What is a logic?

- A formal language
  - Syntax – what expressions are legal
  - Semantics – what legal expressions mean
  - Proof system – a way of manipulating syntactic expressions to get other syntactic expressions (which will tell us something new)

Propositional Logic Syntax

Syntax: what you're allowed to write

- for (thing t = fizz; t == fuzz; t++){ … }
- *Colorless green ideas sleep furiously*.

Sentences (wffs: well formed formulas)

- true and false are sentences
- Propositional variables are sentences: P,Q,R,Z
- If ϕ and ψ are sentences, then so are
  (ϕ), $\neg\phi$, $\phi \vee \psi$, $\phi \wedge \psi$, $\phi \rightarrow \psi$, $\phi \leftrightarrow \psi$
- Nothing else is a sentence

Semantics

- Meaning of a sentence is truth value {**t, f**}
- Interpretation is an assignment of truth values to the propositional variables
- $\models_i \phi$ [Sentence ϕ is **t** in interpretation i]
- $\not\models_i \phi$ [Sentence ϕ is **f** in interpretation i]

Semantic Rules

- $\models_i$ <u>true</u> for all i
- $\not\models_i$ <u>false</u> for all i [the sentence <u>false</u> has truth value **f** in all interpret.]
- $\models_i \neg\phi$ if and only if $\not\models_i \phi$
- $\models_i \phi$ Æ ψ if and only if $\models_i \phi$ and $\models_i \psi$ [conjunction]
- $\models_i \phi$ Ç ψ if and only if $\models_i \phi$ or $\models_i \psi$ [disjunction]
- $\models_i P$ iff $i(P) = \mathbf{t}$

Terminology

- A sentence is valid iff its truth value is **t** in all interpretations

  Valid sentences: true, $\neg$ false, $P \vee \neg P$
- A sentence is satisfiable iff its truth value is **t** in at least one interpretation

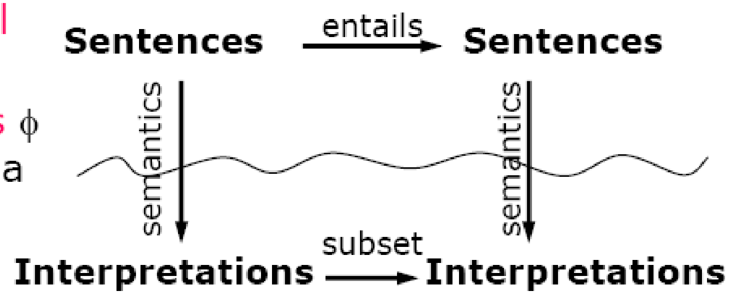
  Satisfiable sentences: P, true, $\neg P$
- A sentence is unsatisfiable iff its truth value is **f** in all interpretations

  Unsatisfiable sentences: $P \wedge \neg P$, false, $\neg$ true

All are finitely decidable.

Models and Entailment

- An interpretation i is a model of a sentence ϕ iff $\models_i \phi$
- A set of sentences KB entails ϕ iff every model of KB is also a model of ϕ



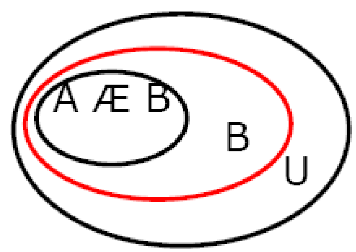
$$KB \models \phi \text{ iff } \models KB \rightarrow \phi$$

KB entails ϕ if and only if ($KB \rightarrow \phi$) is valid

KB = A Æ B

ϕ = B

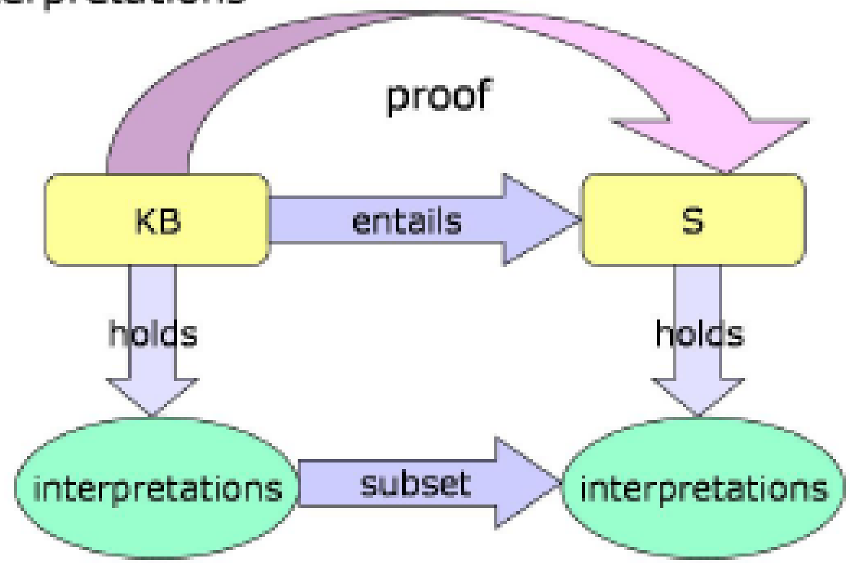
A Æ B $\models$ B



$$KB \models \Phi \Leftrightarrow KB \wedge (\neg \Phi) \text{ is unsatisfiable}$$

Entailment and Proof

A proof is a way to test whether a KB entails a sentence, without enumerating all possible interpretations



Proof

- Proof is a sequence of sentences
- First ones are premises (KB)
- Then, you can write down on the next line the result of applying an inference rule to previous lines
- When S is on a line, you have proved S from KB

- If inference rules are *sound*, then any S you can prove from KB is entailed by KB

- If inference rules are *complete*, then any S that is entailed by KB can be proved from KB

KB $\vdash \phi$ (means "ϕ can be proved from KB")

- Soundness: if KB $\vdash \phi$ then KB $\models \phi$
- Completeness: if KB $\models \phi$ then KB $\vdash \phi$

Natural Deduction

Proof is a sequence of sentences

First ones are premises (KB)

Then, you can write down on line j the result of applying an inference rule to previous lines

When ϕ is on a line, you know KB $\vdash \phi$

If inference rules are sound, then KB $\models \phi$

$$\frac{\alpha \to \beta \qquad \alpha}{\beta}$$

Modus ponens

$$\frac{\alpha \to \beta \qquad \neg \beta}{\neg \alpha}$$

Modus tolens

$$\frac{\alpha \qquad \beta}{\alpha \text{ Æ } \beta}$$

And-introduction

$$\frac{\alpha \text{ Æ } \beta}{\alpha}$$

And-elimination

Natural Deduction

Proof is a sequence of sentences

First ones are premises (KB)

Then, you can write down on line j the result of applying an inference rule to previous lines

When ϕ is on a line, you know KB $\vdash \phi$

If inference rules are sound, then KB $\models \phi$

Some inference rules:

$$\frac{\alpha \rightarrow \beta \quad \alpha}{\beta}$$

Modus ponens

$$\frac{\alpha \rightarrow \beta \quad \neg \beta}{\neg \alpha}$$

Modus tolens

$$\frac{\alpha \quad \beta}{\alpha \wedge \beta}$$

And-introduction

$$\frac{\alpha \wedge \beta}{\alpha}$$

And-elimination

Natural deduction example

Prove S

Step	Formula	Derivation
1	$P \wedge Q$	Given
2	$P \rightarrow R$	Given
3	$(Q \wedge R) \rightarrow S$	Given
4	P	1 And-Elim
5	R	4,2 Modus Ponens
6	Q	1 And-Elim
7	$Q \wedge R$	5,6 And-Intro
8	S	7,3 Modus Ponens

Example

When it is raining, the ground is wet. When the ground is wet, it is slippery. It is raining. Prove that it is slippery.

1.	$raining \Rightarrow wet$	Premise
2.	$wet \Rightarrow slippery$	Premise
3.	$raining$	Premise
4.	wet	MP : 1, 3
5.	$slippery$	MP : 2, 4

Horn Clauses

A *Horn clause* is a clause containing at most one positive literal.

Example: $\{r, \neg p, \neg q\}$
Example: $\{\neg p, \neg q, \neg r\}$
Example: $\langle p \rangle$
Non-Example: $\{q, r, \neg p\}$

NB: Every Horn clause can be written as a "rule".

$$\{\neg p, \neg q, r\} \quad \rightarrow \quad p \wedge q \Rightarrow r$$

First-Order Logic

- Propositional logic only deals with "facts", statements that may or may not be true of the world, e.g. "It is raining". But, one cannot have variables that stand for books or tables.
- In first-order logic variables refer to things in the world and, furthermore, you can quantify over them – to talk about all of them or some of them without having to name them explicitly.

FOL motivation

- Statements that cannot be made in propositional logic but can be made in FOL.
  - When you paint a block with green paint, it becomes green.
    - In propositional logic, one would need a statement about every single block, one cannot make the general statement about all blocks.
  - When you sterilize a jar, all the bacteria are dead.
    - In FOL, we can talk about all the bacteria without naming them explicitly.
  - A person is allowed access to this Web site if they have been formally authorized or they are known to someone who has access.

FOL syntax

- Term
  - Constant symbols: Fred, Japan, Bacterium39
  - Variables: x, y, a
  - Function symbol applied to one or more terms: f(x), f(f(x)), mother-of(John)
- Sentence
  - A predicate symbol applied to zero or more terms: On(a,b), Sister(Jane, Joan), Sister(mother-of(John), Jane)
  - $t_1=t_2$
  - If v is a variable and Φ is a sentence, then $\forall v.\Phi$ and $\exists v.\Phi$ are sentences.
  - Closure under sentential operators: $\wedge$ $\vee$ $\neg$ $\rightarrow$ $\leftrightarrow$ ( )

FOL Interpretations

- Interpretation I
  - U set of objects; domain of discourse; universe
  - Maps constant symbols to elements of U
  - Maps predicate symbols to relations on U (binary relation is a set of pairs)
  - Maps function symbols to functions on U

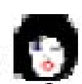
Basic FOL Semantics

Denotation of terms (naming)

- I(Fred) if Fred is constant, then given
- I(x) undefined
- I(F(term)) I(F)(I(term))

$\models_I P(t_1, \ldots, t_n)$ iff $<I(t_1), \ldots, I(t_n)> \in I(P)$

brother(John, Joe)??

- I(John) =  [an element of U]
- I(Joe) =  [an element of U]
- I(brother) = {<, >,< , >, <...,...>, ... }
- $\models_I$ brother(John, Joe)

Semantics of Quantifiers

Extend an interpretation I to bind variable x to element $a \in U$: $I_{x/a}$

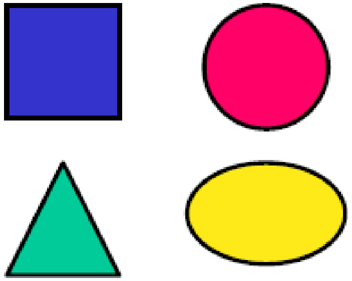
- $\models_I \forall x.\Phi$ iff $\models_{Ix/a} \Phi$ for all $a \in U$
- $\models_I \exists x.\Phi$ iff $\models_{Ix/a} \Phi$ for some $a \in U$

- Quantifier applies to formula to right until an enclosing right parenthesis:

$$(\forall x.p(x) \vee q(x)) \wedge \exists x.r(x) \rightarrow q(x)$$

FOL Example Domain

- U = { , , , }
- Constants: Fred
- Preds: above^2, circle^1, oval^1, square^1
- Function: Hat
- I(Fred) =
- I(above) = {< , >,< , >}
- I(circle) = {< >}
- I(oval) = {< >,< >}
- I(Hat) = {< , >,< , >}
- I(square) = {< >}



The Real World

FOL Example

- I(Fred) = ▲
- I(above) = {<■, ▲>, <●, ⬬>}
- I(circle) = {<●>}
- I(oval) = {<●>, <⬬>}
- I(Hat) = {<▲, ■>, <⬬, ●>}
- I(square) = {<▲>}

- $\models_I$ square(Fred)?

- $\not\models_I$ above(Fred, Hat(Fred))?
  - I(Hat(Fred)) = ■
  - $\not\models_I$ above(▲, ■) ?

- $\models_I \exists$x. oval(x)?
  - $\models_{I x/●}$ oval(x)

FOL Example: Continued

- I(Fred) = △
- I(above) = {<■, △>, <●, ⬬>}
- I(circle) = {<●>}
- I(oval) = {<●>, <⬬>}
- I(Hat) = {<△, ■>, <⬬, ●>}
- I(square) = {<△ >}

- $\models_I \forall x.\ \exists y.\ above(x,y) \vee above(y, x)$
  - $\models_{I x/△}\ \exists y.\ \ldots$
  - $\models_{I x/△,\ y/■}\ above(x,y) \vee above(y,x)$
- $\not\models_I \forall\ x.\ \forall\ y.\ above(x,y) \vee above(y,x)$
  - $\not\models_{I x/■,\ y/●}\ above(x,y) \vee above(y,x)$

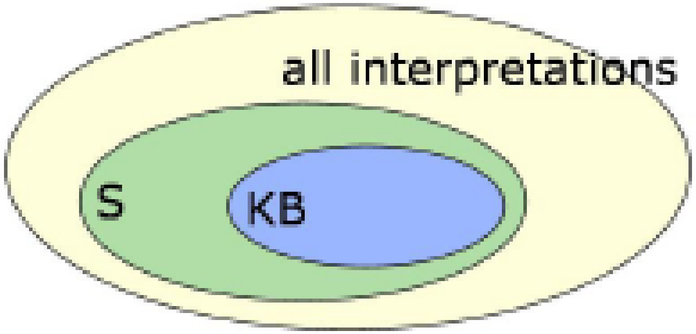
Writing FOL

- Cats are mammals [Cat^1, $Mammal^1$]
  - $\forall x.\ Cat(x) \rightarrow Mammal(x)$
- Jane is a tall surveyor [$Tall^1$, $Surveyor^1$, Jane]
  - $Tall(Jane) \wedge Surveyor(Jane)$
- A nephew is a sibling's son [$Nephew^2$, $Sibling^2$, Son^2]
  - $\forall xy.\ [Nephew(x,y) \leftrightarrow \exists z\ .\ [Sibling(y,z) \wedge Son(x,z)]]$
- A maternal grandmother is a mother's mother [functions: mgm, mother-of]
  - $\forall xy.\ x=mgm(y) \leftrightarrow$
    $\exists z.\ x=mother\text{-}of(z) \wedge z=mother\text{-}of(y)$

- Nobody loves Jane
  - $\forall x.\ \neg\ Loves(x,Jane)$
  - $\neg\exists x.\ Loves(x,Jane)$
- Everybody has a father
  - $\forall\ x.\ \exists\ y.\ Father(y,x)$
- Everybody has a father and a mother
  - $\forall\ x.\ \exists\ yz.\ Father(y,x) \wedge Mother(z,x)$
- Whoever has a father, has a mother
  - $\forall\ x.[[\exists\ y.\ Father(y,x)] \rightarrow [\exists\ y.\ Mother(y,x)]]$

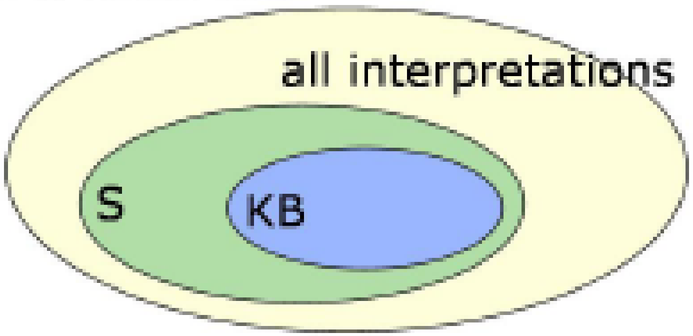
Entailment in First-Order Logic

- KB entails S: for every interpretation I, if KB holds in I, then S holds in I



Entailment in First-Order Logic

- KB entails S: for every interpretation I, if KB holds in I, then S holds in I



- Computing entailment is impossible in general, because there are infinitely many possible interpretations
- Even computing holds is impossible for interpretations with infinite universes

Resolution

For Propositional Logic

-

Resolution rule:

$$\frac{\alpha \vee \beta \qquad \neg\beta \vee \gamma}{\alpha \vee \gamma}$$

Resolution refutation:

- Convert all sentences to CNF
- Negate the desired conclusion (converted to CNF)
- Apply resolution rule until either
  - Derive false (a contradiction)
  - Can't apply any more

Example Problem

Prove R

1	$(P \rightarrow Q) \rightarrow Q$
2	$(P \rightarrow P) \rightarrow R$
3	$(R \rightarrow S) \rightarrow \neg(S \rightarrow Q)$

Convert to CNF

- $\neg(\neg P \vee Q) \vee Q$
- $(P \wedge \neg Q) \vee Q$
- $(P \vee Q) \wedge (\neg Q \vee Q)$
- $(P \vee Q)$

- $\neg(\neg P \vee P) \vee R$
- $(P \wedge \neg P) \vee R$
- $(P \vee R) \wedge (\neg P \vee R)$

- $\neg(\neg R \vee S) \vee \neg(\neg S \vee Q)$
- $(R \wedge \neg S) \vee (S \wedge \neg Q)$
- $(R \vee S) \wedge (\neg S \vee S) \wedge (R \vee \neg Q) \wedge (\neg S \vee \neg Q)$
- $(R \vee S) \wedge (R \vee \neg Q) \wedge (\neg S \vee \neg Q)$

Resolution Proof Example

Prove R

1	$(P \to Q) \to Q$
2	$(P \to P) \to R$
3	$(R \to S) \to \neg (S \to Q)$

1	$P \vee Q$	
2	$P \vee R$	
3	$\neg P \vee R$	
4	$R \vee S$	
5	$R \vee \neg Q$	
6	$\neg S \vee \neg Q$	
7	$\neg R$	Neg
8	S	4,7
9	$\neg Q$	6,8
10	P	1,9
11	R	3,10
12	•	7,11

First-Order Resolution

$$\frac{\forall x.\ P(x) \rightarrow Q(x) \qquad P(A)}{Q(A)}$$

Syllogism:
All men are mortal
Socrates is a man
Socrates is mortal

uppercase letters: constants

lowercase letters: variables

$$\frac{\forall x.\ \neg P(x) \vee Q(x) \qquad P(A)}{Q(A)}$$

Equivalent by definition of implication

$$\frac{\neg P(A) \vee Q(A) \qquad P(A)}{Q(A)}$$

Substitute A for x, still true

then

Propositional resolution

The key is finding the correct substitutions for the variables.

Substitutions

P(x, F(y), B) : an atomic sentence

Substitution instances	Substitution $\{v_1/t_1, \ldots, v_n/t_n\}$	Comment
P(z, F(w), B)	{x/z, y/w}	Alphabetic variant
P(x, F(A), B)	{y/A}	
P(G(z), F(A), B)	{x/G(z), y/A}	
P(C, F(A), B)	{x/C, y/A}	Ground instance

Applying a substitution:

subst({A/y},P(x, F(y), B)) = P(x, F(A), B)

P(x, F(y), B) {A/y} = P(x,F(A),B)

Unification

- Expressions ω_1 and ω_2 are unifiable iff there exists a substitution s such that ω_1 s = ω_2 s
- Let ω_1 = x and ω_2 = y, the following are unifiers

s	ω_1 s	ω_2 s
{y/x}	x	x
{x/y}	y	y
{x/F(F(A)), y/F(F(A))}	F(F(A))	F(F(A))
{x/A, y/A}	A	A

Most General Unifier

g is a most general unifier of ω_1 and ω_2 iff for all unifiers s, there exists s′ such that $\omega_1\ s = (\omega_1\ g)\ s'$ and $\omega_2\ s = (\omega_2\ g)\ s'$

ω_1	ω_2	MGU
P(x)	P(A)	{x/A}
P(F(x), y, G(x))	P(F(x), x, G(x))	{y/x} or {x/y}
P(F(x), y, G(y))	P(F(x), z, G(x))	{y/x, z/x}
P(x, B, B)	P(A, y, z)	{x/A, y/B, z/B}
P(G(F(v)), G(u))	P(x, x)	{x/G(F(v)), u/F(v)}
P(x, F(x))	P(x, x)	No MGU!

Resolution Theorem Proving: First Order Logic

Resolution with variables

Clausal form

Inference using Unification

$$\frac{\forall x.\ \neg P(x) \lor Q(x) \qquad P(A)}{Q(A)}$$

For universally quantified variables, find MGU {x/A} and proceed as in propositional resolution.

$\alpha \vee \phi$ [rename]

$\neg\psi \vee \beta$ [rename] $\mathrm{MGU}(\phi,\psi) = \theta$

$$\frac{\alpha \vee \phi \qquad \neg\psi \vee \beta}{(\alpha \vee \beta)\theta}$$

$$\frac{\begin{array}{c} P(x) \vee Q(x,y) \\ \neg\, P(A) \vee R(B,z) \end{array}}{(Q(x,y) \vee R(B,z))\theta}$$

$Q(A,y) \vee R(B,z)$

$\theta = \{x/A\}$

Resolution with Variables

$\alpha \vee \phi$ [rename]

$\neg\psi \vee \beta$ [rename]

$$\frac{\alpha \vee \phi \qquad \neg\psi \vee \beta}{(\alpha \vee \beta)\theta}$$

$\text{MGU}(\phi,\psi) = \theta$

All vars implicitly univ. quantified

$\forall xy.\ P(x) \vee Q(x,y)$

$\forall z.\ \neg P(A) \vee R(B,z)$

$(Q(x,y) \vee R(B,z))\theta$

$Q(A,y) \vee R(B,z)$

$\theta = \{x/A\}$

$\forall xy.\ \ P(x) \vee Q(x,y)$

$\forall x.\ \neg P(A) \vee R(B,x)$

Scope of var is local to a clause.
Use renaming to keep vars distinct

$\forall x_1y.\ \ P(x_1) \vee Q(x_1,y)$

$\forall x_2.\ \neg P(A) \vee R(B,x_2)$

$(Q(x_1,y) \vee R(B,x_2))\theta$

$Q(A,y) \vee R(B,x_2)$

$\theta = \{x_1/A\}$

Clausal Form

A *literal* is either an atomic sentence or a negation of an atomic sentence.

$$p, \neg p$$

A *clausal sentence* is either a literal or a disjunction of literals.

$$p, \neg p, \; p \vee q$$

A *clause* is a set of literals.

$$\{p\}, \{\neg p\}, \{p,q\}$$

Example: Converting to clausal form

a. John owns a dog
∃ x. D(x) Æ O(J,x)
D(Fido) Æ O(J, Fido)

b. Anyone who owns a dog is a lover-of-animals
∀ x. (∃ y. D(y) Æ O(x,y)) → L(x)
∀ x. (¬∃ y. (D(y) Æ O(x,y)) v L(x)
∀ x. ∀ y. ¬(D(y) Æ O(x,y)) v L(x)
∀ x. ∀ y. ¬ D(y) v ¬ O(x,y) v L(x)
¬ D(y) v ¬ O(x,y) v L(x)

c. Lovers-of-animals do not kill animals
∀ x. L(x) → (∀ y. A(y) → ¬ K(x,y))
∀ x. ¬ L(x) v (∀ y. A(y) → ¬ K(x,y))
∀ x. ¬ L(x) v (∀ y. ¬ A(y) v ¬ K(x,y))
¬ L(x) v ¬ A(y) v ¬ K(x,y)

More converting to clausal form

d. Either Jack killed Tuna or curiosity killed Tuna
K(J,T) v K(C,T)

e. Tuna is a cat
C(T)

f. All cats are animals
¬ C(x) v A(x)

Curiosity Killed the Cat

	D(Fido)	a
	O(J,Fido)	a
	¬ D(y) v ¬ O(x,y) v L(x)	b
	¬ L(x) v ¬ A(y) v ¬ K(x,y)	c
	K(J,T) v K(C,T)	d
	C(T)	e
	¬ C(x) v A(x)	f

Curiosity Killed the Cat

	D(Fido)	a
	O(J,Fido)	a
	¬ D(y) v ¬ O(x,y) v L(x)	b
	¬ L(x) v ¬ A(y) v ¬ K(x,y)	c
	K(J,T) v K(C,T)	d
	C(T)	e
	¬ C(x) v A(x)	f
	¬ K(C,T)	Neg
	K(J,T)	5,8
	A(T)	6,7 {x/T}
	¬ L(J) v ¬ A(T)	4,9 {x/J, y/T}
	¬ L(J)	10,11
	¬ D(y) v ¬ O(J,y)	3,12 {x/J}
	¬ D(Fido)	13,2 {x/Fido}
	•	14,1