CISC 204 Class 29

Semantic Entailment

Text Correspondence: pp. 128–129

Main Concepts:

- *Semantic entailment: an "if-then" relation in predicate semantics*

Recalling propositional logic, we defined the semantics of a formula ϕ by assigning **T** or **F** to each propositional atom in the formula. We called this assignment a *valuation*. We then worked the assignments through the logical connectives and evaluated the formula ϕ as either **T** or **F**.

For a set of formulas $\Gamma = \{\phi_1, \phi_2, \ldots, \phi_n\}$ we wrote the the semantic entailment as

$$\Gamma \models \psi$$

The meaning of semantic entailment in propositional logic is: if, for every valuation in which every $\phi_i \in \Gamma$ evaluates to **T**, the formula ψ also evaluates to **T**, then we say that $\Gamma \models \psi$.

When we think of extending semantic entailment to predicate logic, we must face the difficulty of extending a valuation to a predicate formula. In propositional logic we needed to examine every valuation in which every premise formula in Γ evaluated to **T**. This can be done automatically: for k propositional atoms there are 2^k valuations, which are rows in a truth table. To check every one of the n premises, and the conclusion, we need to evaluate $2^k \cdot (n+1)$ valuation-formula combinations. This grows exponentially but is algorithmic.

What happens in predicate logic? The parallel concept for a valuation is an interpretation, which is the combination of a model and a logical environment. So, it seems that we must reason over every possible interpretation, which means reasoning over every possible model $\mathcal{M}$ and every possible logical environment l.

Then, for each interpretation, we must reason over every premise in Γ and the conclusion ψ. It seems possible that there are an infinite number of models; certainly, if the universe A is infinite then there might be an infinite number of environments; and there could well be an infinite number of premises in the set Γ for us to manage.[1]

In propositional logic, we also defined the concepts of validity and satisfiability. That a propositional formula ϕ was satisfiable meant that ϕ evaluated to **T** for at least one valuation. This concept also has a parallel in predicate logic, in which a predicate formula ϕ evaluates to **T** for some interpretation.

[1]One useful infinite set of premises Γ is the set of all predicate theorems.

That a propositional formula ϕ was valid meant that ϕ evaluated to **T** for every valuation, which in practice meant that ϕ evaluated to **T** for every row in the truth table. This, too, extends to predicate logic with interpretations.

We can formalize these concepts in four definitions. We will begin with satisfiability and end with semantic entailment, advancing from concept to concept.

We will compare and contrast the definitions after we have stated all of them.

Definition: satisfiable formula ϕ

A formula ϕ is *satisfiable* means that:

there exists an interpretation $\mathcal{I}$ such that the satisfaction $\mathcal{M} \models_l \phi$ evaluates to **T**.

Definition: consistent set of formulas Γ

A set $\Gamma = \{\phi_1, \phi_2, \ldots, \phi_n\}$ is *consistent* means that:

there exists an interpretation $\mathcal{I}$ such that, for all $\phi_i \in \Gamma$, the satisfaction $\mathcal{M} \models_l \phi_i$ evaluates to **T**.

A set of formulas Γ that is consistent may also be called a *satisfiable* set.

Definition: valid formula ψ

A formula ψ is *valid* means that:

for all interpretations $\mathcal{I}$ in which ψ can be evaluated, the satisfaction $\mathcal{M} \models_l \psi$ evaluates to **T**.

Definition: semantic entailment $\Gamma \models \psi$

A sequent $\Gamma \models \psi$ is a *semantic entailment* means that:

for all interpretations $\mathcal{I}$ in which ψ and every ϕ_i can be evaluated, if for all $\phi_i \in \Gamma$ the satisfaction $\mathcal{M} \models_l \psi$ evaluates to **T**, then ψ evaluates to **T**.

Computationally, these definitions are successively difficult to automate. The satisfiability of a formula amounts to an existence proof. If there is at least one model and environment in which the satisfaction $\mathcal{M} \models_l \phi$ evaluates to **T**, then the formula ϕ is satisfiable. This may require an infinite loop over an infinite universe A of values. Showing that a formula ϕ cannot be satisfied is harder, because this amounts to showing that $\neg\,\phi$ is valid.

We can often demonstrate specific instances of satisfiability by finding a positive example; this can often be done by translating a symbolic formula into English and using our reasoning ability to find an example. We use a similar line of reasoning for some simple formulas in the previous class; we selected base-4 arithmetic as the model, then we reasoned about satisfiability by writing some brief algorithms for evaluation.

Showing that a set of formulas Γ is consistent requires that we find a model $\mathcal{M}$ and an environment l in which *every* formula in Γ is satisfiable. This, unlike its parallel definition in propositional logic, cannot be automated. An immediate problem is that the set Γ could be infinite, so checking every $\phi_i \in \Gamma$ cannot be programmed.

We can, however, often show that a set of formulas is *not* consistent in one of two ways. A semantics approach is to show that for every interpretation $\mathcal{I}$, at least one formula evaluates to **F**. For example, the set of formulas

$$\{\exists x \neg Q(x)\ ,\ \forall y\,(P(y) \to Q(y))\}$$

is inconsistent in an interpretation in which some value is both in set P and in set Q.

A syntactic approach to inconsistency is to use the rules of predicate logic to deduce a contradiction, that is, prove

$$\Gamma \vdash \bot$$