CS323: AI (Examples on Procedural Interpretation.) **Spring 2016**

Lecture 7: January 20, 2016

Lecturer: K.R. Chowdhary *: Professor of CS (VF)*

Disclaimer: *These notes have not been subjected to the usual scrutiny reserved for formal publications. They may be distributed outside this class only with the permission of the Instructor.*

7.1 Procedural Interpretation Examples

Example 7.1 *Appending two lists.*

Let a term $cons(x, y)$ is interpreted as a list whose first element, the *head*, is x and whose *tail* y is the rest of the list. The constant nil denotes the empty list. The terms u, x, y, and z are variables. The predicate *append(x,y,z)* denotes the relationship: z is obtained by appending y to x.

The following two clauses constitute a program for appending two lists.

$$append(nil, x, x). \tag{7.1}$$

$$append(cons(x, y), z, cons(x, u)) \vee \overline{append}(y, z, u). \tag{7.2}$$

The clause in statement (7.1) represents halt statement. In (7.2) there is a positive literal for procedure name, and negative literal(s) for the procedure body, both together it is procedure declaration. The positive literal means, if $cons(x, y)$ is appended with z, it results to x appended with u such that u is, y appended with z. The later part is indicated by the complementary (negative) term. Note that clausal expression (7.2) is logically equivalent to the expression $append(y, z, u) \rightarrow append(cons(x, y),\, z, cons(x, u))$.

Suppose it is required to compute the result of appending list $cons(b, nil)$ to the list $cons(a, nil)$. Therefore, the goal statement is,

$$append(cons(a, nil), cons(b, nil), v), \tag{7.3}$$

where v (a variable) and a, b (constants), are the "atoms" of the lists. To prove using resolution, we add the negation of the goal,

$$\overline{append}(cons(a, nil), cons(b, nil), v), \tag{7.4}$$

into the set of clauses. The program is activated by this *goal statement* to carry out the append operation. With this goal statement the program is deterministic, because only one choice is available for matching. The following computation follows with a goal directed theorem prover as interpreter: The goal statement,

$$C_1 = \overline{append}(cons(a, nil), cons(b, nil), v). \tag{7.5}$$

matches with the clause statement (7.2) with matchings: $x = a, y = nil, z = cons(b, nil)$. Also, $v = cons(x, u) = cons(a, u)$, i.e., there exists a unifier $\theta_1 = \{cons(a, w)/v\}$. The variable u has been renamed as w. On unifying clauses (7.5) and (7.2), the next computation C_2 is :

$$C_2 = \overline{append}(nil, cons(b, nil), w)\theta_1. \tag{7.6}$$

Keeping θ_1 accompanying the predicate in above is for the purpose that if C_2 is to be unified with some other predicate, the matching of the two shall be subject to the same unifier θ_1.

As next matching, C_2 can be unified with (7.1) using a new unifier $\theta_2 = \{cons(b, nil)/w\}$ to get next computation,

$$C_3 = []\theta_2. \tag{7.7}$$

The result of the computation is value of v in the substitution, i.e.,

$$\begin{aligned} v &= cons(a, u) \\ &= cons(a, w) \\ &= cons(a, cons(b, nil)). \end{aligned}$$

The above result is equal to goal: $append(cons(a, nil), cons(b, nil), v)$. □