

Reverse XSLT

Seminararbeit im Seminar XSLT. ZHAW, Zürich, 2. Juli 2014. Florian Lüthi

Umkehrbarkeit von Konvertierungen

XSL-Transformationen werden als Instanzen von Konvertern angesehen. Da Konvertierungen im Allgemeinen Informationsverlust zur Folge haben (wobei Informationsanreicherung mit Informationsverlust gleichzusetzen ist), sind solche Konvertierungen (und ergo XSL-Transformationen) im Allgemeinen nicht bijektiv. Es ist sogar so, dass im Normalfall jedes Element der Wertemenge auf unendlich viele Elemente der Definitionsmenge abgebildet werden kann.

... aber trotzdem!?

In Spezialfällen ist es möglich, nicht-bijektive Teil-Transformationen, welche auf denselben Daten operieren, so im Definitionsbereich einzuschränken, dass die Umkehrung eine endliche Anzahl an Resultaten produziert. Das Hauptinstrument hierzu ist Prädikatenlogik und brute force. Ebenfalls hilfreich ist der Zuzug des formalen Schemas der Input-Daten, um beispielsweise Datentypen einschränken zu können.

Die Frage nach der Existenz der Umkehrung für jede XSL-Transformation kann bejaht werden, da XSL Turing-vollständig ist und ein Algorithmus zur Invertierung von Turingmaschinen existiert. Eine solche Umkehrung führt aber im Allgemeinen zu unendlich vielen Resultaten.

Konstruktion der Umkehrung

Das grundsätzliche Vorgehen zur Umkehrung von XSL-Transformationen besteht darin, aus Template-Definitionen \XPath es zu konstruieren und umgekehrt. Da die Abhängigkeiten der Template-Definitionen untereinander nur implizit gegeben sind, führt eine vollständige Konstruktion zu einer exponentiell wachsenden Anzahl resultierender Templates und dadurch zu brute force während der Ausführung der so konstruierten Umkehrung. Teilweise Abhilfe schafft wiederum das Hinzuziehen des Schemas der Input-Daten.

Schlussfolgerungen/Ausblick

Der Fokus für eine Weiterführung dieser Arbeit könnte sich dahingehend verschieben, dass nicht die Umkehrung selber im Vordergrund steht, sondern eher die Entscheidung, ob eine Konvertierung bijektiv (und daher im Grunde unnötig) ist oder nicht.