

## Lecture 10: January 23, 2014

Lecturer: K.R. Chowdhary

: Professor of CS (GF)

**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.*

## 10.1 Substitutions and Unification

The following definitions are concerned with the operation of *instantiation*, i.e., substitutions of terms for variables in the well-formed expressions and in sets of well-formed expressions.

**Substitution Components.** A substitution component is any expression of the form  $T/V$ , where  $V$  is any variable and  $T$  is any term different from  $V$ .  $V$  is called variable of component  $T/V$ , and  $T$  is called term of the component.

**Substitutions.** A substitution is any finite set (possibly empty) of substitution components, none of the variables of which are same. If  $P$  is any set of terms, and the terms of the components of the substitution  $\theta$  are all in  $P$ , we say that  $\theta$  is a substitution over  $P$ . We write the substitution where components are  $T_1/V_1, \dots, T_k/V_k$  as  $\{T_1/V_1, \dots, T_k/V_k\}$ , with the understanding that order of components is immaterial. We will use lowercase Greek letters  $\theta, \lambda, \mu$  denote substitutions.

**Instantiations.** If  $E$  is any function string of symbols, and  $\theta = \{T_1/V_1, \dots, T_k/V_k\}$  is any substitution, then the instantiation of  $E$  by  $\theta$  is the operation of replacing each occurrence of variable  $V_i, 1 \leq i \leq k$ , in  $E$  by occurrence of the term  $T_i$ . The resulting string denoted by  $E\theta$  is called an instance of  $E$  by  $\theta$ , i.e., if  $E$  is the string  $E_0V_{i_1}E_1 \dots V_{i_n}En$ , then  $E\theta$  is the string  $E_0T_{i_1}E_1 \dots T_{i_n}En$ . Here, none of the substrings  $E_j$  of  $E$  contain occurrences of variables  $V_1, \dots, V_k$ , some of  $E_j$  are possibly *null*,  $n$  is possibly 0, and each  $V_{i_j}$  is an occurrence of one of the variables  $V_1, \dots, V_k$ . Any string  $E\theta$  is called an instance of the string  $E$ .

If  $C$  is any set of strings and  $\theta$  is substitution, then the instance of  $C$  by  $\theta$  is the set of all strings  $E\theta$ , where  $E$  is in  $C$ . We denote this set by  $C\theta$ , and say that it is an instance of  $C$ .

**Composition of Substitutions.** If  $\theta = \{T_1/V_1, \dots, T_k/V_k\}$  and  $\lambda$  are any two substitutions, then the composition of  $\theta$  and  $\lambda$  denoted by  $\theta\lambda$  is union  $\theta' \cup \lambda'$ , defined as follows:

The  $\theta'$  is set of all components  $T_i\lambda/V_i, 1 \leq i \leq k$ , such that  $T_i\lambda$  ( $\lambda$  substituted in  $\theta$ ) is different from  $V_i$ , and  $\lambda'$  is set of all components of  $\lambda$  whose variables are not among  $V_1, \dots, V_k$ .

**Example 10.1** Find out the composition of  $\{x/y, w/z\}, \{v/x\}$ , and  $\{A/v, f(B)/w\}$ .

**Solution.** Let us assume that  $\theta = \{x/y, w/z\}$ ,  $\lambda = \{v/x\}$  and  $\mu = \{A/v, f(B)/w\}$ . To find the composition  $\lambda\mu$ ,  $A$  is substituted for  $v$ , and  $v$  is then substituted for  $x$ . Thus,  $\lambda\mu = \{A/x, f(B)/w\}$ . When result of  $\lambda\mu$  is substituted in  $\theta$ , we get composition  $\theta\lambda\mu = \{A/y, f(B)/z\}$ .  $\square$

**Example 10.2** Find out the composition of  $\{g(x, y)/z\}$ , and  $\{A/x, B/y, C/w, D/z\}$ .

**Solution.**  $\{g(A, B)/z, A/x, B/y, C/w\}$ .  $\{D/z\}$  has not been included in the resultant substitution set, because otherwise, there will be two terms for the variable  $z$ , one  $g(A, B)$  and other  $D$ .  $\square$

One of the important property of substitution is that, if  $E$  is any string, and  $\sigma = \theta\lambda$ , then  $E\sigma = E\theta\lambda$ . It is straight forward to verify that  $\epsilon\theta = \theta\epsilon = \theta$  for any substitution  $\theta$ . Also, composition enjoys the associative property  $(\theta\lambda)\mu = \theta(\lambda\mu)$ , so we may omit the parentheses in writing multiple compositions of substitutions. The substitutions are not in general commutative; i.e., it is generally not the case that  $\theta\lambda = \lambda\theta$ .

The point of the composition operation on substitution is that, when  $E$  is any string, and  $\sigma = \theta\lambda$ , the string  $E\sigma$  is just the string  $E\theta\lambda$ , i.e., the instance of  $E\theta$  by  $\lambda$ . The composition also have distributive property.

### 10.1.1 Unification

If  $A$  is any set of well-formed expressions and  $\theta$  is a substitution, then  $\theta$  is said to unify  $A$ , or to be a unifier of  $A$ , if  $A\theta$  is singleton. Any set of well-formed expressions which has a unifier is said to be unifiable.

For Unification, a variable can be replaced by any term, including other variables and function expressions of arbitrary complexity, even including expressions that themselves contain variables. For example, the function expression,  $father(jack)$ , may be substituted for  $x$  in  $human(x)$  to get  $human(father(jack))$ . Following are some valid substitution instances of  $foo(x, a, zoo(y))$  :

1.  $foo(fred, a, zoo(z))$ , where  $fred$  is substituted for  $x$  and  $z$  for  $y$ , i.e.,  $\lambda_1 = \{fred/x, z/y\}$ . Thus,

$$foo(fred, a, zoo(z)) = foo(x, a, zoo(y))\lambda_1.$$

2.  $foo(w, a, zoo(jack))$ , where  $\lambda_2 = \{w/x, jack/y\}$ ; hence

$$foo(w, a, zoo(jack)) = foo(x, a, zoo(y))\lambda_2.$$

3.  $foo(z, a, zoo(moo(z)))$ , where  $\lambda_3 = \{z/x, moo(z)/y\}$ , hence;

$$foo(z, a, zoo(moo(z))) = foo(x, a, zoo(y))\lambda_3.$$

We use the notation  $x/y$  to indicate that  $x$  is substituted for the variable  $y$ . We also refer to these as bindings, so  $y$  is bound to  $x$ . A variable cannot be unified with a term containing that variable. So  $p(x)$  cannot be substituted for  $x$ , because this would create an infinite regression:  $p(p(p(\dots x)\dots))$ .

If  $S$  is any unifier of a set of expressions  $E$ , and  $G$  is the most general unifier (i. e., simplest one) of that set of expressions, then for  $S$  applied to  $E$  there exists another unifier  $S'$  such that  $ES = EGS'$  (where  $GS'$  is the composition of unifiers). In simple words, a most general unifier *mgu* is just what its name implies: the most general set of substitutions that can be found for the two expressions.

**Example 10.3** Given a unifier, obtain a more general unifier.

Suppose you have two expressions  $p(x)$  and  $p(y)$ . One way to unify these is to substitute any constant expression for  $x$  and  $y$ :  $\{fred/x, fred/y\}$ . But this is not the most general unifier, because if we substitute

any variable for  $x$  and  $y$ , we get a more general unifier:  $\{z/x, z/y\}$ . The first unifier is a valid unifier, but it would lessen the generality of inferences that we might want to make.

Let,  $E = \{p(x), p(y)\}$

Let,  $S = \{fred/x, fred/y\}$

Let,  $G = \{z/x, z/y\}$

Now, let  $S' = \{fred/z\}$

Then  $ES = \{p(fred), p(fred)\}$

and  $GS' = \{fred/x, fred/y\}$

and therefore  $EGS' = \{p(fred), p(fred)\} = ES$ .  $\square$

So, given a unifier, you can always create a more general unifier. When both of these unifiers are composed and instantiate the expression  $E$ , you get the same instance as it was obtained with the earlier unifier.

## References

- [RN05] S. RUSSEL AND P. NORVIG, "Artificial Intelligence - A Modern Approach," *2nd Edition, Pearson Education, India*, 2005, Chapter 9.
- [GFL09] D. GEORGE F. LUDGER, "Artificial Intelligence - Structures and Strategies for Complex Problem Solving," *5th Edition, Pearson Education, India*, 2009, Chapter 2.
- [PAT01] DAN W. PATTERSON, "Introduction to Artificial Intelligence and Expert Systems," *PHI India*, 2001, Chapter 4.
- [RIC02] ELAN RICH AND KEVIN KNIGHT, "Artificial Intelligence," 2nd Edition, Tata McGraw-Hill, India, 2002, Chapter 5
- [NIL80] NILS J. NILSSON, "Principles of Artificial Intelligence," Narosa Publishers, India, 1980, Chapter 4, 5.