FCA$_{\text{Type}}$ – a System for Type Signature Induction

Wiebke Petersen

Institute of Language and Information
Heinrich-Heine-Universität Düsseldorf
petersew@uni-duesseldorf.de

Abstract. Type signatures are common in modern linguistic theories. Their construction and maintenance is intricate, and therefore, an automatic induction method is desirable. We present FCA$_{\text{Type}}$ a system that automatically induces type signatures from so-called decomposition lattices of sets of untyped feature structures.

It is common in linguistic formalisms to represent lexical information in the form of recursively built attribute-value-matrices, called *feature structures* (see Fig. 1). In order to organize the lexicon, to avoid redundancy, and to capture generalizations, a strict type discipline has been developed (see [1]): Types are assigned to the feature structures and all their embedded structures. The types are organized in a hierarchy which is enriched by *appropriateness conditions*, stating for each type the appropriate attributes and restricting their values by prescribing the most general type they may belong to. A *type signature* is a type hierarchy enriched by appropriateness conditions. A feature structure is considered *totally well-typed* w.r.t. a type signature if none of its attribute-value pairs conflicts with the appropriateness conditions and if the structure contains every pair which is prescribed by the appropriateness conditions.

Linguistic feature structures which encode all the necessary phonological, morphological, syntactic, and semantic information of a lexical entry are huge, and type signatures which cover generalizations about such feature structures become extremely complex. A system (like FCA$_{\text{Type}}$) which induces a type signature from a set of untyped feature structures supports grammar development. The key idea of FCA$_{\text{Type}}$ is to construct the *decomposition lattice* to a set of untyped feature structures. The decomposition lattice consists of (1) the feature structures themselves, (2) the embedded substructures, and (3) all most specific generalizations about structures from (1) and (2). All those structures are partially ordered by the specifity order (subsumption order) as in Fig. 2. The

$$\text{Uther} = \begin{bmatrix} \text{CAT} & \text{np} \\ \text{HEAD} & \begin{bmatrix} \text{AGR} & \begin{bmatrix} \text{PERS} & \text{third} \\ \text{NUM} & \text{sing} \end{bmatrix} \end{bmatrix} \end{bmatrix} \qquad \text{sleeps} = \begin{bmatrix} \text{CAT} & \text{vp} \\ \text{HEAD} & \begin{bmatrix} \text{FORM} & \text{finite} \\ \text{SUBJ} & \begin{bmatrix} \text{CAT} & \text{np} \\ \text{HEAD} & \begin{bmatrix} \text{AGR} & \begin{bmatrix} \text{PERS} & \text{third} \\ \text{NUM} & \text{sing} \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

$$\text{knights} = \begin{bmatrix} \text{CAT} & \text{np} \\ \text{HEAD} & \begin{bmatrix} \text{AGR} & \begin{bmatrix} \text{PERS} & \text{third} \\ \text{NUM} & \text{plur} \end{bmatrix} \end{bmatrix} \end{bmatrix} \qquad \text{sleep} = \begin{bmatrix} \text{CAT} & \text{vp} \\ \text{HEAD} & \begin{bmatrix} \text{FORM} & \text{finite} \\ \text{SUBJ} & \begin{bmatrix} \text{CAT} & \text{np} \\ \text{HEAD} & \begin{bmatrix} \text{AGR} & \begin{bmatrix} \text{PERS} & \text{third} \\ \text{NUM} & \text{plur} \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

Fig. 1: Example lexicon with small untyped feature structures

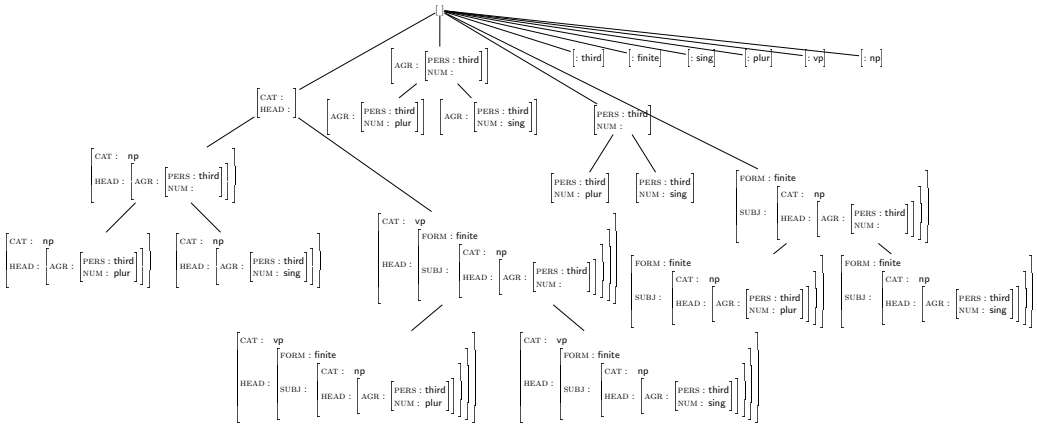

Fig. 2: The decomposition lattice for the structures of Fig. 1

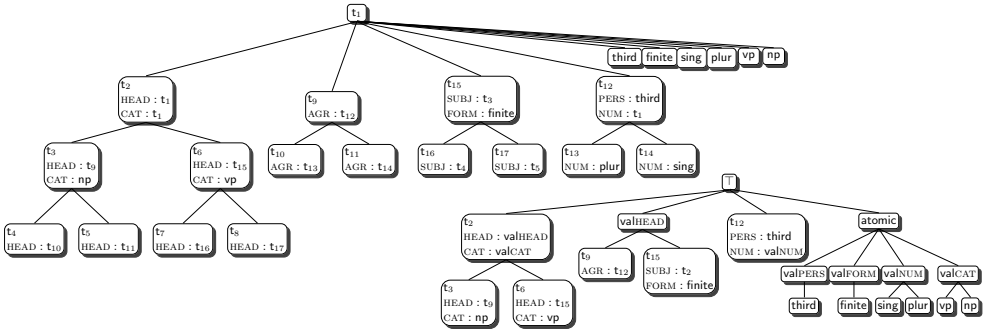

Fig. 3: Unfolded type signature (left) and maximally folded, value-controlled type signature (right) for the structures of Fig. 1 (each box shows the type label in the first line followed by the appropriateness conditions)

decomposition lattice can be straightforwardly transformed into a well-formed type signature. Fig. 3(left) shows the resulting type signature for the input data of Fig. 1. Each input structure can be typed in such a way that it becomes totally well-typed w.r.t. this type signature.

However, the type signature in Fig. 3(left) still has two undesirable properties: First, the induced appropriateness conditions are not restrictive enough (e.g. the appropriateness condition 'NUM:t_1' permits that the attribute NUM takes a complex feature structure of type t_2 as value). Second, some types are superfluous (the set of totally well-typed feature structures would not change substantially if the types t_4,t_5,t_7,t_8,t_{10},t_{11},t_{16}, and t_{17} were deleted). The first problem is solved by adding additional types to control the values and the second one by 'folding up' the signature. Fig. 3(right) shows the maximally folded, value-controlled type signature induced by FCAType from the input data of Fig. 1. A detailed description of the induction process will be given in [2].

References

1. Carpenter, B.: The Logic of Typed Feature Structures. CUP (1992)
2. Petersen, W.: Induktion von Typsignaturen mit Mitteln der Formalen Begriffsanalyse. PhD thesis (in preparation)