Ex Sps $L = \{c, f, R\}$ (binary)

(const. symb., ternary)

Then:

$$\exists z \left((f(c,x,y) = z) \wedge \forall x \exists w \, R(x,w) \right)$$

is a wff in this language

$$\exists z ((f(c,x,y) = z) \wedge \forall x \exists w \, R(x,w))$$

$$(f(c,x,y) = z \wedge \forall x \exists w \, R(x,w))$$

$$f(c,x,y) = z \qquad \forall x \exists w \, R(x,w)$$

$$\exists w \, R(x,w)$$

$$R(x,w)$$

- quantifiers in a wff A apply only to the subfmla they were first attached to:

Defn Sps A is a wff, Q is a quantifier ($\exists$ or $\forall$), x is a variable, and Qx appears in A

There is a unique subfmla B of A s.t. for some (non wff) strings S, T we have

$$A = S \, Qx \, B \, T$$

Call B the **scope** of Qx

- we'll take fact for granted; not hard to prove
- point is: in building A at some stage we formed B then added Qx to get QxB; thus B is the scope of Qx

Ex: Consider

$$\exists z((f(c,x,y)=z) \wedge \forall x \exists w R(x,w))$$

scope of $\exists w$

scope of $\forall x$

scope of $\exists z$

Def'n an occurrence of a variable x in a wff A is bound if it occurs in the scope of a quantifier Qx or it is free

Note: x can have both free and bound occurrences in A

Ex Consider again

$$\exists z((f(c,x,y)=z) \wedge \forall x \exists w R(x,w))$$

free occurrences

bound

bound

bound

Def'n A is a sentence if it has no free variables

(i.e. free occurrences of variables)

Ex (i) $\exists z((f(c,x,y)=z) \wedge \forall x \exists w R(x,w))$
is not a sentence (x, y have free occurrences)

(ii) $\forall x \exists y ((x<y) \vee (y<x))$
is a sentence.

- intuitively a sentence is either true or false
- a wff A w/ free variables only becomes True/False once these variables are specified (more in a moment)

Notation. We write $A(x_1, \ldots, x_n)$ to indicate free variables of A are among $x_1, \ldots, x_n$ (may not included)
↪ e.g. might write $A(x,y)$ to denote $\exists z((f(c,x,y)=z) \wedge \forall x \exists w R(x,y))$

Semantics of FOL

↪ In order to decide whether a wff is true or false, we need to interpret the symbols in A over a structure.

Def'n Sps $L = \{F_1, F_2, \ldots; R_1, R_2, \ldots\}$ is a language. An L-structure is any structure M in the language L.

Explicitly, $\mathcal{M}$ consists of:

(i) a nonempty set M (universe)

(ii) for each n-ary function symbol $f \in L$, an n-ary function $f^{\mathcal{M}}: M^n \to M$

(iii) for each m-ary rel'n symb. $R \in L$, an m-ary rel'n $R^{\mathcal{M}} \subseteq M^m$

Note: for 0-ary function symbols c, i.e. constant symbols, $c^{\mathcal{M}}$ is just an element of M.

- we call $f^{\mathcal{M}}, R^{\mathcal{M}}$ the interpretations of the symbols f, R.

Ex: (i) Sps $L = \{c, f\}$ (c: const., f: binary) and G is a group w/ identity e and group operation $\cdot$.

Define $c^{\mathcal{M}} = e$ and $f^{\mathcal{M}} = \cdot$.

Then $\mathcal{G} = \langle G, c^{\mathcal{M}}, f^{\mathcal{M}} \rangle = \langle G, e, \cdot \rangle$ is an L-structure.

(ii) We sometimes confuse symbols w/ their intended interpretations (warning! notation abuse)

- e.g. sps $L_{ar} = \{0, S, +, \cdot, <\}$ is lang. of arithmetic

(here $0, S, +, \cdot, <$ are just function/relation symbols)

- the standard structure of arithmetic is $\mathcal{N} = \langle \mathbb{N}; 0^{\mathcal{N}}, S^{\mathcal{N}}, +^{\mathcal{N}}, \cdot^{\mathcal{N}}, <^{\mathcal{N}} \rangle$
  where $0^{\mathcal{N}} = 0$ (the actual number, not symbol)
  $S^{\mathcal{N}}$ = actual successor function S
  $+^{\mathcal{N}} = +$
  $\cdot^{\mathcal{N}} = \cdot$
  $<^{\mathcal{N}} = <$
- as a result we sometimes write
  $\mathcal{N} = \langle \mathbb{N}; 0, S, +, \cdot, < \rangle$
  though we should write superscripts to indicate interpretations

(iii) We could interpret these symbols differently.
- E.g. $\mathcal{M} = \langle \mathbb{N}; 0^{\mathcal{M}}, S^{\mathcal{M}}, +^{\mathcal{M}}, \cdot^{\mathcal{M}}, <^{\mathcal{M}} \rangle$
  where $0^{\mathcal{M}} = 5$
  $S^{\mathcal{M}}(x) = x^2$
  $x +^{\mathcal{M}} y = xy$
  $x \cdot^{\mathcal{M}} y = x^y$
  $x <^{\mathcal{M}} y$ iff $x > y$
  is also and L_{ar}-structure!

- once we've interpreted function symbols in a structure $\mathcal{M}$, we can interpret every term.

Def'n Sps $\mathcal{M} = \langle M; f_1^{\mathcal{M}}, f_2^{\mathcal{M}}, \dots; R_1^{\mathcal{M}}, R_2^{\mathcal{M}}, \dots \rangle$ is a structure.

For each term $t = t(x_1, \dots x_n)$ (vars among $x_1, \dots, x_n$) in this language we define a function

$$t^{\mathcal{M}}: M^n \to M$$

recursively as follows.

(i) if $t = x_i$, $t^{\mathcal{M}}(a_1, \dots, a_n) = a_i$ (ith projection function)

(ii) if $t = c$, $t^{\mathcal{M}}(a_1, \dots, a_n) = c^{\mathcal{M}}$ (constant function)

(iii) if $t = f(t_1, \dots, t_m)$ where f is m-ary func. symb. and $t_1, \dots, t_m$ are terms, then

$$t^{\mathcal{M}}(a_1, \dots, a_n) = f^{\mathcal{M}}(t_1^{\mathcal{M}}(a_1, \dots, a_n), \dots, t_m^{\mathcal{M}}(a_1, \dots, a_n))$$

Ex Sps $Lar = \{0, S, +, \cdot, <\}$ and $\mathcal{N} = \langle N; 0, S, +, \cdot, < \rangle$ is stand. struct. of arithmetic.

Then $t := t(x,y) := (x \cdot x + y) + S(S(0))$ is a term, and

$t^{\mathcal{N}}: N^2 \to N$ is defined by

$$t^{\mathcal{N}}(x,y) = x^2 + y + 2$$

$$x \cdot^{\mathcal{N}} x \qquad \text{or} \quad S^{\mathcal{N}}(S^{\mathcal{N}}(0))$$

Note: if $t = t(x_1, \dots, x_n)$, then also $t = t(x_1, \dots, x_n, x_{n+1}, \dots, x_{n+k})$ for any k (vars in t still among $x_1, \dots, x_n, x_{n+1}, \dots, x_{n+k}$)

- so really we have infinitely many interpretations $t^{\mathcal{M}}: M^{n+k} \to M$
- but since $t^{\mathcal{M}}$ only depends on first n inputs (at most), this doesn't make a practical difference

---

Evaluation of wffs

- we can now evaluate the truth of a wff A in a structure $\mathcal{M}$ (once we've assigned values to A's variables...)

Def'n Sps L is a lang, A is a wff in this lang., $\mathcal{M}$ is an L-structure and for every variable x_i we assign some $a_i \in M$ (write: $x_i \mapsto a_i$)

Then: we say A is **true** under this assignment, and write

$$\mathcal{M}, a_1, a_2, \ldots \models A$$

if we have the following:

(i) Sps A is atomic

(a.) if $A = R(t_1, \ldots, t_n)$ for an n-ary (vars among $x_1, \ldots, x_m$) rel'n symbol R and terms $t_1, \ldots, t_n$ then $\mathcal{M}, a_1, a_2, \ldots \models A$ if

$$R^{\mathcal{M}}(t_1^{\mathcal{M}}(a_1, \ldots, a_m), \ldots, t_n^{\mathcal{M}}(a_1, \ldots, a_m))$$

(b.) if A is $t_1 = t_2$

then

$M, a_1, a_2, \ldots \models A$ if $t_1^M(a_1, \ldots, a_m) = t_2^M(a_1, \ldots, a_m)$

(ii) Now recursively define:

a) $M, a_1, a_2, \ldots \models \neg A$ if $M, a_1, \ldots \not\models A$

b) $M, a_1, a_2, \ldots \models (A \wedge B)$
if $M, a_1, \ldots \models A$ and $M, a_1, \ldots \models B$

c) $M, a_1, \ldots \models (A \vee B)$
if $M, a_1, \ldots \models A$ or $M, a_1, \ldots \models B$

d) $M, a_1, \ldots \models (A \Rightarrow B)$
if $M, a_1, \ldots \not\models A$ or $M, a_1, \ldots \models B$

e) $M, a_1, \ldots \models (A \Leftrightarrow B)$
if ($M, a_1, \ldots \models A$ and $M, a_1, \ldots \models B$)
or ($M, a_1, \ldots \not\models A$ and $M, a_1, \ldots \not\models B$)

(iii) Also recursively define:

a) $M, a_1, a_2, \ldots \models \exists x_i A$ if there exists $b_i \in M$ s.t. if we modify our assignment by letting $x_i \mapsto b_i$
then $M, a_1, \ldots, a_{i-1}, b_i, a_{i+1}, \ldots \models A$

b) $M, a_1, a_2, \ldots \models \forall x_i A$ if for every $b_i \in M$ we have
$M, a_1, \ldots, a_{i-1}, b_i, a_{i+1}, \ldots \models A$

Notice - if $A = A(x_1, \ldots, x_n)$ (i.e. A's free vars among $x_1, \ldots, x_n$) then definition of $M, a_1, \ldots \models A$ only depends on $a_1, \ldots, a_n$

- We'll also write
  $\mathcal{M} \models A[a_1, \ldots, a_n]$ to abbreviate
  $\mathcal{M}, c_1, c_2, \ldots \models A$

Defn observe: if A is a sentence (no free vars) then either $\mathcal{M} \models A$ or $\mathcal{M} \not\models A$ regardless of assignment.
If $\mathcal{M} \models A$, say "A is true in $\mathcal{M}$" or "$\mathcal{M}$ satisfies A" or "$\mathcal{M}$ models A"

Ex Sps $L = \{R\}$ ← binary rel'n symb
- Consider $\mathcal{N} = \langle \mathbb{N}, < \rangle$ (i.e. $R^{\mathcal{N}} = <$ is usual ordering on $\mathbb{N}$)
- Let $A(x,y)$ denote the atomic wff $R(x,y)$
  $B(x,y)$ denote $(x = y)$
- will also write A as $(x < y)$
  (notation abuse! confusing syntax R w/ semantics $R^{\mathcal{N}} = <$)
- Then:
  $\mathcal{N} \models A[1,2]$ i.e. A is true in $\mathcal{N}$ under $x \mapsto 1$ $y \mapsto 2$
  could cheat and write $\mathcal{N} \models 1 < 2$
  (notation abuse! ↑ not a wff implicitly defining an assignment)

- Also $\mathcal{N} \models (A \vee B)[1,2]$
  i.e. $\mathcal{N} \models ((x < y) \vee (y < x))[1,2]$

why? because $(1<2) \vee (2<1)$ is true in $\mathcal{N}$, i.e. $\mathcal{N} \models A[1,2]$ or $\mathcal{N} \models B[1,2]$
↑ this one!

- But: $\mathcal{N} \not\models (A \wedge B)[1,2]$ since $\mathcal{N} \not\models B[1,2]$

(ii) Sps $L = \{c, f, R\}$ (c: const, f: binary, R: binary)

- then:

$A := \forall x ((c = x) \vee R(c,x))$

$B := \forall x \exists y (f(y,y) = x)$

are sentences in this language

- consider $\mathcal{N} = \langle \mathbb{N}, 0, \cdot, <\rangle$ an L-struct. ($0^{\mathcal{N}}$, $\cdot^{\mathcal{N}}$, $<^{\mathcal{N}}$)
- then $\mathcal{N} \models A$

why? for every assignment $x \mapsto n$ have $\mathcal{N} \models (0 = n) \vee (0 < n)$

- But $\mathcal{N} \not\models B$

why? not true that for every ass'n $x \mapsto n$ have $\mathcal{N} \models \exists y (y \cdot y = n)$

∴ for $x \mapsto 2$ this fails! (2 has no sqrt in $\mathbb{N}$)

- Now: consider another L-structure

$\mathcal{R} = \langle \mathbb{R}^+, 1, \cdot, <\rangle$

$\{r \in \mathbb{R} : r \geq 0\}$

- then $\mathcal{R} \not\models A$

Why? if $x \mapsto 1/2$ then $\mathcal{R} \not\models (1 = x) \vee (1 < x)$

- But $\mathcal{R} \models B$

Why? for every $x \mapsto r$ can find $y \mapsto s$

(namely $s = \sqrt{r}$) s.t. $\mathcal{R} \models (y \cdot y = x)[s, r]$
i.e. $\mathcal{R} \models s \cdot s = r$

- sentences in FOL behave like prop'l variables in PL (either true or false)

Def'n if Σ is a set of L-sentences and $\mathcal{M}$ is an L-structure we write
$\mathcal{M} \models \Sigma$ ("$\mathcal{M}$ models Σ")
if $\mathcal{M} \models A$ for every $A \in \Sigma$

Def'n if Σ is a set of sentences and A is a sentence we write
$\Sigma \models A$ ("Σ logically implies A")
if every $\mathcal{M}$ s.t. $\mathcal{M} \models \Sigma$ also $\mathcal{M} \models A$

Def'n if $\Sigma = \emptyset$ we write
$\models A$ ("A is logically valid")
to mean $\emptyset \models A$, i.e. A is true in every L-structure $\mathcal{M}$

Note: I posted a pdf of useful valid finds on Canvas (Appendix B)

Def'n Sentences A, B are logically equivalent if $\{A\} \models B$ and $\{B\} \models A$;
equivalently if $\models A \Leftrightarrow B$
Write: $A \equiv B$.

- just like in PL, we'll be interested in knowing when a set of sentences has a model.

Defn A set of sentences S is **satisfiable** if there is a structure $\mathcal{M}$ s.t. $\mathcal{M} \models S$

Note: usually interested in sentences but can extend defns to general wffs
- if S a set of wffs, A a wff
  $S \models A$ means whenever $\mathcal{M}, a_1, a_2, \ldots \models S$ then $\mathcal{M}, a_1, a_2 \ldots \models A$
- a wff $A = A(x_1, \ldots x_n)$ is **valid** if for every $\mathcal{M}$ and every assignment we have $\mathcal{M}, a_1, a_2, \ldots \models A$
- $A \equiv B$ defined analogously for wffs.

Ex (i) An instance of a tautology is a wff obtained by subbing the variables in some PL tautology w/ wffs
- such wffs are always valid
  e.g. $p \vee \neg p$ is a PL tautology
  hence for any wff A in FOL,
  $A \vee \neg A$ is an inst. of taut.
  and clearly $\models A \vee \neg A$

likewise $\neg(A \vee B) \Leftrightarrow (\neg A \wedge \neg B)$ is valid for any wffs A, B
(instance of $\neg(p \vee q) \Leftrightarrow (\neg p \wedge \neg q)$)

(ii) wffs of the form $\forall x \exists y A(x,y)$ and $\exists y \forall x A(x,y)$ are not logically equivalent in general

e.g. sps $\mathcal{M} = \langle \mathbb{R}, < \rangle$
and $A(x,y) := (y < x)$

Then $\mathcal{M} \models \forall x \exists y (y < x)$ (for every real there is a smaller real)

but $\mathcal{M} \not\models \exists y \forall x (y < x)$ (... since there is no smallest real y)

hence $\mathcal{M} \not\models \forall x \exists y (y < x) \Leftrightarrow \exists y \forall x (y < x)$
so fmlas are not equiv.

Substitutions

- sometimes useful to substitute variables, terms, etc in wffs with other ones
- have to be more careful in FOL -- because of quantifiers.

Def'n For A a wff, x a variable, t a term we write $A[x/t]$ for wff obtained by replacing every free occurrence of x in A with t.