Languages and structures

Introduction

- Model theory is the part of logic that studies the relationships between formal languages - we'll be focusing on **first-order languages** - and mathematical structures.

- Philosophical relevance: get a better understanding of fundamental semantic concepts, such as **reference**, **truth** and **representation**.

Signature of a language

- A signature $\mathscr{L}$ is a (possibly empty) set of non-logical symbols that characterize a first-order language. This is the language's non-logical vocabulary. These symbols are of three kinds :

  - **Constant symbols**. These are important proper names for the language. For example, an arithmetic language has the symbol "0" in its signature, as this proper noun is essential for representing arithmetic structures.

  - **Relation symbols**. Continuing with arithmetic: "<" is an example of a relation symbol that can be found in the signature of an arithmetic language.

  - **Function symbols**. For example, the addition symbol "+", or the multiplication symbol "." in the case of arithmetic.

$\mathscr{L}$-structures

- An $\mathscr{L}$-structure is a model for a language with $\mathscr{L}$ as its signature.
- Definition: An $\mathscr{L}$-structure, or a model, $\mathscr{M}$, is define as a pair $< D, I >$ where:
  - D is a non-empty set called the domain of discourse
  - I is an interpretation function which sends:
    - Each constant symbol *c* to and object *I(c)* in *D*
    - Each n-ary relation symbol *R* to a n-ary relation in D^n
    - Each n-ary symbol function to a function in D^n

- Notation : D^n is the set of n-tuples $\{ < x_1, \ldots, x_n > \mid x_1 \in D, \ldots, x_n \in D\}$

- An n-ary relation is any subset of D^n :

  - $I(R) \subseteq D^n$

- There is something new relative to the first semester: the concept of a function.

Relations

- A binary relation on a set A is a subset of $A \times A = A^2$. If $R \subseteq A^2$ is a binary relation on A and $<x,y> \in A$, we sometimes write Rxy (or xRy) for $<x,y> \in R$.
    - Some useful properties of binary relations:
        - Reflexivity: R is reflexive iff $\forall x Rxx$
        - Transitivity: R is transitive iff $\forall x \forall y \forall z((Rxy \wedge Ryz) \rightarrow Rxz)$
        - Symmetry: R is symmetric iff $\forall x \forall y(Rxy \rightarrow Ryx)$
- More generally: an n-ary relation on a set A is a subset of A^n.

Functions

- A function $f : A \to B$ is a mapping of each member of A to a member of B.

- A is called the domain of the function; B is called its co-domain.

- A function with domain A and co-domain B can also be defined, in set-theory, as a special kind of binary relation R, satisfying the following conditions:

    - For every x in A, there is a y in B such that $< x, y > \in R$

    - For every x, y and z: if $< x, y > \in R$ and $< x, z > \in R$, then y=z.

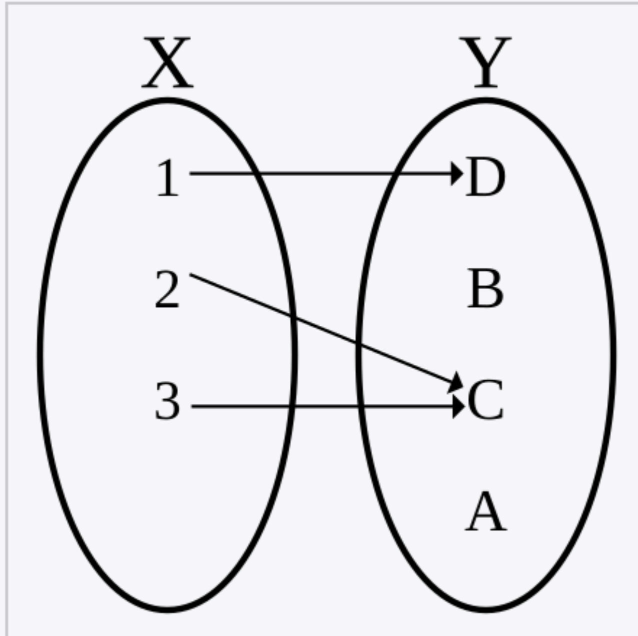


Diagram of a function

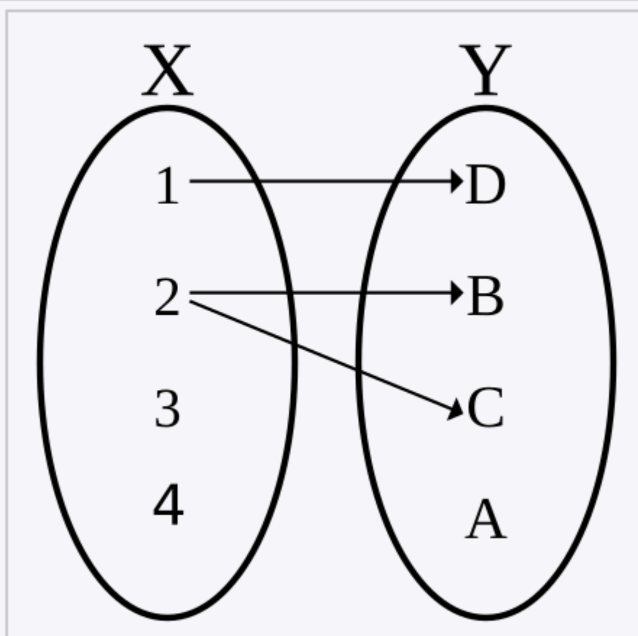
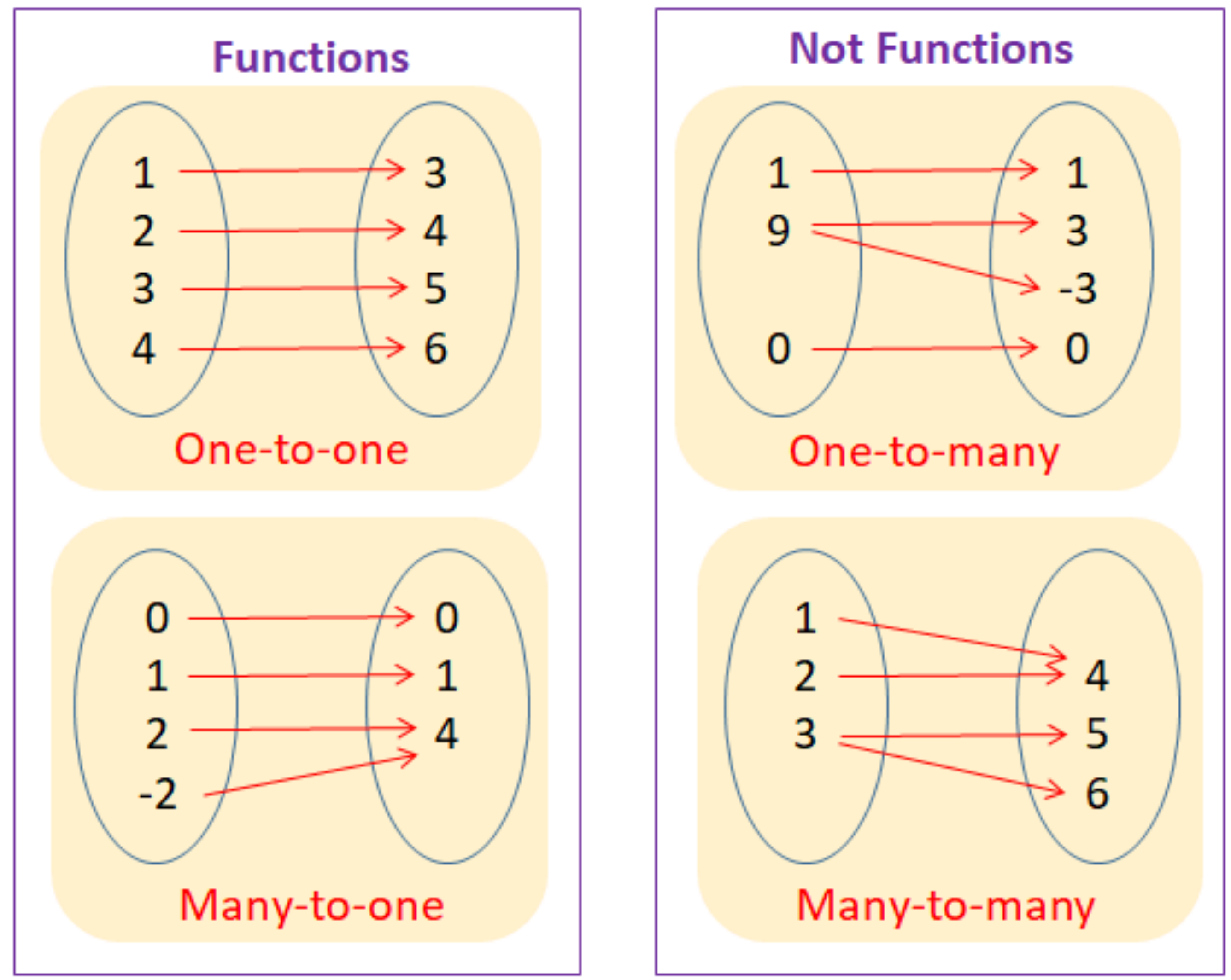


Diagram of a relation that is not a function. One reason is that 2 is the first element in more than one ordered pair. Another reason is that neither 3 nor 4 are the first element (input) of any ordered pair therein

Relations

A relation shows a relationship between two values.
A function is a relation where each input has only one output.



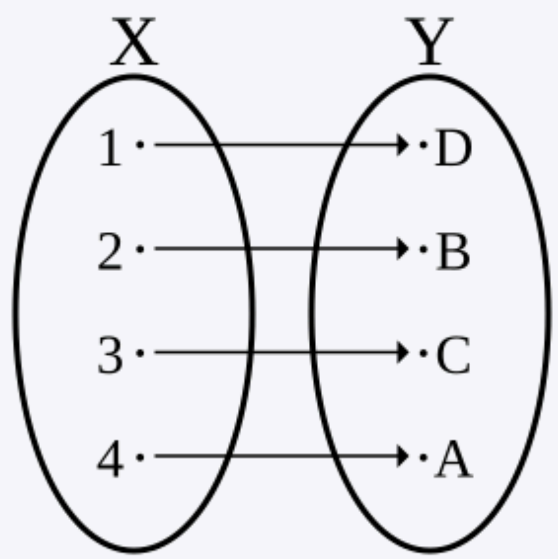
Kinds of functions

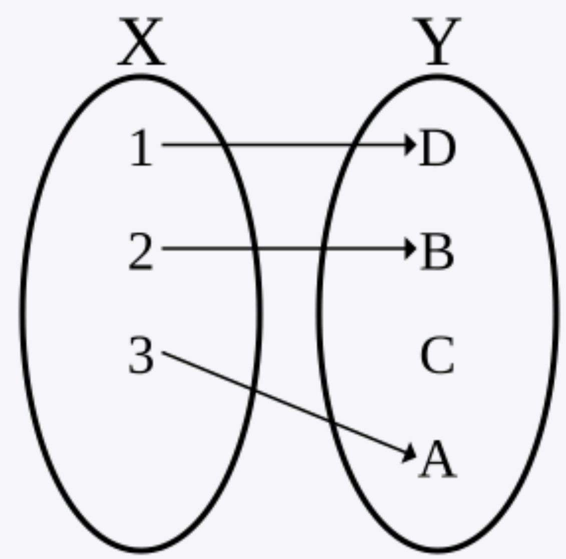
- **Surjective functions**: A function f : A —> B is surjective iff for every y ∈ B there is **at least** one x ∈ A such that f(x)=y.

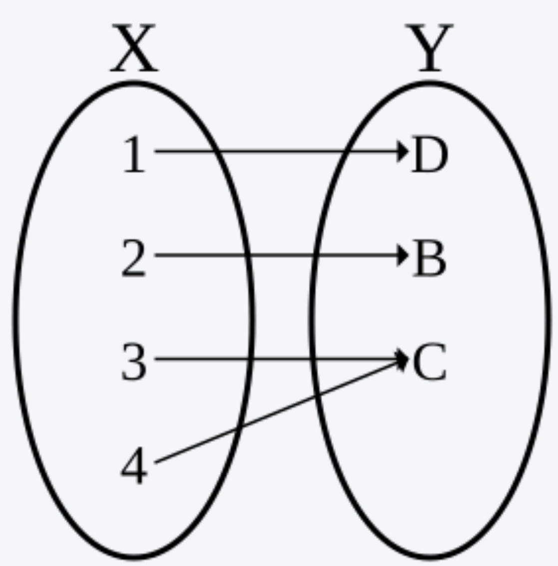
- **Injective functions**: A function f : A —> B is injective iff for each y ∈ B there is **at most** one x ∈ A such that f(x)=y.

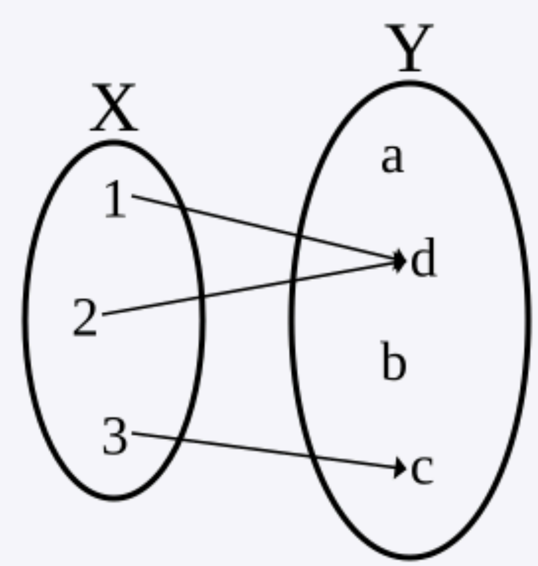
- **Bijective functions**: A function f : A —> B is bijective iff it is **both injective and surjective**;

h

<table>
<tr><th></th><th>surjective</th><th>non-surjective</th></tr>
<tr><td>injective</td><td>

bijective</td><td>

injective-only</td></tr>
<tr><td>non-injective</td><td>

surjective-only</td><td>

general</td></tr>
</table>

Wikipedia

Notational convention

- To describe a $\mathscr{L}$-structure one often just lists:
  - Its domain D
  - The objects to which the constants in the signature refer
  - The n-ary relations (defined on Dn) to which the relation symbols in the signature refer
  - The functions to which the function symbols in the signature refer

Illustrations

- $(\mathbb{N}, <)$: the $\mathscr{L}$-structure (or model) whose domain is the set of all natural numbers, and whose interpretation function $\mathscr{I}$ sends a relation-symbol in the language, let's say <, to a binary relation defined on $\mathbb{N} \times \mathbb{N}$, $\{ < x, y > \mid x \in \mathbb{N}, y \in \mathbb{N}, x < y\}$.

- $(\mathbb{N}, 0, ', +, .)$: the « standard model of arithmetics », whose domain is the set of all natural numbers, with the number 0 as the referent of the symbol « 0 », and three functions (the successor function $'$, the addition, and the multiplication) as the denotations of three function-symbols in the formal language.

- $(\mathbb{R}, 0, 1, +, ., ^{-1})$: the field of real numbers.

Object language vs meta-language

- To be fully rigorous, we would need distinct notations for :
  - A symbol in the object formal language
  - Its denotation in an $\mathscr{L}$-structure
- For instance : $\underline{0}$ for the formal symbol in the langage of arithmetics ; 0 for the symbol in the meta-language denoting the number 0.
  - Here : in $(\mathbb{N}, 0, ', +, .)$, $\mathscr{I}(\underline{0}) = 0$
- This is cumbersome: most of the time we'll just use the same symbols in the object-language and in the meta-language and we'll let the context do it's job!

Identity conditions for structures

- Two $\mathscr{L}$-structures $\mathscr{M}$ and $\mathscr{N}$ are identical iff:
  - They have the same domains
  - They interpret the same signatures
  - There interpretation functions are the same, which means that:
    - Constants symbols have the same referents
    - Relations symbols have the same denotations
    - Functions symbols have the same denotations

First-order language

- A first-order formal language $\mathscr{L}$ is a set of formal symbols, some of which are simple, and some others complex.
- The simple symbols of such a language comprise:
  - The symbols in the signature, which are called the non-logical symbols of the language: constants symbols, relations symbols and functions symbols.
  - Logical symbols:
    - Variables: x, y, z …
    - Quantifiers: $\forall$, $\exists$
    - Identity: = (NB: it can be debated whether or not this is a logical symbol …)
    - Propositional symbols: $\neg$, $\vee$, $\wedge$, $\rightarrow$, $\leftrightarrow$, $\perp$
    - Parenthesis: (,)

Complex symbols, 1: Terms

- The set of terms for $\mathscr{L}$ is recursively defined:
    - For any constant symbol *c*, *c* is a term
    - For any variable symbol *x*, *x* is a term
    - If $t_i \ldots t_n$ are *n* terms of $\mathscr{L}$, and if f^n is a symbol for an n-ary function in the signature, then $f^n(t_1 \ldots t_n)$ is a term
- Nothing else is a term of $\mathscr{L}$.

Complex symbols, 2 : atomic formulae

- If t_i and t_j are terms, $t_i = t_j$ is an atomic formula ;
- If R^n is a symbol for an n-ary relation in the signature, and $t_1 \ldots t_n$ are terms, $R^n t_1 \ldots t_n$ is an atomic formula.

Complex symbols 3 : Well-formed formulae

- Atomic formulae are wff
- If ϕ and ψ are wff, $\neg\phi$, $\phi \wedge \psi$, $\phi \vee \psi$, $\phi \to \psi$, $\phi \leftrightarrow \psi$, $\bot$ are well-formed formulae
- If ϕ is a wff, and if x is a variable, $\forall x\phi$ is a wff
    - If x is a free variable in ϕ, its free occurrences are said to be **bound** by the quantifier.
- If ϕ is a wff, and if x is a variable, $\exists x\psi$ is a wff.
    - If x is a free variable in ϕ, its free occurrences are said to be **bound** by the quantifier.
- Nothing else is a wff.

Free variables in a term

- Given any term, one can recursively define the **set** $var(t)$ **of the variables that occur free in it**:
  - If t is a constant c, $var(t) = \emptyset$
  - If t is a variable x, $var(t) = \{x\}$
  - If t is a complex term $f^n(t_1 \ldots t_n)$, $var(t) = var(t_1) \cup var(t_2) \ldots \cup var(t_n)$

Free variables in a formula

- If ϕ is an atomic formula $R^n t_1 \dots t_n$, the set of its free variables is the union of the sets of free variables in each term that occurs in it :
  - $Varf(\phi) = var(t_1) \cup \dots \cup var(t_n)$
- If ϕ = $\neg\psi$, $Varf(\phi) = Varf(\psi)$
- If $\phi = \psi \to \theta$ (or : $\psi \wedge \theta$, or $\psi \vee \theta$), $Varf(\phi) = Varf(\psi) \cup Varf(\theta)$
- If $\phi = \forall x \psi (or : \exists x \psi)$, $Varf(\phi) = Varf(\psi) - \{x\}$
- A formula without any free variable is called a **closed formula**, or a **statement**.
  - $\exists y Rxy$ is an open formula.
  - $\forall x \exists y Rxy$ is a closed formula.

- $\forall x R x z$

- $\forall x R x z$

- $\forall x(Pz \to Rxz)$

Truth in a model (or: satisfaction) for atomic sentences

- $M \vDash \phi$ is read: ϕ is true in M, or ϕ is satisfied by M.
- $M \vDash t_i = t_j$ iff $I(t_i) = I(t_j)$
- $M \vDash R^n(t_1 \ldots t_n)$ iff $< I(t_1) \ldots I(t_n) > \in I(R^n)$.

Illustrations

- Let us consider a formal language for arithmetics with the following signature: $\{\underline{0}, S, \underline{+}\}$, and let us interpret it in the standard model $\mathscr{M} : (\mathbb{N}, 0, ', +)$.

  - $\mathscr{M} \vDash \underline{0} = \underline{0}$ because $\mathscr{I}(\underline{0}) = 0, 0 = 0$; thus $\mathscr{I}(\underline{0}) = \mathscr{I}(\underline{0})$

  - $\mathscr{M} \vDash S\underline{0} \,\underline{+}\, S\underline{0} = SS\underline{0}$ because: $\mathscr{I}(S) ='$; $I(S\underline{0}) = 0' = 1$; $\mathscr{I}(\underline{+}) = +$; $\mathscr{I}(SS\underline{0}) = 0'' = 2$ and we know that$1 + 1 = 2$;

- Now let us add a relation symbol to the signature: Inf^2, and let us consider a new model $\mathscr{M} = (\mathbb{N}, 0, ', +, <)$. So in this model $\mathscr{I}(Inf^2) = <$.

- Let us show that : $\mathscr{M} \models Inf^2 \underline{0} S \underline{0}$

  - $\mathscr{I}(\underline{0}) = 0, \mathscr{I}(Inf^2) = <, \mathscr{I}(S) = ', 0' = 1$so $\mathscr{M} \models Inf^2 \underline{0} S \underline{0}$ iff $< 0,1 > \in <$, which is the case since 0<1.

What about quantified formulae?

- It's tricky to specify a truth value for quantified formulae, like $\forall x Fx$.
- We would like the formula to be true in a model iff the every element of the domain is in the extension of $\mathscr{I}(F)$.
- But how can we say this precisely? On Frege's substitutional approach, $\forall x Fx$ is true in a model iff Fa is true for every constant symbol a.
    - But this does not work: we can't presuppose that we have a constant symbol for every element in the domain, especially since some domains are infinite!
- We can't replace x in $\forall x Fx$ by every object in the domain either, since an element of a domain is not (always) a symbol of first-order logic!

Tarski's solution

- We don't fix truth-values for formulae absolutely, but rather relative to a context called a **variable assignment** (or: an assignment ; an assignment function), that let the variables temporarily refer to elements of the domain.

- **Definition**: σ is a **variable assignment** for a model $\mathscr{M} = < D, I >$ iff σ is a function that assigns an object in the domain D to each variable of the language.

Interpreting terms

- Variables can occur free in terms, like in $x \pm y$.

- So we can't interpret our terms just by using the interpretation function of a model: we need to supplement it with an assignment function: if I is an interpretation function and σ an assignment function, I_σ is an interpretation relative to the assignment that is defined thus:

  - If t is a variable, $I_\sigma(t) = \sigma(t)$

  - Si t is a constant symbol, $I_\sigma(t) = I(t)$

  - Si t is a complex term $f(t_1 \ldots t_n)$:

    - $I_\sigma(f(t_1 \ldots t_n)) = I(f)(I_\sigma(t_1) \ldots I_\sigma(t_n))$

Tarski's truth definition for atomic sentences

- $M \models_s t_i = t_j$ iff $I_\sigma(t_i) = I_\sigma(t_j)$.
- $M \models_\sigma R(t_1 \ldots t_n)$ ssi $< I_\sigma(t_1), \ldots, I_\sigma(t_n) > \in I(R)$
- Note that the truth of atomic formulae is relative to:
  - A model and its interpretation function …
  - … but also an assignment function σ.

Truth-definition for propositional formulae

- Nothing new:
- $M \vDash_\sigma \neg\phi$ iff $M \nvDash_\sigma \phi$
- $M \vDash_\sigma \phi \wedge \psi$ iff $M \vDash_\sigma \phi$ and $M \vDash_\sigma \psi$
- $M \vDash_\sigma \phi \vee \psi$ iff $M \vDash_\sigma \phi$ or $M \vDash_\sigma \psi$
- $M \vDash_\sigma \phi \rightarrow \psi$ iff $M \nvDash_\sigma \phi$ or $M \vDash_\sigma \psi$

Variant of an assignment

- We need a further bit of notation. Let x be a variable, o some object in the domain of a model, and σ an assignment.

- We note : $\sigma[x \rightarrow o]$ the function that is just like σ, except maybe that it assigns o to x (I say « maybe » because if σ does already assign o to x, σ will be the same function as $\sigma[x \rightarrow o]$).

- In words: **the σ variant** that assigns o to x

Truth-definitions for quantifiers

- $M \vDash_\sigma \forall x \phi x$ iff for all variants $\sigma[x \to o]$of σ, $M \vDash_{\sigma[x \to o]} \phi x$
  - This is like saying that any object o in the domain satisfies ϕx in M
- $M \vDash_\sigma \exists x \phi x$ iff for at least one variant $\sigma[x \to o]$ of σ, $M \vDash_{\sigma[x \to o]} \phi x$
  - This is like saying that at least one object o in the domain satisfies ϕx in M

Truth in a model, simpliciter

- $M \vDash \phi$ iff for any assignment σ, $M \vDash_\sigma \phi$