CS 228 tut 5

13.4

FO more powerful than propositional

FO signature

$p_0/0, p_1/0, \ldots.$

countably infinite 0-ary predicates.

Propositional logic model m_P

$Vars = \{p_0, p_1, \ldots\}$

$p_i \in \{0, 1\}$

FO logic model m_F

Assigns interpretation to 0-ary predicates

$p_1 \wedge p_2$

FO model

$p_1 \to T$ $\quad$ $p_2 \to T$ $\quad$ Rest: Don't care

Key: An FO model MUST assign interpretation to ALL predicates & functions in vocab.

13.5

13.6

An exercise in unrolling the definition of $\models$.

$$\neg\forall x\, P(x) \iff \exists x.\, \neg P(x)$$

Goal:

$m, \nu \models \neg\forall x\, P(x)$

iff

$m, \nu \models \exists x.\, \neg P(x)$

$m, \nu \models \underline{\neg}\forall x\, P(x)$

$\iff m, \nu \not\models \forall x\, P(x)$ defn $\models \neg F$

$\iff$ It is not the case that every $u_0 \in D$
$m, \nu[x \mapsto u_0] \models P(x)$ defn $\models \forall x F$

$\iff$ For some u_0
$m, \nu[x \mapsto u_0] \not\models P(x)$ trivial

$\iff m, \nu[x \mapsto u_0] \models \neg P(x)$ $\models \neg F$

$\iff m, \nu \models \underline{\exists} x.\, \neg P(x)$ u_0 is the "witness"

$\exists x.\, P \vee Q$

$m, \nu \models \exists x\, P \vee Q$

for some u_0. Fix it!

Definition GUARANTEES THAT WE can fix appropriate u_0

$\iff m, \nu[x \to u_0] \models P \vee Q$

$\Leftrightarrow \; \underbrace{m, \nu[x \to u_0]}_{\nu'} \models P$ or $m, \nu' \models Q$

$m, \nu \models \exists x\, P \qquad\qquad m, \nu \models \exists x\, Q$

$\Leftrightarrow \; m, \nu \models \exists x\, P \lor \exists x\, Q$ defn of $\models \;\; \lor$

$\forall x\; P \land Q$

any arbit u_0 $\quad m, \nu' \models P \land Q$

$m, \nu' \models P$ and $m, \nu' \models Q$

$m \models \forall x\, P$ and $m \models \forall x\, Q$

here, again note that this works for any choice of u_0

14.8 Finite vs Infinite.

a. Key: Some total order, where there is always someone bigger

If domain is finite, this is a problem.

$\forall x\, \exists y\;\; x < y$

Define some axioms about binary predicate $<$

$\forall x, y, z$
$(x < y) \land (y < z)$
$\Rightarrow x < z$
$\land$
$\forall x\;\; \neg(x < x)$
$\land$

Transitive $x < y, y < z \Rightarrow x < z$
Irreflexive $\neg(x < x)$
Antisymmetric
$x < y \Rightarrow \neg(y < x)$

$\forall x, y \;\; x<y \Rightarrow \neg(y<x)$

$\wedge$

$\forall x \; \exists y \;\; x<y$

Can't be satisfied by finite set. (N

$x_0 < x_1 < x_2 \dots$

Can't have a chain longer than N.

Pigeonhole

$x_i < \dots < x_i$

Transitivity

$x_i < x_i$

F in the chat for irreflexivity

An infinite model that satisfies

$\mathbb{N}, <$ the usual

An infinite model that does not satisfy.

$\mathbb{N} \cup \{\omega\}$ $\quad \forall n \in \mathbb{N} \;\; n < \omega$

$\omega <$ nothing.

14.9 Cardinality constraints.

There is no way of picking three distinct objects from a set of size two

can pick 3 $\iff$ you have at least 3

$\forall x, y, z \; (x=y \vee y=z \vee x=z)$

Can we do it without $=$?

To prove that one can't (for any arbit signature)

- Take a domain with < 3 elements

- Take a domain with ≥ 3 elements.
Somehow show that no sentence can distinguish –

By domain I mean Universe, complete with interpretations for all functions and predicates.

Signature. $P_0/k_0, P_1/k_0, \ldots$ countably many
$f_0/l_0, f_1/l_0 \ldots$

$|\mathcal{A}|$ $\{a, b\}$ $P_i(a, b, \ldots) = T$, $f_j(\quad) = \_\_$ } assign any interpretation

$|\mathcal{B}|$ $\{a, b, c\}$ over here, will assign interpretation carefully.

$P(a, b)$, $f(a, b)$, ... } if only a, b are arguments, then same as above

If c is an argument to the predicate/func. in question, then.
Define the result to be what we obtain by replacing c with a

I'm "cheating"
Formally c and a are different
But there's no = pred to tell me that!

It is OUR choice to decide what EACH

of the predicates mean.

But with our predicates, we can make c and a indistinguishable.

So why do $\mathcal{A}$ and $\mathcal{B}$ agree.

Induction

on what?

Sentences only use $\exists$.

$$\forall x\, F \Leftrightarrow \neg\exists x. \neg F$$

Induction on number/nesting of $\exists$.

If no variables at all:

using constants / 0-ary predicates

by construction, same interpretation

$$\exists x. \underbrace{F(x)}_{\text{no quantifiers here}}$$

$\mathcal{A} \models \exists x. F(x)$

$x \mapsto a.$

$\mathcal{A} \models F(a)$

By construction

$\mathcal{B} \models F(a)$

Conversely

$\mathcal{B} \models \exists \bar{x}. G(x)$

$x_1 \mapsto a$

$x_2 \mapsto c$

$\vdots$

Claim by construction

Whatever the witness maps to c, you can also

WLOG every witness uses just a, b

map to a, and continue to be a witness.

k a witness to $A \models$

Induction step:

The key is to ensure that variable x occurs only within its scope.

i.e. simplify as in Exercise 14.12

then can apply inductive hypothesis

i.e. if it works for nesting level n, then works for n+1

$$\underbrace{\exists x\, F(x)}_{\text{indistinguish}} \wedge \underbrace{\neg \exists y\, F(y)}_{\text{indistind}}$$

$\wedge$

ind

Equivalences.

Kinda sorta an exercise in unrolling definitions.

$m, v \models \exists x\, G$

$\Updownarrow$ ← this part is not being used

$m, v[x \mapsto v_0] \models G$

x does not occur in G | as far as G is concerned, you don't care what happening to x

$m, v \models G$

for that matter

$m, v[x \mapsto v'] \models G$

$m, \mathcal{V}[x \mapsto u] \models G$

arbit

$m, \mathcal{V} \models \forall x\, G$

$\exists x.\, F(x) \wedge G$

$\Leftrightarrow m, \mathcal{V} \models \exists x\, F(x)$ and $m, \mathcal{V} \models G$

$\Updownarrow$ fix u_0 appropriately

$m, \mathcal{V}[x \mapsto u_0] \models F(x)$

Key: we don't care about x here

$m, \mathcal{V}[x \mapsto u_0] \models G$

$m, \mathcal{V}' \models F(x) \wedge G$

$m, \mathcal{V} \models \exists x (F(x) \wedge G)$

- Break formula down
  Look at $\models$ defn for topmost connective
- Apply key obs.
- Build reqd formula again from defn

15.12

1) tricky, have to basically have a formal proof of 14.12

You will see it as running eg. in Lecture 16.

FORESHADOWING: CNF for FO-logic

In practice, we are only going to encounter automation friendly techiques we will develop next week

2. $\phi \vdash t = t$ eq. reflexive

we not do y/t

$t = z$ t/z y/z this "imagined" formula is upto you at least for $\exists$

no t here

$\phi \vdash \exists y.\ t = y$ $\exists$. intro

$\exists y\ z = y$ no t here t/z makes sense

$\phi \vdash \forall x \exists y\ x = y$

4. $\phi \vdash f(g(c), c) = f(g(c), c)$ $t = t$

x_3

$f(g(c), c) = f(z, c)$ $g(c)/z$ x_3/z

x_2

$\phi \vdash \exists x_3\ f(g(c), c) = f(x_3, c)$

$\exists x_3\ f(g(c), z) = f(x_3, c)$

x_1

$\vdash \exists x_2 \exists x_3\ f(g(c), x_2) = f(x_3, c)$

$\exists x_2 x_3\ f(g(x_1), z) = f(x_3, x_1)$

$\vdash \exists x_1 x_2 x_3\ f(g(x_1), x_2) = f(x_3, x_1)$

2. Repeated application of eq. sub.

$\{a = b \wedge f(b) = g(b)\} \vdash a = b$ Assumption

$\vdash f(b) = g(b)$.

$f(z) = g(b)$

b/z

$\vdash f(a) = g(b)$ eq. sub .

$\vdash h(g(b)) = h(g(b))$ eq. reflex

$h(z) = h(g(b))$

$\vdash h(f(a)) = h(g(b))$

$\Rightarrow$ intro

$$\phi \vdash (a = b \wedge f(a) = g(b)) \Rightarrow h(f(a)) = h(g(b))$$

univ qt y / all b

univ qt x / all a

$$\phi \vdash \forall x, y((x = y \wedge f(y) = g(y)) \Rightarrow h(f(x)) = h(g(y)))$$

15.15 Modelling equivalences.

For formal proofs, might be good idea to wait for the simplications next week

In practice, you want to work with normalised stuff, not relying too much on ingenuity

Best to prove formally in normalised frameworks

1. Singleton domain ← Key

   2, 3 contradict whether $R(x, x)$ holds

3. The second formula

$\forall x, y\ E(x,y)$ typo. In slides

1. reflexive E
3. symmetric E
5. transitive E

2. the whole domain is in the same equivalence class

4. R is irreflexive.

6. R non-empty

$E(x,y)$

$E(\underline{x},x)$ $E(\underline{y},x)$ $R(\underline{x,y})$ clever pattern matching

$\Downarrow$

$R(x,x)$

if $R(x,y)$ then $\neg E(x,y)$ and $E(x,y)$ ← key to derive contradiction

$E(x,y)$ happens all the time (2)

$\Rightarrow R(x,y)$ should never happen

this isn't there in 15.15.

6. $\exists x,y\ R(x,y)$ says "Hello there"

15.15.2

$\{a,b\}$

$E(a,a)$
$E(a,b)$
$E(b,a)$
$E(b,b)$

$R(a,b)$