Logic for Computer Science 23rd January, 2024

Lecture 3: Relevance Lemma

Lecture: Sujata Ghosh *Scribe: Ritam M Mitra*

1 Free and Bound Variables

The introduction of variables and quantifiers allows us to express the notions of *all* ... and *some* ... [2]Intuitively, to verify that $\forall x Q(x)$ is true amounts to replacing x by any of its possible values and checking that Q holds for each one of them. There are two important and different senses in which such formulas can be 'true.' First, if we give concrete meanings to all predicate and function symbols involved we have a **model** [1] and can check whether a formula is true for this particular **model**. For example, if a formula encodes a required behaviour of a hardware circuit, then we would want to know whether it is true for the model of the circuit. To begin with, we need to understand that variables occur in different ways.

1.1 Parse Tree

Consider the formula.

$$\forall x((P(x) \to Q(x)) \wedge S(x, y)).$$

We draw its parse tree with two sorts of nodes :

- The quantifiers $\forall x$ and $\exists y$ form nodes and have, like negation, just one subtree.
- Predicate expressions, which are generally of the form $P(t_1, t_2, \ldots, t_n)$, have the symbol P as a node, but now P has n many subtrees, namely the parse trees of the terms $t_1, t_2, \ldots, t_n$.

You can see that variables occur at two different sorts of places. First, they appear next to quantifiers $\forall$ and $\exists$ in nodes like $\forall x$ and $\exists z$; such nodes always have one subtree, subsuming their scope to which the respective quantifier applies. The other sort of occurrence of variables is leaf nodes containing variables. If variables are leaf nodes, then they stand for values that still have to be made concrete.

There are two principal such occurrences:

- In this example, we have three leaf nodes x. If we walk up the tree beginning at any one of these x leaves, we run into the quantifier $\forall x$. This means that those occurrences of x are actually bound to $\forall x$ so they represent, or stand for, any possible value of x.
- In walking upwards, the only quantifier that the leaf node y runs into is $\forall x$ but that x has nothing to do with y; x and y are different place holders. So y is free in this formula. This means that its value has to be specified by some additional information, for example, the contents of a location in memory.

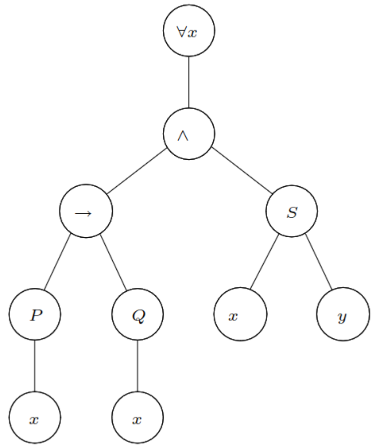

Figure 1: A parse tree of a predicate logic formula.

Definition 1.1 *Let ϕ be a formula in predicate logic. An occurrence of x in ϕ is free in ϕ if it is a leaf node in the parse tree of ϕ such that there is no path upwards from that node x to a node $\forall x$ or $\exists x$. Otherwise, that occurrence of x is called bound. For $\forall x\phi$, or $\exists x\phi$, we say that ϕ– minus any of ϕ's subformulas $\exists x\psi$, or $\forall x\psi$– is the scope of $\forall x$, respectively $\exists x$.*

Thus, if x occurs in ϕ, then it is bound if, and only if, it is in the scope of some $\exists x$ or some $\forall x$; otherwise it is free. In terms of parse trees, the scope of a quantifier is just its subtree, minus any subtrees which re-introduce a quantifier for x; e.g. the scope of $\forall x$ in $\forall x(P(x) \rightarrow \exists x Q(x))$ is $P(x)$.

1.2 Reasoning by Induction on Terms and Formulas(Relevance Lemma

When comparing different variable assignments and their effect on a given formula ϕ, those variables that do not occur in the formula or are not free variables in the formula should not affect the fact that a given structure A logically implies the formula or not. This intuitive result is formalized as the **Relevance Lemma**, and can be stated as:

Lemma 1.2 *Let A be a structure, ϕ a formula, and α_1, α_2 be variable assignments such that:*

$$\alpha_1|_{FV(\phi)} = \alpha_2|_{FV(\phi)}.$$

Then,

$$A, \alpha_1 \models \phi \iff A, \alpha_2 \models \phi.$$

Often proofs about first-order logic involve induction on the structure of terms and formulas. We prove this lemma by way of example.
Proof. We first prove the following claim.

Claim 1.3 *Let $t \in$ Term, and let A be a structure and α_1 and α_2 be variable assignments. Then, if $\alpha_1|_{FV(t)} = \alpha_2|_{FV(t)}$. then $\overline{\alpha_1}(t) = \overline{\alpha_2}(t)$.*

Proof. We give a proof by structural induction.

- **Base case:** If t is a variable x then $\alpha_1(x) = \alpha_2(x)$. Notice also that by definition of α in this case $\overline{\alpha_1}(t) = \alpha_1(t)$ and the same is true for $\alpha_2(t)$. Thus,

$$\overline{\alpha_1}(t) = \alpha_1(t) = \alpha_2(t) = \overline{\alpha_2}(t).$$

  and therefore $\alpha_1(t) = \alpha_2(t)$. Here the first and the third equalities follow from the definition of α, and the second equality comes from the assumption of the lemma.

- **Inductive case:** The other option for t is to be a function $f^k(t_1, \ldots, t_k)$, where $t_1, \ldots, t_k$ are terms. By definition of $Vars$(set of variables), $Vars(f^(k)(t_1, ..., t_k) = \cup_{i=1}^{k} Vars(t_i)$ and so $Vars(t_i) \subset Vars(t)$. Then by inductive hypothesis we have that $\alpha_1(t_i) = \alpha_2(t_i)$ for all i, and thus

$$\overline{\alpha_1}(f^k(t_1, \ldots, t_k)) = \text{(by definition)}$$

$$f^A(\overline{\alpha_1}(t_1), \ldots, \overline{\alpha_1}(t_k)) = \text{(by I.H.)}$$

$$f^A(\overline{\alpha_2}(t_1), \ldots, \overline{\alpha_2}(t_k)) = \text{(by definition)}$$

$$\overline{\alpha_2}(f^k(t_1, \ldots, t_k)).$$

□

We prove the lemma using this claim.

Again we use induction.

- **Base case:** Suppose that ϕ is a k-ary predicate (relation) $P^(k)(t_1, \ldots, t_k)$. By definition $FVars(\phi) = \cup_{i=1}^{k} FVars(t_i)$. We now have the following proof sequence:

$$A, \alpha_1 \vDash \phi$$

$$\text{iff } \langle \alpha_1(t_1), \ldots, \alpha_1(t_k) \rangle \in P^A \text{ (by definition)}$$

$$\text{iff } \langle \alpha_1(t_1), \ldots, \alpha_1(t_k) \rangle \in P^A \text{ (by claim)}$$

$$\text{iff } A, \alpha_2 \vDash \phi \text{ (by definition)}$$

- **Inductive case :**

  1. $\phi = \neg\psi$. By definition $A,\ \alpha_1 \vDash \neg\psi$ iff $A,\ \alpha_1 \nvDash \psi$. Also by definition $FVars(\phi) = FVars(\psi)$. Thus, by inductive hypothesis we have $A, \alpha_2 \nvDash \psi$ which leads to $A, \alpha_2 \vDash \phi$.
  2. $\phi = \psi \wedge \theta$. By definition $A, \alpha_1 \vDash \phi$ iff $A, \alpha_1 \vDash (\psi \wedge \theta)$ iff $A, \alpha_1 \vDash \psi \text{and} A, \alpha_1 \vDash \theta$. Clearly $FVars(\psi) \subset FVars(\phi)$ and $FVars(\theta) \subset FVars(\phi)$, so by inductive hypothesis $A, \alpha_2 \vDash \psi$ and $A, \alpha_2 \vDash \theta$. This is another way of saying $A, \alpha_2 \vDash \phi$.

3. $\phi = (\exists x)\psi$. $A, \alpha_1 \models (\exists x)\psi$ by definition means that there is some $d \in D$ such that $A, \alpha_1[x \to d] \models \psi$. To use the inductive hypothesis we need agreement on the free variables. Observe that $FVars(\phi) = FVars(\psi) - \{x\}$ and therefore $FVars(\psi) \subset FVars(\psi) \bigcup x$. Since $\alpha_1|_{FVars(\alpha)} = \alpha_2|_{FVars(\alpha)}$ it must also be true that$\alpha_1[x \to d]|_{FVars(\psi)} = \alpha_2[x \to d]|_{FVars(\psi)}$ which is the agreement on free variables that we need. We quote once again the inductive hypothesis and conclude that $A, \alpha_2[x \to d] \models \phi$ which is equivalent to $A, \alpha_2 \models (\exists x)\phi$

The proof for the remaining connectives and one quantifier can be derived from those above stated by using DeMorgan's laws.

□

References

[1] Herbert B Enderton. *A mathematical introduction to logic*. Elsevier, 2001.

[2] Michael Huth and Mark Ryan. *Logic in Computer Science: Modelling and reasoning about systems*. Cambridge university press, 2004.

Logic for Computer Science	January 19 2024

Lecture 4: Consequence Relation

Lecture: Sujata Ghosh *Scribe: Basudeb Roy, Ritam Manna Mitra*

Topics for this lecture

In this lecture, we shall talk about the following

1. Proof of corollary
2. Truth in a structure
3. Expressivity
4. (Semantic)Consequence relation
5. validity and Satisfiability

Let us begin this lecture with the proof of the following corollary.

Corollary 0.1 *Let ϕ be a sentence and $(\mathcal{D}, \mathcal{I})$ be a structure. Then, either for all assignment functions $\mathcal{G}$, $(\mathcal{D}, \mathcal{I}, \mathcal{G}) \models \phi$ or $(\mathcal{D}, \mathcal{I}, \mathcal{G}) \not\models \phi$.*

Proof. Let's assume, $(\mathcal{D}, \mathcal{I}, \mathcal{G}) \models \phi$, that means we are done. Now if it is the case that, $(\mathcal{D}, \mathcal{I}, \mathcal{G}) \not\models \phi$ that means, it is the case that $(\mathcal{D}, \mathcal{I}, \mathcal{G}) \models \neg\phi$. □

1 Truth in a structure

1. When ϕ is a sentence, we say ϕ is true in a structure $(\mathcal{D}, \mathcal{I})$, which is expressed in first-order language by $[(\mathcal{D}, \mathcal{I}) \models \phi]$.
2. When ϕ is a sentence, we say ϕ is false in a structure $(\mathcal{D}, \mathcal{I})$, which is expressed in first-order language by $[(\mathcal{D}, \mathcal{I}) \not\models \phi]$.

Next natural question comes what can we express using the first-order language? To answer this question, let us first consider a first-order language such that $\mathcal{C}_{\mathcal{L}} = \mathcal{F}_{\mathcal{L}} = \mathcal{P}_{\mathcal{L}} = \Phi$. With these assumptions we can talk only about sets.

2 On Expressivity:-

For this discussion, let us assume that $\mathcal{D}$ can be empty or non-empty. Let us try to talk about the sets with such language.

1. Empty Set:- $\forall x \neg(x = x)$.
2. All sets whose cardinality is ≤ 1:- $\forall x \forall y (x = y)$.

3. $SingletonSets : - = \exists x \forall y(x = y)$
$$= \exists x(x = x) \wedge \forall x \forall y(x = y)$$

4. Set containing exactly 2 elements:- $\exists x \exists y(\neg(x = y) \wedge \forall z(z = x \vee z = y))$

5. Set containing exactly 3 elements:- $\exists x \exists y \exists z(\neg(x = y) \wedge \neg(y = z) \wedge \neg(z = x) \wedge \forall w(w = x \vee w = y \vee w = z))$

Similarly, we can express sets having exactly k elements, for any finite k.

3 (Semantic)Consequence relation:-

Let Γ be a set of formulas and ϕ be a formula. We say that ϕ is a semantic consequence of Γ, if for all models $\mathcal{M}$, $\mathcal{M} \models$ v for all $v \in \Gamma$ imply $\mathcal{M} \models \phi$. It is denoted by $\Gamma \models \phi$.

Example:- $\Gamma = \{P^1x \rightarrow Q^1x, Q^1 \rightarrow R^1x\}$ and $\phi = \{P^1 \rightarrow R^1x\}$.

Example:-

1. **Γ=$\{P^1x \rightarrow Q^1x, Q^1 \rightarrow R^1x\}$ and $\phi = \{P^1 \rightarrow R^1x\}$**

   Proof. To show $\Gamma \models \phi$, we need to show for all models $\mathcal{M}$ if $\mathcal{M} \models \Gamma$ then $\mathcal{M} \models \phi$. Now take any model $\mathcal{M} : -(\mathcal{D}, \mathcal{I}, \mathcal{G})$. Suppose, $\mathcal{M} \models \Gamma$, that is suppose $\mathcal{M} \models P^1x \rightarrow Q^1x$ and $\mathcal{M} \models Q^1x \rightarrow R^1x$.

   To show $\mathcal{M} \models P^1x \rightarrow R^1x$. LEt us assume that $\mathcal{M} \models P^1x$. To show $\mathcal{M} \models R^1x$. Since, $\mathcal{M} \models P^1x \rightarrow Q^1x$ we have $\mathcal{M} \models Q^1x$. Now, since $\mathcal{M} \models Q^1x \rightarrow R^1x$, then we have $\mathcal{M} \models R^1x$. This completes the proof. □

2. Suppose that $\Gamma \models \Phi$, then when we have $\Gamma \models \phi$, we basically have $\Phi \models \phi$. It is denoted by $\models \phi$ and we say 'ϕ is satisfied by all models'.

4 validity and Satisfiability:-

1. A formula ϕ is said to be **valid** if for every model $\mathcal{M}, \mathcal{M} \models \phi$.

2. A formula ϕ is said to be **satisfiable** if there is a model $\mathcal{M}$, such that $\mathcal{M} \models \phi$.

Example of valid and satisfiable formulas:-

Consider a first-order language $\mathcal{L}$ with $\mathcal{C} = \Phi$, $\mathcal{F} = \Phi$ and $\mathcal{P} = P^2$. Also consider a structure $(\mathcal{D}, \mathcal{I})$ where $\mathcal{D}$ being a non-empty. Let us define three interpretations of P^2

1. $\mathcal{I}_1(P^2) = \mathcal{D} \times \mathcal{D} \iff \phi_1 : \forall x \forall y P^2 xy$.
2. $\mathcal{I}_2(P^2) = \Phi \iff \phi_2 : \forall x \forall y \neg P^2 xy$.
3. $\mathcal{I}_3(P^2) =$ a serial relation $\iff \phi_3 : \forall x \forall y \neg P^2 xy$.

Examples of valid and satisfiable formulas in $\mathcal{L}$

1. ∀x ∀y (P^2 xy ∨¬P^2 xy) : Valid.
2. ∀x ∀y (P^2 xy ∧¬P^2 xy): which is true in empty model.
3. ∃x ∃y (P^2 xy ∨¬P^2 xy): which is unsatisfiable.

Note: A serial relation for each d ∈ $\mathcal{D}$ there is d' ∈ $\mathcal{D}$ such that (d,d') ∈ $\mathcal{I}_3(P^2)$ = R say.

Exercise

1. Let $\Gamma = \{\phi, \not\phi\}$. Then show that $\Gamma \models \psi$ for all formulas ψ.
2. Show that if $\phi \in \Gamma$, then $\Gamma \models \phi$.
3. Let ϕ be a formula, then **prove or disprove that** ϕ is valid iff $/\phi$ is not satisfiable.
4. If $\Gamma_1 \subseteq \Gamma_2$ and $\Gamma_1 \models \phi$, then show that $\Gamma_2 \models \phi$.