Handout 2: Formal Foundations[1]

Annemarie van Dooren (University of Maryland)
avdooren@umd.edu

Goal of this course:
How does sentence meaning depend on its parts?

Today:
An introduction into sets, relations and functions.

Why do we need this? Remember, the field of formal semantics makes use of the techniques of logic. The basic model of the specific truth-conditional semantics we study is based on sets, relations and functions.

Sets

Three properties of sets:
1. A set is an abstract collection of objects. Anything can be a set: the numbers from 1 to 10, the students in this class, or the number 2, 7 and one student in this class.
2. As it is an abstract collection, it doesn't mean anything if you're in the same set twice: once you're in the set, you're there.
3. Sets are unordered.

Notation:
{x, y, z} = the set of the items x, y, and z
{x: x is an apple} = the set of all apples
{∅} = the empty set

How do we talk about it:
{x, y, z} : x, y, and z are the *members* or *elements* of the set.
{x: x is an apple} the set of all x such that x is an apple

The *cardinality* of a set is the number of members it has.

Notation:
|{x, y, z}| = 3
|{x: x is a natural number}| = ∞
|{∅}| = 0
|{x}| = 1

How do we talk about it:
|{x, y, z}| = the cardinality of the set containing x, y, and z
|{∅}| = the cardinality of the empty set

[1] This handout is based on Alexander Williams's handouts presented in LING410: *Meaning and Grammar*, UMD 2018 and Partee, Ter Meulen & Wall (1990) *Mathematical methods in Linguistics*, chapter 1 and 2.

$|\{x\}|$ = the singleton set

Set-theoretic relations

Membership is a one-way relation between an object and a set.

Notation

$x \in \{x, y, z\}$
$v \notin \{x, y, z\}$

How do we talk about it:

$x \in \{x, y, z\}$ = x is a member of the set containing x, y, z
$v \notin \{x, y, z\}$ = v is not a member of the set containing x, y, z

Subset is a one-way relation between two sets.
A is a subset of B just when every member of A is also a member of B.

Notation

$\{x\} \subseteq \{x, y, z\}$
$\{x, y, z\} \subseteq \{x, y, z, a, b\}$
$\{a\} \nsubseteq \{x, y, z\}$
$x \nsubseteq \{x, y, z\}$

How do we talk about it:

$\{x\} \subseteq \{x, y, z\}$ = the set containing x is a subset of the set containing x, y, and z

Proper subset is a one-way relation between two sets.
A is a proper subset of B just when every member of A is also a member of B, *and also* A and B are not equal.

Notation

$\{x\} \subset \{x, y, z\}$
$\{x, y, z\} \subset \{x, y, z, a, b\}$
$\{a\} \not\subset \{x, y, z\}$
$x \not\subset \{x, y, z\}$

$\{x, y\} \subseteq \{x, y\}$
$\{x, y\} \not\subset \{x, y\}$

Operations on sets

Union
The union operation takes two sets as input, and outputs a set.
The output is the set of all objects that are members of either input set (or of both).

Notation
{x, y} ∪ {z} = {x, y, z}
{x, y} ∪ {x} = {x, y}

How do we talk about it:
{x, y} ∪ {z} = the union of the set containing x and y and the set containing z

Intersection
The intersection operation takes two sets as input, and outputs a set.
The output is the set of all objects which are members of both input sets.

Notation
{x, y, z} ∩ {z} = {z}
{x, y} ∩ {z} = {∅}

How do we talk about it:
{x, y} ∩ {z} = the intersection of the set containing x, y, and z and the set containing z

Set complementation
The set complementation operation takes an *ordered pair of sets* as input, and outputs a set.
The output is the set of all objects that are members of the first, but not of the second set.

Notation
{x, y} – {x} = {y}
{x, y} – {x, y} = {∅}

General complementation
The general complementation operation takes a single set as input, and outputs a set.
The output is the set of all objects in the domain that are not members of the input. The domain is the set of all objects presumed to exist.

Notation
Domain = {x, y, z}
Input set A = {x, y}
A' = {z}

<u>Power set</u>: ℘
The powerset operation takes a single set as input, and outputs a set.
The output is the set of subsets of the input.

Notation
℘({x, y, z}) = {{∅, {x}, {y}, {z}, {x, y}, {y, z}, {x, z}, {x, y, z}}

Sequences

A sequence is an *ordered* list of objects (in contrast to sets).
Sequences can meaningfully contain the same object *twice* (in contrast to sets).

Notation:
<x, y, z> = the ordered list of x, followed by y, followed by z
(x, y, z) = the ordered list of x, followed by y, followed by z

<x, y, z> ≠ <z, y, x>
<x, x, y> ≠ <x, y, y>

How do we talk about it:
<x, y, z> = the sequence of x, y, and z.
<x, y, z> = a 3-tuple.
<x, y> = an ordered pair

Cartesian product
The Cartesian product operation takes an ordered pair of sets as input, and outputs a set. The output is the set of all ordered pairs <x, y> such that x is a member of the first input set and y is a member of the second input set.

Notation
{x, y} x {z} = {(x, z), (y, z)}

Exercise
True or false?

1. a ∈ {a, b, c}
2. {a} ∈ {a, b, c}
3. {a} ⊂ {a, b, c}
4. {a} ∈ {{a}, b, c}
5. |{x: x is a student in this class}| < 30
6. {a, b} x {a, b} = {a,b}

Relations

In daily life, we talk about relations: Who is who's neighbor, who is who's mother, who is who's teacher. As a mathematical notion, a relation between objects in set A and in set B is a subset of A x B. As such, a relation between set A and set B is a subset of the Cartesian product of A and B: It takes an object of set A and set B as the input, and returns an ordered pair.

Let's look at a concrete example:

Set A = {x: x lives on the top floor of 33 Meshkisvili street} = {Irakli, Nino}
Set B = {x: x lives on the bottom floor of 33 Meshkisvili street} = {Lela, Zura}

A = a 1-place relation, or a *property*, between the members of the set and the property of *living on the top floor of 33 Meshkishvili street*.
B = a 1-place relation, or a *property*, between the members of the set and the property of *living on the top floor of 33 Meshkishvili street*.

L = a 2-place relation of *living above* between the members of set A and the members of set B = A x B ={<Irakli, Lela>, <Irakli, Zura>, <Nino, Lela>, <Nino, Zura>}

Notation:
Rab = the relation R that holds between elements a and b

How do we talk about it:
For a relation R, a set A and a set B, we say that

- R is *from* set A *to* set B
- A is its *domain* and B is its *range*
- A is its *domain* and B is its *codomain*
- A is its *set of origin* and B is its *set of destination*

<u>Properties of relations</u>
1. Reflexivity: In a set A, a relation is *reflexive* if each member has the R-relation to itself. Example → *equals*
2. Symmetry: In set A and set B, a relation is *symmetric* if for each pair aRb, bRa also holds. Example → *being a neighbor*
3. Transitivity: In a set A and set B, a relation is *transitive* if aRb and bRc, aRc also holds. Example → *being an ancestor*

Exercise
For set A and B above, define the relation N 'is a neighbor of'
N = …

<u>Functions</u>
Functions are a special kind of relation. A relation Rab is a function if and only if each element in the *domain* is paired with just one element in the *range*. So, a relation is not a function if:

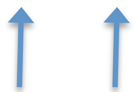
- not all elements from the domain are mapped (these are called *partial functions*)
- one element from the domain is mapped to two elements in the *range*

Notation:
Fab = the function F that holds between elements a and b

How do we talk about it:
For a function F, a set A and a set B, we say that

- F is *from* set A *to* set B
- A is its *domain* and B is its *range*
- A is its *domain* and B is its *codomain*
- A is its *set of origin* and B is its *set of destination*

Exercises
An example of a function is (a subset of) the Cartesian product.
{x, y} x {z} = {(x, z), (y, z)}



domain range
input output

Not all Cartesian products are functions – why is the following not a function?

{x,y} x {v,w} = {(x,v),(x,w),(y,v),(y,w)}

domain range
input output

Exercise
Consider the sets A = {a,b,c} and B = {1,2,3,4}. Which of the following relations are functions from A to B?
P = {(a,1), (b,2), (c,3)}
Q = {(1,a), (2,b), (3,c)}
R = {(a,1), (a,2), (b,2), (b,3)}

Characteristic functions
A special type of functions are characteristic functions, which are special in that their *range* are T and ⊥, or "green light" and "red light", meaning that for a domain S, any element maps to T if that element is in S, and maps to ⊥ otherwise.

Notation:
cf(S) is a characteristic function from an element a to {T, ⊥} such that
1. $\langle x, \top \rangle \in cf(S)$ just in case $x \in S$, and
2. $\langle x, \bot \rangle \in cf(S)$ just in case $x \notin S$

How do we talk about it
The characteristic function of S maps any x to T if x is an element of S, and maps x to ⊥ otherwise.

Examples
For a set S {a,b,c}, cf(S) = {(a,T), (b, T), (c, T), (d,⊥)….}
For a set S {x:x is a city in Georgia}, cf(S) = {(Tbilisi, T), (Amsterdam, ⊥), (Kutaisi, T)…}

Given all this, we can switch effortlessly between the following equivalent statements:
1. $x \in S$
2. $\langle x, \top \rangle \in cf(S)$
3. cf(S) maps x to ⊤
4. the value of cf(S) at x is ⊤

<u>Lambda notation</u>
Finally, let's learn a handy new piece of notation, called a λ (lambda) term.

Notation
$\lambda x \in D$. some description of the value at x
λx . some description of the value at x
λy . some description of the value at y

How do we talk about it
A function which maps any x in domain D to the value described by the given description.

Examples
$\lambda x \in$ natural numbers $\mathbb{N}$. x^2 = {(1,1), (2,4), (3,9), …}
$\lambda x \in$ cities . x is a city in Georgia = {Tbilisi, Kutaisi, …}
$\lambda x \in$ cities . T iff x is a city in Georgia = {(Tbilisi, T), (Amsterdam, ⊥), (Kutaisi, T)…}

<u>Function Application</u>
The application of a function f to some x in its domain is called *function application*.

Notation
f(x) f(x) = y

How do we talk about it
f(x) is the application of f to x. f(x) = y is the application of f to x giving y

Examples
$P = \lambda x \in \mathbb{N}. x^2$ P(2) = 4

$Q = \lambda x \in$ cities . T iff x is a city in Georgia Q(Kutaisi) = T
Q(Amsterdam) = ⊥

Take-home message:
Formal notions like sets, relations, and functions can be used to describe certain parts of language. Some of these you've already seen in action, and some of these we'll return to in later classes.