First Order Predicate Calculus

Propositional logic v. FOPC

- Propositional calculus deals only with ***facts***
  - P : I-love-all-dogs
  - facts are either true or they are false
- Predicate calculus makes a stronger commitment to what there is (ontology)
  - ***objects*:** things in the world (no truth value)
  - ***properties & relations*** of and between objects (truth value)
- FOPC breaks facts down into objects & relations
  - can be seen as an extension of propositional logic

Predicate calculus

- ***Terms*** refer to ***objects*** in the world
  - John, Mary, etc.
  - ***functions*** (one-to-one mapping)
  - these terms do *not* have a truth value assigned to them
- ***Predicates*** [propositions with arguments]
  - marriedTo(John, Mary)
- ***Quantifiers***
  - $\forall x$[valuable(x)]
  - $\exists x$[valuable(x)]
- ***Equality:*** do two terms refer to the same object?

Terms

- Logical expressions that refer to objects
  - **Constants** (by convention, capitalized)
    - e.g., Sue
  - **Variables** (by convention, lower case)
    - used with quantifiers
    - e.g., x
  - **Functions**
    - **MotherOf(Sue)**
    - since functions represent objects, we can nest them
      - **MotherOf(MotherOf(Ann))**
    - don’t need explicit names
      - **LeftFootOf(John)**

Sentences

- Just as in propositional logic, sentences have a truth-value
- In FOPC, only **relations** (*predicates*) have truth-values
- Thus, ***terms*** alone are not wffs
- They must be (part of) an argument to a ***predicate***

Making sentences

- Atomic sentences: a single predicate
  - married(Sue, FatherOf(Ann))
- Complex sentences
  - just as in propositional logic, we can make more complicated sentences by combining predicates using connectives
    - or, and, implies, equivalence, not
  - quantifiers
  - equality

Quantifiers

- Allows us to express properties of categories of objects without listing all of the objects
- **Universal**
  - **∀x[P(x)] :** T if P(x) is true for *every* object in our interpretation
  - e.g., all men are mortal
- **Existential**
  - **∃x[P(x)] :** T if P(x) is T for *some* object in our interpretation
  - e.g., I love some dog

Equality

- An in-fix predicate
  - but a predicate all the same [returns true or false]
  - e.g., FatherOf(John) = Henry
- termA = termB is shorthand for equal(termA, termB)
  - doesn't have to be in-fix; convenient
- Will be true if termA & termB refer to the same object

Backus-Naur form

Sentence → AtomicSentence

| Sentence Connective Sentence

| Quantifier Variable,… Sentence

| ¬ Sentence

| (Sentence)

Atomic Sentence → Predicate (Term,…)

| Term = Term

BNF (cont.)

Term $\rightarrow$ Function (Term,…)

| Constant

| Variable

Connective $\rightarrow \Rightarrow \mid \wedge \mid \vee \mid \Leftrightarrow$

Quantifier $\rightarrow \forall \mid \exists$

BNF (cont.)

Constant $\rightarrow$ A | X_1 | John | . . .

Variable $\rightarrow$ a | x | s | . . .

Predicate $\rightarrow$ before | hasColor | raining | . . .

Function $\rightarrow$ MotherOf | LeftLegOf | . . .

Well Formed Formulas (WFFs)?

- tall(john)
- MotherOf(john)
- mother(john, mary)
- John = brother(Bill)
  - equal(john, brother(Bill))
- F(p(1) ^ q(2))

English → FOPC

- “Every rational number is a real number”
- $\forall x[\text{rational}(x) \Rightarrow \text{real}(x)]$
- What about
  - $\forall x[\text{rational}(x) \wedge \text{real}(x)]$?
  - $\exists x[\text{rational}(x) \wedge \text{real}(x)]$?
  - $\forall x[\neg\text{real}(x) \vee \text{rational}(x)]$?

More English → FOPC

- There is a prime number greater than 100
  - $\exists x[prime(x) \wedge greaterThan(x, 100)]$
  - $\exists x[prime(x) \wedge x > 100]$
- There is no largest prime
  - no-largest-prime
  - $\forall x[prime(x) \Rightarrow \exists y[prime(y) \wedge greaterThan(y, x)]]$
- Every number has an additive inverse
  - $\forall x[number(x) \Rightarrow \exists y[equal(Plus(x, y), 0)]$

Mixing quantifiers

- Everyone likes a dog
  - $\forall x[\text{human}(x) \Rightarrow \exists y[\text{dog}(y) \wedge \text{likes}(x, y)]]$
- There's one dog everyone likes
  - $\exists y[\text{dog}(y) \wedge \forall x[\text{human}(x) \Rightarrow \text{likes}(x, y)]]$
- Everyone likes a different dog
  - $\forall x[\text{human}(x) \Rightarrow \exists y[\text{dog}(y) \wedge \text{likes}(x, y) \wedge \forall z[\text{human}(z) \wedge \text{likes}(z, y)] \Rightarrow x = z]]$

Location of quantifiers

- Everyone likes a dog
  - $\forall x[human(x) \Rightarrow \exists y[dog(y) \wedge likes(x, y)]]$
- What about
  - $\forall x[\exists y[(human(x) \wedge dog(y)) \Rightarrow likes(x, y)]]$
    - human(Jim) : T; human(Spot) : F;
    - dog(Jim) : F; dog(Spot) : T;
    - likes(Jim,Jim) : T; likes(Jim,Spot) : F;
    - likes(Spot,Jim) : F; likes(Spot,Spot) : T
- In general, never use $\exists x$ with $\Rightarrow$, and don't use $\forall x$ with $\wedge$

What is the truth-value of FOPC wffs? FOPC interpretations

- The ***"user"*** must provide a finite list of objects in the world
  - "universe of discourse"
- For each ***function***, a mapping from "parameter setting" to an object in the world
  - e.g., Father(John) maps to "Bill"
- For each ***predicate***, a mapping from each "parameter setting" to true or false

Determining truth-value of FOPC wff

- Specify an interpretation, I
- Obtain truth-values of
  - predicates
    - look up functions until only constants remain & then look up the truth value of the predicate
  - termA = termB
    - look up functions until only constants remain; true if same constant; false otherwise
  - wffA connective wffB
    - compute the truth-value of the wffs
    - use connective's truth table to determine the truth-value of compound wff (same for $\neg$)

Truth-value for quantifiers

- $\forall x$ wff(x)
  - successively replace x by each constant in the interpretation
  - if wff(constant) is true for every case, then $\forall x$(wff) is true
- $\exists x$ wff(x)
  - same as above, but wff(constant) has only to be true once
- assume constants list is never empty

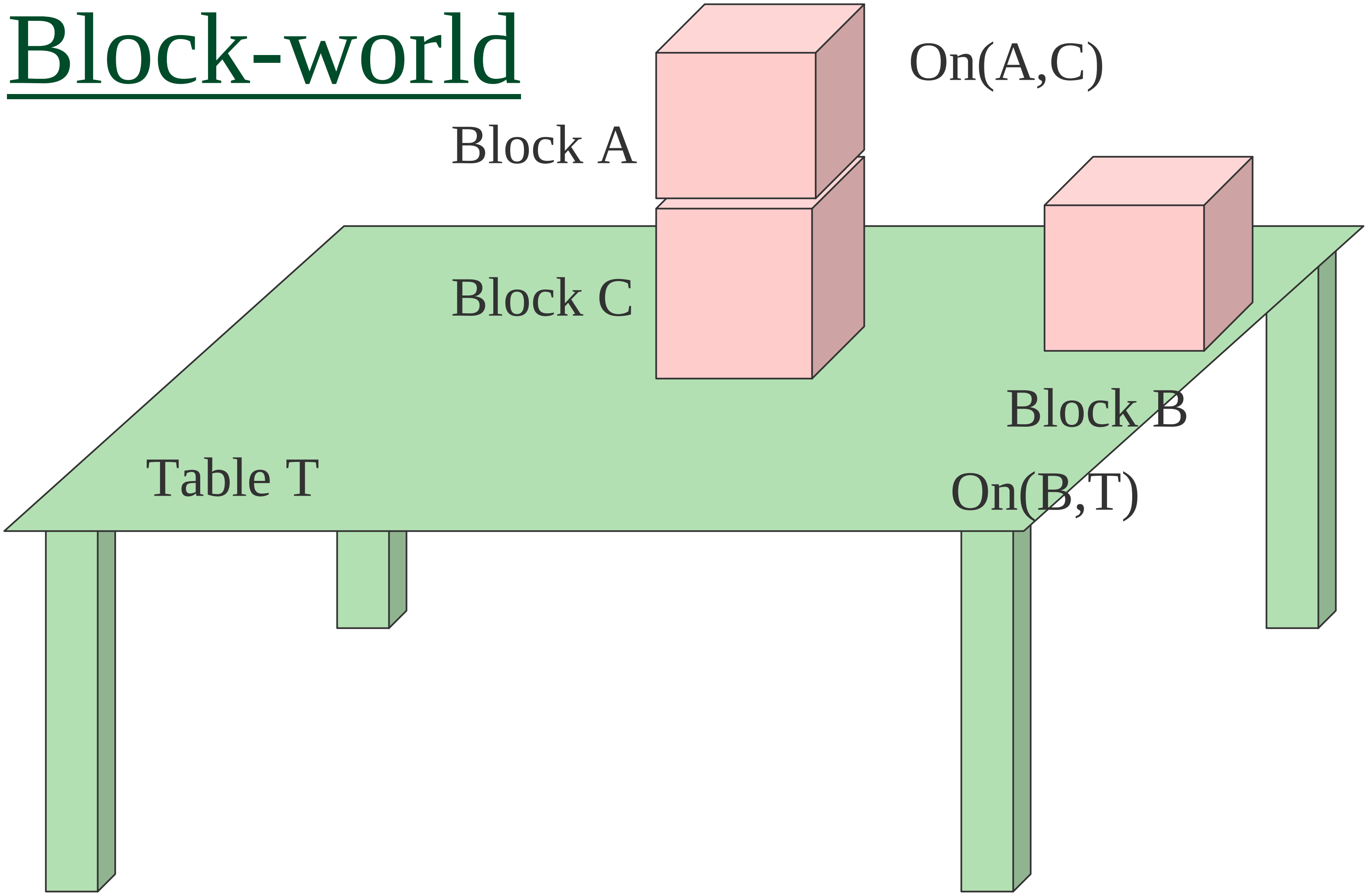
Representing change

- On(BlockA, BlockB)
  - this is either T or F
  - there is no way to change this fact in “basic” FOPC
- Solution: “time stamp” wffs
  - add one more parameter to all predicates indicating *when* they are true
  - On(BlockA, BlockB, S0)
  - On(BlockA, BlockC, S1)
    - where S0 & S1 are situations or states

Changing the world

- Acting (operator applications) changes states (situations) into other states
- We need a name for the new state
  - Use functions!
  - In particular, the function *Result*
    - maps an action and a state to a new state
    - Result(<action>, <state>) ⇒ <state>
    - simply a fancy name for a state, just as FatherOf(---) is a fancy name for some man

Block-world



Block-world example

- $\forall x,y,z,s[\text{block}(x) \wedge \text{block}(y) \wedge \text{table}(z) \wedge \text{state}(s) \wedge \text{on}(x, z, s) \wedge \text{clear}(x, s) \wedge \text{clear}(y, s)] \Rightarrow$
  $\text{state}(\text{Result}(\text{Stack}(x, y), s)) \wedge$
  $\text{on}(x, y, \text{Result}(\text{Stack}(x, y), s)) \wedge$
  $\text{clear}(x, \text{Result}(\text{Stack}(x, y), s)) \wedge$
  $\neg\text{clear}(y, \text{Result}(\text{Stack}(x, y), s)$
- Could now almost use deductions to produce plans (sequences of action)

What’s missing?

- What do we know about c in the new state?
- This is a case of the *frame problem*: knowing what stays the same as we move from state to state (like frames in a movie)
- “Blocks stay clear unless something is placed on them during stacking”
- $\forall u,x,y,s[clear(u, s) \wedge \neg(u = y) \Rightarrow clear(u, Result(Stack(x, y), s))$

Example

- Painting a house does not change who owns it
- $\forall s,h,p[state(s) \wedge house(h) \wedge human(p) \wedge owns(p, h, s) \Rightarrow owns(p, h, Result(paint(h), s))]$

Alternate approach

- Say properties stay the same unless a specific action performed
- $\forall$u,x,s,a [block(u) ^ state(s) ^ action(a) ^ clear(u, s) ^ ¬(a = Stack(x, u)) ^ ¬(a = CoverWithBlanket(u)) ^ ¬(a = Smash(u)) => clear(u, Result(a, s)]
- This usually leads to fewer rules, but it is less modular
  - when new actions defined, we have to double check every such rule to see if it needs editing

Problems with formalization

- ***Qualification*** problem
  - can we ever really write down all the "preconditions" for a real-world action?
  - E.g., starting a car
- ***Ramification*** problem
  - need to represent implicit consequences of actions
  - moving car from A to B also moves its steering wheel, spare tire, etc.
  - can be handled but becomes tedious

Sources

- ***Computer Science Lab***
- ***University of Wisconsin, Madison***