Lecture 2
Logic and Proof

David Garlan
Carnegie Mellon University

Models of Software Systems
15-671
Fall 1994

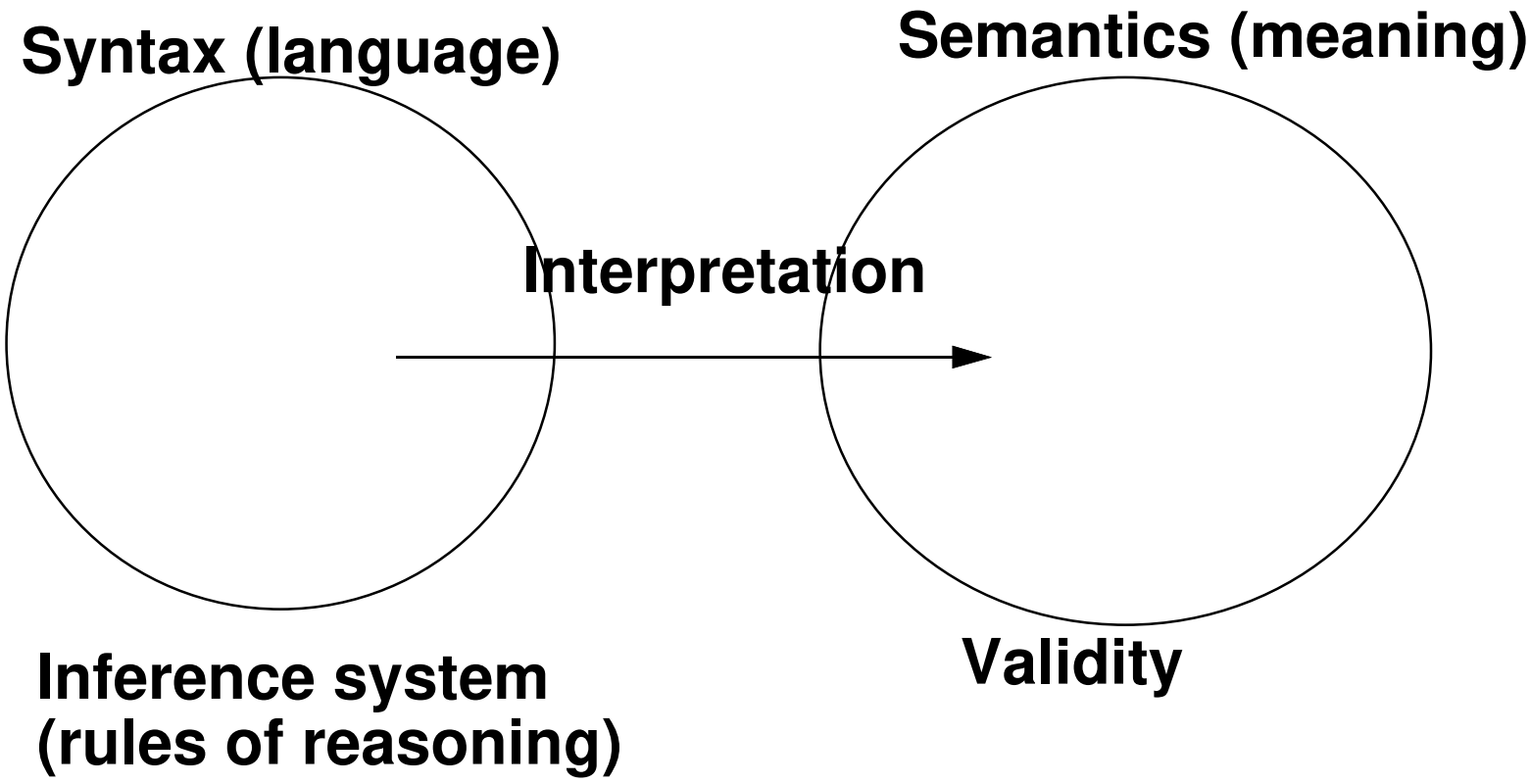
© Garlan, 1994

The Story So Far

- **Formal System**
  - > language: alphabet + grammar
  - > inference system: axioms + rules of inference
  - > theory

    Theorem, Proof, Deduction
- **Semantics**
  - > interpretation
  - > soundness (consistency), completeness

Basic Elements



This Lecture

- **This lecture we will look at two specific formal systems:**
  - **Propositional logic**
  - **Predicate logic**
- **We will also briefly examine some of the main proof styles used in these systems**
  - **Equational reasoning**
  - **Proof by contradiction**
  - **Proof by case analysis**
  - **and others**

Propositions

A proposition is a statement that is either true or false (but not both).

Examples

- **Pittsburgh is the home of the Pirates**
- **Pittsburgh is the capital of Pennsylvania**

The same proposition can often be expressed in several ways.

Examples

- **five is bigger than four**
- **four is less than five**
- **5 > 4**

(Qn: what's an example of a statement that is neither true nor false?)

Propositional Logic: Syntax

- **Alphabet: {P,Q,R, ..., $\wedge$, $\vee$, ~, $\Rightarrow$, $\Leftrightarrow$}**
- **Grammar:**

  sentence = "P" | "Q" | "R" | ...

  | "~", sentence

  | "(", sentence, "$\vee$", sentence, ")"

  | "(", sentence, "$\wedge$", sentence, ")"

  | "(", sentence, "$\Rightarrow$", sentence, ")"

  | "(", sentence, "$\Leftrightarrow$", sentence, ")"

  we usually drop the parentheses when it the resulting sentence is unambiguous

© Garlan, 1994

Propositional Logic: Semantics

- **The meaning of a propositional wff is given as follows:**
  - > **each primitive symbol (P, Q, R, ...) is given an interpretation as a proposition (with an associated truth value)**

    example: P => today is Thursday, ...
  - > **the meaning (truth or falsity) of a compound wff is then determined by a set of standard tables:**

    example:

p	q	$p \Rightarrow q$
T	T	T
T	F	F
F	T	T
F	F	T

(NB: four interpretations)

  - > **In short, we use "truth tables"**

Terminology & Special Cases

- **A sentence of the form**
  - **A $\wedge$ B is called a *conjunction***
  - **A $\vee$ B is called a *disjunction***
  - **A $\Rightarrow$ B is called an *implication***
- **Some sentences always interpret to true.**
  - **these are called *tautologies***
  - **example: a sentence of the form A $\vee$ ~A, where A is any proposition**
- **Some sentences always interpret to false.**
  - **these are *contradictions,* or *inconsistencies***
  - **example: a sentence of the form A $\wedge$ ~A, where A is any proposition**

© Garlan, 1994

Propositional Calculus

- To complete our system we need a set of inference rules (called a calculus, here)
- One way to represent these rules is in the form:

  if we have proved these facts

  ---

  we can conclude these facts

- Examples

$$\frac{A \wedge B}{A} \qquad \frac{A}{A \vee B} \qquad \frac{A,\ A \Rightarrow B}{B}$$

- These are given names (resp):

  $\wedge$-elimination, $\vee$-introduction, $\Rightarrow$-elimination

Example

- **Prove:** $(P \wedge Q) \vdash (P \vee Q)$
- **Derivation:**
  - $P \wedge Q$ — premise
  - P — $\wedge$-elimination
  - $P \vee Q$ — $\vee$-introduction

About Proofs

- There are many styles (or techniques) of proof.
- SEM presents a style called “natural deduction”
- In this style a proof is constructed as a tree.
- We will not be using this style.
- However there are a lot of other styles that we will use as needed.
- Many of these center around the application a particular inference rule.

Some Proof Styles

- **Style 1: simple application of rules**
  - > we have already seen this a couple of times
  - > note for each step of proof write justification in terms of rule and lines on which it depends
- **Style 2: proof by contradiction**
  - > assume that the conclusion is not true and derive something that we know to be false
- **Style 3: proof by cases**
  - > break proof in to separate parts and then combine
- **Style 4: modus ponens**
  - > Show P and P $\Rightarrow$ Q. Then conclude Q.

© Garlan, 1994

Equivalence

- We augment our formal system with a new symbol
- Two sentences are *equivalent* ($\equiv$) iff their truth value is the same under every interpretation.
- Example: $(P \Rightarrow Q) \equiv (\sim P \vee Q)$
- There is a collection of properties of equivalence, summarized on page 26 of SEM.
- These can be used in an "equational" style to do proof.

Example

Prove: ~ (P $\wedge$ ~Q) $\equiv$ (Q $\vee$ ~P)

Proof:

~ (P $\wedge$ ~Q)	
$\equiv$	DeMorgan
~ P $\vee$ Q	
$\equiv$	$\vee$–Commutative
Q $\vee$ ~P	

Predicate Logic

- Sometimes want to assert that all elements of some set (S) satisfy a common property (P).
- For example if S == {Larry, Joe, Moe}, we might want to claim that all elements of S are stooges.
- We could write

  (Larry is a stooge)$\wedge$ (Joe is a stooge)$\wedge$ (Moe is a stooge)
- Alternatively we can create a template: Stooge( )

  Stooge(Larry)$\wedge$ Stooge (Joe)$\wedge$ Stooge (Moe)
- Stooge() is called a predicate.

Predicates (2)

- But for large sets this becomes unwieldy (or impossible).
- So we use a shorthand:

  $\forall$ x:S • Stooge(x)

- Similarly, to assert the existence of an element of some set (S) that satisfies a property (P) we use

  $\exists$ x:S • P(x)

  which is shorthand for

  P(s1) $\vee$ P(s2) $\vee$ P(s3) $\vee$ ...

Predicates (3)

- For the moment we will assume that the sets that are quantified over are understood by context

  examples: $\forall$ x • P(x), $\exists$ x • Q(x)

- You can think of a predicate as a boolean valued function: when applied to an argument it returns true or false.
- Predicates may have n arguments
  - examples: >, is-pythagorian

Predicate Logic: Syntax

- **Alphabet: as with propositional logic, but includes quantifiers (∀, ∃) and •**
  - **∀ is called a "universal" quantifier**
  - **∃ is called an "existential" quantifier**
- **Grammar: see SEM**
  - **but note every quantified expression has a •**
- **Can nest quantified expressions**
  - **∀ x • (∀ y • (∃ z • P(x,y,z)))**
  - **usually leave out the parens**
  - **combine adjacent variables with same quantifier.**
  - **so this becomes ∀ x, y • ∃ z • P(x,y,z)**

Predicate Logic: Semantics

- Considerably more involved than propositional logic
- Basic idea is that sentence
  - $\forall$ x • P(x) is true iff P is true of all values of x, and
  - $\exists$ x • Q(x) is true iff P is true for some value of x

Predicate Calculus

- As with Propositional Logic we introduce a set of deductive rules.
- Similar to situation before
  - > SEM uses natural deduction style
  - > We won't focus on this, but instead use other techniques as appropriate

Equality

- We now add one more symbol to Predicate Logic: = to represent that two entities are the same.
- This is equivalent to having a special predicate “equals”, but we use the more common infix notation.
- We also introduce several new rules to our deductive apparatus.
- The most important of these is:

  Substitution rule:

$$\frac{m = n,\ S(n)}{S[m/n]}$$

© Garlan, 1994

Theories

- We can do a lot with just logic.
- But sometimes we want to introduce special axioms to describe a specific phenomenon.
- Then we are interested in the consequences of those axioms (given the other rules of deduction).
- These consequences are called a theory.

Examples

- Some general theories that we will encounter in this course
  - > set theory
  - > theory for sequences
  - > theory for concurrent processes
- Other theories we will make up on the spot to characterize a software system
  - > example: theory of telephones for a switching system
  - > example: theory of devices and event processing for real-time system
  - > homework problems explore some of these