

Using Charts for Transfer in MT

Jan W. Amtrup

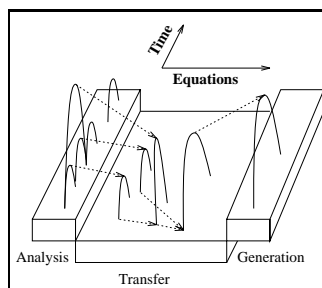
University of Hamburg, CS Dept., Natural Language Division, Vogt-Kölln-Str. 30,
D-22527 Hamburg. email: amtrup@informatik.uni-hamburg.de

Abstract. We present an adaption of chart-parsing techniques to the application of transfer in machine translation. The resulting flexibility enables the investigation of several architectural features when integrating a transfer stage into an incremental, interactive architecture.

The architecture for an automatic interpreting system capable of translating spontaneously produced dialogue contributions has to obey several restrictions concerning the overall design and the conception of individual components. The features include *incrementality*, *component interaction*, and — if possible — *time synchronicity* and *parallelization*. It is highly desirable for a transfer module to follow these restrictions in order to augment system usability and behaviour.

Charts have mainly been used for parsing. The ability to store incompletely analyzed constituents as well as the great flexibility implemented by means of an *agenda* enable chart parsing mechanisms to investigate several choices of parsing and search strategies.

We adapted the chart schema for transfer to benefit from these properties. *Active edges* correspond to partially translated source language constituents, the fundamental rule combines two partial translations to form a more complete one thus yielding a tree of transfer edges rooted at every inactive analysis edge. The result is a twodimensional structure of time and solved transfer equations. Transfer rules are formulated within a typed feature structure formalism and carry out structural transfer on a syntactic level.



The main features of the approach are: Translations can be *reused*, they don't have to be reanalyzed from scratch when backtracking. The transfer mechanism works *incrementally*, which enables implementation of e.g. left-to-right strategies or best-first approaches. In principle, there are no barriers which prevent interaction between modules to reduce ambiguity, which is an ultimate goal of an incremental, interactive architecture. It is also possible to parallelize the transfer stage using well-known techniques from chart parsing.

A first experimentation system using Lisp and CLOS has been built. Subsequent work will concentrate on the integration of a more elaborate transfer concept into an architectural prototype which follows the architectural lines as indicated above.