see layers of $\{\ \}$, first go to the inner layer.

$\{y\text{: } y \in \{x\text{: x kissed Eve}\}\}$

Set A is a set of objects who kissed Eve.

$A = \{x\text{: x kissed Eve}\}$

We replace the inner layer set with A.

$\{y\text{: } y \in \{x\text{: x kissed Eve}\}\} = \{y\text{: } y \in A\}$

Now things look way more clear:
The outer layer set is still the set of objects who kissed Eve.

Exercise: Set Equivalence

Is (6) true?

(6) $\{x : x \in A\} = A.$

Yes, $\{x : x \in A\}$ is a set which contains all members of A, therefore equal to A.

Exercise: Set Equivalence

Is (7) true?

(7) $\{x : \text{a kissed } x\} = \{y : \text{b kissed } y\}$

Only if either

- a $=$ b. If they are, then both sets denote the set of objects kissed by a (or b for that matter).

- or a $\neq$ b, but a and b kissed exactly the same objects.

Exercise: Set Equivalence

Is (8) true?

(8) $\{$Eve: Eve saw Mia$\}$ = $\{$Mia: Mia saw Eve$\}$

Since we don't use Eve and Mia as variables, but only to refer to individuals, both sets are ill-defined. You cannot abstract over an individual.

Exercise: Set Equivalence

Is (9) true?

(9) $\{x : \{y : x \text{ loves } y\} = \varnothing\} = \{y : \{x : y \text{ loves } x\} = \varnothing\}$

Yes,

First the inner layer set on the **left**.

> $\{y : x \text{ loves } y\} = \varnothing$ means: the set of objects loved by x is the empty set.

Then the outer layer.

> $\{x : \{y : x \text{ loves } y\} = \varnothing\}$ is the set of objects x such that the set of objects loved by x is the empty set, i.e., **the set of objects who do not love anything**

Exercise: Set Equivalence

Is (9) true?

(9) $\big\{x : \{y : x \text{ loves } y\} = \varnothing\big\} = \big\{y : \{x : y \text{ loves } x\} = \varnothing\big\}$

Then the inner layer set on the **right**.

> $\{x : y \text{ loves } x\} = \varnothing$ means: the set of objects loved by y is the empty set.

Then the outer layer.

> $\big\{y : \{x : y \text{ loves } x\} = \varnothing\big\}$ is the set of objects y such that the set of objects loved by y is the empty set, i.e., **the set of objects who do not love anything**

Exercise: Set Equivalence

Is (10) true? **Under which condition?**

(10) $\{x : \{y : x \text{ loves } y\} = \varnothing\} = \{x : \{y : y \text{ loves } x\} = \varnothing\}$

First the inner layer set on the **left**.

> $\{y : x \text{ loves } y\} = \varnothing$ means: the set of objects loved by x is the empty set.

Then the outer layer.

> $\{x : \{y : x \text{ loves } y\} = \varnothing\}$ is the set of objects x such that the set of objects loved by x is the empty set, i.e., **the set of objects who do not love anything**

Exercise: Set Equivalence

Is (10) true? **Under which condition?**

(10) $\{x : \{y : x \text{ loves } y\} = \varnothing\} = \{x : \{y : y \text{ loves } x\} = \varnothing\}$

Then the inner layer set on the **right**.

$\{y : y \text{ loves } x\} = \varnothing$ means: the set of objects who loves x is the empty set.

Then the outer layer.

$\{x : \{y : y \text{ loves } x\} = \varnothing\}$ is the set of objects x such that the set of objects who loves x is the empty set, i.e., **the set of objects who are not loved by anything**

Exercise: Set Equivalence

Is (10) true? **Under which condition?**

(10) $\{x : \{y : x \text{ loves } y\} = \varnothing\} = \{x : \{y : y \text{ loves } x\} = \varnothing\}$

(10) is true when the people who do not love anything are the same people who are not loved by anything.

Functions

Functions

Think about functions a special type of **relations**:

It is like a machine, for each input (**argument**) you feed it, it returns a **unique value** as its output.

This returned value is written as **F(A)**.

This is how you read it:
'F applied to A' or 'F of A'

Defining a function: Domain and Range

Since function is mapping from arguments to values, to define a function, you need to specify its domain and range:

Domain: The set of objects the function can take as inputs.

Range: The set of objects the function returns as outputs.

For example, if a function is $f(x) = x + 2$, when the domain is the set of all natural numbers, the range would the set of all natural numbers.

You feed it 2, it gives you the unique value 4.

Ways of defining functions

We may define functions with lists, tables, or words.

- $F = \{< a, b >, < c, b >, < d, e >\}$
- $F = \begin{bmatrix} a \to b \\ c \to b \\ d \to e \end{bmatrix}$

- *F* is a function *f* with domain {a, c, d} such that *f(a)=f=(c)=b* and *f(d)=e*

Exercise: Relations vs. functions

Name examples of ordered pairs that are in relation R:

(11) $R := \{\langle x, y \rangle : x \text{ is a country and } y \text{ is the national capital of } x\}$

E.g.:

$\langle$Germany, Berlin$\rangle \in R$
$\langle$India, New Delhi$\rangle \in R$
$\langle$Norway, Oslo$\rangle \in R$
$\langle$China, Beijing$\rangle \in R \ldots$

Exercise: Relations vs. functions

(12) Is R a function?

$R := \{\langle a, b\rangle, \langle c, d\rangle\}$

Yes. Every left component of an ordered pair in R has exactly one right component.

Exercise: Relations vs. functions

(13) Is R a function?

$R := \{\langle x, a\rangle, \langle x, b\rangle\}$

Not generally, only if $a = b$.

If $a \neq b$, x has two distinct right components. In that case R is a mere relation.

Exercise: Relations vs. functions

(14) Is R a function?

$$R := \{\langle x, a\rangle, \langle y, a\rangle\}$$

Yes. Every left component in a pair in R corresponds to exactly one right component. I.e., a right component may correspond to more than one left component.

Exercise: Relations vs. functions

(15) Is R a function?

$R := \{\langle x, y \rangle : x \text{ is a person and } y \text{ is a sibling of } x\}$

No, it is possible for people to have no or more than one siblings, i.e., every left component does not have to correspond to exactly one right component.

Exercise: Relations vs. functions

(16) Is R a function?

$R := \{\langle x, y\rangle : x \text{ is a person and } y \text{ is the biological mother of } x\}$

Yes, a person can only have one biological mother, i.e., every left component corresponds to exactly one right component.

Exercise: Defining domain and range

(17) Define the domain and range of F:

$F := \{\langle x, y\rangle : x$ is an author and y is x's best-selling book$\}$

Domain of $F = \{x : x$ is an author who published books $\}$

Range of $F = \{x : x$ is a book and x is the best-selling book of an author$\}$

Exercise: Defining domain and range

(18) Define the domain and range of F:

$F := \{\langle x, y\rangle : x$ is an object and y is x's year of birth$\}$

Domain of $F = \{x : x$ is an object such that there is a year in which they were born $\}$

Range of $F = \{x : x$ is a year and some object was born in $x\}$

Exercise: Defining functions

(19) Define the function F in words:

$F := \{\langle x, y \rangle : x$ is an object and y is x's year of birth$\}$

Domain of F is the set A = $\{x : x$ is an object such that there is a year in which they were born $\}$

Range of F is the set B = $\{x : x$ is a year and some object was born in $x\}$

Let F be the function f from A to B, and for every argument x $\in$ A is mapped by f to its year of birth.

Exercise: Defining functions

(20) Define the function F in words:

$F := \{\langle x, y\rangle : x$ is a person and y is x's favourite kid$\}$

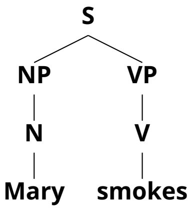
Domain of F is the set A = $\{x : x$ is a person who has a favourite kid$\}$

Range of F is the set B = $\{x : x$ is a kid and some person's favourite kid is $x\}$

Let F be the function f from A to B, and for every argument x $\in$ A is mapped by f to its favourite kid.

Proof of truth-conditions

top-down vs bottom-up

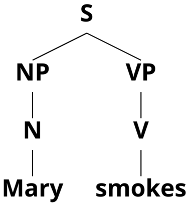


Top-down: Start with the **S-node** and de-compose all the way down to the terminal nodes *Mary* and *smokes*.

Bottom-up: Start with **the terminal nodes *Mary* and *smokes*** and compose all the way up to the S-node.

Three things we already know

- **Sentences denotes a truth-value (0 or 1).**



Assumption: $[\![S]\!] = 1$ iff Mary smokes

- **Composinality** tells us $[\![S]\!]$ is determined by $[\![NP]\!]$ and $[\![VP]\!]$ and how they combine.

Three things we already know

- **proper names denote individuals**

Native speakers use a **proper name** to **refer** to a particular individual/entity.

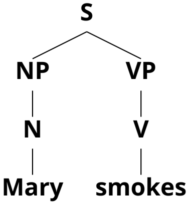
$[\![\text{NP}]\!] = [\![\text{N}]\!] = [\![\text{Mary}]\!] = \text{Mary}$

NP
|
N
|
Mary

NP inherits denotation from N (Rule S2)

N inherits denotation from Mary (Rule S4)

What's the VP denotation



⟦VP⟧ combines with ⟦NP⟧ and the output is ⟦S⟧.

⟦VP⟧ should be a function that **take an individual as its input** and **output a truth-value**.

Intransitive verbs as functions

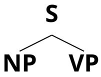
Elements of D: set of actual individuals/entities

Elements of $\{0, 1\}$: set of truth-values

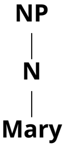
Intransitive verbs like *smoke* is the function that maps any individual in a situation to 1 iff that individual smokes in that situation.

$[\![\textbf{smokes}]\!] = f : D \to \{0, 1\}$
For all $x \in D, f(x) = 1$ iff x smokes

Functional application and Non-Branching Nodes



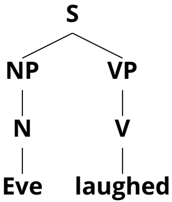
When you combine two **branching nodes**, one of them, like ⟦**VP**⟧, needs to be a function that takes the other one, like ⟦**NP**⟧, as its argument.



Non-branching nodes inherit the denotations of their daughters.

Exercise: Top-Down Computation

Using rules S1 to S5 prove top-down that ⟦S⟧ = 1 iff Eve laughed.



Assumption: ⟦S⟧ $=$ 1 iff Eve laughed

Exercise: Top-Down Computation

Here are the 5 rules. You don't need to memorize them.

S1 If $\boldsymbol{\alpha}$ has the form $[_{\mathbf{S}}\ \boldsymbol{\beta}\ \boldsymbol{\gamma}]$, then $[\![\boldsymbol{\alpha}]\!] = [\![\boldsymbol{\gamma}]\!]([\![\boldsymbol{\beta}]\!])$.

S2 If $\boldsymbol{\alpha}$ has the form $[_{\mathbf{NP}}\ \boldsymbol{\beta}]$, then $[\![\boldsymbol{\alpha}]\!] = [\![\boldsymbol{\beta}]\!]$.

S3 If $\boldsymbol{\alpha}$ has the form $[_{\mathbf{VP}}\ \boldsymbol{\beta}]$, then $[\![\boldsymbol{\alpha}]\!] = [\![\boldsymbol{\beta}]\!]$.

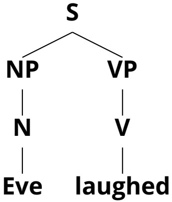
S4 If $\boldsymbol{\alpha}$ has the form $[_{\mathbf{N}}\ \boldsymbol{\beta}]$, then $[\![\boldsymbol{\alpha}]\!] = [\![\boldsymbol{\beta}]\!]$.

S5 If $\boldsymbol{\alpha}$ has the form $[_{\mathbf{V}}\ \boldsymbol{\beta}]$, then $[\![\boldsymbol{\alpha}]\!] = [\![\boldsymbol{\beta}]\!]$.

S1 is functional application for branching nodes, S2-S5 are for non-branching nodes.

Exercise: Top-Down Computation

Let's start from $[\![\text{S}]\!]$ and try to de-compose it into $[\![\text{NP}]\!]$ and $[\![\text{VP}]\!]$

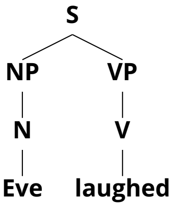


Assumption: $[\![\text{S}]\!] = 1$ iff Eve laughed

$$[\![\textbf{S}]\!] = [\![\textbf{VP}]\!]([\![\textbf{NP}]\!]) \tag{S1}$$

Exercise: Top-Down Computation

Then go down the non-branching nodes.



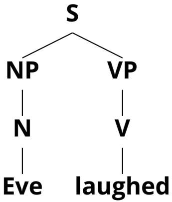
Assumption: $\llbracket S \rrbracket = 1$ iff Eve laughed

$$\llbracket \textbf{S} \rrbracket = \llbracket \textbf{VP} \rrbracket(\llbracket \textbf{NP} \rrbracket) \quad \text{(S1)}$$

$$\begin{aligned} \llbracket \textbf{NP} \rrbracket &= \llbracket \textbf{N} \rrbracket && \text{(S2)} \\ &= \llbracket \textbf{Eve} \rrbracket && \text{(S4)} \end{aligned}$$

$$\begin{aligned} \llbracket \textbf{VP} \rrbracket &= \llbracket \textbf{V} \rrbracket && \text{(S3)} \\ &= \llbracket \textbf{laughed} \rrbracket && \text{(S5)} \end{aligned}$$

Exercise: Top-Down Computation



Assumption: $[\![\mathbf{S}]\!] = 1$ iff Eve laughed

$$
\begin{aligned}
[\![\mathbf{S}]\!] &= [\![\mathbf{VP}]\!]([\![\mathbf{NP}]\!]) && \text{(S1)} \\
&= [\![\mathbf{laughed}]\!]([\![\mathbf{Eve}]\!]) && ([\![\mathbf{VP}]\!], [\![\mathbf{NP}]\!]) \\
&= \left[\begin{array}{l} f : D \to \{0,1\} \\ \text{For all } x \in D, f(x) = 1 \text{ iff } x \text{ laughed} \end{array} \right] (\text{Eve}) && (\textit{lexicon}) \\
&= 1 \text{ iff Eve laughed}
\end{aligned}
$$

$[\![\mathbf{S}]\!]$= 1 iff Eve laughed.

End of proof