15-819K: Logic Programming

Lecture 16

Substitution Semantics

Frank Pfenning

October 24, 2006

In this lecture we introduce a semantics with an explicit substitution, in preparation for presenting various program analyses later. The semantics will have the property that its goals appear exactly as in the original programs, with a separate substitution as part of the judgment. We also reviewed potential project areas during lecture, which is not represented in these notes.

16.1 Semantic Levels

When designing a program analysis we need to consider which level of semantic description is appropriate. This is relevant both for designing and proving the correctness of the analysis, which could be either simple or difficult, depending on our starting point. By the "level of semantic description" we mean here the spectrum from the logical semantics (in which we can only talk about truth), through one where subgoal order is explicit, to one with a failure continuation. An additional dimension is if a substitution is explicit in the semantics.

Which kind of semantics is appropriate, for example, for defining mode analysis? We would like to stay as high-level as possible, while still being able to express the property in question. Because mode analysis depends on subgoal order, one would expect to make subgoal order explicit. Moreover, since groundedness is a property of the substitution that is applied, we also should make a substitution explicit during computation. On the other hand, modes do not interact with backtracking, so we do not expect to need a failure continuation.

We take here a slight shortcut, using a semantics with an explicit substitution, but *not* a subgoal stack. Of course, it is possible to give such a

semantics as well. Omitting the subgoal stack has the advantage that we can relatively easily talk about the successful return of a predicate.

16.2 A Substitution Semantics

The semantics we give takes a goal G under a substitution τ. It produces a substitution θ with the invariant that $G\tau\theta\sigma$ for any grounding substitution σ. We define it on the fully residuated form, where for every predicate p there is exactly one clause, and this clause has the form $\forall \mathbf{x}.\, p(\mathbf{x}) \leftarrow G$.

In the judgment $\tau \vdash G \;/\; \theta$ we maintain the invariant that $\mathrm{dom}(\tau) \supseteq \mathrm{FV}(G)$ and that $G\tau\theta \;\; true$ where we mean that there is a proof parametric in the remaining free variables. Moreover, θ substitutes only for logic variables X.

An important point[1] is that τ should substitute *exactly* for the originally quantified variables of G, and *not* for logic variables introduced during the computation. This is the role of θ which substitutes *only* for logic variables. The rule for atomic predicates is one place where this is important.

$$\frac{(\forall \mathbf{x}.\, p(\mathbf{x}) \leftarrow G) \in \Gamma \quad \mathbf{t}\tau/\mathbf{x} \vdash G \;/\; \theta}{\tau \vdash p(\mathbf{t}) \;/\; \theta}$$

We see that, indeed, the substitution on the premise accounts for all variables in G by the assumption of a closed normal form for programs.

The rule for conjunction presumes a subgoal order via the threading of θ_1, without using a subgoal stack.

$$\frac{\tau \vdash G_1 \;/\; \theta_1 \quad \tau[\theta_1] \vdash G_2 \;/\; \theta_2}{\tau \vdash G_1 \wedge G_2 \;/\; \theta_1\theta_2}$$

Here we have used a variant of the composition operator in order to maintain our invariant on the input substitution. $\tau[\theta_1]$ applies θ_1 to every element of τ, but does not extend it. That is,

$$(t_1/x_1, \ldots, t_n/x_n)[\theta] = (t_1[\theta]/x_1, \ldots, t_n[\theta]/x_n)$$

Truth is straightforward, as are the rules for disjunction and falsehood.

$$\frac{}{\tau \vdash \top \;/\; (\cdot)} \qquad \frac{\tau \vdash G_1 \;/\; \theta}{\tau \vdash G_1 \vee G_2 \;/\; \theta} \qquad \frac{\tau \vdash G_2 \;/\; \theta}{\tau \vdash G_1 \vee G_2 \;/\; \theta} \qquad \begin{array}{c}\text{no rule for} \\ \tau \vdash \bot \;/\; \_\end{array}$$

[1] I missed this point in lecture, which is why the system I gave did not work quite as well to prove the correctness of mode analysis.

The existential quantifier introduces a fresh logic variable X. This logic variable can somehow "escape" in that it may occur in the domain or co-domain θ. Intuitively, this makes sense because a logic variable that is not instantiated during the solution of G will remain after success.

$$\frac{X \notin \mathrm{FV}(\tau) \quad \tau, X/x \vdash G \mid \theta}{\tau \vdash \exists x.\, G \mid \theta}$$

Finally, equality reduces to unification.

$$\frac{\mathbf{t}\tau \doteq \mathbf{s}\tau \mid \theta}{\tau \vdash \mathbf{t} \doteq \mathbf{s} \mid \theta}$$

16.3 Correctness

The substitution semantics from the previous section is sound and complete in relation to the logical semantics of truth. First, the soundness. As remarked above, truth of a proposition with free variables is defined parametrically. That is, there must be one deduction with free variables every ground instance of which is true under the usual ground interpretation.

Theorem 16.1 *If $\tau \vdash G \mid \theta$ for $\mathrm{dom}(\tau) \supseteq \mathrm{FV}(G)$ then $G\tau\theta$ true.*

Proof: By induction on the deduction $\mathcal{D}$ of $\tau \vdash G \mid \theta$. For unification, we rely on soundness of unification. □

Completeness follows the usual pattern of lifting to deduction with free variables.

Theorem 16.2 *If $G\tau\sigma$ true where $\mathrm{dom}(\tau) \supseteq \mathrm{FV}(G)$ and $\mathrm{cod}(\sigma) = \emptyset$ then $\tau \vdash G \mid \theta$ and $\sigma = \theta\sigma'$ for some θ and σ'.*

Proof: By induction on the deduction of $G\tau\sigma$ θ. For unification we invoke the property that unification returns a most general unifier. □

16.4 An Asynchronous Substitution Semantics

Instead of giving substitution on goals for the residuated semantics, we can also give it directly on programs and goals if a normal form for programs is not desired or needed. There will be two judgments: $\tau \vdash A \mid \theta$ where A functions as a goal, and $\tau; A \ll P \mid \theta$ where A is formula under focus.

$$\frac{\tau \vdash A_1 \;/\; \theta_1 \quad \tau[\theta_1] \vdash A_2 \;/\; \theta_2}{\tau \vdash A_1 \wedge A_2 \;/\; \theta_1\theta_2} \qquad \frac{}{\tau \vdash \top \;/\; (\cdot)}$$

$$\text{omitted here}\quad \tau \vdash A_1 \supset A_2 \;/\; \_ \qquad \text{omitted here}\quad \tau \vdash \forall x.\, A \;/\; \_$$

$$\frac{\tau; A \ll P \;/\; \theta \quad A \in \Gamma}{\tau \vdash P \;/\; \theta}$$

$$\frac{\tau; A_1 \ll P \;/\; \theta}{\tau; A_1 \wedge A_2 \ll P \;/\; \theta} \qquad \frac{\tau; A_2 \ll P \;/\; \theta}{\tau; A_1 \wedge A_2 \ll P \;/\; \theta} \qquad \text{no rule for}\quad \tau; \top \ll P \;/\; \_$$

$$\frac{X \notin \mathrm{FV}(\tau) \quad (\tau, X/x); A \ll P \;/\; \theta}{\tau; \forall x.\, A \ll P \;/\; \theta} \qquad \frac{P'\tau \doteq P\tau \mid \theta}{\tau; P' \ll P \;/\; \theta}$$

$$\frac{\tau; A_1 \ll P \;/\; \theta_1 \quad \tau[\theta_1] \vdash A_2 \;/\; \theta_2}{\tau; A_2 \supset A_1 \ll P \;/\; \theta_1\theta_2}$$

We have not treated here implication and universal quantification as a goal. Implication is straightforward (see Exercise 16.1). Universal quantification in goals (which we have mostly avoided so far) creates difficulties for unification and is left to a future lecture.

The correctness theorem for this version of the semantics is left to Exercise 16.2.

16.5 Exercises

Exercise 16.1 *Extend the substitution semantics to permit dynamic assumptions* Γ *and goals of the form* $A_1 \supset A_2$*. Take care to account for the possibility that that dynamic assumptions may contain free variables.*

Exercise 16.2 *State and prove the correctness theorems for the asynchronous substitution semantics.*