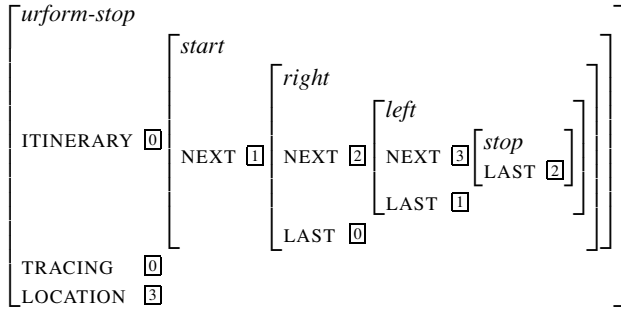


(38) *The first solution*

This structure differs from the original query structure in only two points:

- 1) The root type has been replaced by the type *urform-stop*, which results from the unification of the root types of the query *urform* and the *urform-stop* constraint.
- 2) The type at path *LOCATION*, which is the same as the one at path *ITINERARY|NEXT|NEXT|NEXT*, has been replaced with the type *stop*, which results from the unification of *limit* and *stop*, the first being the original value of the query structure at path *LOCATION*, the second the value of the same path of the *urform-stop* constraint.

The sequence encoded in this structure is $\langle \text{start}, \text{right}, \text{left}, \text{stop} \rangle$, which corresponds to the first image of the Urform language:

(39) *Urform I*

Having demonstrated how the token sequence corresponding to the first solution can be generated, the paper will now explain in more detail the process responsible for generating solutions (*resolution*).

3.3 Resolution

The TH and TCs of a constraint system can be understood as a kind of knowledge base: Questions to this knowledge base can be posed in the form of *queries*, which are answered by applying the knowledge stored in the knowledge base. The mechanism responsible for this procedure is called *resolution*.

The resolution algorithm of the constraint system used for this paper is simple: The type subsumption relations and the TCs are applied to the query structure until

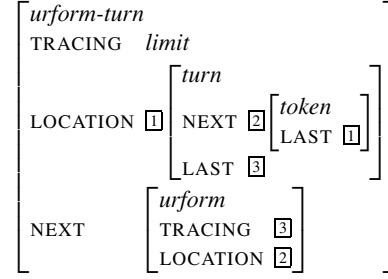
- 1) all types in the query structure are minimal in the sense, that they have no subtypes;
- 2) all TCs matching types in the solution structure have been applied.

In the TCS discussed here, the type subsumption relations and TCs are applied in a depth-first order and backtracking is used whenever the unification of a TC fails. The order in which subtypes of a type are tried, and features inside a

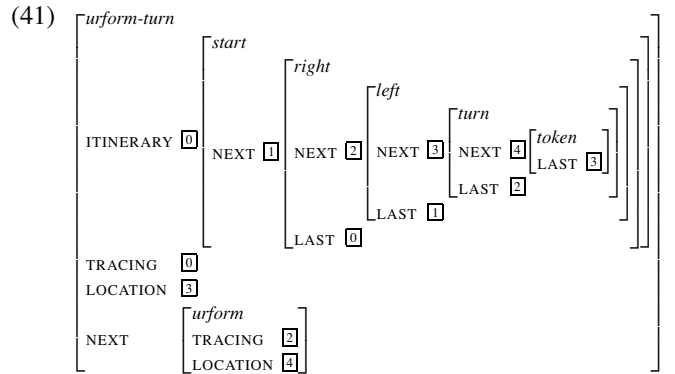
structure are resolved, follows their definition order unless otherwise stated.

The following section, which explains the generation of the second solution, demonstrates the resolution process in more detail.

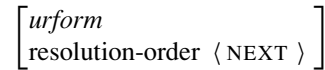
Generating the second solution Besides *urform-stop*, the second subtype of *urform* with a type constraint matching the query structure (34) is *urform-turn*. The complete TC for *urform-turn* is:

(40) *Constraint for urform-turn*

Unifying it with the query results in the following structure:



The root type *urform-turn* is a minimal type and therefore already resolved. The resolution procedure continues with its feature values. In the case of structures with the root type *urform* or one of its subtypes, the resolution order for its features has been given explicitly:

(42) *Resolution order of urform structures*

Resolution-order is a meta-attribute which can be used to specify the resolution order of the features: the listed features are resolved first; features not listed are resolved later, using the order in which they have been defined.

Looking at the value of feature *NEXT*, it can be seen to have the same type *urform* and features *TRACING* and *LOCATION* as the original query. The feature *TRACING*, however, now points to the last token of the first level, and the *LOCATION* feature points to the new end of the token sequence, extended by the type *turn*. This time only the type constraint for *urform-left* matches, which in its complete form looks as follows (the features *NEXT* and *LAST* have been abbreviated in most cases as *N* and *L* respectively):