16.3.1 The HOL deductive system

The deductive system of the HOL logic is specified by eight rules of inference, given below. The first three rules have no hypotheses; their conclusions can always be deduced. The identifiers in square brackets are the names of the ML functions in the HOL system that implement the corresponding inference rules (See Section 18.4). Any side conditions restricting the scope of a rule are given immediately below it.

Assumption introduction [ASSUME]

$$\frac{}{t \vdash t}$$

Reflexivity [REFL]

$$\frac{}{\vdash t = t}$$

Beta-conversion [BETA_CONV]

$$\frac{}{\vdash (\lambda x.\ t_1)t_2 = t_1[t_2/x]}$$

- Where $t_1[t_2/x]$ is the result of substituting t_2 for x in t_1, with suitable renaming of variables to prevent free variables in t_2 becoming bound after substitution.

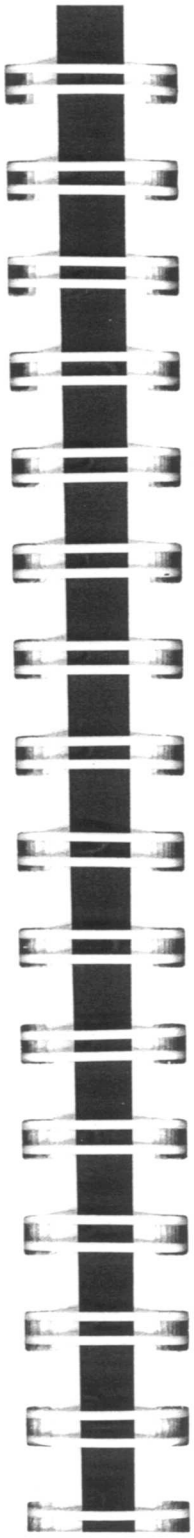
Substitution [SUBST]

$$\frac{\Gamma_1 \vdash t_1 = t_1' \quad \cdots \quad \Gamma_n \vdash t_n = t_n' \qquad \Gamma \vdash t[t_1, \ldots, t_n]}{\Gamma_1 \cup \cdots \cup \Gamma_n \cup \Gamma \vdash t[t_1', \ldots, t_n']}$$

- Where $t[t_1, \ldots, t_n]$ denotes a term t with some free occurrences of subterms t_1, ..., t_n singled out and $t[t_1', \ldots, t_n']$ denotes the result of replacing each selected occurrence of t_i by t_i' (for $1 \leq i \leq n$), with suitable renaming of variables to prevent free variables in t_i' becoming bound after substitution.

Abstraction [ABS]

$$\frac{\Gamma \vdash t_1 = t_2}{\Gamma \vdash (\lambda x.\ t_1) = (\lambda x.\ t_2)}$$



Type instantiation [INST_TYPE]

$$\frac{\Gamma \vdash t}{\Gamma \vdash t[\sigma_1, \ldots, \sigma_n/\alpha_1, \ldots, \alpha_n]}$$

- Where $t[\sigma_1, \ldots, \sigma_n/\alpha_1, \ldots, \alpha_n]$ is the result of substituting, in parallel, the types σ_1, ..., σ_n for type variables α_1, ..., α_n in t, with the two restrictions: (i) none of the type variables α_1, ..., α_n occur in Γ; and (ii) no distinct variables in t become identified after the instantiation.[2]

Discharging an assumption [DISCH]

$$\frac{\Gamma \vdash t_2}{\Gamma - \{t_1\} \vdash t_1 \Rightarrow t_2}$$

- Where $\Gamma - \{t_1\}$ is the set subtraction of $\{t_1\}$ from Γ.

Modus Ponens [MP]

$$\frac{\Gamma_1 \vdash t_1 \Rightarrow t_2 \qquad \Gamma_2 \vdash t_1}{\Gamma_1 \cup \Gamma_2 \vdash t_2}$$

In addition to these eight rules, there are also five *axioms* which could have been regarded as rules of inference without hypotheses. This is not done, however, since it is most natural to state the axioms using some defined logical constants and the principle of constant definition has not yet been described. The axioms are given in Section 16.4.3 and the definitions of the extra logical constants they involve are given in Section 16.4.2.

The particular set of rules and axioms chosen to axiomatize the HOL logic is rather arbitrary. It is partly based on the rules that were used in the LCF logic PPλ, since HOL was implemented by modifying the LCF system. In particular, the substitution rule SUBST is exactly the same as the corresponding rule in LCF; the code implementing this was written by Robin Milner and is highly optimized. Because substitution is such a pervasive activity in proof, it was felt to be important that the system primitive be as fast as possible. From a logical point of view, it would be better to have a simpler substitution primitive, such as 'Rule R' of Andrews' logic $\mathcal{Q}_0$, and then to derive more complex rules from it.

[2] The ML function implementing INST_TYPE in the HOL system fails if side condition (i) is violated, but instead of failing if (ii) is violated, it automatically renames any variable whose type is instantiated if the variable is preceded in t by a different variable with the

16.3.1 The HOL deductive system

The deductive system of the HOL logic is specified by eight rules of inference, given below. The first three rules have no hypotheses; their conclusions can always be deduced. The identifiers in square brackets are the names of the ML functions in the HOL system that implement the corresponding inference rules (See Section 18.4). Any side conditions restricting the scope of a rule are given immediately below it.

Assumption introduction [ASSUME]

$$\frac{}{t \vdash t}$$

Reflexivity [REFL]

$$\frac{}{\vdash t = t}$$

Beta-conversion [BETA_CONV]

$$\frac{}{\vdash (\lambda x.\ t_1)t_2 = t_1[t_2/x]}$$

- Where $t_1[t_2/x]$ is the result of substituting t_2 for x in t_1, with suitable renaming of variables to prevent free variables in t_2 becoming bound after substitution.

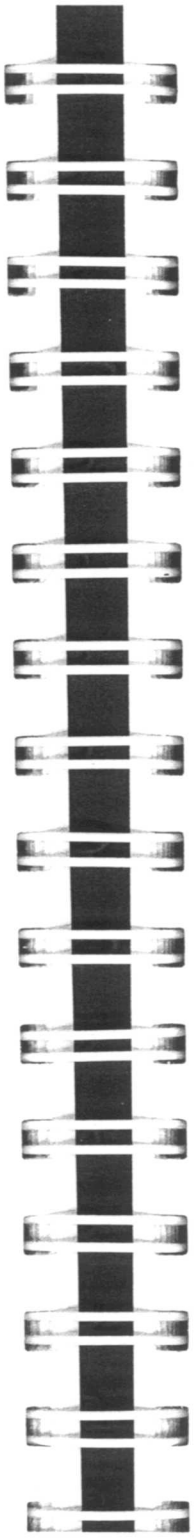
Substitution [SUBST]

$$\frac{\Gamma_1 \vdash t_1 = t_1' \quad \cdots \quad \Gamma_n \vdash t_n = t_n' \qquad \Gamma \vdash t[t_1, \ldots, t_n]}{\Gamma_1 \cup \cdots \cup \Gamma_n \cup \Gamma \vdash t[t_1', \ldots, t_n']}$$

- Where $t[t_1, \ldots, t_n]$ denotes a term t with some free occurrences of subterms t_1, ... , t_n singled out and $t[t_1', \ldots, t_n']$ denotes the result of replacing each selected occurrence of t_i by t_i' (for $1 \leq i \leq n$), with suitable renaming of variables to prevent free variables in t_i' becoming bound after substitution.

Abstraction [ABS]

$$\frac{\Gamma \vdash t_1 = t_2}{\Gamma \vdash (\lambda x.\ t_1) = (\lambda x.\ t_2)}$$



Type instantiation [INST_TYPE]

$$\frac{\Gamma \vdash t}{\Gamma \vdash t[\sigma_1, \ldots, \sigma_n/\alpha_1, \ldots, \alpha_n]}$$

- Where $t[\sigma_1, \ldots, \sigma_n/\alpha_1, \ldots, \alpha_n]$ is the result of substituting, in parallel, the types σ_1, ... , σ_n for type variables α_1, ... , α_n in t, with the two restrictions: (i) none of the type variables α_1, ... , α_n occur in Γ; and (ii) no distinct variables in t become identified after the instantiation.[2]

Discharging an assumption [DISCH]

$$\frac{\Gamma \vdash t_2}{\Gamma - \{t_1\} \vdash t_1 \Rightarrow t_2}$$

- Where $\Gamma - \{t_1\}$ is the set subtraction of $\{t_1\}$ from Γ.

Modus Ponens [MP]

$$\frac{\Gamma_1 \vdash t_1 \Rightarrow t_2 \qquad \Gamma_2 \vdash t_1}{\Gamma_1 \cup \Gamma_2 \vdash t_2}$$

In addition to these eight rules, there are also five *axioms* which could have been regarded as rules of inference without hypotheses. This is not done, however, since it is most natural to state the axioms using some defined logical constants and the principle of constant definition has not yet been described. The axioms are given in Section 16.4.3 and the definitions of the extra logical constants they involve are given in Section 16.4.2.

The particular set of rules and axioms chosen to axiomatize the HOL logic is rather arbitrary. It is partly based on the rules that were used in the LCF logic PPλ, since HOL was implemented by modifying the LCF system. In particular, the substitution rule SUBST is exactly the same as the corresponding rule in LCF; the code implementing this was written by Robin Milner and is highly optimized. Because substitution is such a pervasive activity in proof, it was felt to be important that the system primitive be as fast as possible. From a logical point of view, it would be better to have a simpler substitution primitive, such as 'Rule R' of Andrews' logic $\mathcal{Q}_0$, and then to derive more complex rules from it.

[2] The ML function implementing INST_TYPE in the HOL system fails if side condition (i) is violated, but instead of failing if (ii) is violated, it automatically renames any variable whose type is instantiated if the variable is preceded in t by a different variable with the