Realisierung paradigmenbasierter Derivationsmorphologie in finite-state Umgebungen

Christian Fischbach und James Kilbury

1 Einleitung

Sowohl linguistisch-theoretische als auch anwendungsorientierte Überlegungen erfordern eine Strukturierung und Codierung derivationsmorphologischer Informationen des Deutschen, die nicht nur die lexikalisierten Formen und Verallgemeinerungen über sie, sondern auch Verallgemeinerungen über neue, im Lexikon noch nicht verzeichnete Formen erfaßt. Zum Kern der Generalisierungen über Derivationsmorphologie zählt eine detaillierte Beschreibung morphotaktischer Strukturen, die allerdings ohne Rückgriff auf automatische Verfahren zur Wissensextraktion aus Lexikondatenbanken kaum erreichbar sein dürfte.

Vor diesem Hintergrund präsentieren wir effiziente finite-state Codierungen des in [Kilbury, 1992] entwickelten Ansatzes zur paradigmenbasierten Derivationsmorphologie, die eine effiziente und robuste morphotaktische Verarbeitung der Derivationsstrukturen des Deutschen ermöglichen. Wir zeigen ferner, wie diese Codierungen in einem vollautomatischen, generischen Verfahren aus der Lexikondatenbank CELEX [Baayen et al., 1995] extrahiert werden können.

2 Linguistische Motivation

Die linguistische Motivation für die in den Abschnitten 4 und 5 vorgestellten Implementierungen ergibt sich aus Grundannahmen von Kilbury [Kilbury, 1992] über die Morphotaktik der Derivation in einem

morphembasierten Lexikon. Hierzu zählt insbesondere die Annahme einer gegenseitigen, quasi komplementären Subkategorisierung (hier verstanden als Kombinationspotential) von Affixen (für Argument- und Ergebniswortklasse) und Stämmen (für Affixe). So subkategorisiert etwa das Suffix *–lich* die Argumentwortklasse V(erb) und die Ergebniswortklasse A(djektiv), die verbale Wurzel *wirk* subkategorisiert komplementär das Suffix *–lich*. Die Subkategorisierung der Affixe drückt ihr allgemeines Kombinationspotential aus, während die der Wurzeln bzw. Stämme die tatsächlich lexikalisierten Formen erfaßt (z.B. *wirklich* aber nicht *entwirken*).

Diese gegenseitige morphologische Subkategorisierung induziert rekursiv aufeinander anwendbare Affigierungsstrukturen und damit Hüllen von lexikalisierten Formen. Für die Wurzel $bind_V$ bildet z.B. {*verbind*, *verbindlich*, *unverbindlich*, ...} eine Teilhülle. Die Annahme, daß die einzelnen Wurzelhüllen Gemeinsamkeiten aufweisen, wie etwa die Wurzeln *leg* und *setz* mit Teilhüllen {*beleg*, *verleg*, *zerleg*, ...} bzw. {*besetz*, *versetz*, *zersetz*, ...}, führt schließlich zur Idee des Derivationsparadigmas als Template (Muster), das den gemeinsamen Teil verschiedener Wurzelhüllen repräsentiert.

Mit unserem Ansatz schaffen wir die Grundlagen für ein Lexikon, das (a) die Intuition erfaßt, daß Formen wie *entsetzen* morphologisch komplexe Stämme haben, (b) dennoch eine Adressierung der idiosynkratischen lexikalischen Informationen definiert und (c) die für die Beschreibung von Neubildungen erforderlichen Verallgemeinerungen über Affixe ausdrückt. Der formale Ansatz läßt sich auf triviale Weise von Affix- auf Template-basierte Modelle der Morphologie wie das von [Riehemann, 1998] im Rahmen der HPSG übertragen.

3 Zur finite-state Technik

In unserem Ansatz verwenden wir zwar Transduktoren, aber sie realisieren keine phonologischen Transduktionen zwischen Ebenen im Sin-

ne von Two-Level Morphology (vgl. [Sproat, 1992]). Die Bänder des Basisgerüsts (vgl. Schritt (b) in Abschnitt 4) sind zugleich Ein- *und* Ausgabeband: die den Affigierungsstrukturen inhärenten Reihenfolgeconstraints werden bandübergreifend erfaßt; die Ausgabe von Werten („Auszahlungen“ oder „pay-offs“) erfolgt in bandübergreifend verteilte Variablen. Diese Variablen simulieren weitere Bänder, führen allerdings nicht aus der Klasse der endlichen Automaten heraus, weil die instantiierte Information im Automaten nicht propagiert wird (wodurch die implizite Realisierung von Kellerstrukturen ausgeschlossen ist).

Die Kernimplementierungen nutzen das finite-state Toolkit *FSA* [FSA-URL, 1999], während die Automaten mit dem Graphvisualisierungstool *daVinci* [Werner, 1998] graphisch dargestellt werden.

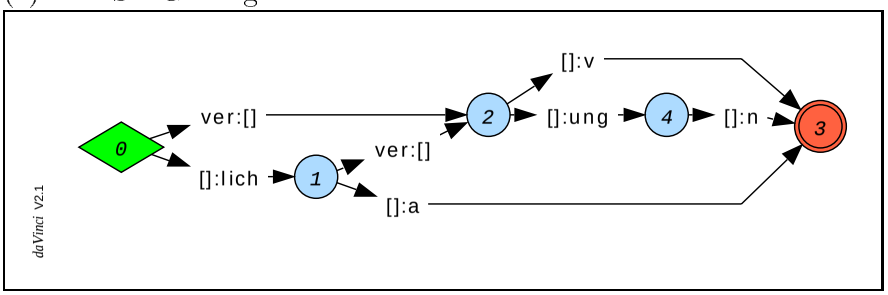
4 Codierung und Verarbeitung

In Anlehnung an [Kilbury, 1992] besteht unser morphotaktisches Parsing aus zwei nichtdeterministischen Analyseschritten: (a) reguläre Zerlegung eines Wortes in Morph(em)e, und (b) rekursive Affigierung gemäß der morphologischen Subkategorisierung. Allerdings operationalisieren wir beide Schritte – nicht nur den ersten – durch endliche Automaten.

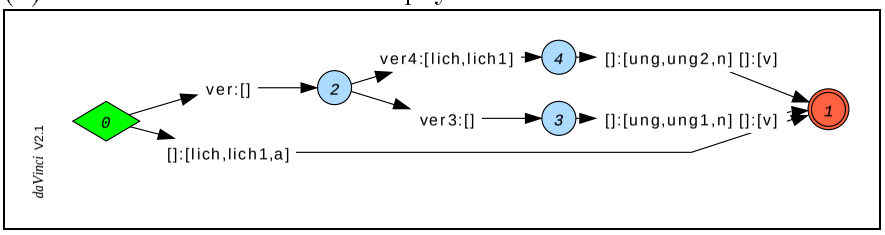
Die in Schritt (a) verwendeten Lettertrees (d.h. Diskriminationsnetze, die mit fortschreitender zeichenweiser Wortanalyse „von links nach rechts“ immer mehr Lemmata als Ergebnis ausschließen) können (nach einfachen Transformationen) direkt durch Transduktoren modelliert werden. Solche Transduktoren existieren für Präfix-, Wurzel- und Suffix-Lettertrees. Der pay-off der Affixtransduktoren enthält z.B. die Affixzeichenkette (d.h. deren phonetische oder orthographische Transkription) und Argument- und Ergebniswortklasse der Subkategorisierung. Der Wurzeltransduktor liefert zu einer bekannten Wurzel ihre Zeichenkette und einen Verweis auf einen Automaten, der ihre

Derivationshülle beschreibt. Unbekannte Wurzeln werden im Automaten zur Zeit auf eine spezielle Konstante abgebildet, aber in einer künftigen Version soll ein zusätzlicher Automat die Phono- bzw. Graphotaktik von wohlgeformten Wurzeln erfassen.

Abbildung 1: Teilmenge der derivationellen Hülle von $wirk_V$
(a) als FST-Grundgerüst ...



(b) erweitert mit lokalem Affix pay-off.



In Schritt (b) prüft ein Transduktor, der die Derivationshülle einer in Schritt (a) ermittelten Wurzel w beschreibt, ob die in (a) rein zeichensyntaktisch identifizierten Affixe eine bzgl. w morphotaktisch gültige Affigierungsfolge bilden. Somit wird festgestellt, daß die nach (a) prinzipiell wohlgeformte Wortform *verbildlich* gemäß (b) nicht lexikalisiert und nicht wohlgeformt ist. Dieser Hüllen-Transduktor ist, abhängig von den gewünschten pay-offs, in seiner Komplexität skalierbar. Das Basisgerüst ist durch Reduktion von Affixstacks auf zwei Bänder, dem Präfix- und dem Suffixband, charakterisiert und liefert

die Kategorie der abgeleiteten Form. Die Abb. 1(a) und 2(a) zeigen den Graphen und die Konstruktion eines solchen FST-Grundgerüsts für eine Teilmenge der derivationellen Hülle von $wirk_V$.

Abbildung 2: Erzeugung der FSTs aus Abb.1 mit *FSA*
(a) das FST-Grundgerüst,

```
% {...,...} Vereinigung    [...,...] Konkatenation  [] = Epsilon

fsa -r '{\
  [ver:[], {[]:v,\                                % verwirk V
           [[]:ung, []:n]}],\                     % verwirkung N
  [[]:lich, {[]:a,\                               % wirklich A
            [ver:[], {[]:v,\                      % verwirklich V
                     [[]:ung, []:n]}]}]}'\        % verwirklichung N
  fst_T1
```

(b) erweitert mit lokalem Affix pay-off.

```
fsa -r '{\
  [[ver, ver3]:[], {[]:v,\
                   [[]:[ung,ung1], []:n]}],\
  [[]:[lich,lich1], {[]:a,\
                    [[ver,ver4]:[], {[]:v,\
                                     [[]:[ung,ung2], []:n]}]}]}'\
  fst_T2
```

Adäquatere morphotaktische und semantische Analysen beruhen auf einer feineren Affixindizierung, insbesondere auf der Unterscheidung verschiedener semantischer Affixausprägungen mit denselben syntaktischen Subkategorisierungseigenschaften. So zeigt z.B. *ver–* in *verbinden* und *sich verschlucken* zwar das gleiche morphosyntaktische Verhalten aber unterschiedliche semantische Funktionen. Eine Transduktorerweiterung ermöglicht deshalb einen lokalen pay-off für Affixe in Variablen, die auf den Bändern systematisch direkt hinter dem Affix positioniert werden. Die Abb. 1(b) und 2(b) illustrieren eine sol-

che Erweiterung für das FST-Grundgerüst zur Teilhülle von $wirk_V$ aus 1(a) und 2(a). Weitere Abstraktionen sind notwendig, falls Information über die Reihenfolge der pay-offs benötigt wird, beispielsweise für die Weiterverwendung der pay-offs in einem nichtmonotonen Rahmen.

Die Schritte (a) und (b) erfolgen derzeit streng seriell. Die einzelnen Lettertrees aus (a) werden durch die regulären Operatoren * (Kleenescher Abschluß) und $\cdot$ (Konkatenation) gemäß dem Ausdruck $p^* \cdot r \cdot s^*$ zu einem Gesamttransduktor zusammengefaßt, der die erforderliche Zerlegung leistet. Pay-offs aus (a) werden mittels eines morphologischen Interpreters direkt in Bandbeschriftungen für den Derivationshüllen-Transduktor in (b) umgesetzt. Es bleibt zu klären, ob eine versetzt parallele Arbeitsweise der Transduktoren aus (a) und (b) prinzipiell effizienter ist.

5 Inferenz aus der Lexikondatenbank CELEX

Die multilinguale (Deutsch, Englisch, Niederländisch) Lexikondatenbank CELEX (CD Release 2) enthält u.a. Dateien, in denen die hierarchische Morphemsegmentierung mit Wortklassenangaben für jedes Wort in linearer Repräsentation codiert wird.[1] Unser vollautomatisches, generisches Verfahren zur Inferenz von Derivationshüllen bestimmt zunächst durch ein *Gawk*-Script[2] für eine Wurzel alle Hüllenformen und extrahiert zu diesen jeweils den hier relevanten Feldeintrag zur Morphemsegmentierung. Ein *Flex*/*Bison*-Compiler filtert dann aus diesen quasi „inside-out“ codierten CELEX-Affigierungsstrukturen die Wurzelbeschreibung heraus und übersetzt sie in „left-to-right“

[1] Hier konkret die Datei `german/gml/gml.cd` (German Morphology, Lemmas), Feld Nr. 14. (StrucLab – structured segmentation, word class labels).

[2] Softwarepakete und Dokumentationen der Programmierungstools *Gawk* (GNU Project Implementierung der Programmiersprache *AWK* für Patternerkennung und -verarbeitung), *Flex* (Fast lexical analyzer generator) und *Bison* (General-purpose Parsergenerator für LALR CFGs) gibt es z.B. unter `ftp://ftp.gnu.org/gnu`.

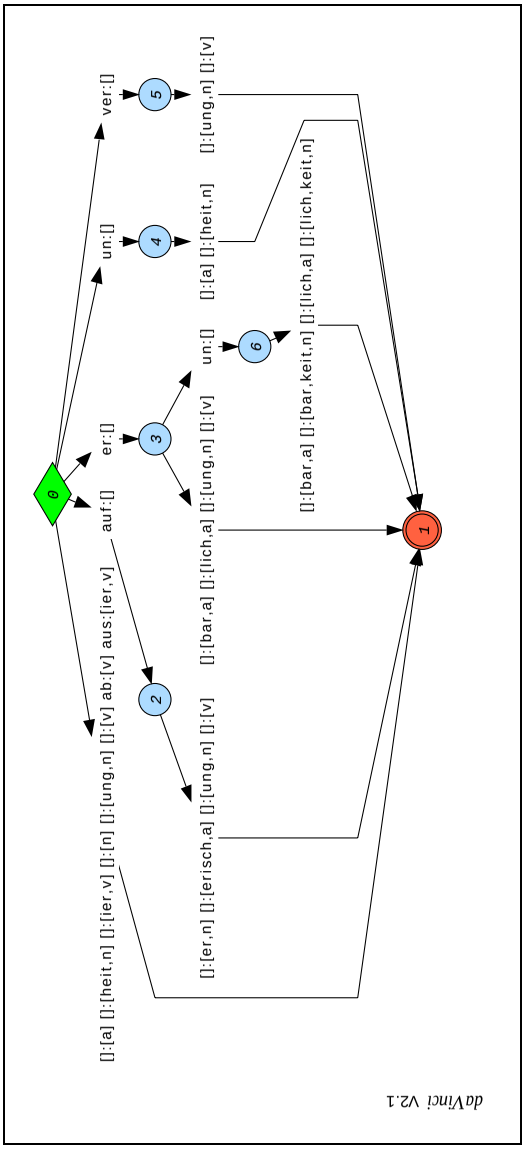


Abbildung 3: Aus CELEX inferierte Derivationshülle von $klar_A$.

codierte Glieder einer verallgemeinerten Vereinigung in einem *FSA*-Ausdruck. Der minimierte Automat, der aus diesem *FSA*-Ausdruck generiert wird, repräsentiert schließlich die gesuchte Derivationshülle und kann z.B. in *Prolog*-Klauseln auscompiliert und visualisiert werden.[3]

Ein Beispiel: Für die gegebene Wurzel $klar_A$ bestimmt das Verfahren u.a. auch $unerklärlich_A$ als Hüllenform und extrahiert die Information

```
((un)[A|.A],(((er)[V|.V],((klar)[A])[V])[V],(lich)[A|V.])[A])[A] .
```

Das aus den genannten Transformationen resultierende FST-Vereinigungsglied ist dann

```
[er:[], {[]:v, [[]:lich, {[]:a, [un:[], {[]:a}]}]}] .
```

Abb. 3 zeigt die 24 Formen umfassende, aus CELEX inferierte Derivationshülle von $klar_A$.[4]

6 Ausblick

Mit dem in Abschnitt 5 vorgestellten Verfahren können Derivationshüllen beliebiger Wurzelmengen automatisch aus CELEX inferiert werden. Es liefert damit die Basis für einen Ansatz zur Identifikation derivationeller Paradigmen. Da die inferierten Wurzelhüllen in der formalen Struktur endlicher Transduktoren vorliegen, könnten Algorithmen zur geeigneten Klassifikation gerichteter Graphen den Kern eines solchen Ansatzes bilden.

[3] Eine unrestringierte Online Demonstration des Verfahrens ist derzeit über `http://www.phil-fak.uni-duesseldorf.de/sfb282/B3` erreichbar.

[4] Stammallomorphie und speziell Umlautung vermerkt CELEX separat in den Tabellenfeldern Nr. 30 (StrucAllo – Stem allomorphy, any level) bzw. Nr. 32 (StrucUml – Umlaut, any level); diese Prozesse können also bei Bedarf besonders berücksichtigt werden (vgl. Fußnote 1).

Literatur

[Baayen et al., 1995] Baayen, H., Piepenbrock, R., and van Rijn, H. (1995). The CELEX Lexical Database, Release 2 (CD-ROM). Linguistic Data Consortium, University of Pennsylvania, Philadelphia, PA.
`http://www.kun.nl/celex`.

[FSA-URL, 1999] FSA-URL (1999). FSA Homepage von G. van Noord an der Universität Groningen/NL.
`http://odur.let.rug.nl/~vannoord/fsa/fsa.html`.

[Kilbury, 1992] Kilbury, J. (1992). Paradigm-based derivational morphology. In Görz, G., editor, *KONVENS 92*, pages 159–168. Springer, Berlin et al.

[Riehemann, 1998] Riehemann, S. (1998). Type-based derivational morphology. *Journal of Comparative Germanic Linguistics*, 2:49–77.

[Sproat, 1992] Sproat, R. (1992). *Morphology and computation*. MIT Press, Mass. et al.

[Werner, 1998] Werner, M. (1998). daVinci V2.1.x Online Documentation Universität Bremen.
`http://www.tzi.de/~davinci/doc_V2.1/`.