

for by a verb may appear in the verb phrase or may be realized apart from the verb, as a **long-distance dependency**.

- Feature structures can be **typed**. The resulting **typed feature structures** place constraints on which type of values a given feature can take; they can also be organized into a **type hierarchy** to capture generalizations across types.

Bibliographical and Historical Notes

The use of features in linguistic theory comes originally from phonology. Anderson (1985) credits Jakobson (1939) with being the first to use features (called **distinctive features**) as an ontological type in a theory, drawing on previous uses of features by Trubetskoi (1939) and others. The semantic use of features followed soon after; see Chapter 19 for the history of componential analysis in semantics. Features in syntax were well established by the 1950s and were popularized by Chomsky (1965).

The unification operation in linguistics was developed independently by Kay (1979) (feature structure unification) and Colmerauer (1970, 1975) (term unification) (see page 47). Both were working in machine translation and looking for a reversible formalism for combining linguistic information. Colmerauer's original Q-system was a bottom-up parser, based on a series of rewrite rules that contained logical variables, designed for an English-to-French machine translation system. The rewrite rules were reversible to allow them to work for both parsing and generation. Colmerauer, Fernand Didier, Robert Pasero, Philippe Roussel, and Jean Trudel designed the Prolog language, based on extending Q-systems to full unification using the resolution principle of Robinson (1965), and implemented a French analyzer based on it (Colmerauer and Roussel, 1996). The modern use of Prolog and term unification for natural language with **Definite Clause Grammars** was based on Colmerauer's (1975) metamorphosis grammars, and was developed and named by Pereira and Warren (1980). Meanwhile, Martin Kay and Ron Kaplan had been working with Augmented Transition Network (**ATN**) grammars. An ATN is a Recursive Transition Network (RTN) in which the nodes are augmented with feature registers. In an ATN analysis of a passive, the first NP would be assigned to the subject register, then when the passive verb was encountered, the value would be moved into the object register. To make this process reversible, they restricted assignments to registers so that certain registers could only be filled once, that is, couldn't be overwritten once written. They thus moved toward the concepts of logical variables without realizing it. Kay's original unification algorithm was designed for feature structures rather than terms (Kay, 1979). The integration of unification into an Earley-style approach given in Section 15.5 is based on Shieber (1985b).

See Shieber (1986) for a clear introduction to unification, and Knight (1989) for a multidisciplinary survey of unification.

*Definite Clause
Grammars*

ATN