CS2120 F22

From Propositional to Predicate Logic

The Expressiveness of a Logic

- The expressiveness of a logic refers to the range of ideas it can encode
  - Propositional logic
    - Syntax
      - The basic truth value constants (true, false)
      - Boolean variables for propositions (X, Y, Z, NiftyIsACat, TomLivesInCVille, etc.)
      - The basic logical connectives
    - Semantics
      - An *interpretation* gives a Boolean value to each variable in an expression
      - The logical connectives are also interpreted: as specifying Boolean functions
      - The meaning of a larger proposition is composed from the meanings of its parts
  - Propositional logic extended with *background theories*
    - Syntax:
      - Same logical connectives with the same meaning
      - Elementary propositions can be expressed in terms of *background theories* (e.g., arithmetic)
    - Semantics:
      - Richer notion *interpretation*: e.g., variables can now also refer to objects such as numbers
      - Think about our circle/triangle/square puzzle: these funny looking variables refer to *numbers*
      - Can specify much richer ideas, e.g., "no two numbers in a row or column can be *equal"*

Limitations on the Expressiveness of Propositional Logic

- Variables can't refer to objects of arbitrary types; no notion of types at all
  - *Boolean*, sure
  - But *person, electron, mathematical group, school, class, employee, principal,* … no
- Cannot *quantify* over sets of objects, or over functions, n-ary relations, etc
  - <u>*Every*</u> interpretation for proposition P makes P true (P is valid)
  - <u>*Some*</u> interpretation for proposition P makes P true (P is satisfiable)
  - <u>*No*</u> interpretation for proposition P makes P true (P is unsatisfiable)
- No mechanism for *parameterizing* propositions
  - Yes: *SocratesIsHuman*, *KantIsHuman*, *SearleIsHuman, X, Y,, Z,* etc.
  - No: IsHuman(X), where X can be any person: *isHuman(Socrates), isHuman(Kant),* etc.
  - A *parameterized* proposition is called a *predicate:* e.g., isHuman, isEven, isMortal, …
  - Predicates can have multiple parameters: e.g., *=(m,n); <(m,n); relativelyPrime(m,n)*; etc.
  - A *proposition* is a degenerate predicate with no parameters, e.g., *isHuman(Socrates)*

Predicate Logic

- Predicate logic addresses each limitation vastly expanding its expressiveness
  - Variables can range over values of *arbitrary types*
  - Syntax adds *universal* and *existential quantifiers*
    - ∀ x, P x – for *all*, for *any*, for *every* value of x, P(x) is true
    - ∃ x, P x – there *exists*, for *some* value of x, P(x) is true
  - Predicate logic has *predicates*, denoting whole families of propositions
- But this increase in expressiveness comes with heavy costs
  - Vastly more complex notions of *interpretation* and *truth*
  - No longer any possibility of an automated satisfiability solver
  - Can no longer rely on truth table checking as a test of validity
  - One must use *deductive reasoning* (using *inference rules*) to check validity
  - A chain of deductive inferences showing validity is what we usually mean by a *proof*

Rest of today

- Quick reminder of what a natural deduction proof looks like
- Quick exposure to variants of predicate logic
  - Intuitionistic or constructive logic (a.k.a. *type theory*)
  - First-order predicate logic
- Practice with predicates and quantifiers

Proofs in propositional and predicate logic

- How would you prove that this proposition is valid? $X \land Y \to Y \land X$
- If it's propositional logic, use a truth table (evaluate it for each interpretation)
- In predicate logic you will generally need to use deductive reasoning
  - Assume H: $X \land Y$ (hypothesis, or premise)
  - Goal: In this context, show $Y \land X$
  - Apply *and-elimination* to H to deduce X and Y separately
  - Apply *and-introduction* to Y and X in that (reverse) order to show $Y \land X$
  - Having shown that *if* $X \land Y$ then $Y \land X$ apply *arrow introduction to* conclude $X \land Y \to Y \land X$
- *Arrow intro*: If in a context with H, you can prove K, you can deduce $H \to K$
- A proof in predicate logic is a very different beast than in propositional logic!

Major Variants of Predicate Logic

- First-order predicate logic
  - No explicit types
    - everything is just an object
    - use predicates in lieu of types
    - E.g., $\forall$ x, isHuman x $\rightarrow$ isMortal x
    - Read that as, "With x being any *object,* if x is Human then x is mortal"
  - You can quantify over sets of objects but not over functions, relations, predicates, propositions
  - Predicates can have objects but not predicates, functions, relations, propositions as arguments
- Higher-Order constructive logic
  - Generally come with an exceptionally expressive *type systems*, including user-defined types
  - E.g., if Human is a defined type, you can now write *$\forall$ (x : Human), isMortal x*
  - In FOPL, you can apply isMortal to any object, in HOCL applying it to non-human is a type error
  - In FOPL, you can write *isMortal oxygen*; in HOCL it's just a type error
  - In HOCL, you can quantify over functions, propositions, etc.
  - HOCL can serve as both a logic and a typed functional programming language
  - FOPL is a subset of HOCL (as propositional logic is a subset of predicate logic)

Let’s play with quantifiers: formalize in HOCL

- Every person is mortal
- Some person is happy
- Someone likes everyone
- Everyone likes someone
- There is someone everyone likes
- For any persons, P, Q, R, if P likes Q, and Q likes R, then P might like R
- The enemy of your enemy is your friend
- For any natural number, n, if n is even then n+1 is odd
- With n being any natural number, if n >= 2, then if no number in 2,...,n/2 divides n evenly, then n is prime
- No one likes everyone
- If there’s a person everyone likes, then everyone likes someone