Universität Hamburg
Fachbereich Informatik
Arbeitsbereich NatS

Inkrementelle Analyse natürlicher Sprache mit Constraint Satisfaction

Studienarbeit

Michael Schulz
Breitenfelder Straße 56
20251 Hamburg
Matrikel-Nr. 4561994
micha@nats.informatik.uni-hamburg.de

Der Informatiker sagt: oh, das performt nicht,
der Musiker sagt: das groovt nicht,
der Kognitionswissenschaftler sagt: das ist nicht adäquat,
der Mathematiker sagt: na und.

Inhaltsverzeichnis

Kapitel 1

Einleitung

Die vorliegende Arbeit ist im Rahmen des DFG-Projektes DAWAI (Menzel und Schröder, 1998) entstanden und basiert auf dem in Schröder, Foth, und Schulz (1999) beschriebenen Experimentiersystem. Es erweitert dieses um ein Verfahren zur inkrementellen Analyse natürlicher Sprache. Zielsetzung ist es zu zeigen, ob ein constraintbasierter Parsing-Ansatz sich grundsätzlich zur inkrementellen Verarbeitung eignet.

Problembeschreibung

Doch was ist Inkrementalität? Es unterscheiden sich grundsätzlich zweierlei Arten von Problemen, zum Einen solche, die als Ganzes vollständig vorliegen und als solche bearbeitet werden können. Sie entsprechen dem typischen Schema der Informationsverarbeitung: Eingabe, Verarbeitung und Ausgabe. Was aber, wenn sich ein Problem erst im Laufe der Zeit ergibt und man nicht abwarten will, bis es sich möglicherweise nicht mehr verändert? Wie wird mit Problemen umgegangen, deren Natur es ist, nie zu enden und sich ständig zu wandeln? Derartige Aufgaben lassen sich nicht im klassischen Schema verarbeiten. Vielmehr ist man gezwungen, das sich verändernde Problem zu beobachten, neu eintreffende Daten in bereits vorliegende zu integrieren und stetig aus der Summe beider ein Ergebnis zu berechnen bzw. ein altes Ergebnis zu revidieren. Probleme dieser Art können also nur dann verarbeitet werden, wenn Eingabe-, Verarbeitungs- und Ausgabe-Phasen ineinander verschränkt sind. In diesem Sinne betrachten wir also inkrementelle Informationsverarbeitung als fortschreitende Problemaquirierung und -bewältigung.

Typische technische Systeme, die in diesem Sinn Information inkrementell verarbeiten, sind Regelungs- und Steuerungssysteme, deren Aufgabe es ist, einen stetigen Abgleich von Mess- und Steuerwerten vorzunehmen. Kritisch ist hier der Begriff der Zeit, die benötigt wird, auf veränderte äußere Bedingungen zu reagieren. Deshalb bezeichnet man solche Systeme auch als Echtzeit-Systeme, um die unmittelbare Koppelung von Problemveränderung und Problembewältigung zu beschreiben. Im Folgenden sprechen wir demnach Online- und Offline-System.

Unmittelbar auf der Hand liegen hier die Fragen, ob oder ab wann ein System die Flut der einströmenden Informationen nicht mehr bewältigen kann oder welche Strategien nötig sind, um standzuhalten. Ein derartiges Online-System muß davon ausgehen, daß die Anzahl der Problemveränderungen nicht beschränkt ist, also ein Ende der Eingabe nicht in Sicht ist. Daraus folgt, daß in irgendeiner Weise eine Auswahl aus den einkommenden Daten im Verhältnis zum voraussichtlichen Aufwand der damit verbundenen Berechnungen getroffen werden muß. Das heißt allerdings auch, daß eine mögliche Fehlberechnung in Kauf genommen wird, die nicht stattfindet, falls mehr Informationen berücksichtigt werden. Dies stellt den Kern der Problematik inkrementeller Informationsverarbeitung dar.

Doch ein viel näherliegendes Beispiel inkrementell verarbeitender Informationssysteme stellt der Mensch selbst dar. In unablässiger Weise ist er seiner Umwelt ausgesetzt, die er ständig wahrnimmt und zu interpretieren versucht, um rechtzeitig zu reagieren oder Einspruch zu erheben. Das Problem stetiger Stimuli unterschiedlichster Sinneswahrnehmung stellt sich dem Menschen in ganz unmittelbarer Form.

In dieser Hinsicht kann Sprachverarbeitung als ein Echtzeit-Problem verstanden werden: Sprache wird vom Menschen also unmittelbar inkrementell verarbeitet. Dabei versucht er nicht nur, wie oben beschrieben, neue in bereits analysierte Informationen zu integrieren, sondern formuliert aufgrund seiner gesamten Kenntnisse Erwartungen an zukünftige Informationen auf verschiedenen sprachlichen Ebenen. In welcher Weise und wie stark die sprachlichen Ebenen miteinander interagieren und Erwartungen der Ebenen untereinander propagiert werden, ist Gegenstand aktueller Forschungen.

Aus linguistischer Sicht sind unter anderem Gartenpfadsätze von Interesse, die der Mensch trotz genügendem grammatischen Wissen nur schwerlich oder gar nicht versteht. Um derartige sprachliche Phänomene zu modellieren, sind bereits verschiedene Parsing-Modelle entworfen worden, denen eine inkrementelle Verarbeitung der Sprache zugrunde liegt. In der vorliegenden Arbeit wird versucht zu zeigen, daß constraint-basiertes Parsing in der Art, wie es Maruyama (1990) vorschlägt, inkrementell verarbeitbar ist. Vorerst liegt eine Antwort auf diese Frage nicht auf der Hand, da klassischer Weise *Constraint-Satisfaction* eliminativ verarbeitet wird. Ein eliminatives Verfahren verfährt im Gegensatz zu konstruktiven Verfahren, die Lösungen eines Problems von Grund auf berechnet, derart, daß alle möglichen Lösungskandidaten generiert werden und mit Hilfe von Wohlgeformtheitsbedingungen (Constraints) ungültige Kandidaten gelöscht werden. Es bleibt also zu zeigen, inwieweit sich Inkrementalität und Eliminativität miteinander vereinbaren lassen.

Strukturierung

In Kapitel 2 werden grundlegende Begriffe des Constraint-Satisfaction zusammengefaßt und eine formale Basis für die Modellierung zeitabhängiger Probleme gegeben.

Kapitel 3 beschreibt die Verwendung von Constraint-Satisfaction zum Parsen natürlicher Sprache und gibt einen Einblick in den in Schröder, Foth, und Schulz (1999) gegebenen Grammatik-Formalismus.

Schließlich wird in Kapitel 4 das grundlegende Verfahren um unterspezifizierte Dependenzen erweitert. Desweiteren wird der Versuch unternommen, Satzenden zu prognostizieren und so das Parsing-Problem zu beschränken.

Im Anhang sind einige Beispielsitzungen des Experimentiersystems abgedruckt, welches begleitend zu dieser Arbeitet implementiert worden ist.

Danksagung

Ich möchte an dieser Stelle denjenigen danken, die diese Arbeit erst möglich gemacht haben. An erster Stelle ist Ingo Schröder zu nennen, der mir stets durch seinen fortwährenden fachlichen Beistand und seine Hartnäckigkeit sehr geholfen hat. Desweiteren danke ich Prof. Wolfgang Menzel, der mir erst die Chance und den Spielraum zu dieser Arbeit gegeben hat. Ich möchte auch Damir Cavar für die zahlreichen Diskussionen danken, sowie Kilian Foth, der mich mit seinem flinken Verstand und seinem außerordentlichen Humor immer wieder erneut in Erstaunen versetzt. Und natürlich danke ich meiner Freundin Constanze Daum für ihre Geduld, oft nur einen Rücken vor einem Bildschirm zu ertragen. Sie alle haben wesentlich zum Gelingen dieser Arbeit beigetragen.

Michael Schulz.

Kapitel 2

Grundlagen

Dieses Kapitel gibt zunächst einen Überblick über gängige Begriffe des *Constraint-Satisfaction* (kurz *CSP*) sowie eine informelle Beschreibung wohlbekannter Algorithmen. Anschließend wird das allgemeine *CSP* auf zeitlich veränderbare Probleme erweitert.

2.1 Das allgemeine *CSP*

Die ursprüngliche Definition des *CSP* geht auf Waltz (1975) bzw. Montanari (1974) zurück und wird von Tsang (1994) lehrbuchartig dargestellt. Die hiesige Verwendung von Variablen und Namenskonventionen orientiert sich weitestgehend an Maruyama (1990), Menzel (1994) und Schröder (1996).

2.1.1 Definition

Das allgemeine *CSP* stellt die Aufgabe dar, eine wohlgeformte Wertebelegung einer endlichen Menge von Variablen zu finden. Wohlgeformtheit ist bestimmt durch eine endliche Menge von Bedingungen (engl. *Constraints*).

Sei

$$
\begin{aligned}
CSP &:= \langle Z, D, C \rangle && (2.1)\\
&\ \text{das gestellte Problem, wobei} \\
Z &:= \{z_1, \ldots, z_n\} && (2.2)\\
&\ \text{eine endliche Menge von Variablen,} \\
D &:= \{D_i \mid D_i := \{d_i^1, \ldots, d_i^{m_i}\} \ \wedge \ i \in \{1, \ldots, n\}\} && (2.3)\\
&\ \text{Domänen der Variablen in } Z \text{ und} \\
C &:= \{C_{i_1,\ldots,i_j} \mid C_{i_1,\ldots,i_j} \subset D_{i_1} \times \cdots \times D_{i_j} \ \wedge \ j, i_k \in \{1, \ldots, n\}\} && (2.4)\\
&\ \text{die Menge der Constraints auf } j\text{–Tupel von Variablen-} \\
&\ \text{belegungen.}
\end{aligned}
$$

Wir beschränken uns hier auf *CSPs* mit finiten, diskreten Domänen, sowie auf unäre und binäre Constraints C_x und $C_{x,y}$.[1] Die Menge aller möglichen Belegungen der Variablen ist gegeben durch:

$$S := \{A_q \mid \quad A_q := \{\langle z_1, d_1^{k_1}\rangle, \ldots, \langle z_n, d_n^{k_n}\rangle\} \quad \wedge \tag{2.5}$$
$$z_i \in Z,\ k_i \in \{1, \ldots, m_i\} \quad \wedge \quad d_i^{k_i} \in D_i\}. \tag{2.6}$$

$A \in S$ ist eine Lösung des *CSP*, wenn alle Constraints erfüllt sind, also:

$$\forall_{\langle z_i, a_i\rangle \in A} \forall_{C_k \in C} \quad : \quad a_i \in C_k \qquad \wedge \tag{2.7}$$
$$\forall_{(\langle z_i, a_i\rangle, \langle z_j, a_j\rangle) \in A \times A} \forall_{C_{k,l} \in C} \quad : \quad \langle a_i, a_j\rangle \in C_{k,l} \tag{2.8}$$

2.1.2 Darstellung

Durch die Beschränkung auf unäre und binäre Constraints läßt sich ein *CSP* als ein Graph darstellen, in dem die Knoten des Graphen die Variablen und die Kanten binäre Constraints repräsentieren.

Sei ein Constraintnetz wie folgt definiert:

$$CN := \langle V, E\rangle \tag{2.9}$$
$$V := \{V_i \mid V_i := \langle z_i, D_i\rangle\} \tag{2.10}$$
$$E := \{E_{i,j} \mid E_{i,j} := \{x_{s,t} \mid x_{s,t} := \mathit{compatible}(\langle v_i, d_i^s\rangle, \langle v_j, d_j^t\rangle)\}\tag{2.11}$$

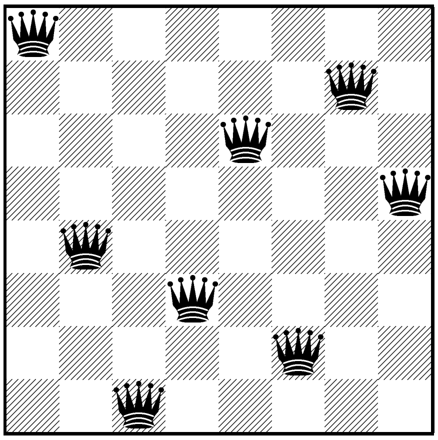
V werden *Constraintknoten* und E *Constraintkanten* genannt. *compatible* ist eine Funktion, die Paaren von Variablenbelegungen boolesche Werte zuordnet; sie sind bestimmt durch Constraints in C.

Es bestehen unterschiedliche Möglichkeiten, Constraints zu formulieren, je nach Art des betrachteten Problems. Constraints auf numerischen Werten sind zum Beispiel in Form von Gleichungen wie $x + y < 10$ oder 'Die Summe zweier Zahlen sei ungerade.' formulierbar.

Allgemein können unäre und binäre Constraints in booleschen Matrizen einheitlich dargestellt werden: Matrizen binärer Constraints stellen in jeder Zelle die Verträglichkeit der assoziierten Wertepaare dar; eine Matrix, die ein unäres Constraint repräsentiert, enthält in der Diagonalen Einsen bzw. Nullen je nachdem, ob das Element mit dem entsprechenden Index in der assoziierten Domäne enthalten ist oder nicht.[2]

[1] Siehe Nudel (1983) für eine Betrachtung derartig beschränkter Probleme.

[2] Der Vorteil dieser Darstellung besteht darin, daß sich alle Constraints miteinander multiplizieren lassen (engl. *relations composition*), was in Verfahren zur Berechnung von Pfadkonsistenz genutzt wird.



$$A = \{\langle z_1, 1\rangle, \langle z_2, 15\rangle, \langle z_3, 21\rangle, \langle z_4, 32\rangle, \langle z_5, 34\rangle, \langle z_6, 44\rangle, \langle z_7, 54\rangle, \langle z_8, 59\rangle\}$$

Abbildung 2.1: Lösung des 8-Damen Problems

2.1.3 Beispiel

Wir betrachten nun eines der bekanntesten *CSP*-Benchmark Probleme[3], das n-Damen Problem. Die Aufgabe besteht darin, n Damen derart auf einem Schachbrett zu positionieren, daß sie sich nicht gegenseitig bedrohen[4].

Dieses Problem läßt sich auf verschiedene Arten formalisieren. Zum einen ist es möglich, den Zeilen (oder Spalten) des Schachbrettes jeweils eine Variable zuzuordnen, wobei jede Variable mit den Zahlen eins bis n belegt werden kann und angibt, daß dort eine Dame in der Spalte (oder Zeile) plaziert ist. Diese Darstellung ist deshalb geschickt, da wir somit schon gewährleistet haben, daß in einer Zeile nur eine Dame plaziert wird. Eine andere Möglichkeit soll hier aufgezeigt werden: wir ordnen jeder Dame eine Variable zu und belegen diese mit Werten zwischen 1 und n^2, welche die Platzierung der Damen auf dem Brett angeben. Alle weiteren Aussagen über zulässige Stellungen werden in Constraints ausgedrückt. Sei das Problem wie folgt definiert, wobei wir uns auf acht Damen beschränken:

$$\begin{aligned}
CSP_1 &:= \{Z, D, C\} \\
Z &:= \{z_1, z_2, \dots, z_8\} \\
D &:= \{D_i \mid D_i = \{1, 2, \dots, 64\} \wedge i \in \{1, \dots, 8\}\} \\
C &:= \{C^1_{x,y}, C^2_{x,y}, C^3_{x,y}\} \\
C^1_{x,y} &:= (x \div 8) \neq (y \div 8) \\
&\ \text{„Zwei Damen befinden sind nicht in derselben Zeile“} \\
C^2_{x,y} &:= (x \; mod \; 8) \neq (y \; mod \; 8) \\
&\ \text{„Zwei Damen befinden sind nicht in derselben Spalte“} \\
C^3_{x,y} &:= |(x \div 8) - (y \div 8)| \neq |(x \; mod \; 8) - (y \; mod \; 8)|
\end{aligned}$$

[3] *CSP*-Benchmarks sind immer mit Vorsicht zu genießen, da ihre Aussagekraft bzgl. der Güte eines Lösungsverfahrens im Allgemeinen sich nicht auf Real-Live Probleme übertragen läßt.

[4] Eine Dame schlägt andere Spielfiguren in der gleichen Zeile, Spalte oder Diagonale.

```
funct compatible_queens(x, y) ≡
  row_1 := x/8;
  row_2 := y/8;
  col_1 := x mod 8;
  col_2 := y mod 8;
  if ((row_1 = row_2) ∨
     (col_1 = col_2) ∨
     (row_1 − row_2) = (col_1 − col_2) ∨
     (row_1 − row_2) = (col_2 − col_1))
    then
        false;
     else
        true;
  fi.
```

Abbildung 2.2: Kompatibilitätskriterium im 8-Damen Problem

„Zwei Damen befinden sind nicht in derselben Diagonalen“

Die Constraints $C^1_{x,y}$, $C^2_{x,y}$ und $C^3_{x,y}$ drücken das Wissen über die Gültigkeit der Stellung zweier Damen zueinander aus, was ebenso durch die in Abbildung 2.2 dargestellte zweistellige Funktion *compatible_queens*(x, y) beschrieben wird. Dabei geben x und y die Nummer des Feldes zwischen 1 und 64 an, auf dem die Dame positioniert werden soll. Eine gültige Belegung des CSP_1 ist in Abbildung 2.1 dargestellt.

Weitere einschlägige Beispiele für Problemstellungen, die sich gut bzw. vorzugsweise mit der Technik des *Constraint-Satisfaction* lösen lassen, sind das Graph-Colouring-Problem, Scene-Labeling (Tsang, 1994, Seite 17ff) sowie Planning- oder Scheduling-Probleme. Die Liste der möglichen Einsatzgebiete von *CSP* ließe sich beliebig weiter führen. Allen gemeinsam ist die Schwierigkeit, ein algebraisches Lösungsverfahren anzugeben, allerdings Aussagen über wohlgeformte Lösungen getroffen werden können.

2.1.4 Lösungsverfahren

Auf einfachste Art läßt sich ein *CSP* lösen, indem alle möglichen Variablenbelegungen auszuprobiert werden (GT – Generate and Test). Obwohl diese Herangehensweise im Vergleich zu anderen Lösungsverfahren ineffizient ist, bleibt das *CSP* ungeachtet dessen ein NP-hartes Problem.[5]

Um ein adäquates Laufzeitverhalten zu erlangen, wurden diverse Lösungsverfahren entworfen, die sich in mehrere Hauptklassen unterteilen lassen.

[5] Beweis: Siehe Schröder (1995)

2.1.4.1 Suche

Eine Suche im *CSP* besteht darin, sukzessive Variablen mit Werten zu belegen und zu prüfen, ob diese Belegung verträglich ist mit den vorherig instanziierten Variablen. Verletzt eine Wertebelegung einer Variable die Constraints, so ist diese Belegung rückgängig zu machen und durch eine andere Belegung aus der Domäne zu ersetzten. Existieren keine weiteren möglichen Werte in dieser Domäne, so setzt *Backtracking* ein, d.h. es werden vorherige Instanziierungen revidiert. Ist eine vollständige Belegung aller Variablen gefunden, die keine Constraints verletzt, so stellt sie eine Lösung des *CSP* dar.

Die Vielfalt der Suchverfahren ist in mehreren Eigenschaften von *CSP*-Suchräumen, d.h. der Menge aller Teilbelegungen, begründet:

1. Die Größe des *CSP*-Suchraums ist finit[6].
2. Die Tiefe des (vollständigen) Suchraumes ist überall gleich.
3. Untersuchräume sind ähnlich.

Punkt 3 führt dazu, daß Folgen von Variablen wiederholt mit gleichen Werten belegt und wieder revidiert werden. Dieses Verhalten nennt man *Trashing.*[7] Es existiert eine große Anzahl von Backtrackingvarianten, die dieses Problem dahingehend adressieren, daß sie lokale Konsistenzbedingungen berechnen oder *Erkenntnisse* aus einem Untersuchraum in anderen Bereichen wiederverwenden. Eine Einführung in einfache Backtracking-Verfahren geben Golomb und Baumert (1965), Bitner und Reingold (1975). Weiterführende Varianten des Backtracking finden sich in Gaschnig (1977, 1978), Haralick und Elliott (1980), Dechter (1990) und Prosser (1993).

2.1.4.2 Problemreduktion

Problemreduktion bedeutet im weitesten Sinne, daß man ein global schwierig erscheinendes Problem vorab lokal zu lindern versucht. Im engeren Sinne bedeutet Problemreduktion, die Größe einer Domäne $|D_i|$ der Variablen z_i zu verringern, indem Konsistenzeigenschaften des *CSP* überprüft werden. Eine Lösung des *CSPs* wird jedoch auf diese Weise selten ermittelt. Konsistenzprüfungen können als eine Art Präprozeß angesehen werden, dem anschließend Verfahren zur Generierung von vollständigen Lösungstupeln folgen. In einigen Fällen ist es möglich, vorab die Unlösbarkeit eines *CSP* durch die Überprüfung von Konsistenzeigenschaften festzustellen.

Je nach Größe des geprüften Umfeldes eines Constraintnetzes werden Knotenkonsistenz, Kantenkonsistenz, Pfadkonsistenz und k-Konsistenz unterschieden.

Knotenkonsistenz: Ein Constraintknoten V_i ist *knotenkonsistent*, wenn alle Werte $d_i^k \in D_i$ alle unären Constraints erfüllen. Ein Constraintnetz ist *knotenkonsistent*, wenn alle Knoten $V_i \in V$ knotenkonsistent sind. Knotenkonsistenz wird auch als 1-Konsistenz bezeichnet.

[6] siehe Tsang (1994)
[7] siehe Gaschnig (1979)

Kantenkonsistenz: Eine Constraintkante $E_{i,j}$ ist *kantenkonsistent*, wenn für alle Werte $d_i^k \in D_i$ ein Wert $d_j^l \in D_j$ existiert, sodaß alle unären und binären Constraints erfüllt sind. Ein Constraintnetz ist *kantenkonsistent*, wenn alle Kanten $E_{i,j} \in E$ kantenkonsistent sind.

$k-$Konsistenz: Ein *CSP* ist *k-konsistent* für $k > 1$, wenn für alle $(k-1)$-konsistenten Variablenbelegungen $A := \{a_1, \ldots, a_{k-1}\}$ alle Variablen z_{i_k} mit einem Wert $d_{i_k}^j$ belegt werden können, sodaß $A' := \{a_1, \ldots, a_{k-1}, \langle z_{i_k}, d_{i_k}^j \rangle\}$ alle Constraints erfüllt.

Pfadkonsistenz: Ein *CSP* ist *pfadkonsistent* genau dann, wenn es 3-konsistent ist.

Die Herstellung von Knotenkonsistenz ist in der Regel Teil der *Generate*-Phase. Sie wird also bereits im Verlauf des Aufbauens eines Constraintnetzes erreicht. Da jede Domäne jeder Variable einmalig überprüft wird, beträgt die Komplexität dieser Operation $O(an)$, wobei $a := \max_{1 \le i \le |D|}(|D_i|)$ und $n = |Z|$ ist. In dieser Phase werden bereits alle unären Constraints ausgewertet. Alle höherstelligen Constraints wirken erst in späteren Phasen der Analyse.

Das Problem der Berechnung höherwertiger Konsistenzeigenschaften wie Kantenkonsistenz besteht darin, daß die Bereitstellung einer einzelnen konsistenten Kante möglicherweise die Konsistenz benachbarter Knoten und Kanten gefährdet und somit deren erneute Überprüfung nach sich zieht. Dieses Problem der *Interkonnektivität* — der gegenseitigen Abhängigkeit der Konsistenzeigenschaft — wurde an verschiedenen Stellen der Literatur thematisiert. Der *Waltz filtering*-Algorithmus in Waltz (1975) stellt eine Möglichkeit der Herstellung von Knotenkonsistenz dar. Ein Überblick über optimierte Verfahren ist in Mackworth (1977) zu finden.

Desweiteren existieren Verfahren zur Berechnung von Pfadkonsistenz und k-Konsistenz, die in der Klasse der Problemreduktionsverfahren die aufwendigsten darstellen. Eine genauere Betrachtung von Pfadkonsistenz nimmt Mackworth (1977) vor. Hier finden sich ebenfalls Hybridalgorithmen, die Konsistenz- und Backtrackingverfahren miteinander verbinden. Fortführende Betrachtung dieser Algorithmen sind in Mohr und Henderson (1986), Han und Lee (1988), Nadel (1989) und in Hentenryck (1989) zu finden.

Tabelle 2.3 gibt einen kurzen Überblick über die Komplexität verschiedener Versionen entworfener Verfahren. NC bezeichnet hier Knotenkonsistenz, AC Kantenkonsistenz und PC Pfadkonsistenz.[8]

2.1.4.3 Unvollständige Methoden

Die Berechnung einer vollständigen, allen Constraints entsprechenden Lösung eines *CSPs* kann unter Umständen sehr lange dauern. In einigen Applikationen sinkt jedoch mit der Dauer der Berechnung deren Nutzen, sodaß eine teilweise korrekte

[8]Cohen und Feigenbaum (1982) bezeichnen Algorithmen zur Herstellung von Konsistenz als *relaxation algorithms*.

Algorithmus	Zeit	Platz
NC–1	$O(an)$	$O(an)$
AC–1	$O(a^3ne)$	$O(e+na)$
AC–3	$\Omega(a^2e), O(a^3e)$	$O(e+na)$
AC–4	$O(a^2e)$	$O(a^2e)$
PC–1	$O(a^5n^5)$	$O(n^3a^2)$
PC–2	$\Omega(a^3n^3), O(a^5n^5)$	$O(n^3+n^2a^2)$
PC–4	$O(a^3n^3)$	$O(a^3n^3)$

$n := |Z|$, $a := \max_{1\leq n}(|D_i|)$, $e := |C_{i,j}|$

Abbildung 2.3: Komplexität verschiedener *relaxation algorithms*

Lösung, die frühzeitig gefunden wird, von höherem Nutzen ist, als eine total korrekte Lösung, die erst nach langer Zeit gefunden wird. Es existieren eine Reihe von Verfahren, die zu Gunsten des Nutzens auf die Vollständigkeit verzichten.

Hill-Climbing stellt einen Ansatz dar, der im Unterschied zu Suchverfahren nicht einen Lösungstupel stückweise generiert, sondern mit einer vollständigen Instanziierung aller Variablen eines *CSPs* beginnt, diese mit Hilfe der Constraints bewertet und Konfliktstellen, also inkompatible Variableninstanziierungen, zu beheben versucht. In Minton et al. (1990, 1992) werden derartige Strategien auch als *heuristic repair methods* bezeichnet. Der Vorteil eines solchen Verfahrens besteht darin, daß zu jedem Zeitpunkt bereits eine – wenn auch suboptimale – vollständige Wertebelegung existiert und durch die fortschreitende Reparatur der Konfliktstellen idealerweise die Qualität der Lösung monoton steigt. Allerdings birgt Hill-Climbing den Nachteil in sich, aus lokalen Maxima und *Plateaus* (Bereiche gleicher Bewertungen) nur ausbrechen zu können, wenn auf die Monotonieeigenschaft zwischenzeitlich verzichtet wird. Desweiteren sind Maßnahmen zur Vermeidung zirkulärer Instanziierungsreihenfolgen zu treffen. Eine Studie der Effekte, die in Hill-Climbing-Verfahren anzutreffen sind, findet sich in Morris (1992).

Im Unterschied zum Hill-Climbing versuchen konnektionistische Verfahren unverträgliche Variablenkonstellationen zu lernen. Die Darstellung von *CSPs* in Constraint-Netzen legt bereits eine Verwandtschaft zu konnektionistischen Verfahren nahe. Constraint-Netze lassen sich als rekurrente Netzwerke simpler Prozessoren auffassen, die die Konsistenz ein oder mehrerer Wertebelegungen überwachen. Ansätze dieser Art finden große Aufmerksamkeit, da sie das Potential zur massiven Parallelisierung in sich bergen. Allerdings sind hier unter anderem die Sicherstellung der Konvergenz der Netze sowie die Vermeidung oszillierender Aktivierungen im Netz problematisch. In Wang und Tsang (1991) wird ein Verfahren namens GENET vorgestellt, welches ein konnektionistisches Netz zur Lösung von *CSPs* simuliert. Wang und Tsang (1992) beschreibt weiterführend einen Entwurf zur Implementierung von GENET in Hardware.

2.2 Das partielle *CSP*

Für viele Anwendungen ist es notwendig, eine Lösung zu finden, die in jeder Hinsicht den Constraints entspricht. Das heißt, daß die Verletzung eines einzigen Constraints dazu führt, daß die gesamte Variablenbelegung scheitert. Mitunter erweist es sich allerdings als sehr schwierig, die Wohlgeformtheit einer Lösung in absoluten Aussagen, das heißt mit *harten* Constraints zu formulieren. Man könnte sich vorstellen, daß es in bestimmten Anwendungen akzeptabel ist, ein oder mehrere Constraints für einen bestimmten Preis zu verletzen, um so ein *optimales* Ergebnis zu erzielen. Man spricht hier von der *Aufweichung* ehemals harter Constraints zu *weichen* Constraints unterschiedlicher Granularität. Doch ebenso, wie ein vorher überbestimmtes *CSP* aufgeweicht werden kann, ist eine Herangehensweise denkbar, in der ein *CSP* mit einer großen Anzahl Lösungen mit zusätzlichen weichen Constraints ausgestattet werden kann, welche *Präferenzen* vergleichbarer Lösungen ausdrücken.

In vielen Anwendungen ist die Verwendung von weichen Constraints eine ganz natürliche Art der Problembeschreibung. Erst die Verwendung von *PCSP* beim Parsing ermöglicht robuste Sprachverarbeitung, das heißt, daß trotz fehlerhafter oder fragmentarischer Eingabe ein Analyseergebnis geliefert wird. (vgl. Menzel (1994, 1995, 1997))

2.2.1 Definition *PCSP*

Die hier vorgestellte Definition ist Tsang (1994) entliehen, geht aber auf die erste formale Definition von Freuder (1989) zurück. In Lawler und Wood (1966), Hall (1971), Reingold et al. (1977), Aho et al. (1983) wurde gezeigt, daß sich *PCSPs* zur Lösung einer ganzen Reihe von Optimierungsproblemen eignen.

Ein *partielles Constraint-Satisfaction Problem PCSP* ist ein Quadrupel

$$PCSP := \langle Z, D, C, g \rangle \tag{2.12}$$

wobei $\langle Z, D, C \rangle$ ein *CSP* ist und g eine Funktion

$$g : S \mapsto \mathcal{R} \tag{2.13}$$

die jeder Belegung $A \in S$ einen numerischen Wert zuweist. Im Allgemeinen wird g derart erweitert, daß bereits Teilbelegungen bewertet werden können. A ist eine Lösung des *PCSP* genau dann, wenn A eine vollständige Belegung aller Variablen und $g(A)$ optimal ist.

2.2.2 Bewertungsfunktionen

Je nach Art der gewählten Bewertungsfunktion unterscheiden sich *PCSPs* erheblich voneinander. Motiviert durch das Scheduling-Problem unterteilt Maruyama (1990)

PCSPs in zwei Klassen: Minimale-Verletzungs-Probleme (MVP) und Maximal-Nützlichkeits-Probleme (MUP). Die Wahl der Bewertungsfunktionen hat unmittelbaren Einfluß auf die Bedeutung der Constraints, mit Hilfe derer ein Problem modelliert wird: in *MVPs* drücken Constraints die Wohlgeformtheit von Lösungen in Form von fehlerhaften Konstellationen aus; einschlägige Constraints in *MUPs* drücken eine Art Unterstützung aus, die einer Lösung zukommt. In konkrete Anwendungen kann eine Kosten-Nutzen-Abwägung als eine Mischung beider Arten von Bewertungsfunktionen modelliert werden.

Minimale-Verletzungs Probleme – *MVP*
Scheduling-Probleme können als MVP definiert werden, wenn es möglich ist, zusätzliche Ressourcen $r \in R$ zur Problembewältigung für einen gewissen Preis anzufordern. Ressourcen können zum Beispiel Kosten für den Zugriff auf einen externen Speicher oder externe Informationen sein (Cachespeicher, Hauptspeicher, Festplattenspeicher, Bandlaufwerke, Archive, Konsultation anderer Informationssysteme). Sei

$$c : C \mapsto \mathcal{N} \tag{2.14}$$

eine Funktion, die jeder Ressource $C_r \in C$ einen Preis zuweist. Somit ergibt sich die Bewertung g als Summe aller Kosten, die eine Belegung A mit sich bringt.

$$g(A) := \sum_{C_r \in C \ \wedge \ \neg compatible(A, C_r)} c(C_r) \tag{2.15}$$

Die Aufgabe besteht darin, g zu *minimieren*.

Maximal-Nützlichkeits Probleme – *MUP*
Scheduling kann auch in anderer Hinsicht modelliert werden, wenn nämlich Constraintverletzungen inakzeptabel sind. Dies ist z.B. der Fall, wenn Speicherbegrenzungen nicht wie im vorherigen Abschnitt angedeutet aufgehoben werden können. Das Problem wird derart modelliert, daß nicht alle Variablen belegt sein müssen, um eine Lösung darzustellen. Vielmehr ist ein k-stelliger Wertetupel A gesucht, wobei $k \leq |Z|$ ist und A maximal nützlich ist. Sei

$$n : Z \mapsto \mathcal{N} \tag{2.16}$$

eine Funktion, die den Nutzen einer Ressource $z \in Z$ angibt. Auch hier ergibt sich die Bewertungsfunktion summativ:

$$g(A) \quad := \sum_{z \in A_{|Z} \ \wedge \ compatible(A, C)} n(z) \quad , \tag{2.17}$$

wobei $A_{|Z}$ die Menge aller Variablen in A denotiert. Die Aufgabe besteht darin, g zu *maximieren*.

2.2.3 Lösungsverfahren

PCSP lassen sich mit ähnlichen Methoden lösen wie *CSPs*. Zusätzlich bietet eine Teilbewertung von Variablenbelegungen die Möglichkeit, Heuristiken zur Steuerung eines Lösungsverfahrens zu entwickeln. Ein derartiges Verfahren ist zum Beispiel *Branch & Bound.*

Branch & Bound ist ein Suchverfahren, welches nur anwendbar ist, wenn die Bewertungsfunktion g monoton ist, also

$$\begin{array}{ll} \bigwedge\limits_{|A'|\leq|A''|} g(A') \leq g(A'') & \vee \\ \bigwedge\limits_{|A'|\geq|A''|} g(A') \geq g(A'') & \end{array}$$

Ist dies der Fall, so läßt sich im Verlauf der Suche der noch zu durchsuchende Raum durch die Bewertung der optimalsten Teilbelegung begrenzen (engl. *to bound*). Dabei verfährt man wie folgt:

1. Beginne mit der Leeren Bindungsliste, bewerte sie optimal und füge sie der Agenda von Suchzuständen hinzu.
2. Entnehme der Agenda den ersten Suchzustand.
3. Ist der aktuelle Suchzustand schlechter bewertet als die bisherigen Lösungen, so lösche diesen Suchzustand und fahre mit Punkt 2 fort. (*bound*)
4. Stellt dieser Suchzustand eine Lösung dar, das heißt, sind alle Variablen instanziiert und ist diese Lösung gleich bewertet wie die bisher optimalste Lösung, so nehme sie in die Liste aller Lösungen auf und fahre mit Punkt 2 fort.
5. Ist der aktuelle Suchzustand eine Lösung und besser bewertet als die der Liste der bisher optimalsten Lösungen, so lösche diese und füge den aktuellen Suchzustand als neues Element ein. Fahre mit Punkt 2 fort.
6. Ist der Suchzustand noch keine vollständige Belegung, so wähle eine noch zu instanziierende Variable aus und erzeuge für jede mögliche Belegung einen neuen Suchzustand. (*branch*)
7. Bewerte jeden neuen Suchzustand und füge ihn der Güte nach sortiert der Agenda hinzu.
8. Lösche den aktuellen Suchzustand und springe zu Punkt 2, falls die Agenda nicht leer ist.

Nach Beendigung des Verfahrens erhalten wir eine Liste der optimalsten Variablenbelegungen des *CSPs*.

2.3 Das dynamische *CSP*

Dynamische *CSPs* (*DCSP*) sind *PCSPs*, die sich mit der Zeit verändern. Dies geschieht, indem sie Werte, Variablen oder Constraints verlieren oder hinzugewinnen oder die Bewertung der Belegungen sich ändern. Man kann weiterhin zwischen *DCSPs* unterscheiden, die temporären oder permanenten Veränderungen unterliegen sind. Frühere Arbeiten beschäftigten sich weitestgehend mit letzterem Fall (Dechter und Dechter, 1988, Bessière, 1991, 1992). Temporäre Veränderungen sind solche, die zu einer Ungültigkeit einer einmal gefunden Lösungen führen, aber zu einem späteren Zeitpunkt wieder rückgängig gemacht werden können (Ausfall einer Maschine, eines Speichermediums, Krankheitsfall eines Angestellten). Dabei ist nicht gewährleistet, daß durch den Einsatz zwischenzeitlich abhanden gekommener Ressourcen nachher wieder die gleiche Lösung gefunden wird. Permanente Veränderungen von Problemen stellen im Gegensatz zu temporären einen irreversiblen Zusammenhang zwischen Problemen dar.

Einen weiteren Spezialfall dynamischer *CSPs* stellen inkrementelle *CSPs* dar. Inkrementalität wird derart verstanden, daß ein Problem bereits im Verlaufe seines Entstehens analysiert wird. Veränderungen geschehen hier nicht in willkürlicher sondern in stetiger Weise, so daß zum Beispiel mit jedem Zeitschritt dem *CSP* eine Variable hinzugefügt wird. Oft spricht man bei Verfahren, die inkrementelle *CSPs* lösen, von einem inkrementellen Berechnungsmodus. Im Unterschied dazu wird ein Berechnungsmodus *Batch-Mode* oder *Offline-Betrieb* genannt, wenn das gesamte Problem bereits vor Rechenbeginn vollständig spezifiziert ist.

2.3.1 Definition *DCSP*

Ein *DCSP* läßt sich als ein Tupel von *PCSPs* auffassen,

$$DCSP \; := \prod_{i \in \{1,\dots,max\}} P(t_i) \tag{2.18}$$

wobei das Ausgangsproblem $P(t_1)$ ein *PCSP* ist und

$$P(t_i) \; := \; P(t_{i-1}) \oplus \Delta_P(t_i) \quad \text{für } i > 1 \tag{2.19}$$

rekursiv definiert ist.[9] Seien zwei aufeinander folgende *PCSPs* definiert als

$$\begin{aligned} P(t_{i-1}) \; &:= \; \langle Z(t_{i-1}), D(t_{i-1}), C(t_{i-1}), g(t_{i-1}) \rangle \quad \text{und} && (2.20) \\ P(t_i) \; &:= \; \langle Z(t_i), D(t_i), C(t_i), g(t_i) \rangle \quad \text{mit} && (2.21) \\ g(t_i) \; &: \; S(t_i) \mapsto \mathcal{R} && (2.22) \\ & \; \text{eine Funktion, die zu jedem Zeitpunkt } t_i \text{ Belegung} \\ & \; A(t_i) \in S(t_i) \text{ bewertet,} \end{aligned}$$

[9]O.B.d.A. gelten alle Betrachtungen auch für $max = \infty$.

so bezeichnet

$$\begin{aligned}
\Delta_Z(t_i) &:= Z(t_i) \ominus Z(t_{i-1}) && (2.23)\\
&\quad \text{die Veränderung der Menge der Variable,} \\
\Delta_D(t_i) &:= \langle D_1(t_i), \ldots, D_n(t_i)\rangle \ominus \langle D_1(t_{i-1}), \ldots, D_n(t_{i-1})\rangle && (2.24)\\
&\quad \text{Veränderung der Domänen,} \\
\Delta_C(t_i) &:= C(t_i) \ominus C(t_{i-1}) && (2.25)\\
&\quad \text{Veränderung der Constraints und} \\
\Delta_g(t_i) &:= g(t_i) \ominus g(t_{i-1}) && (2.26)\\
&\quad \text{Veränderung der Gewichte der Constraints}
\end{aligned}$$

Folglich läßt sich die Veränderung eines *PCSPs* zum Zeitpunkt t_i darstellen als

$$\Delta_P(t_i) = \Delta_Z(t_i) \oplus \Delta_D(t_i) \oplus \Delta_C(t_i) \oplus \Delta_g(t_i) \tag{2.27}$$

2.3.2 Lösungsverfahren

Die Lösungen eines *DCSPs* zum Zeitpunkt t_i stellen sich als $A(t_i) \in S(t_i)$ dar. Besonders hervorzuheben ist die letzte Lösung eines *DCSPs* $A(t_{max})$, falls t_{max} endlich ist. Allgemein gilt nicht, daß diese gleich einer Lösung A_{batch} eines *Batch-Mode*-Algorithmus zur Lösung von $P(t_{max})$ ist, da nicht unbedingt

$$A(t_{max}) = A(t_1) \oplus \Delta_A(t_2) \oplus \cdots \oplus \Delta_A(t_{max}) = A_{batch} \tag{2.28}$$

gewährleistet ist. Trotzdem will man geeignete Verfahren entwickeln, die meistens der Bedingung (2.28) genügen.

Anstatt *max CSPs* unabhängig voneinander zu lösen, wollen wir vermeiden, die Menge aller gefundenen Lösungen $A(t_{i-1})$ des Problems $P(t_{i-1})$ zu verlieren, und sie statt dessen zur Lösung von $P(t_i)$ heranziehen. Ein *DCSPs* zu lösen besteht also darin, aufgrund der Problemveränderungen $\Delta_P(t_i)$ Lösungsveränderungen $\Delta_A(t_i)$ vorzunehmen, sodaß $A(t_{i-1}) \oplus \Delta_A(t_i)$ eine Lösung von $P(t_{i-1}) \oplus \Delta_P(t_i)$ ist. Dies läßt sich nochmals an Abbildung 2.4 veranschaulichen, welche mit notationellen Adaptionen Wirén (1993, 1994) entliehen ist und deren Notation wiederum Alpern et al. (1990) ähnelt.

Wirén betrachtet ein formelles Maß für „die Menge der Dinge, die sich ändern", indem er $\Delta_{A_{min}}$ definiert als das Minimum an Änderungen, die ein Algorithmus vornehmen muß, um eine weitere Lösung zu finden. $\Delta_{A_{min}}$ läßt sich beispielsweise angeben als die „Differenz" zweier Lösungen $A_{batch}(t_i)$ und $A_{batch}(t_{i+1})$, die ohne Berücksichtigung vorheriger Probleminstanzen direkt berechnet werden. Es wird nun angenommen, daß ein inkrementelles Lösungsverfahren nicht unterhalb der Schranke $\Delta_{A_{min}}$ Zwischenlösungen generieren kann. Idealerweise hängt die Komplexität von der Größe der Veränderungen ab, denen ein Problem unterliegt, und nicht von der

$$
\begin{tikzcd}
P(t_{i-1}) \arrow[r, "\Delta_P(t_i)"] \arrow[d] & P(t_i) \arrow[d] \\
A(t_{i-1}) \arrow[r, "\Delta_A(t_i)"] & A(t_i)
\end{tikzcd}
$$

Abbildung 2.4: Problemveränderung versus Lösungsveränderungen

Größe des momentanen Inputs. Alpern et al. definieren dahingehend die *intrinsische* Größe einer Veränderung als

$$\delta := |\Delta_P| + |\Delta_{A_{min}}| \tag{2.29}$$

Diese Formulierung ist durch die Tatsache motiviert, daß einerseits eine kleine Problemveränderung zu einer großen Lösungsveränderung führen kann, und daß andererseits eine große Problemveränderung durch geringe Anstrengungen mit Hilfe der vorherigen Lösung bewältigt werden kann. Ein inkrementeller Algorithmus wird laut Wirén als *beschränkt*[10] betrachtet, wenn er alle Veränderungen in $O(f(\delta))$ Zeit berechnen kann. Algorithmen verschiedenster Art lassen sich so nach dem Grad ihrer Beschränktheit klassifizieren (siehe Ramalingam und Reps (1991)). Zum Beispiel wird ein Algorithmus, der sich linear zu δ verhält, als *asymptotisch optimal* bezeichnet, und als *unbeschränkt*, falls für ein gegebenes δ die Zeit zur Berechnung von $\Delta_A(t_i)$ beliebig sein kann.

Nachdem wir in Kapitel 3 constraintbasiertes Parsing auf das Lösen von *CSP* zurückgeführen, stellen wir in Abschnitt 4.4 eine Modifikation des Branch & Bound Suchverfahrens vor, welches sich zum inkrementellen Parsing eignet.

[10](engl. *bounded*)

Kapitel 3

Constraintbasiertes Parsing

Sprachverarbeitung durch Constraint-Satisfaction beschreibt erstmalig Maruyama (1990) und führt dazu einen neuen Grammatikformalismus namens *Constraint Dependency Grammar* (*CDG*) ein, in dem grammatikalische Regeln als Constraints auf Dependenzen zwischen Worten eines Satzes dargestellt werden. Die gewählte Formalisierung erlaubt es, das Parsing-Problem als *CSP* aufzufassen und Lösungsverfahren des *CSP* anzuwenden.

Zu Beginn stellen wir die klassische Definition von Maruyama vor. In Schröder, Foth, und Schulz (1999) wird der Formalismus um lexikalische Informationen und Begriffshierarchien erweitert. Hier seien allerdings nur die Grundzüge der Möglichkeiten erläutert, die dort beschrieben werden. Desweiteren wird die Gewichtung von Constraints näher betrachtet, die – wie bereits in Menzel (1995) beschrieben – genutzt werden kann, um Sprache robust zu verarbeiten. Wir betrachten hier ebenfalls dynamische Bewertungen von Constraintverletzungen im Unterschied zu statisch gewichteten Constraints, welche in späteren Abschnitten eine wichtige Rolle zur Modellierung von Satzende-Hypothesen spielen.

In Abschnitt 3.2 stellen wir eine Beispielgrammatik vor, die einige grundlegende Mechanismen des *CDGs* präsentiert.

3.1 Definition *CDG*

Sei $S = \langle w_1, w_2, \ldots, w_n \rangle$ ein Satz über einem finiten Alphabet Σ, dessen Worte w_i ein oder mehrere *Rollen* einnehmen. Eine Rolle ist eine Beschreibungsebene eines Satzes, wie z.B. Syntax oder Semantik. Ein *CDG* ist ein Quadrupel der Art

$$CDG := \langle \Sigma, L, R, C \rangle \tag{3.1}$$

$$\Sigma := \text{finites Alphabet} \tag{3.2}$$

$$R := \{r_1, \ldots, r_k\} \quad \text{Menge von Rollen} \tag{3.3}$$

$$L := \{a_1, \ldots, a_l\} \quad \text{Menge von Labeln} \tag{3.4}$$

$$C := \text{Menge von Constraints} \tag{3.5}$$

Eine Analyse des Satzes S besteht darin, jeder Rolle jeden Wortes Werte zuzuweisen, sodaß alle Constraints unverletzt sind. Eine Analyse ist eine Menge von Rollenwertzuweisungen der Art

$$A : S \times R \mapsto L \times (S \cup \{nil\}) \tag{3.6}$$

und wird in Form von Dependenzbäumen dargestellt. Constraints werden in Form von logischen Ausdrücken über Rollenwertzuweisungen $x \in A$ angegeben.

3.1.1 Lexikalische Informationen

Schröder, Foth, und Schulz (1999) erweitert hier die klassische Definition des *CDG* um ein Lexikon $\mathcal{D}$, einer Menge von Feature-Strukturen, die das lexikalische Wissen beinhalten, auf welches sich Constraints beziehen. In Maruyama (1990) finden wir zwar keine explizite Erwähnung lexikalischen Wissens in Form einer Lexikoneinheit, halten es aber in diesem Rahmen für erwähnenswert, da seine Struktur unmittelbar mit der Formulierung der Constraints korreliert. Entsprechend ist die Definition (3.6) der Rollenwertzuweisung anzupassen, da sie jetzt nicht auf Wortformen, sondern auf den jeweiligen Lexemen definiert ist.

$$A : \mathcal{D} \times R \mapsto L \times (\mathcal{D} \cup \{nil\}) \tag{3.7}$$

Zum Beispiel hat ein Lexikoneintrag $\text{Flöte}_{sg} \in \mathcal{D}$ folgende Form:

$$\text{Flöte}_{sg} := \begin{bmatrix} word & : & \text{Flöte} \\ syn & : & \begin{bmatrix} cat & : & noun \\ num & : & sg \\ case & : & nom, acc, dat, gen \\ pers & : & 3 \\ gen & : & fem \end{bmatrix} \\ sem & : & \begin{bmatrix} cat & : & \text{Flöte} \end{bmatrix} \end{bmatrix}$$

Eine freie Variable x in einem Constraint repräsentiert eine Kante des Dependenzbaumes[1], sodaß mit „$x.label$“ auf den Label der Dependenz zugegriffen wird, „$x\downarrow$“ das modifizierende Lexem (Modifier) und “$x\uparrow$“ das modifizierte Lexem (Modifiee) bezeichnet. Ferner können auf lexikalische Eigenschaften der beteiligten Worte zugegriffen werden, indem die Pfade innerhalb der Featurestrukturen des Lexems angegeben werden. So denotiert der Term „$x\uparrow$*syn:cat*“ die syntaktische Kategorie des Modifiees. Im Falle der obigen Definition von Flöte_{sg} wäre dies die Zeichenfolge „*noun*“.

[1]nicht zu verwechseln mit den Kanten des Constraintnetzes; Knoten des Constraintnetzes korrelieren mit den Kanten des Dependenzbaumes

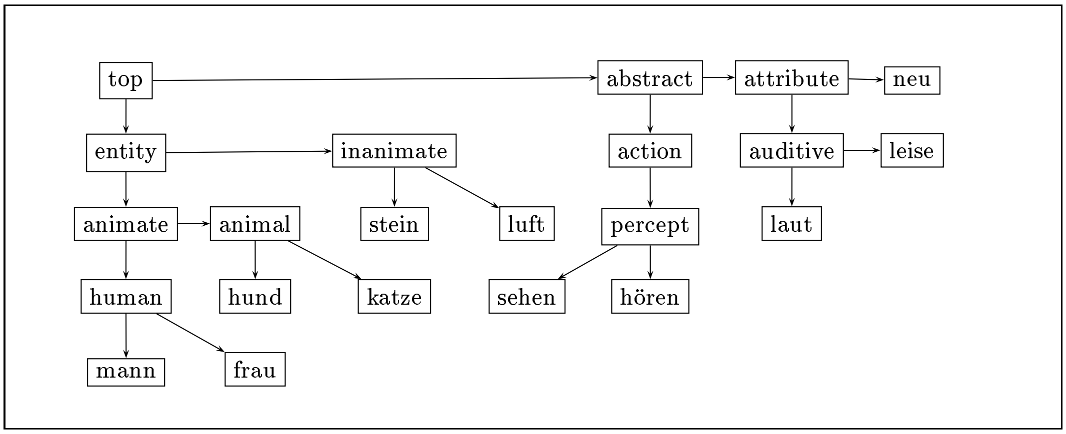


Abbildung 3.1: Ausschnitt einer Ontologie

3.1.2 Hierarchien

Hierarchien dienen hier der Definition von Ontologien, sodaß Abstraktionen von Begriffen bezeichnet werden können.[2] So ist es zum Beispiel wünschenswert, belebte und unbelebte Dinge oder Pflanzen, Tiere und Menschen zu unterscheiden. Einzelne Lexeme besitzen jeweils semantische Kategorien (*sem:cat*), die in der Ontologie vermerkt sein können. Die Verben wiederum notieren sortale Restriktionen im Lexikon , sodaß die semantischen Kategorien der Argumente entsprechend subsumiert werden. Im Verlauf des Kapitels 3.2 werden Constraints zur Formulierung sortaler Restriktionen genauer betrachtet. Abbildung 3.1 zeigt eine Beispielontologie, die auch später Verwendung findet. Diese Hierarchie dient lediglich den Anforderungen der Beispielgrammatik aus Kapitel 3.2; Anspruch auf Vollständigkeit wird nicht erhoben.

Auf die gegebene Ontologie sei ein zweistelliges Prädikat $subsumes(a, b)$ definiert, welches den booleschen Wert *true* liefert, falls die Kategorie a die Kategorie b subsumiert.

3.1.3 Bewertung von Belegungen

Wie bereits erwähnt, ist eine Vielzahl von Möglichkeiten gegeben, Variablenbelegungen eines *CSP* zu bewerten. Darum sei hier eine Möglichkeit erwähnt, die im Kontext dieser Arbeit verwendet wird.

Entscheidend für die Bewertung einer Belegung ist die Menge der verletzten Constraints. Jedem Constraint sei ein Gewicht zugeordnet, welches die Funktion eines Straffaktors besitzt, falls dieses Constraint verletzt ist. Somit wird hier das Parsingproblem als *MVP* modelliert, das heißt, das ein Constraint, welches den logischen Wert *false* liefert, den Fehlerfall, also ungewollte Variablenbelegungen, modelliert.

Seien die Gewichtsfunktionen der Constraints

$$w \;:\; C \times A \times A \;\mapsto\; [0;\ 1] \tag{3.8}$$

[2]Ontologien sind hier strikte Hierarchien.

frei definierbar. Allerdings sei

$$w(C_{x,y}, a_i, a_j) = 1 \quad \text{falls } \langle a_i, a_j \rangle \in C_{x,y} \tag{3.9}$$

immer gegeben, das heißt, das das Gewicht eines Constraints nur in Betracht gezogen wird, wenn es verletzt ist. Zur Bewertung einer Menge von Variablenbelegungen A von n Variablen sei g als Produkt aller Gewichte unärer und binärer Constraints definiert:

$$g(A) := (\prod_{a_i \in A} \prod_{C_x \in C} w(C_x, a_i, a_i)) * \tag{3.10}$$

$$(\prod_{a_i, a_j \in A, i \neq j} \prod_{C_{x,y} \in C} w(C_{x,y}, a_i, a_j)) \tag{3.11}$$

Da alle Gewichte in das Intervall [0; 1] abbilden, ist g monoton, das heißt, daß durch eine weitere Variablenbelegung die Gesamtbewertung nicht verbessert werden kann. Eine weitere Konsequenz der Wahl der Bewertungsfunktion g besteht darin, daß ein Constraint, welches mit dem Wert 0 gewichtet ist, eine Belegung gänzlich aus dem Suchraum ausschließt, falls das Constraint verletzt wird.

3.2 Eine Beispielgrammatik

Im folgenden wollen wir eine Dependenzgrammatik für einen minimalen Sprachausschnitt definieren, die in der Lage ist, die syntaktische Ambiguität von (1) aufgrund der sortalen Restriktionen des Verbs zu reduzieren. Wir betrachten eine Gruppe verschiedener Constraints, wie sie in ähnlicher Art in den meisten Constraint-Dependenzgrammatiken vorkommen.

(1) die Flöte hört die Frau.

Sei CDG_1 wie folgt definiert, wobei die einzelnen Constraints im Verlauf des Beispiels angegeben werden:

$$\begin{aligned}
\Sigma &:= \{\text{die, Frau, Flöte, hört}\}\\
R &:= \{syn, sem\}\\
L &:= \{det, obj, subj, verb, agens, theme, link\}\\
C &:= \{C_x^1, C_x^2, \ldots\}\\
\mathcal{D} &:= \{\text{die}_{nsf}, \text{die}_{asf}, \text{die}_{np}, \text{die}_{ap}, \text{hört}_{trsg}, \text{hört}_{trpl}, \text{Frau}_{sg}, \text{Flöte}_{sg}\}\\
\mathcal{H} &:= \text{Ontologie aus Abb. 3.1}
\end{aligned}$$

Die gewünschte Dependenzanalyse ist in Abbildung 3.2 dargestellt. So ist zum Beispiel die Flöte_{sg} auf der syntaktischen Ebene dem Wort hört_{trsg} als Objekt untergeordnet, sprich Flöte_{sg} modifiziert hört_{trsg} mit dem Label *obj*. Das Verb hört_{trsg} ist

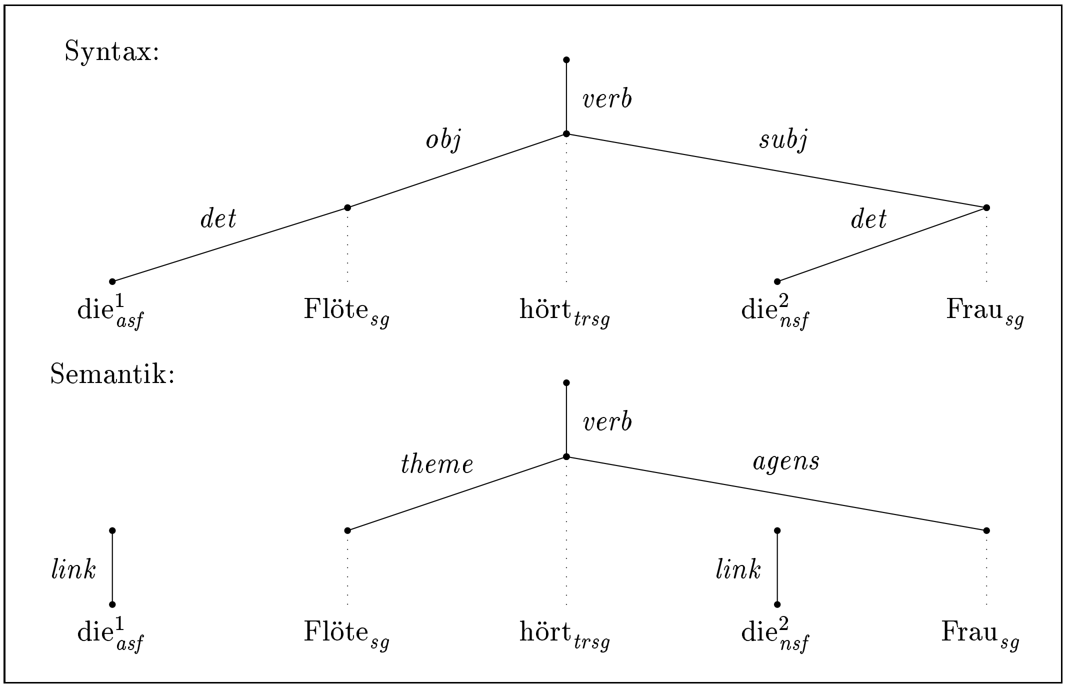


Abbildung 3.2: Lösung aus Beispiel 3.2

mit einer senkrechten Kante angebunden und bildet die Wurzel des Baumes, was als Bindung an *nil* dargestellt wird. Die vollständige Liste der Dependenzen ist gegeben durch:

$$
\begin{aligned}
A \;=\{\; & \langle \text{die}_{asf}, syn, det, \text{Flöte}_{sg}\rangle, \langle \text{Flöte}_{sg}, syn, obj, \text{hört}_{trsg}\rangle, \\
& \langle \text{hört}_{trsg}, syn, verb, nil\rangle, \langle \text{die}_{nsf}, syn, det, \text{Frau}_{sg}\rangle, \\
& \langle \text{Frau}_{sg}, syn, subj, \text{hört}_{trsg}\rangle, \langle \text{die}_{asf}, sem, link, nil\rangle, \\
& \langle \text{Flöte}_{sg}, sem, theme, \text{hört}_{trsg}\rangle, \langle \text{hört}_{trsg}, sem, link, nil\rangle, \\
& \langle \text{die}_{nsf}, sem, link, nil\rangle, \langle \text{Frau}_{sg}, sem, agens, \text{hört}_{trsg}\rangle \;\}
\end{aligned}
$$

Zu Beginn des Verfahrens erhalten wir vorerst die Menge aller möglichen Unterordnungsbeziehungen, kompakt dargestellt als Constraintnetz, wie es in Abbildung 3.3 angedeutet ist. Hier ist zu sehen, daß pro Wort des Satzes alle lexikalischen Lesarten aus dem Lexikon extrahiert werden und jeweils einen eigenen Knoten des Netzes bilden. An jeder der dargestellten Kanten wäre die Menge aller möglichen Labels zu verzeichnen, worauf der Übersicht halber verzichtet wurde.

3.2.1 Wortklassen

Als erstes wenden wir Constraints zur Beschreibung der Wortklassen an. So ist eine Artikelanbindungen definiert als eine Beziehung zwischen zwei Worten mit dem Label *det*, wobei der Modifier ein Artikel und der Modifiee ein Nomen ist. Formell ließe

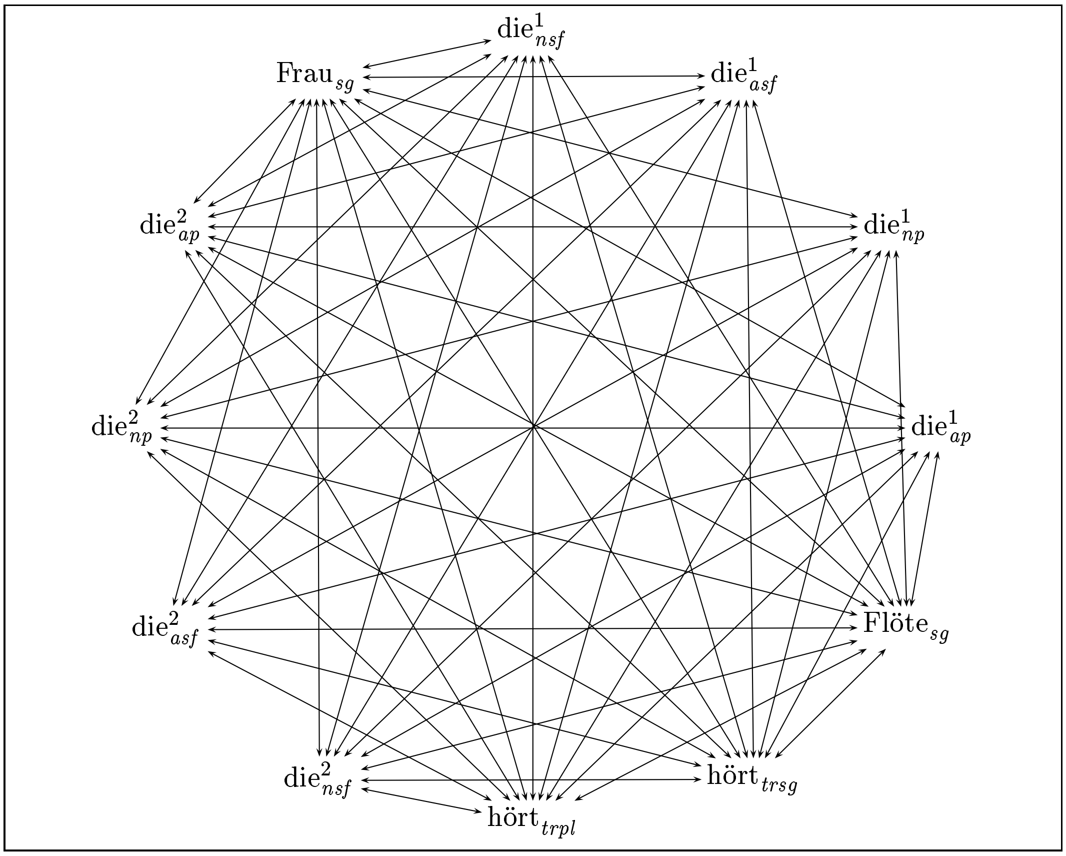


Abbildung 3.3: Initiales Constraintnetz der Syntax-Ebene

sich dies in etwa durch folgende Formel[3] ausdrücken:

$$\begin{aligned} C^1_x \quad := \quad & x.role = syn \Rightarrow \\ & (\ x.label = det \ \Leftrightarrow \ x{\downarrow}syn{:}cat = article \ \wedge \ x{\uparrow}syn{:}cat = noun\ ) \\ & w(C^1_x, x, x) = 0 \end{aligned}$$

Die Verwendung des Terms *x.role* liefert die Rollenzugehörigkeit einer Constraintkante. Dieses Constraint ist mit dem Straffaktor 0 versehen, da Constraint C^1_x, so wie alle Constraints zur Beschreibung von Wortarten, *hart* ist, also eine obligatorische Forderung ausdrückt. Gleichermaßen definieren wir Subjekte und Objekte als Beziehungen zwischen Nomen und Verben. Finite Verben bilden die Wurzel des Dependenzbaumes auf der Syntaxebene:

$$\begin{aligned} C^2_x \quad := \quad & x.role = syn \Rightarrow \\ & (\ x.label = verb \Leftrightarrow \ x{\downarrow}syn{:}cat = verb \ \wedge \ root(x{\uparrow}id)\ ) \\ C^3_x \quad := \quad & x.role = syn \ \wedge \ x.label = subj \end{aligned}$$

[3]Die Syntax der Constraint-Logik sei hier nur soweit erläutert, daß ein intuitives Verständnis ermöglicht wird; näheres dazu in Schröder, Foth, und Schulz (1999). Falls im Verlaufe des Beispiels Besonderheiten auftreten sollten, so wird darauf nochmals speziell eingegangen.

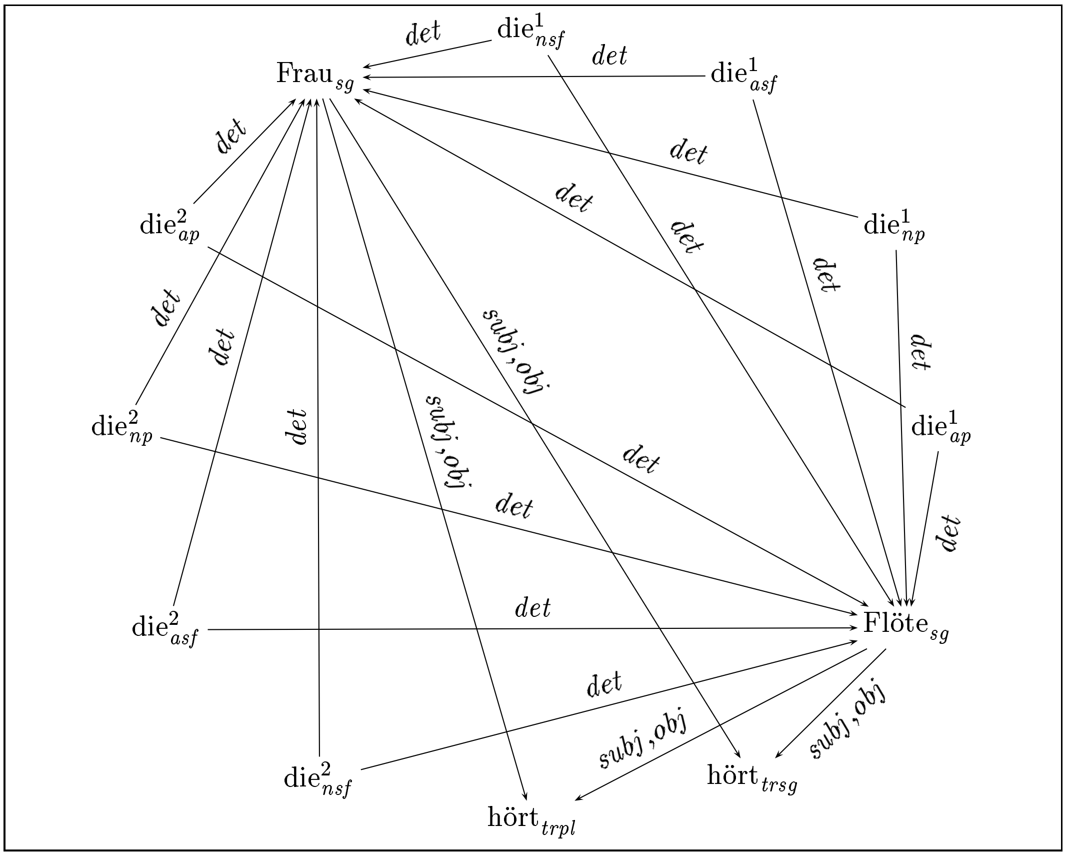


Abbildung 3.4: Constraintnetz nach der Auswertung von C_x^1, C_x^2, C_x^3 und C_x^4

$$
\begin{array}{rcl}
 & & \Rightarrow\ x{\downarrow}syn{:}cat = noun\ \wedge\ x{\uparrow}syn{:}cat = verb \\
C_x^4 & := & x.role = syn\ \wedge\ x.label = obj \\
 & & \Rightarrow\ x{\downarrow}syn{:}cat = noun\ \wedge\ x{\uparrow}syn{:}cat = verb
\end{array}
$$

$$
\begin{array}{l}
w(C_x^2, x, x) = 0 \\
w(C_x^3, x, x) = 0 \\
w(C_x^4, x, x) = 0
\end{array}
$$

Hier gebrauchen wir ein Prädikat namens „*root*()“, welches den booleschen Wert *true* liefert, falls das übergebene Argument eine Wurzelanbindung ist. Damit reduzieren sich die möglichen Variablenbelegungen auf die in Abbildung 3.4 angegebenen Werte. Auf die Darstellung der Anbindungen möglicher Verben wurde hier verzichtet.

3.2.2 Kongruenzen

Aus der Abbildung 3.4 ist zu entnehmen, daß mit den bis jetzt vorhandenen Constraints noch nicht entschieden werden konnte, welcher Artikel zu welchem Nomen gehört, welches Nomen das Subjekt und welches das Objekt des Satzes bildet. Ganz bewußt wurde in den ersten Constraints noch nicht vollständig spezifiziert, was ein

Subjekt ausmacht. Wir verteilen zur besseren Übersicht das grammatikalische Wissen auf mehrere Constraints, was ebenfalls den Vorteil hat, daß sie einzeln gewichtet werden können. Zur weiteren strukturellen Disambiguierung definieren wir also die Kongruenzeigenschaften der Nomen modifizierenden Wortarten wie Artikel, Adjektive und Partizipien:

$$\begin{aligned}
C_x^5 &:= x.role = syn \;\wedge\; x{\uparrow}syn{:}cat = noun \\
&\quad \Rightarrow\; x{\downarrow}syn{:}num = x{\uparrow}syn{:}num \\
C_x^6 &:= x.role = syn \;\wedge\; x{\uparrow}syn{:}cat = noun \\
&\quad \Rightarrow\; x{\downarrow}syn{:}gen = x{\uparrow}syn{:}gen \\
C_x^7 &:= x.role = syn \;\wedge\; x{\uparrow}syn{:}cat = noun \\
&\quad \Rightarrow\; x{\downarrow}syn{:}case = x{\uparrow}syn{:}case \\
&\quad w(C_x^5, x, x) = 0.16 \\
&\quad w(C_x^6, x, x) = 0.17 \\
&\quad w(C_x^7, x, x) = 0.18
\end{aligned}$$

Ebenso müssen die Kongruenzeigenschaften zwischen dem Verb und seinen Argumenten, dem Subjekt und seinen Objekten, erfüllt sein.

$$\begin{aligned}
C_x^8 &:= x.role = syn \;\wedge\; x.label = subj \\
&\quad \Rightarrow\; x{\downarrow}syn{:}pers = x{\uparrow}syn{:}pers \\
C_x^9 &:= x.role = syn \;\wedge\; x.label = subj \\
&\quad \Rightarrow\; x{\downarrow}syn{:}num = x{\uparrow}syn{:}num \\
C_x^{10} &:= x.role = syn \;\wedge\; x.label = subj \\
&\quad \Rightarrow\; x{\downarrow}syn{:}case = x{\uparrow}arg1{:}case \\
C_x^{11} &:= x.role = syn \;\wedge\; x.label = obj \\
&\quad \Rightarrow\; x{\downarrow}syn{:}case = x{\uparrow}arg2{:}case \\
&\quad w(C_x^8, x, x) = 0.15 \\
&\quad w(C_x^9, x, x) = 0.16 \\
&\quad w(C_x^{10}, x, x) = 0.18 \\
&\quad w(C_x^{11}, x, x) = 0.18
\end{aligned}$$

Man beachte bei dieser Gruppe die letzten beiden Constraints C_x^{10} und C_x^{11}, die die Kasusforderung eines Verbs an seine Argumente implementieren. Der entsprechende

Lexikoneintrag des Wortes hört$_{trsg}$ gestaltet sich wie folgt:

$$\text{hört}_{trsg} := \begin{bmatrix} word & : & \text{hört} \\ syn & : & \begin{bmatrix} cat & : & verb \\ num & : & sg \\ pers & : & 3 \\ form & : & finit \\ temp & : & präsens \end{bmatrix} \\ sem & : & \begin{bmatrix} cat & : & \text{hören} \end{bmatrix} \\ arg1 & : & \begin{bmatrix} case & : & nom \\ sort & : & animate \\ role & : & agens \end{bmatrix} \\ arg2 & : & \begin{bmatrix} case & : & acc \\ sort & : & entity \\ role & : & theme \end{bmatrix} \end{bmatrix}$$

Auf diese Weise lassen sich im Lexikon verschiedene Kasusrahmen und damit verbundene Rollenzuweisungen definieren, die dann mit Hilfe entsprechender allgemeiner Constraints, wie zum Beispiel C_x^{10}, wirken.[4] Ebenfalls haben wir unterschiedliche Staffaktoren für Numerus-, Kasus-, Genus- und Person-Verletzungen eingeführt, sodaß eine gewisse Toleranz bzgl. syntaktischer Fehler gegeben ist.

3.2.3 Projektion

Wenn man sich das Problem der Artikelanbindung genauer anschaut, so fällt auf, daß einige Anbindungen gewisse Projektionseigenschaften des Dependenzbaumes verletzen. Wir formulieren deshalb ein Constraint, welches verbietet, eine Konstituente *innerhalb* ihrer selbst anzubinden. Dies wäre nämlich der Fall, wenn die$^1_{nsf}$ als Artikel zu Frau$_{sg}$ analysiert würde und man zusätzlich versuchte, diese Konstituente als Bestandteil des Verbs einzubinden, denn das Verb steht in diesem Satz *zwischen* dem Artikel und dem zugehörigen Nomen. Diese Beobachtung läßt sich mit folgendem Constraint verallgemeinern:

$$\begin{aligned} C_{x,y}^{12} \quad := \quad & x.role = y.role \\ & \wedge\ x{\uparrow}id = y{\downarrow}id \ \wedge\ \neg\ root(y{\uparrow}id) \\ & \Rightarrow\ y{\uparrow}from < min(x{\downarrow}from, y{\downarrow}from) \\ & \vee\ y{\uparrow}from > max(x{\downarrow}from, y{\downarrow}from) \\ & w(C_x^{12}, x, y) = 0 \end{aligned}$$

Betrachten wir dieses Constraint genauer, so sehen wir, daß durch die Prämisse der Implikation zwei zusammenhängende Kanten x und y betrachtet werden, wobei die

[4] Die Benennung der thematischen Rollen im Lexikon ist nicht zu verwechseln mit den Rollen des *CDGs*.

obere Kante y keine Wurzelanbindung darstellen soll. Gefordert wird dann, daß das Wort am oberen Ende der y-Kante entweder vor dem vordersten oder nach dem letzten Wort angebunden wird, was der Anschauung von *außerhalb* entspricht. Die Terme „$y{\uparrow}from$" bzw. „$y{\downarrow}from$" seien hier so verstanden, daß die Position des Wortes im Satz gemeint ist.[5] Damit ist in jedem Fall gewährleistet, daß Variablenbelegungen wie

$$\begin{array}{rcl} A' & = & \{\ldots, \langle syn, \mathrm{die}^1_{nsf}, det, \mathrm{Frau}_{sg}\rangle, \langle syn, \mathrm{Frau}_{sg}, subj, \mathrm{h\ddot{o}rt}_{trsg}\rangle, \ldots\} \text{ oder} \\ A'' & = & \{\ldots, \langle syn, \mathrm{die}^2_{asf}, det, \mathrm{Fl\ddot{o}te}_{sg}\rangle, \langle syn, \mathrm{Fl\ddot{o}te}_{sg}, obj, \mathrm{h\ddot{o}rt}_{trsg}\rangle, \ldots\} \end{array}$$

nicht vorgenommen werden.

Ebenso wie sich Projektionseigenschaften zweier verbundener Kanten formulieren lassen, muß für zwei unverbundene Kanten eines Dependenzbaumes gelten, daß eine Kante die andere vollständig überspannt. Betrachten wir den Satz „der große rote Ball gehört Peter", so ist es nicht möglich, die Adjektive der Subjekt-NP an „Peter" anzubinden und gleichzeitig den Artikel dem Wort „Ball" zuzuordnen (vgl. Abbildung 3.8). Diese beiden Kanten überkreuzen sich. Die Artikelkante überspannt hier die gesamte NP und bewirkt, daß alle Adjektive innerhalb dieser Grenzen angebunden werden.

$$\begin{array}{rcl} C^{13}_{x,y} & := & x.role = y.role \\ & & \quad \wedge\ x{\downarrow}id \neq y{\uparrow}id \ \wedge\ x{\uparrow}id \neq y{\downarrow}id \ \wedge\ x{\uparrow}id \neq y{\uparrow}id \\ & & \quad \wedge\ \neg\ root(x{\uparrow}id)\ \wedge\ \neg\ root(y{\uparrow}id) \\ & & \quad \wedge\ min(x{\downarrow}from, x{\uparrow}from) < min(y{\downarrow}from, y{\uparrow}from) \\ & & \Rightarrow\ max(x{\downarrow}from, x{\uparrow}from) > max(y{\downarrow}from, y{\uparrow}from) \\ & & \quad \vee\ max(x{\downarrow}from, x{\uparrow}from) < min(y{\downarrow}from, y{\uparrow}from) \\ & & w(C^{13}_{x}, x, y) = 0 \end{array}$$

Betrachten wir nun das verbleibende Netz in Abbildung 3.5. Wir sehen, daß durch das sukzessive Eliminieren von Kanten, die Constraints verletzen, mehrere Constraintknoten ohne jegliche Möglichkeit, an einer Lösung beteiligt zu werden, übrig geblieben sind und deshalb gelöscht wurden.

3.2.4 Thematische Rollen

Um nun die verbleibenden Ambiguitäten zu lösen, ist es notwendig, eine semantische Ebene zu definieren, die aufgrund der sortalen Restriktionen des Verbs die thematischen Rollen *agens* und *theme* korrekt zuweist. Bis jetzt enthielt unser Constraintnetz lediglich Knoten für syntaktische Wertebelegungen. Für jede definierte Rolle

[5] Tatsächlich werden durch die zugrundeliegende Implementation Wortgraphen verarbeitet und *from* und *to* denotieren Start- und Endpunkt einer Kanten des Wortgraphen, an welcher ein Wort notiert ist. Auf diese Weise sind komplexe Satzhypothesen darstellbar, wobei sich alternative Sätze durch verschiedene Pfade im Wortgraphen ergeben. Eine nähere Betrachtung der Verarbeitung von Wortgraphen wird hier nicht vorgenommen. Es werden nur lineare Wortgraphen behandelt, also solche, die nur eine einzige Satzalternative darstellen.

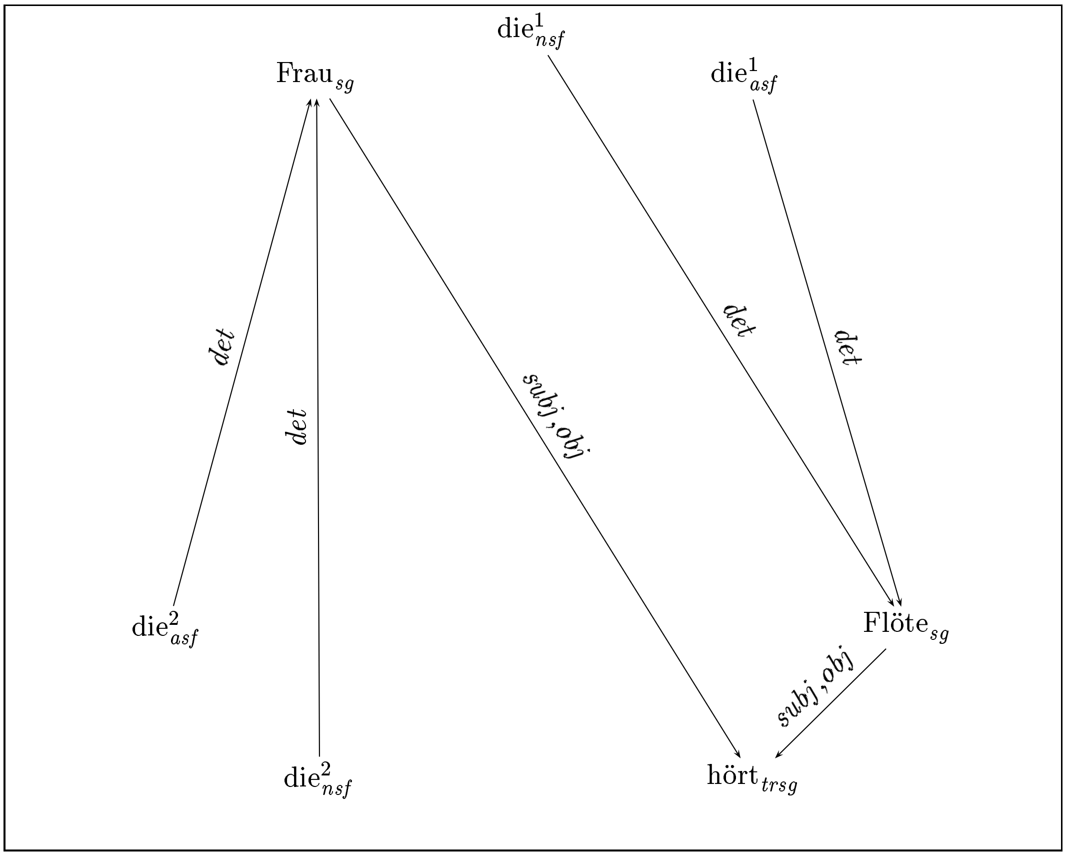


Abbildung 3.5: Constraintnetz nach der Auswertung der Constraints bis einschließlich $C^{13}_{x,y}$

sind diese Knoten jetzt zu verdoppeln, um neben syntaktischen auch semantische Bewertungen vorzunehmen. Der Übersicht halber sei hier auf die gemeinsame Darstellung aller Knoten verzichtet. Wir betrachten vorerst nur die Kanten und Knoten der semantischen Rolle des Constraintnetzes. Es sei jedoch bemerkt, daß, alle Rollen in einem einzigen Constraintnetzes gemeinsam ausgewertet werden, obwohl dies hier nicht den Anschein haben mag. Die Rollenzugehörigkeit jedes Knotens ist eindeutig markiert. In Abschnitt 3.2.5 werden wir sehen, wie Knoten, die sich auf das selbe Wort beziehen, aber verschiedenen Rollen angehören, miteinander in Verbindung treten.

Definieren wir also die grundlegenden Dependenzeigenschaften der Semantikebene:

$$\begin{aligned}
C^{14}_x &:= \mathit{x.role = sem} \;\wedge\; \mathit{x.label = agens} \\
&\; \Rightarrow\; \mathit{x{\uparrow}arg1{:}role = agens} \\
C^{15}_x &:= \mathit{x.role = sem} \;\wedge\; \mathit{x.label = agens} \\
&\; \Rightarrow\; \mathit{subsumes(x{\uparrow}arg1{:}sort, x{\downarrow}sem{:}cat)} \\
C^{16}_x &:= \mathit{x.role = sem} \;\wedge\; \mathit{x.label = theme} \\
&\; \Rightarrow\; \mathit{x{\uparrow}arg2{:}role = theme} \\
C^{17}_x &:= \mathit{x.role = sem} \;\wedge\; \mathit{x.label = theme}
\end{aligned}$$

$$\Rightarrow \; subsumes(x{\uparrow}arg2{:}sort, x{\downarrow}sem{:}cat)$$

$$w(C_x^{14}, x, x) = 0$$
$$w(C_x^{15}, x, x) = 0.2$$
$$w(C_x^{16}, x, x) = 0$$
$$w(C_x^{17}, x, x) = 0.2$$

Hier gebrauchen wir zum ersten Mal die $subsumes(a, b)$ Relation, die im Lexikoneintrag die gewünschte Rollenverteilung ermittelt und mit der semantischen Kategorie des angebundenen Arguments vergleicht. Auch hier verteilen wir die Beschreibung des Agens auf die Constraints C_x^{14} und C_x^{15}, um später eine getrennte Gewichtung der Constraints vornehmen zu können. Wärend C_x^{14} und C_x^{16} nämlich strikt bindend sind, wollen wir robust bzgl. sortaler Verletzungen sein.

Neben den *agens*- und *theme*-Anbindungen werden Artikel und Verben hier nicht weiter spezifiziert und per Default an *nil* gebunden.

$$\begin{aligned} C_x^{18} \;:=\;& x.role = sem \;\wedge \\ & (x{\downarrow}syn{:}cat = verb \;\vee\; x{\downarrow}syn{:}cat = nil) \\ & \Rightarrow \; root(x{\uparrow}id) \\ C_x^{19} \;:=\;& x.role = sem \Rightarrow \\ & x.label = link \;\Leftrightarrow\; root(x{\uparrow}id) \end{aligned}$$

$$w(C_x^{18}, x, x) = 0.00001$$
$$w(C_x^{19}, x, x) = 0$$

Die Beschreibung der *nil*-Bindungen auf der semantischen Ebene sind hier durch das Constraint C_x^{18} etwas aufgelockert, sodaß der Parser die Analyse nicht vollständig abbricht, sondern im Zweifelsfall ein Lexem auf der Semantikebene mit *nil* unter *root* anbinden kann, obwohl es eigentlich semantische Informationen mit sich bringt. Betrachten wir Abbildung 3.6, so sehen wir, daß auf der semantischen Ebene die thematischen Rollen eindeutig zugeordnet werden können.

3.2.5 Zusammenhang der Analyseebenen

Verbleibt nun die Frage, wie die Erkenntnisse der Semantikebene zur syntaktischen Disambiguierung genutzt werden können. Hierzu definieren wir sogenannte Mapping-Constraints, die Aussagen über das Verhältnis zweier Kanten der Syntax und der Semantik formulieren. Wir möchten die Subjekt-Kante auf die Agens-Kante und die Objekt-Kante auf die Theme-Kante abbilden.

$$\begin{aligned} C_{x,y}^{20} \;:=\;& x.role = syn \;\wedge\; y.role = sem \Rightarrow \\ & (\; x.label = subj \;\wedge\; x{\downarrow}id = y{\downarrow}id \\ & \quad \Rightarrow \; y.label = agens \;\wedge\; x{\uparrow}id = y{\uparrow}id \;) \;\wedge \\ & (\; y.label = agens \;\wedge\; x{\downarrow}id = y{\downarrow}id \end{aligned}$$

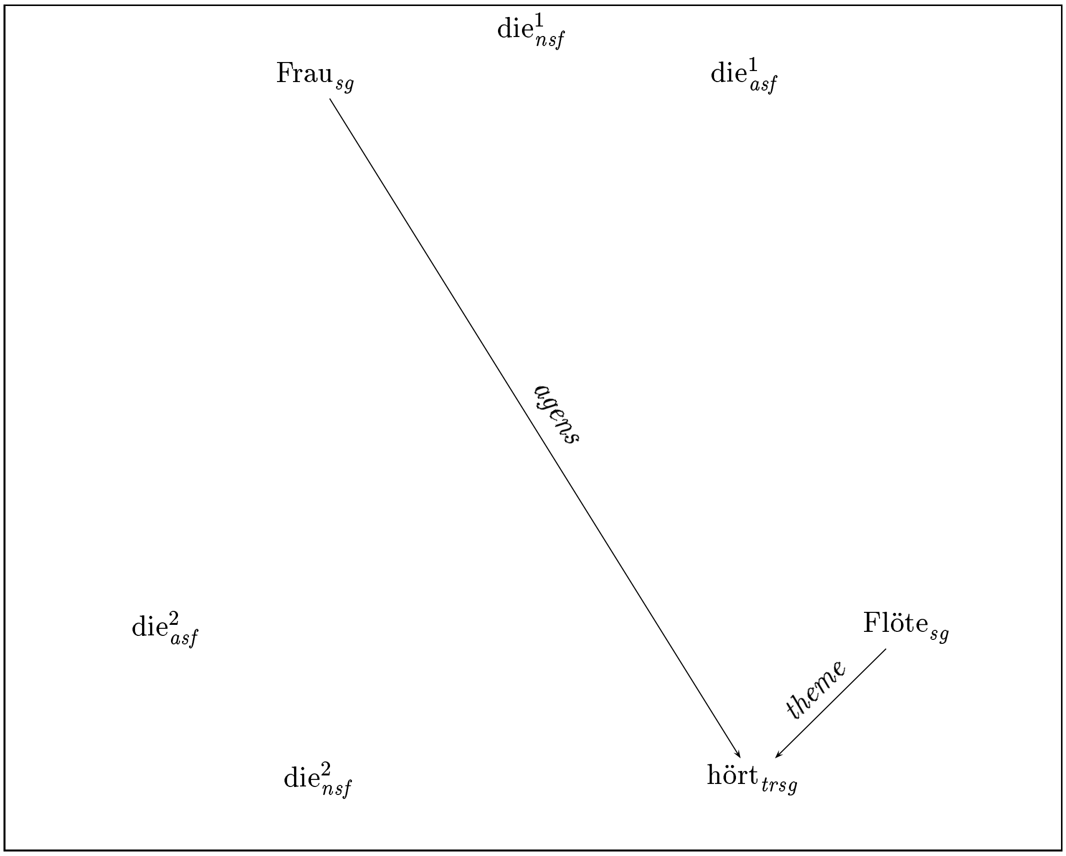


Abbildung 3.6: Constraintnetz mit Knoten der semantischen Rolle

$$
\begin{array}{lcl}
 & & \quad \Rightarrow\ x.label = subj\ \wedge\ x{\uparrow}id = y{\uparrow}id\ ) \\
C^{21}_{x,y} & := & x.role = syn\ \wedge\ y.role = sem \Rightarrow \\
 & & (\ x.label = obj\ \wedge\ x{\downarrow}id = y{\downarrow}id \\
 & & \quad \Rightarrow\ \ y.label = theme\ \wedge\ x{\uparrow}id = y{\uparrow}id\ )\ \wedge \\
 & & (\ y.label = theme\ \wedge\ x{\downarrow}id = y{\downarrow}id \\
 & & \quad \Rightarrow\ x.label = obj\ \wedge\ x{\uparrow}id = y{\uparrow}id\ ) \\
 & & w(C^{20}_x, x, y) = 0 \\
 & & w(C^{21}_x, x, y) = 0
\end{array}
$$

In Constraint $C^{20}_{x,y}$ werden zwei Implikationen konjunktiv verknüpft, sodaß gewährleistet ist, daß aus jeder Ebene Information in die andere fließen kann. Die jeweiligen Prämissen der Implikationen gehen davon aus, daß zwei gleiche Lexeme und der Label der Kante betrachtet werden, und schließen auf den Label der abzubildenden Kante der anderen Ebene und die Identität der Lexeme in Modifiee-Position.

Treten auf beiden Ebenen gegensätzliche Erkenntnisse auf, so setzt sich diejenige durch, deren Constraints härter verletzt würden, falls sie zur Aufgabe ihrer Bedingungen gezwungen wäre. Es wäre ebenfalls vorstellbar, $C^{20}_{x,y}$ und $C^{21}_{x,y}$ jeweils an der Konjunktion der Implikationen aufzuspalten und die getrennten Constraints unter-

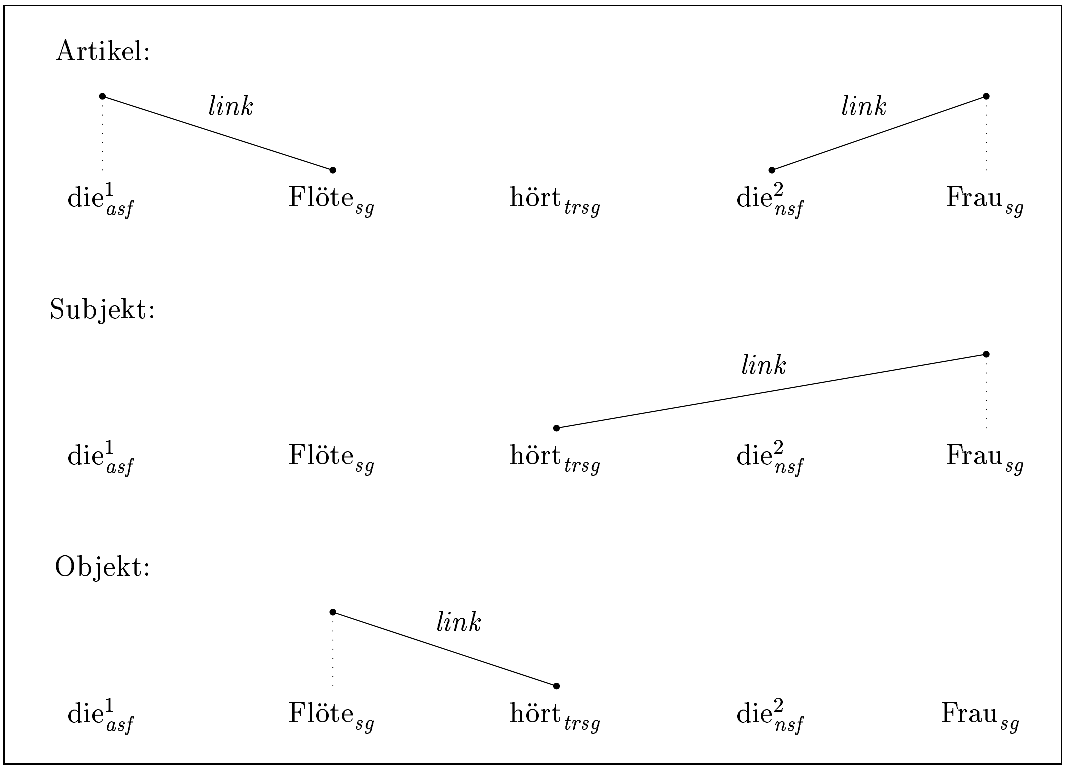


Abbildung 3.7: Spiegelebenen zu Abbildung 3.2

schiedlich zu bewerten, sodaß auf diese Weise zusätzlich das Verhältnis der Ebenen beeinflußt werden kann.

3.2.6 Existenzforderungen

Wir wollen in diesem Abschnitt ein Konstrukt betrachten, welches zur Forderung obligatorischer Bindungen notwendig ist. Verben zum Beispiel benötigen in diesem Sinne obligatorisch ein Subjekt und in Abhängigkeit vom definierten Kasusrahmen evtl. erwartete Argumente. Ein anderes Beispiel stellen Nomina im Singular dar. Sie benötigen obligatorisch einen Artikel. Zur Gewährleistung der Existenz ist es allerdings notwendig, zusätzliche Rollen zu definieren, die die geforderte Bindungen in gespiegelter Richtung vornehmen. Deshalb nennt man solche Rollen auch Spiegelebenen oder Stützebenen. Abbildung 3.7 stellt die Spiegelung der Artikelbindungen, des Subjekts sowie des direkten Objekts des Satzes (1) dar. Diese Spiegelebenen sind lediglich technisch motiviert und dienen nicht der Interpretation wie die Syntax- und Sematikebenen.

Sei die Menge der Rollen erweitert um die Rollen *det*, *subj* und *obj*, die nicht mit den entsprechenden Labeln der Syntaxebene zu verwechseln sind. Folgende Constraints dienen der eindeutigen Spiegelbindung von Artikelbindungen:

$$C^{22}_x \quad := \quad x.role = det \Rightarrow$$

$$
\begin{array}{rcl}
 & & (\neg\ root(x{\uparrow}id) \ \Leftrightarrow\ x{\uparrow}syn{:}cat = article \ \wedge\ x{\downarrow}syn{:}cat = noun\ ) \\
C^{23}_{x,y} & := & x.role = det \ \wedge\ \neg\ root(x{\uparrow}id) \ \wedge\ \neg\ root(y{\uparrow}id) \\
 & & \Rightarrow\ x{\uparrow}id \neq y{\uparrow}id \\
 & & w(C^{22}_x, x, x) = 0 \\
 & & w(C^{23}_x, x, x) = 0
\end{array}
$$

Constraint C^{22}_x definiert Artikelspiegelkanten als solche, die ein Nomen in Modifiee-Position und einen Artikel in Modifier-Position besitzen. Constraint $C^{23}_{x,y}$ ist notwendig, damit nicht zwei Nomen auf der *det*-Ebene einem Artikel zugeordnet werden.

Jetzt ist es möglich, das zentrale Constraint der *det*-Ebene zu formulieren:

$$
\begin{array}{rcl}
C^{24}_x & := & x.role = det \\
 & & \wedge\ x{\downarrow}syn{:}cat = noun \ \wedge\ x{\downarrow}syn{:}num = sg \\
 & & \Rightarrow\ \neg\ root(x{\uparrow}id) \\
 & & w(C^{24}_x, x, x) = 0.01
\end{array}
$$

In anderen Worten: ein Nomen im Singular darf nicht an Root gebunden werden.[6] Sind derartige Kanten definiert, verbleibt nur noch die eindeutige Verbindung der Artikelkante auf der Syntaxebene und der Spiegelebene.

$$
\begin{array}{rcl}
C^{25}_{x,y} & := & x.role = syn \ \wedge\ y.role = det \Rightarrow \\
 & & (\ x.label = det \ \wedge\ x{\uparrow}id = y{\downarrow}id \Leftrightarrow\ x{\downarrow}id = y{\uparrow}id\ ) \\
 & & w(C^{25}_x, x, x) = 0
\end{array}
$$

Man beachte, daß auf der *det*-Ebene keine Kongruenzforderungen in Form von einstelligen Constraints aufgestellt werden. Aus Performanzgründen könnte man die entsprechenden Bedingungen der Syntaxebene ebenfalls auf der *det*-Ebene fordern, um bereits in der Generate-Phase des Constraintnetzes Mehrdeutigkeiten auszuschließen. Der Nachteil, der dabei entsteht ist, daß im Falle einer Kongruenzverletzung zwei Gewichtungen ins Spiel kommen, wo vorher nur ein Constraint verletzt war und somit ungewollt aus einem einzigen Grund zwei Constraints verletzt werden. Zweitens haben wir Constraint C^{24}_x *weich* formuliert (Straffaktor ≥ 0), sodaß Sätze wie „Pferd frißt Gras.“ zwar bestraft, aber nicht unmöglich werden.

Der Vollständigkeit halber seien die Constraints der *subj*-Ebene und der *obj*-Ebene angegeben. Sie unterscheiden sich nicht wesentlich von der *det*-Ebene.

$$
\begin{array}{rcl}
C^{26}_x & := & x.role = subj \Rightarrow \\
 & & (\ x{\uparrow}syn{:}cat = noun \ \wedge\ x{\downarrow}syn{:}cat = verb \ \Leftrightarrow\ \neg\ root(x{\uparrow}id)\ ) \\
C^{27}_x & := & x.role = subj \Rightarrow \\
 & & (\ x{\downarrow}syn{:}cat = verb \ \wedge\ exists(x{\downarrow}arg1{:}sort) \ \Leftrightarrow\ \neg\ root(x{\uparrow}id)\ )
\end{array}
$$

[6] „Bindung an *root*“ ist eine andere Sprechweise für Wurzelbindungen oder „Bindung an *nil*“.

$$
\begin{aligned}
C^{28}_{x,y} \quad &:= \quad x.role = syn \ \wedge \ y.role = subj \Rightarrow \\
&\quad\ (\ x.label = subj \ \wedge \ x{\uparrow}id = y{\downarrow}id \ \Leftrightarrow \ x{\downarrow}id = y{\uparrow}id\ )
\end{aligned}
$$

$$
\begin{aligned}
&w(C^{26}_x, x, x) = 0 \\
&w(C^{27}_x, x, x) = 0.01 \\
&w(C^{28}_x, x, y) = 0
\end{aligned}
$$

Der Term $exists(x{\downarrow}arg1{:}sort)$ dient lediglich der Überprüfung, ob das Verb ein Feature *arg1* hat. Die tatsächliche Eintragung des Features *sort* ist nicht entscheidend. Allerdings ist es notwendig, alle Verben mit einem entsprechenden Lexikoneintrag zu versehen. Entsprechend führt ein Lexikoneintrag *arg2* zur Existenzforderung eines Objekts auf der *obj*-Ebene.

$$
\begin{aligned}
C^{29}_{x} \quad &:= \quad x.role = obj \Rightarrow \\
&\quad\ (\ x{\uparrow}syn{:}cat = noun \ \wedge \ x{\downarrow}syn{:}cat = verb \ \Leftrightarrow \ \neg\ root(x{\uparrow}id)\ ) \\
C^{30}_{x} \quad &:= \quad x.role = obj \Rightarrow \\
&\quad\ (\ x{\downarrow}syn{:}cat = verb \ \wedge \ exists(x{\downarrow}arg2{:}sort) \ \Leftrightarrow \ \neg\ root(x{\uparrow}id)\ ) \\
C^{31}_{x,y} \quad &:= \quad x.role = syn \ \wedge \ y.role = obj \Rightarrow \\
&\quad\ (\ x.label = obj \ \wedge \ x{\uparrow}id = y{\downarrow}id \ \Leftrightarrow \ x{\downarrow}id = y{\uparrow}id\ )
\end{aligned}
$$

$$
\begin{aligned}
&w(C^{29}_x, x, x) = 0 \\
&w(C^{30}_x, x, x) = 0.01 \\
&w(C^{31}_x, x, y) = 0
\end{aligned}
$$

3.2.7 Distanz-Constraints

Wir betrachten einen Typ von Constraints, der eine Bindung um so mehr bestraft, je weiter Modifier und Modifiee voneinander entfernt sind. Hierdurch wird versucht zu modellieren, daß von einigen Menschen unter Umständen lange Dependenzen schwerer zu verstehen sind als kurze.[7] Solche Phänomene lassen sich allerdings nur dadurch ausdrücken, daß Constraintgewichtungen *dynamisch* berechnet werden. Deshalb seien solche Constraints als *Distanz-Constraints* bezeichnet. Im hiesigen Kontext sind Distanz-Constraints vorerst nicht motiviert, spielen aber in Kapitel 4 eine entscheidende Rolle. Distanz-Constraints sind Teil des in Schröder, Foth, und Schulz (1999) beschriebenen Formalismus.

Betrachten wir beispielhaft ein Constraint, welches Artikelbindung aufgrund ihrer Länge bestraft:

$$C^{32}_{x} \quad := \quad x.role = syn \ \wedge \ x.label = det \ \Rightarrow \ length(x) \leq 1$$

Wir verwenden hier eine Funktionen $length()$, die den Abstand von Modifier und Modifiee einer Dependenz mißt. Dieses Constraint also immer dann verletzt, wenn

[7] Diese Behauptung ist nicht durch psycholinguistische Experimente fundiert und nur spekulativer Natur, um die Möglichkeiten des Formalismus zu demonstrieren.

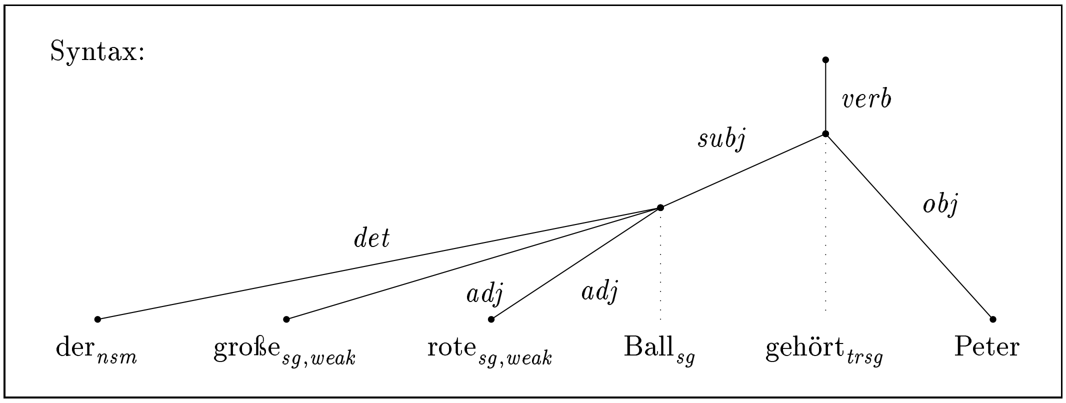


Abbildung 3.8: Beispiel für Distanz-Constraints

eine Entfernung größer als eins überschritten wird. In diesem Fall wird der Straffaktor von C_x^{32} wie folgt berechnet:

$$w(C_x^{32}, x, x) := \frac{1}{length(x)} \tag{3.12}$$

Betrachten wir den Satz

(2) der große rote Ball gehört Peter

so ergibt eine Constraintauswertung der Analyse in Abbildung 3.8 eine Bewertung von $g(A) = 1/3$, da obiges Constraint verletzt ist. Je nach Gewichtung der anderen Constraints einer Grammatik ist eine Dämpfung der Zunahmen des Straffaktors vorzunehmen, damit eine große Nominalphrase richtig analysiert wird und nicht statt einer gestreckten Dependenz eine anderweitige Verletzung in Kauf genommen wird.

3.2.8 Präferenzen

Eine Gruppe weiterer Constraints läßt sich als sogenannte Präferenz-Constraints bezeichnen. Ihr Zweck besteht darin, lexikalische Ambiguitäten zu bewerten, was den Effekt hat, daß der Parser vermeintlich bessere Lesarten eines Wortes bevorzugt, anstatt alle Lesarten gleichwertig zu behandeln. Letzteres hat in einem Branch & Bound Suchverfahren den Effekt der Verbreiterung des Suchraumes zur Folge. Andererseits tritt ein Artefakt solcher Constraints auf, daß der Parser einer Präferenz folge leistet, aber zu einem späteren Zeitpunkt merkt, daß die Lösung des *CSPs* dieser Präferenz widerspricht. Hier sind je nach Grammatik und analysierten Sätzen die Vor- und Nachteile abzuwägen. Die linguistische Motivation solcher Constraints neben der technischen ist zumindest umstrite. Erste Experimente mit derartigen Constraints haben allerdings gezeigt, daß der Suchaufwand mit geeigneten Präferenz-Constraints erheblich verringert werden konnte.

Formulieren wir beispielhaft eine Gruppe von Constraints, die ausdrücken sollen, daß überall Nominative vor Akkusativen, Dativen und Genitiven präferiert werden:

$$
\begin{array}{lcl}
C_x^{33} & := & x.role = syn \ \wedge \ exists(x{\downarrow}syn{:}case) \ \wedge \ x{\downarrow}syn{:}case = acc \\
 & & \Rightarrow \ false \\
C_x^{34} & := & x.role = syn \ \wedge \ exists(x{\downarrow}syn{:}case) \ \wedge \ x{\downarrow}syn{:}case = dat \\
 & & \Rightarrow \ false \\
C_x^{35} & := & x.role = syn \ \wedge \ exists(x{\downarrow}syn{:}case) \ \wedge \ x{\downarrow}syn{:}case = gen \\
 & & \Rightarrow \ false \\
 & & w(C_x^{33}, x, x) = 0.999 \\
 & & w(C_x^{34}, x, x) = 0.998 \\
 & & w(C_x^{35}, x, x) = 0.997
\end{array}
$$

Hier sehen wir, daß die bevorzugte Lesart eines Lexems durch eine geringere Bestrafung des entsprechenden Constraints ausgedrückt wird. Da der Nominativ die von allen Fällen präferierte Lesart ist, existiert kein Constraint, daß ihn bestraft. Man beachte weiterhin dreierlei: erstens, daß die hier ausgedrückten Präferenzen für *alle* Wortarten gelten, solange sie eine Kasusmarkierung besitzen; zweitens, daß selektionale Kriterien einer übergeortneten Konstituente nicht berücksichtigt werden, also die Akkusativforderung eines finiten Verbs obligatorisch ist, zum anderen aber der Präferenz C_x^{33} widerspricht; und drittens, dadurch, daß diese Präferenzen sich unär ausdrücken lassen, der Hypothesenraum frühzeitig reduziert wird.

Kapitel 4

Inkrementelles Parsing

Eine der grundlegenden Eigenschaften menschlichen Sprachvermögens besteht darin, Sätze bereits während ihrer Äußerung zu analysieren und zu versteht. Inkrementelles Parsing stellt den Versuch dar, dieses Verhalten nachzubilden. Dabei werden die Ziele verfolgt, eine Analyse sehr langer sprachlicher Äußerungen zu ermöglichen, sowie die Untersuchung von Sprachphänomene wie Sackgassensätzen.

In diesem Kapitel stellen wir ein Verfahren vor, welches auf dem in Abschnitt 2.2.3 beschriebenen Branch & Bound Suchverfahren basiert. Dazu führen wir zwei neue Konzepte ein, die zum einen der Betrachtung unterspezifizierter Dependenzen und zum Anderen der Beschränkung des inkrementellen Parsingproblems durch eine Satzendeerkennung dienen.

4.1 Verwendung von *DCSPs*

Wie bereits in Kapitel 3 beschrieben, lassen sich zum Parsing *CSPs* verwenden. Der Übergang zum inkrementellen Parsing besteht nun in dem Übergang vom einzelnen *CSP* zum *DCSP*.[1] Dabei verwenden wir weiterhin das in Abschnitt 3.1 behandelte *CDG*.

Inkrementelles Parsing bedeutet hier, daß zu aufeinanderfolgenden Zeitpunkten ein oder mehrere Worte dem Satz hinzugefügt werden. Das heißt, daß wir uns hier auf Worte als kleinste Informationseinheiten beschränken, die der Eingabe entnommen und dem *DCSP* hinzugefügt werden können. Wir beginnen also mit einem Wort, aufgrund dessen das erste $CSP(t_1)$ konstruiert wird, und nehmen im Folgenden weitere Segmente auf, bis das Ende der Eingabe erreicht ist. Derartig entstehende Satzpräfixe werden dargestellt als

$$S(t_i) \quad := \quad \langle w_1, \ldots, w_{n(t_i)} \rangle \tag{4.1}$$

wobei $n(t_i)$ die Länge der Eingabe zum Zeitpunkt t_i ist. Falls das Ende der Eingabe erreicht wird und noch keine *gültige* Lösung vorliegt, wird ein letzter Versuch unternommen, eine andere, bisher dispräferierte Lösung zu finden. Eine Lösung ist jetzt

[1] Wenn wir hier *CSPs* erwähnen, sind eigentlich *PCSPs* gemeint.

ungültig, wenn ein oder mehrere Dependenzen nicht eindeutig analysiert werden können. Derartige Dependenzen werden im weiteren als *unterspezifizierte* Dependenzen bezeichnet (vgl. Abschnitt 4.2). Festzuhalten bleibt, daß, während alle Problemveränderungen $\Delta_P(t_i)$ vor dem Zeitpunkt t_{max} vorgenommen wurden, indem ein weiteres Wort dem *DCSP* hinzugefügt worden ist, jetzt eine andersartige Problemveränderung $\Delta_P(t_{max})$ berechnet wird, indem alle unterspezifizierten Dependenzen aus $CSP(t_{max-1})$ gelöscht werden und $A(t_{max})$ berechnet wird. Ist jedoch die Lösung $A(t_{max-1})$ bereits eine gültige Lösung, so braucht $CSP(t_{max})$ nicht mehr konstruiert zu werden. Das derartig konstruierte *DCSP* umfaßt somit höchstens $n(t_{max}) + 1$ einzelne *CSPs*, besteht also aus der Anzahl einzelner Probleme, die durch die Länge des Satzes plus einer weiteren Berechnung begrenzt ist.

Für jedes Segment wird ein Bearbeitungszyklus vorgenommen, in dem versucht wird $CSP(t_i)$ zu generieren und $A(t_i)$ zu berechnen. Gelingt beides, so sprechen wir von einem *vollinkrementellen* Verfahren: Es werden fortwährend Eingaben entgegengenommen und Ausgaben erzeugt. Im Unterschied dazu nennt man Verfahren *teilinkrementell*, falls zwar $CSP(t_1)$ bis $CSP(t_{max})$ generiert werden, allerdings nur zum Schluß $A(t_{max})$ ausgegeben wird. Grundsätzlich vorstellbar sind Mischformen, die nicht zu jedem *CSP* eine Zwischenlösung berechnen.

Schließlich lassen sich die Problemveränderungen hier betrachteter *DCSPs* genauer spezifizieren:

$$\Delta_Z(t_i) > 0 \tag{4.2}$$

Es werden keine Variablen entfernt.

$$\Delta_D(t_i) \quad \textit{variabel} \tag{4.3}$$

Domänen-Werte werden hinzugefügt und entfernt.

$$\Delta_C(t_i) = 0 \tag{4.4}$$

Es werden keine Constraints verändert oder entfernt.

$$\Delta_g(t_i) \quad \textit{variabel} \tag{4.5}$$

Bewertung von Dependenzen sind veränderbar.

Eigenschaft 4.5 ist dadurch gegeben, daß Gewichte einiger Constraints dynamisch berechnet werden und so die Bewertungen bereits vorgenommener Variablenbindungen sich über die Zeit ändern können. Eine genauere Erläuterung hierzu findet in den folgenden Abschnitten statt.

4.2 Unterspezifizierte Dependenzen

4.2.1 Motivation

Aufgrund des inkrementellen Bearbeitungsmodus besteht das Problem, daß Modifier und Modifiee einer Dependenz sich in verschiedenen Segmenten der Eingabe befinden können. Dies ist zum Beispiel in Abbildung 4.1 der Fall, da das Nomen noch

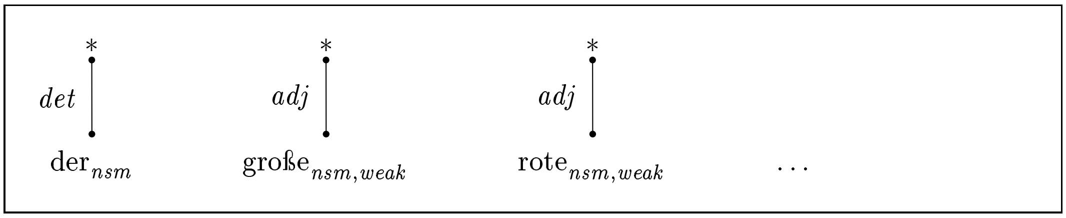


Abbildung 4.1: Satzpräfix mit unterspezifizierten Bindungen

nicht Teil des bisher gelesenen Satzpräfixes ist. Unproblematisch sind solche Dependenzen, in denen der Modifiee vor dem Modifier auftritt. Wird andererseits der Modifiee einer Dependenz in einem späteren Eingabesegment erwartet, wollen wir, wie in Abbildung 4.1 angedeutet, bereits alle möglichen Labelungen des Modifiers formulieren, ohne weiter auf den Modifiee einzugehen. Derartige Variablenbelegungen sind also in der Modifieeposition unterspezifiziert und sollen dort mit einem Wert *nonspec* belegt werden. Sie liefert somit nur einen Teil der Informationen, die eine vollständig spezifizierte Variablenbelegung bietet. Es ist weiterhin alle lexikalischen und Wortstellungs-Informationen des Modifiers, sowie die Art des Labels der Dependenz zu erfragen. Nicht zur Verfügung stehen lexikalische Informationen des Modifiees. Das einzige, was über den Modifiee gesagt werden kann ist, daß er noch nicht gelesen worden ist.

In Abbildung 4.1 und 4.2 werden Bindung an das *Pseudolexem nonspec* als Stern (*) dargestellt. Man beachte, daß man im allgemeinen nicht sagen kann, ob *nonspec*-Lexeme in einem Dependenzbaum eindeutig sind oder nicht. Es bleibt dem Grammatikschreiber überlassen, die Gleichheit unterspezifizierter Modifiees in zwei Dependenzen zu präferien oder nicht. Im Satzpräfix in Abbildung 4.1 scheint es zum Beispiel sinnvoll zu sein, im Verlaufe der Analyse vorerst anzunehmen, daß die bisherig gelesenen unterspezifizierten Lexeme Teil der selben Nominalphrase sind und somit vom selben Nomen abhängen. Ohne diese Annahme wären wir gezwungen, alle lexikalischen Lesarten des Artikels und der Adjektive zu kombinieren, womit wir — gegeben ein entsprechendes Beispiel-Lexikon — $4 * 15 * 15 = 900$ Lösungen erhielten. Eine nicht-inkrementelle Analyse kann hier bereits eine Disambiguierung vornehmen, da genügend Informationen vorliegen; im inkrementellen Fall sind jedoch die in Abschnitt 3.2.2 definierten Constraints C_x^5, C_x^6 und C_x^7 noch nicht einschlägig, und es entsteht die beschriebene kombinatorische Explosion der möglichen Lösungen. In Abschnitt 4.2.3 werden wir näher besprechen, wie diese Constraints umzuformulieren sind.

4.2.2 Definition *nonspec*

An dieser Stelle wollen wir die Constraint-Logik, wie sie in Kapitel 3 angedeutet wurde, um das unterspezifizierte Lexem *nonspec* erweitern. Es zeigt sich, daß fundamentale Eigenschaften der Logik durch unterspezifizierte Variablenbelegungen verletzt werden, so daß unter Beibehaltung der Zweiwertigkeit die Äquivalenz zwischen logisch ineinander umformbaren Formeln nicht mehr gegeben ist. Obwohl dieses Problem hier nicht gelöst wird, stellen wir zusätzliche Prädikate zur Verfügung, die

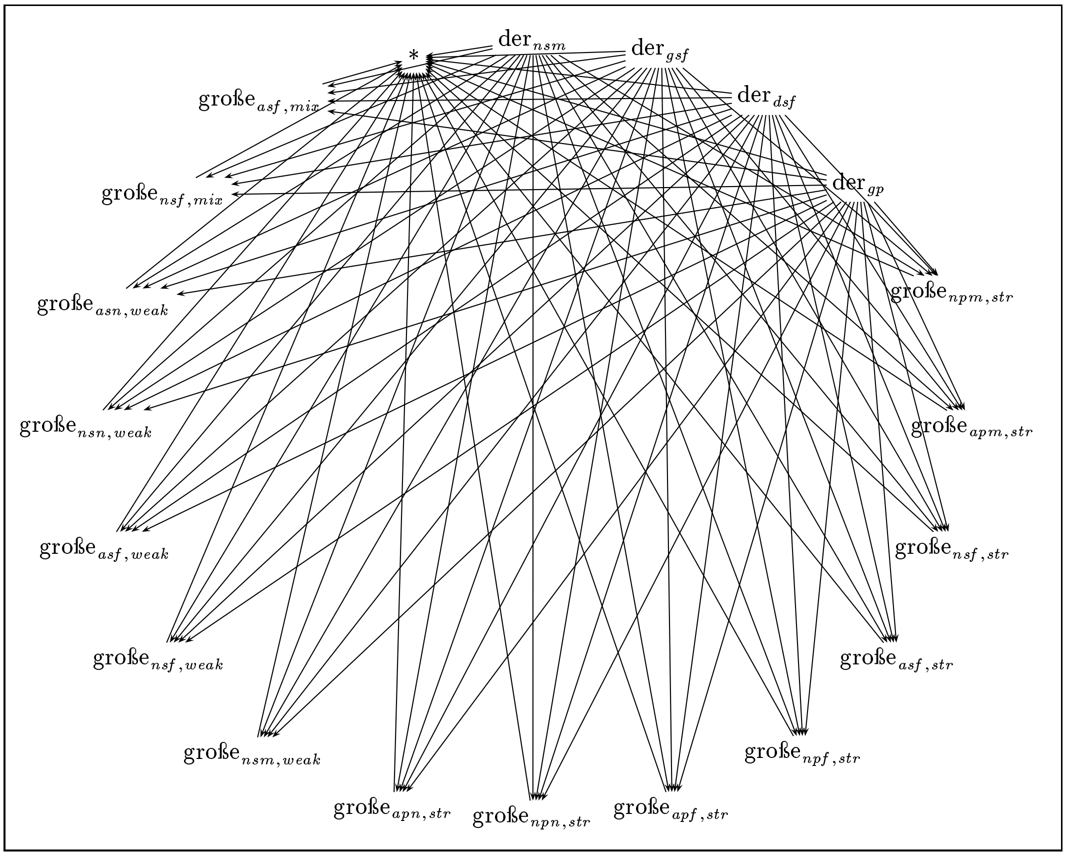


Abbildung 4.2: Kombinatorische Explosion der Ambiguitäten

kritische Bereiche einer Constraint-Formel abschirmen können. Derartige Prädikate werden *Guards* genannt.

Sei die Definition von Variablenbelegungen erweitert um den Wert *nonspec* in Modifieeposition,

$$A : \mathcal{D} \times R \mapsto L \times (\mathcal{D} \cup \{nil, nonspec\}) \tag{4.6}$$

und die Prädikate *nonspec*() und *spec*() definiert durch

$$spec, nonspec \quad : \quad \mathcal{D} \cup \{nil, nonspec\} \mapsto \{true, false\} \tag{4.7}$$

$$nonspec(x) \quad := \quad \begin{cases} true & : \text{ falls } x = nonspec \\ false & : \text{ sonst} \end{cases} \tag{4.8}$$

$$spec(x) \quad := \quad \neg\ nonspec(x) \tag{4.9}$$

Betrachten wir nun Constraint C_x^1 zur Definition von Artikelbindungen, so sehen wir, daß die Information der syntaktischen Kategorie des Modifiees benötigt wird, welche zumindest im Falle eines unterspezifizierten Modifiees nicht zur Verfügung steht. Allgemein gilt, daß ein Fehler auftritt, wenn versucht wird, unbekannte lexikalische Informationen zu verwenden. Da der Fehler allerdings auf *false* abgebildet wird, ist die Äquivalenzunformung in einer derartig definierten Logik nur eingeschränkt gültig, wie folgendes Beispiel demonstriert:

$$\neg\ (x{\uparrow}syn{:}cat = noun) \quad \not\equiv \quad (x{\uparrow}syn{:}cat \neq noun). \tag{4.10}$$

Unter der Voraussetzung, das x eine unterspezifizierte Dependenz bezeichnet, liefert die erste Formel den logisceh Wert *true* und die zweite *false*.

Es sei bemerkt, daß in Schröder, Foth, und Schulz (1999) bereits Guards zur Vermeidung von Fehlerfällen realisiert sind: Eine Abfrage auf die Existenz einer lexikalischen Eigenschaft ist realisiert durch das Prädikat *exists*(); eine *nil*-Bindung wird erfragt durch das Prädikat *root*().

Erweitern wir nun die restlichen Definitionen der in Schröder, Foth, und Schulz (1999) gegebenen Prädikate und Funktionen. Zu unterscheiden sind hier Prädikate, die aufgrund eines Fehlerfalls *false* liefern und diejenigen, die *false* zurückliefern, weil sie tatsächlich nicht erfüllt sind. Letzteres gilt ausschließlich für das Prädikat *root*(), da mit Sicherheit gesagt werden kann, daß $nonspec \neq nil$ ist, weil *nil*-Bindungen eines Lexemes sofort vorgenommen werden können, und somit *nonspec*-Bindungen nie durch *nil* ausspezifiziert werden. Alle anderen Prädikate, einschließlich *exists*(), liefern immer *false*, falls ein *nonspec*-Lexem involviert ist.

Interessanter sind die Definition der Funktionen *distance*(), sowie Terme der Form $x{\uparrow}from$ und $x{\uparrow}to$. Dies sind Terme, die auf Wortstellungsinformationen im Satz referieren. Grundsätzlich wollen wir die maximal optimistische Annahme machen, daß *nonspec* stellvertretend für ein Lexem steht, welches direkt nach dem Ende des bisherig gelesenen Satzpräfixes steht. Falls es sich im Laufe der Analyse zeigt, daß diese Erwartung nicht erfüllt wurde, so sind die bisher vorgenommenen *nonspec*-Bindungen neu zu bewerten. Diese Annahme ist bewußt optimistisch gewählt, da somit die Monotonie der Bewertungen der Zwischenlösungen gewährleistet bleibt. So können zum Beispiel mit geeigneten Distanz-Constraints *nonspec*-Bindungen immer stärker bestraft werden (vgl. Abschnitt 4.2.4), sodaß sie im Vergleich zu voll ausspezifizierten Dependenzen immer stärker abfallen.

4.2.3 Verwendung unterspezifizierter Lexeme in Constraints

Wenden wir nun die im vorherigen Abschnitt eingeführten neuen Prädikate an, und untersuchen die in Kapitel 3 Constraints, inwieweit sie sich bereits zum inkrementellen Parsing eignen.

Betrachten wir also Constraint C_x^1 und formulieren es derart um, daß ein Artikel auch ohne bereits gelesenes Nomen in einer Dependenz zu *nonspec* mit dem Label *det* untergebracht werden kann:

$$\begin{aligned} C_x^1 \quad := \quad & role(x) = syn \Rightarrow \\ & (\ x.label = det \ \Leftrightarrow\ x{\downarrow}syn{:}cat = article \\ & \quad \wedge\ (nonspec(x{\uparrow}id)\ \vee\ x{\uparrow}syn{:}cat = noun)\ ) \end{aligned}$$

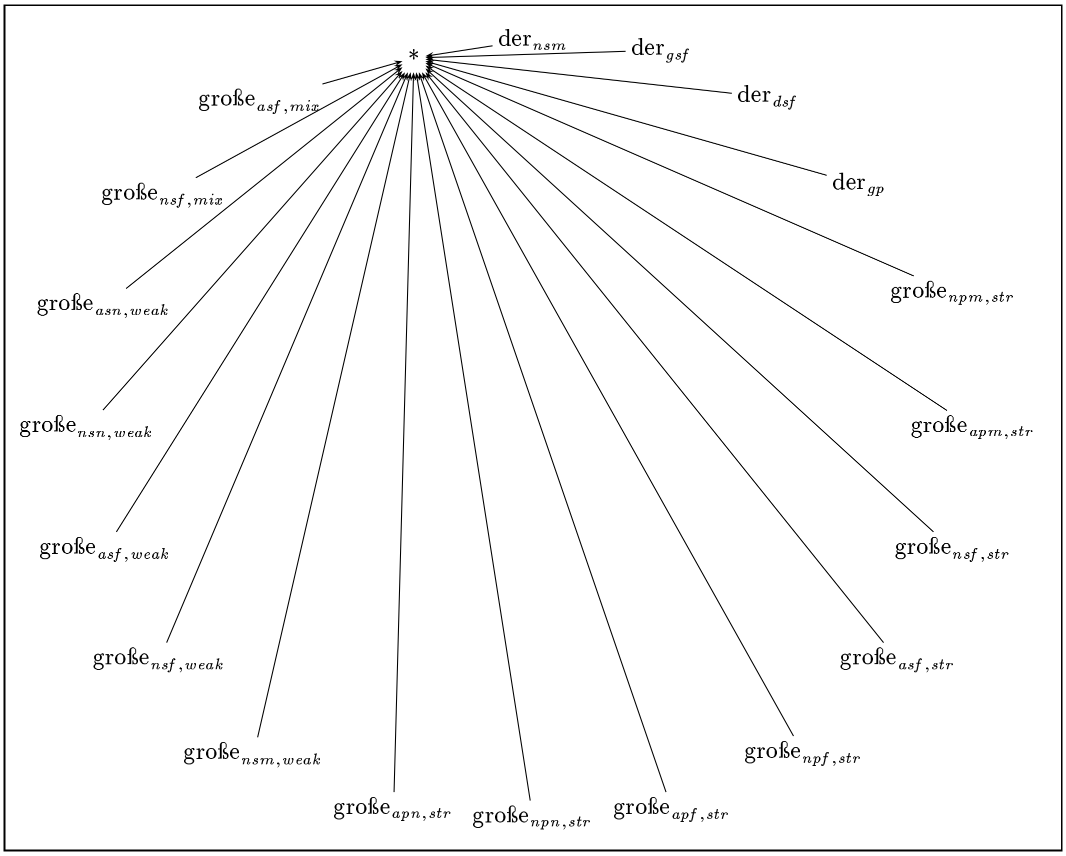


Abbildung 4.3: Wirkung des modifizierten Constraints C_x^1

Wir gehen davon aus, daß Disjunktionen *lazy* evaluiert werden, so daß, falls der Term $nonspec(x{\uparrow}id)$ wahr wird, der zweite Teil der Disjunktion nicht mehr evaluiert wird und dadurch kein Fehler auftritt. Gleiches gilt für logische Operatoren wie die Konjunktion und Implikation.

Betrachten wir die Wirkung des neuen Constraints C_x^1 in Abbildung 4.3, so sehen wir, daß alle möglichen Lexeme lediglich eine Bindung an *nonspec* behalten. Leider sehen wir auch, daß die Anzahl der Kanten sich zwar verringert hat, die Anzahl der Knoten aber noch konstant geblieben ist. Desweiteren ist mit den bisherig vorgestellten Constraints noch nicht die eindeutige Lösung aus Abbildung 4.1 berechenbar, da zwar die Kongruenz des Nomen und seine modifizierenden Wortarten in den Constraints C_x^5, C_x^6 und C_x^7 beschrieben wurde, die Kongruenz der Komplemente untereinander aber nicht überprüft wird.

Folgendes heuristisches Constraint zeigt beispielhaft, wie dies gefordert werden kann:

$$\begin{aligned} C_{x,y}^{36} \;:=\;\; & x.role = syn \;\wedge\; y.role = syn \\ & \wedge\; ((nonspec(x{\uparrow}id) \;\wedge\; nonspec(y{\uparrow}id)) \;\vee\; x{\uparrow}id = y{\uparrow}id) \\ & \wedge\; (x.label = det \;\vee\; x.label = adj) \\ & \wedge\; (y.label = det \;\vee\; y.label = adj) \end{aligned}$$

$$\begin{aligned}
&\wedge \;\; (x.label = det \;\; \Rightarrow \;\; y.label \neq det) \\
&\wedge \;\; exists(x{\downarrow}syn{:}num) \;\; \wedge \;\; exists(y{\downarrow}syn{:}num) \\
&\Rightarrow \;\; x{\downarrow}syn{:}num = y{\downarrow}syn{:}num \\
&w(C^{36}_{x}, x, y) = 0.16
\end{aligned}$$

Im Detail funktioniert dieses Constraint derart, daß zwei Dependenzkanten der Syntaxebene betrachtet werden, die entweder dasselbe Lexem oder beide *nonspec* modifizieren. Beide Kanten sollen mit Labeln versehen sein, die zur Modifizierung von Nomen üblich sind, hier *det* und *adj*. Im nicht-inkrementellen Fall hätte hier auch die syntaktische Kategorie des Modifiees gefordert werden können. Weiterhin dürfen nicht beide Kanten Artikelbindungen sein, was ein Hinweis auf unterschiedliche Modifiees wäre. Falls das Lexikon derart angelegt ist, daß nicht jeder Eintrag ein Numerus-Feature besitzt, so wird mit entsprechenden *exists*-Guards der Fall ausgeschlossen, daß diese Lexeme miteinander verglichen werden. Gelten also all diese Prämissen, so folgern wir, daß die betrachteten Modifier im angegebenen Feature übereinstimmen. Nach dem gleichen Muster lassen sich alle anderen Kongruenzeigenschaften ebenfalls beschreiben. Um insbesondere die Lösung in Abbildung 4.1 herzustellen, ist es notwendig, die Gleichheit der Deklination der Adjektive eines Nomens, sowie die schwache Deklination bei bestimmten Artikeln zu fordern.

Bei diesen Constraints über gemeinsame Eigenschaften wollen wir die Annahme machen, daß die *nonspec*-Bindungen in Abbildung 4.1 später durch dasselbe Nomen ersetzt werden. Wir nehmen in Kauf, daß diese Annahme in Sätzen wie (3) zu einem gewissen Zeitpunkt sich als falsch erweisen.

(3) alte grünes Gras fressende Zebras jagen Löwen.

Zu Beginn einer inkrementellen Analyse von (3) wird unter Umständen angenommen, daß die Adjektive „alte“ und „grünes“ auf das selbe Lexem verweisen werden, obwohl sie nicht miteinander kongruieren:

$$\begin{aligned}
A(t_1) &= \{\langle \text{alte}_{nsf,strg}, syn, adj, nonspec\rangle, \langle \text{grünes}_{nsn,strg}, syn, adj, nonspec\rangle\} \\
g(A(t_1)) &= 0.16
\end{aligned}$$

Im nächsten Schritt wird „Gras“ gelesen, welches zwar mit „grünes“ aber nicht mit „alte“ kongruiert, so daß nur die zweite *nonspec*-Bindung ausspezifiziert wird:

$$\begin{aligned}
A(t_2) &= \{\langle \text{alte}_{nsf,strg}, syn, adj, nonspec\rangle, \langle \text{grünes}_{nsn,strg}, syn, adj, \text{Gras}_{ns}\rangle \\
&\quad\;\; \langle \text{Gras}_{ns}, syn, subj, nonspec\rangle\} \\
g(A(t_2)) &= 1.0
\end{aligned}$$

Da zum Zeitpunkt t_2 Constraint $C^{36}_{x,z}$ nicht mehr einschlägig ist, wird Lösung $A(t_2)$ besser bewertet als $A(t_1)$. Man beachte also, daß die Heuristik, formuliert durch $C^{36}_{x,z}$, nicht in jedem Fall zu einer optimistischen Vorhersage zukünftiger Bindungen geführt hat.

Eine weitere Möglichkeit zur Komplexitätsvermeidung wäre hier, den Begriff der Unterspezifiziertheit auf lexikalische Eigenschaften auszudehnen, so daß in Abbildung 4.3 Knoten gleicher Worte zusammengefaßt werden, solange keine Kongruenz des Kasus, Genus und Numerus der Artikel und Adjektive mit ihrem Nomen stattfindet. Erst in einem weiteren Schritt werden zusammengefaßte Knoten aufgespaltet, falls in unterschiedlichen Belegungsvarianten unterschiedliche Kongruenzen gefordert werden. Foth (1999) zeigt dieses Vorgehen für den nicht-inkrementellen Fall. Hier werden Constraints nach ihrer Rollenzugehörigkeit indiziert, und jede Rolle aufgrund der in ihren Constraints verwendeten lexikalischen Eigenschaften klassifiziert. So wird zum Beispiel in der Grammatik aus Abschnitt 3.2 in der semantischen Rolle keine Kongruenz erzwungen, so daß die entsprechenden Eigenschaften, die die lexikalische Ambiguität eines Wortes ausmachen, unberücksichtigt bleiben können. Dadurch lassen sich alle Lesarten eines Wortes auf dieser Ebene mit einem einzigen Constraintknoten repräsentieren, und es ist gleichzeitig garantiert, daß dieser Knoten später nicht aufgespaltet werden muß. In dem Sinne, daß Lexeme in einigen Rollen nicht näher unterschieden werden müssen, werden mit diesem Verfahren also ebenfalls unterspezifizierte Dependenzen hergestellt.

4.2.4 Beispiel

Im Folgenden wollen wir die Wirkung des Constraints C_x^{32} aus der in Kapitel 3 skizzierten Beispielgrammatik anhand einer inkrementellen Analyse des Satzes (2) betrachten. Constraint C_x^{32} bestraft die Länge einer Artikelbindung, die mehr als ein Lexem überspannt. Nacheinander seien die Worte „der“, „große“, „rote“ und „Ball“ usw. eingelesen und jeweils Zwischenanalysen berechnet worden. Beginnen wir mit dem ersten Wort, so ergibt sich die Belegung:

$$\begin{aligned} A(t_1) &= \{\langle \mathrm{der}_{nsm}, syn, det, nonspec \rangle\} \\ g(A(t_1)) &= 1.0 \end{aligned}$$

Der Übersicht halber sei nur die Syntax-Rolle dargestellt. Das Constraint C_x^{32} schlug bisher noch nicht an, da die Entfernung zwischen dem Lexem der_{nsm} und *nonspec* eins beträgt. $A(t_1)$ ist zu diesem Zeitpunkt die einzige Lösung, da wir durch entsprechende Constraints Nominative und Singulare präferieren (vgl. Abschnitt 3.2.8). Im nächsten Schritt lesen wir das Wort „große“ und erhalten:

$$\begin{aligned} A(t_2) &= \{\langle \mathrm{der}_{nsm}, syn, det, nonspec \rangle, \langle \mathrm{große}_{nsm,weak}, syn, adj, nonspec \rangle\} \\ g(A(t_2)) &= 1/2 \end{aligned}$$

Hier sehen wir, daß die Bindung des Artikels an *nonspec* jetzt schlechter bewertet wurde, da zwischen Artikel und hypothetischem Nomen ein weiteres Wort gefunden wurde. Diese Lösung ist jetzt ebenfalls die einzige, da aufgrund des bestimmten Artikels „große“ schwach dekliniert werden muß. Gleichermaßen wird die Artikelbindung in der nächsten Lösung heruntergestuft, da wiederum nicht das Nomen, sondern ein

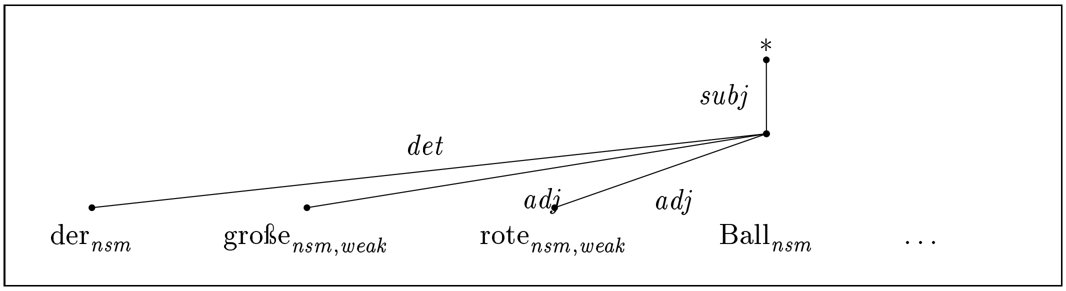


Abbildung 4.4: Dependenzbaum der Zwischenlösung $A(t_4)$

weiteres Adjektiv eintrifft:

$$\begin{aligned} A(t_3) &= \{\langle \text{der}_{nsm}, syn, det, nonspec\rangle, \langle \text{große}_{nsm,weak}, syn, adj, nonspec\rangle, \\ &\quad \langle \text{rote}_{nsm,weak}, syn, adj, nonspec\rangle\} \\ g(A(t_3)) &= 1/3 \end{aligned}$$

Und wieder ist dies die einzige Lösung, da zum Einen schwache Deklination gefordert wird und zum Anderen Kongruenz mit den vermeintlichen Nachbar-Komplementen eines Nomens gefordert wird, welches im nächsten Schritt eintreffen könnte. Tatsächlich wird nun das Wort „Ball“ gelesen (vgl. Abbildung 4.4), und es wird folgende Belegung vorgenommenen:

$$\begin{aligned} A(t_4) &= \{\langle \text{der}_{nsm}, syn, det, \text{Ball}_{nsm}\rangle, \langle \text{große}_{nsm,weak}, syn, adj, \text{Ball}_{nsm}\rangle, \\ &\quad \langle \text{rote}_{nsm,weak}, syn, adj, \text{Ball}_{nsm}\rangle, \langle \text{Ball}_{nsm}, syn, subj, nonspec\rangle\} \\ g(A(t_4)) &= 1/3 \end{aligned}$$

Jetzt konnten alle offenen Valenzen durch den „Ball“ gesättigt werden, und die Annahme, daß sich alle Komplemente auf dasselbe Nomen beziehen, bestätigt sich. Die jetzt einzig offene Valenz ist die der *subj*-Bindung, die durch ein Verb gesättigt werden kann. Man könnte sich fragen, warum nicht angenommen wird, daß der „Ball“ das Objekt eines noch zu lesenden Verbs ist. Dies unterbinden wir jedoch, indem wir entsprechende Präferenz-Constraints für Subjekt- und/oder Objekt-Bindungen formulieren. Erwähnenswert ist hier, daß die Alternativ-Lösungen zu $A(t_4)$ aufgrund von C^{32}_x jetzt höchstens mit dem Gewicht 1/4 versehen sind, da Bindungen über das Nomen hinweg an *nonspec* vorgenommen werden. Ist der Artikel allerdings korrekt angebunden, zwingt er alle Adjektive zur Bindung an das Nomen, da die Artikelbindung sie überspannt und sonst eine Projektionsverletzung von Constraint $C^{13}_{x,y}$ erkannt wird.

Die restliche Analyse des Satzes sei hier übersprungen, und die gewünschte Lösung wird gefunden, die auch ein *batch*-Algorithmus gefunden hätte (vgl. Abbildung 3.8):

$$\begin{aligned} A(t_6) &= \{\langle \text{der}_{nsm}, syn, det, \text{Ball}_{nsm}\rangle, \langle \text{große}_{nsm,weak}, syn, adj, \text{Ball}_{nsm}\rangle, \\ &\quad \langle \text{rote}_{nsm,weak}, syn, adj, \text{Ball}_{nsm}\rangle, \langle \text{Ball}_{nsm}, syn, subj, \text{gehört}_{trsg}\rangle, \\ &\quad \langle \text{gehört}_{trsg}, syn, verb, nil\rangle, \langle \text{Peter}_{dsm}, syn, obj, \text{gehört}_{trsg}\rangle\} \\ g(A(t_6)) &= 1/3 \end{aligned}$$

4.3 Beschränkung des Parsingproblems

Obwohl wir durch eine inkrementelle Analyse die Komplexität des Problems nicht vermindern, können wir Heuristiken entwickeln, die die Eingangsparameter möglichst klein halten. In diesem Abschnitt wird versucht, durch die Definition eines *Workingsets* und der Annahme eines Satzendes dem stetig wachsenden Berechnungsaufwand eines *DCSPs* entgegenzuwirken.

Betrachten wir nochmals den allgemeinen Fall eines *DCSPs* und nehmen wir an, daß t_{max} unendlich ist, also eine nicht endende Kette von sich stetig verändernden *CSPs* darstellt. Diese Annahme ist durchaus gerechtfertigt, da aus der Sicht des Algorithmus das Ende des *DCSPs* nicht absehbar ist und man im Allgemeinen auch sehr lange *DCSPs* verarbeiten will. Nehmen wir weiter an, es gäbe einen beschränkten Algorithmus zur Lösung von *unendlichen DCSPs*, so müssen unendlich viele diskrete Zeitpunkte t_i existieren, zu denen gefundene Lösungen $A(t_i)$ eines $CSP(t_i)$ *stabil* sind. Existieren solche Zeitpunkten nicht, so ist anzunehmen, daß kein beschränkter Algorithmus zur Lösung des *DCSP* existiert.

Ein $CSP(t_i)$ heißt *stabil* genau dann, wenn

$$A(t_i) \subseteq \bigcap_{i<j\leq max} A(t_j) \tag{4.11}$$

also alle späteren Lösungen $A(t_j)$ sich in den Variablenbelegungen, die vor dem Zeitpunkt t_i gefunden worden sind, nicht mehr ändern. Insbesondere heißt dies, daß sich die Domänen der Variablen in $Z(t_i)$ nicht mehr verändern.

Leider gestaltet sich Definition 4.11 derart, daß wir nur im Nachhinein, also wenn bereits alle $A(t_j)$ berechnet wurden, bemerken können, ob irgendein *CSP* stabil war. Kehren wir deshalb zur Parsing-Anwendung von *DCSPs* zurück und betrachten die Rolle der *nonspec*-Bindungen. Eine *nonspec*-Bindung als Element der Domäne einer Variable bedeutet, daß diese Domäne durch die Ankunft weiterer Lexeme erweitert werden kann, indem die *nonspec*-Bindung um die Anzahl neuer Lexeme vervielfältigt wird und die neuen Lexeme statt ihrer eingefügt werden. Das Signal, daß eine Domäne keinerlei *nonspec*-Bindungen mehr anbietet, bedeutet andererseits, daß sie nicht mehr durch die Ankunft neuer Lexeme verändert werden kann. Diese Tatsache wollen wir nutzen und *nonspec*-Bindungen rechtzeitig löschen. Dies ist zum Einen durch die Erkennung eines Satzendes erreichbar, nach dessen Detektion *nonspec*-Bindungen aus den Domänen vorheriger Lexeme gelöscht werden können. Zum Anderen kann man versuchen, ein sogenanntes *Workingset zum Zeitpunkt* t_i zu definieren

$$\mathcal{W}(t_i) \subseteq Z(t_i) \tag{4.12}$$

welches, intuitiv betrachtet, den Aktivitätsbereich eines *DCSPs* einschränkt. Damit wird hier primär versucht, die Problemveränderung $\Delta_P(t_i)$ so gering wie möglich zu halten. In Abschnitt 2.3.2 wurde bereits angedeutet, daß $\Delta_P(t_i)$ nur einen Teil der intrinsischen Problemveränderung darstellt. Workingsets zur Beschränkung der Problembewältigung $\Delta_A(t_i)$ sind ebenfalls vorstellbar. Vorerst hoffen wir, daß in den meisten Fällen eine geringe Problemveränderung auch zu einer weniger aufwendigen Problembewältigung führt.

Grundsätzlich verfahren wir nach Ankunft eines neuen Lexems in zwei Schritten: (1) Domänen bereits vorhandener Variablen werden um Bindungen mit dem neuen Lexem in Modifiee-Position ergänzt und (2) die vorläufigen Domänen neuer Variablen werden berechnet, indem Bindungen zu alten Lexemen in Betracht gezogen werden. Wir bezeichnen dieses zwei Schritte als *Update-Regel.* An dieser Stelle sind verschiedene Heuristiken vorstellbar, die die Anzahl alter betrachteter Lexeme und Domänen einschränken. Fünf seien hier in Form der Definition der Berechnung und Verwendung von Workingsets vorgestellt.

Variante 1: Das Workingset umfaßt alle Variablen, deren Domänen eine *nonspec*-Bindung anbieten:

$$\mathcal{W}(t_i) \;:=\; \{z \in Z(t_i) \mid \exists\, x \in D_j(t_i) : \mathit{nonspec}(x{\uparrow}id)\} \tag{4.13}$$

Variante 2: Das Workingset umfaßt alle Variablen, deren Domänen eine *nonspec*-Bindung anbieten, die nicht schlechter als ein bestimmter Grenzwert p bewertet ist:

$$\begin{aligned}\mathcal{W}(t_i) \;:=\; \{z_j \in Z(t_i) \mid \exists\, x \in D_j(t_i) :& \\ \mathit{nonspec}(x{\uparrow}id) \;\wedge\; x.\mathit{score} \geq p\}&\end{aligned} \tag{4.14}$$

Der Term $x.score$ denotiere hier die Bewertung der Dependenz x durch alle unären Constraints.

Variante 3: Das Workingset umfaßt alle Variablen, deren Domänen eine *nonspec*-Bindung anbieten oder deren Lexeme in einer anderen Rolle des *CDGs* an *nonspec* anbindbar sind. Seien die Domänenwerte aller Variablen, die sich auf das gleiche Lexem wie eine Variable z_j beziehen, gegeben durch die Menge $D_{quer}(t_i, z_j)$.

$$D_{quer}(t_i, z_j) \;=\; \bigcup_{1 \leq k \leq n} D_k(t_i) : a{\downarrow}id = b{\downarrow}id \;,\; a \in D_j \;,\; b \in D_k \tag{4.15}$$

$$\begin{aligned}\mathcal{W}(t_i) \;:=\; \{z_j \in Z(t_i) \mid \exists\, x \in D_{quer}(t_i, z_j) :& \\ \mathit{nonspec}(x{\uparrow}id)\}&\end{aligned} \tag{4.16}$$

Variante 4: Das Workingset umfaßt alle Variablen, die in den besten Lösungen beteiligt sind und mit einer *nonspec*-Bindung instanziiert sind:

$$\begin{aligned}\mathcal{W}(t_i) \;:=\; \{z_j \in Z(t_i) \mid \exists\, x \in D_j(t_i) :& \\ \mathit{nonspec}(x{\uparrow}id) \;\wedge\; x \in A(t_i)\}&\end{aligned} \tag{4.17}$$

Variante 5: Das Workingset umfaßt alle *nonspec*-gebundenen Variablen, die in den p besten Lösungen beteiligt sind, wobei die Lösungen nicht schlechter als q bewertet sein dürfen.

$$\begin{aligned} \mathcal{W}(t_i) \quad := \quad & \{z_j \in Z(t_i) \mid \exists\ x \in D_j(t_i) : \\ & \quad nonspec(x{\uparrow}id) \ \wedge \ x \in A_{p,q-best}(t_i)\} \end{aligned} \tag{4.18}$$

In einem ersten Ansatz ließe sich ein Workingset definieren als Menge aller Variablen, die noch eine unterspezifizierte Bindung anbieten. Die Verwendung von verletzbaren Constraints hat jedoch zur Folge, daß *nonspec*-Bindungen nur selten aus Domänen gelöscht werden. Durch geeignete Constraints kann zwar erreicht werden, daß *nonspec*-Bindungen immer schlechter bewertet werden. Allerdings enthalten dann viele Domänen sehr niedrig bewertete *nonspec*-Bindungen, die nie mehr Verwendung finden. Das hat jedoch zur Folge, daß das Workingset, so wie es in Variante 1 berechnet wird, unnötig groß bleibt.

Variante 2 fügt der ersten Variante einen Parameter hinzu, der bereits in Schröder, Foth, und Schulz (1999) beschrieben ist und Domänen aufgrund der Verletzung unärer Constraints jenseits eines Grenzwertes minimiert (vgl. *unary pruning*). Diese Strategie wird hier speziell auf die Berechnung von *nonspec*-Bindungen angewendet. Dabei wird der Grenzwert p nicht nur innerhalb der Definition des Workingsets verwendet, sondern grundsätzlich bei der Erzeugung und Neubewertung von *nonspec*-Bindungen berücksichtigt. Das heißt, daß neubewertete *nonspec*-Bindungen in jeder Iteration des Verfahrens unabhängig vom Workingset gelöscht werden, falls sie unterhalb des Grenzwertes p fallen. Dieser Wert ist nicht universell bestimmbar, sondern immer in Anbetracht einer Grammatik zu wählen. Experimente mit einer Beispielgrammatik haben gezeigt, daß mit einem Wert von $p = 0.5$ ein erheblich besseres Laufzeitverhalten des Verfahrens erziehlbar ist. Wird jedoch ein höherer Wert gewählt, steigt der Berechnungsaufwand wieder, und es kommt gehäuft zu Fehlanalysen.

Variante 3 ist motiviert durch die Verwendung von Hilfsebenen zur Formulierung von Existenzforderungen. Ein Verb zum Beispiel besitzt in einer Ebene, die die Obligarität des Objekts garantiert, noch eine *nonspec*-Bindung, wodurch der noch nicht gefüllte Kasusrahmen angedeutet wird. In der *syn*-Rolle ist dieses Verb aber nicht mehr weiter anzubinden. Allerdings wollen wir unter Umständen die Variable der *syn*-Rolle ebenfalls in das Workingset aufnehmen. Auf diese Weise bleibt die Aktivierung eines Lexems quer durch seine Rollen erhalten.

Wie bereits bemerkt, kommt es vor, daß eine Variable zwar noch genügend gute *nonspec*-Bindungen in ihrer Domäne besitzt, diese aber in keiner Lösung mehr vorkommen. Zum Einen wurden sie nicht durch Konsistenzverfahren gelöscht, zum Anderen war kein Constraint formulierbar, das ein derartig definiertes Workingset rechtzeitig begrenzt. Die Betrachtung der Bewertungen der *nonspec*-Bindungen ist oft zu lokal, um rechtzeitig eine Löschung der Variablen aus dem Workingset zu bewirken. Deshalb basieren die Definitionen des Workingsets in Variante 4 und 5 nicht mehr auf Domänen, sondern auf den tatsächlich gefundenen Zwischenlösungen und den in ihr enthaltenen *nonspec*-Bindungen. Es ist jedoch zu beachten, daß die

Zwischenlösungen eines *DCSPs* jetzt anders zu modellieren sind, denn eine unterspezifiziert belegte Variable in einem Workingset $\mathcal{W}(t_i)$ kann nur bis zum Zeitpunkt t_{i+1} ausspezifiziert werden. Danach tritt sie nicht mehr in einem Workingset auf, und ihre Domäne wird nicht mehr erweitert. Es war allerdings im Rahmen dieser Arbeit leider nicht möglich zu zeigen, daß für diese Art von Workingsets Constraints formulierbar sind, die akzeptable Ergebnisse liefern.

Variante 5 ist berechnungstechnisch etwas aufwendiger als Variante 4, da die in Abschnitt 4.4 vorgestellte Suche erst abbricht, wenn die p besten Lösungen der Länge $|Z(t_i)|$ gefunden sind, die nicht schlechter als q sind. Allerdings scheinen fehlerhafte Analysen vermeidbar, die durch ein frühzeitiges Ausscheiden von Variablen aus einem Workingset entstehen.

Sei nun an dem Punkt angeknüpft, der uns dazu veranlaßte, Workingsets einzuführen. Der erste Zweck, den sie erfüllen sollen, ist oben bereits mit der Beschränkung der Problemveränderung zum Zeitpunkt t_i angedeutet. Betrachten wir die Situation, daß das Workingset leer wird, erscheint nun die Annahme berechtigt, daß $CSP(t_i)$ stabil ist.

$$\textbf{Annahme:} \qquad \mathcal{W}(t_i) = \emptyset \;\Leftrightarrow\; CSP(t_i)\ \textit{stabil} \tag{4.19}$$

Die Update-Regel garantiert, daß die Rückwärts-Lesart der Biimplikation immer gegeben ist, also jedes stabile *CSP* ein leeres Workingset hat. Die Vorwärts-Lesart deuten wir dahingehend, daß das bisher gelesene Satzpräfix einen vollständigen Satz bildet, so daß die bisherigen Variablenbelegungen eingefroren werden können, indem alle anderen Optionen aus den Domänen der Variablen gelöscht werden. Dieser Prozeß sei *Satzende-Regel* genannt. Wir nehmen in Kauf, daß Annahme 4.19 falsch sein kann und dadurch eine suboptimale Lösung gefunden wird. Dies geschieht zum Beispiel, wenn ein Satzende prognostiziert wird und die verbleibende Eingabe nicht mehr adjungiert werden kann. Sie muß dann als separater Baum analysiert werden und als eigenständiger Satz angenommen werden. Das Einfrieren aller bisherigen Belegungen hat andererseits den Effekt, daß die Problembewältigung keine andere Wahl hat, als diese Belegungen zu akzeptieren: Eine Suche innerhalb dieser Belegungen erübrigt sich. Das heißt, daß an dieser Stelle die Problembewältigung ebenfalls effektiv beschränkt wird.

Im Folgenden wählen wir aus der Menge der oben beschriebenen Varianten zur Handhabung von Workingsets Variante 2 aus. Das heißt, daß das Workingset alle Variablen umfaßt, deren Domänen eine *nonspec*-Bindung anbieten, die besser als ein Grenzwert p bewertet sind. Dabei werden Variablen, die in verschiedenen Rollen auf das gleiche Lexem referieren, nur einmal gezählt. Allerdings wird ein derart gewähltes Workingset zu gewünschten Satzenden nicht leer. Dazu wenden wir eine Heuristik an, die nicht auf dem vorhandenen Workingset basiert, sich aber auf dieses auswirkt: Wir nehmen ein Satzende an, falls in der Zwischenlösung $A(t_i)$ keine *nonspec*-Bindung mehr vorgenommen wird. Das heiß, daß der Annahme 4.19 nur in der Rückrichtung folge geleistet wird. Das Wechselspiel zwischen *nonspec*-Bindungen, Workingset und Satzende besteht also darin, daß *nonspec*-Bindungen aus dem Workingset herausfallen, falls sie zu schlecht bewertet werden oder in Folge einer Satzende-Prognose gelöscht werden.

4.4 Inkrementelles Branch & Bound

Wir erweitern nun das Verfahren aus Abschnitt 2.2.3, so daß es sich zur Lösung von *DCSPs* eignet. Es sei bemerkt, daß das hiesige Verfahren speziell auf das Parsen mit Constraints und auf die Verwendung von Dependenzstrukturen zugeschnitten ist, und daß die in den Abschnitten 4.2 und 4.3 eingeführten Begriffe unmittelbar einfließen. Wir betrachten im Folgenden im Detail die Aspekte der Problemveränderung $\Delta_P(t_i)$ und der Problembewältigung $\Delta_A(t_i)$ eines *DCSPs*.

Betrachten wir nochmals das normale Branch & Bound aus Abschnitt 2.2.3, so fällt auf, daß in Schritt 6 eine beliebige noch uninstanziierte Variable ausgewählt werden kann, die in einem Umlauf der Suche bearbeitet wird. Im Allgemeinen wird bereits eine Links-Rechts-Reihenfolge der Variableninstanziierung vorgenommen, so daß das Verfahren inhärent inkrementell arbeitet! Das inkrementelle unterscheidet sich von dem normalen Branch & Bound im Wesentlichen nur darin, daß der Sichtbarkeitsbereich auf die Eingabe nach rechts beschränkt ist und sich dort stetig erweitert.

Grob bewältigen wir das veränderte *CSP* derart, daß wir pro Segment die nun besten Instanziierungen aller Variablen suchen. Diese werden als Zwischenlösungen repräsentiert. Zusätzlich wird segmentübergreifend der bisher durchlaufene Suchraum aufbewahrt. Dazu verwenden wir wie in Abschnitt 2.2.3 eine Agenda von sortierten Suchzuständen. Die ersten besten Zustände stellen die Zwischenlösungen dar, der Rest ergänzt den verbleibenden Suchraum. Mit anderen Worten wird also in jeder Iteration des Verfahrens die Liste der Suchzustände erweitert, in gute und schlechte unterteilt und wieder zu einer neuen Agenda zusammengeführt.

Somit sind der Reihe nach folgende Schritte zu bearbeiten:

1. Erstelle ein initiales Constraintnetz ohne Kanten und Knoten.
2. Lese das nächste Segment aus der Eingabe. ($\Delta_Z(t_i)$)
3. Wende die Update-Regel an. ($\Delta_D(t_i)$)
4. Gleiche den Suchraum an das veränderte Netz an. ($\Delta_g(t_i)$)
5. Suche die beste Lösung des Netzes. ($\Delta_A(t_i)$)
6. Berechne das Workingset.
7. Wende die Satzende-Regel an.
8. Gebe die Zwischenlösung aus.
9. Falls die Eingabe noch nicht aufgebraucht ist, fahre fort mit Schritt 2.
10. Wenn in der letzten Lösung unterspezifizierte Bindungen verwendet werden, dann lösche sie alle aus dem Netz und suche ein letztes Mal die beste Lösung des Netzes.
11. Gebe die Lösung aus.

Die Schritte 2, 3, 4, 6 und 7 berechnen die Problemveränderung, wobei Schritte 6 und 7 versuchen, das aktuelle Problem zu beschränken, also Veränderungen derart vorzunehmen, so daß die darauf folgende Problemveränderung geringer ausfällt. Schritt 5 dient auf dieser Betrachtungsebene rein der Problembewältigung.

Seien nun die einzelnen Schritte in *depth-first*-Manier beschrieben. Schritt eins sei hier nicht näher erläutert, da er nur der Initialisierung der zugrundeliegenden Datenstrukturen dient. Einen detailierten Überblick über die verwendeten Datenstrukturen bietet Foth und Schulz (1999).

Im dritten Schritt wird das neu gelesene Segment dem Constraintnetz hinzugefügt.

1. Entferne aus dem Segment das nächste Wort und erzeuge eine Liste aller lexikalischen Lesarten des Wortes.

2. Erzeuge für jedes Lexem und jede Rolle des Constraintnetzes einen Con-straint-Knoten.

3. Füge die neuen Constraintknoten zum Netz hinzu.

4. Fahre mit Schritt 1 fort, falls noch weitere Worte im Segment enthalten sind.

In Schritt 2 wird die soweit erzeugbare Domäne der neuen Constraintknoten erzeugt. Diese Domäne beinhaltet alle Dependenzen, welche das zugrundeliegende Lexem rückwärtig aufwärts im Dependenzbaum anbinden. Schritt 3 fügt dem Constraintnetz die neuen Knoten hinzu und erweitert alle Domänen der Knoten des Workingsets. Hier werden die neuen Lexeme als Ersatz für alte *nonspec*-Bindungen ausprobiert. Die Trennung der beiden Schritte 2 und 3 ist nicht unbedingt nötig, ermöglicht aber eine Differenzierung in der Verwendung des Workingsets.

Kehren wir zur Rahmen-Routine zurück. Ein entscheidender Schritt in diesem Verfahren, bevor mit der eigentlichen Suche begonnen werden kann, ist die Anpassung des Suchraumes an das veränderte Netz. Zwar wurden dem Netz neue Constraintknoten hinzugefügt und Domänen alter Constraintknoten erweitert, der Suchraum selbst ist aber ebenfalls den veränderten Umständen anzupassen. Dies geschieht in drei Schritten:

1. Bewerte alle alten *nonspec*-Bindungen in allen Domänen neu.

2. Registriere *offene* Zustände des Suchraumes.

3. Bewerte alle Suchzustände neu, deren Bindungen neu bewertet wurden.

Wie bereits bemerkt, verändert sich hier das *DCSP* über die Zeit in den Constraint-Gewichten $\Delta_g(t_i)$, was zur Folge hat, daß bereits instanziierte *nonspec*-Bindungen neu bewertet werden müssen. Diese Operation wird in Schritt 1 vorgenommen. Mit anderen Worten: Möglicherweise war die Bewertung der Bindungen zu optimistisch und muß jetzt, da neue Eingaben gelesen wurden, revidiert werden. Diese Prozedur kann auf die unterspezifizierten Dependenzen eingeschränkt werden. Man beachte, daß die Bewertung der neuen *nonspec*-Bindungen den Grenzwert p unterschreiten kann und damit ein Ausscheiden des Wertes aus der Domäne bewirkt. Die Neubewertung einzelner Bindungen zieht ebenfalls die Neubewertung der Suchzustände nach sich, die sie verwenden. Zuvor ist Schritt 2 notwendig, um die neuen Werte der Domänen in den Suchraum zu integrieren. Es sei in diesem Zusammenhang bemerkt, daß jeder Suchzustand eindeutig mit einer Variablenbindung assoziiert wird. Die Liste aller in diesem Suchzustand gültigen Variablenbindungen ergibt sich auf dem Pfad vom Startpunkt des Suchraumes bis zu diesem Zustand. Ein Suchzustand wird nun als *offen* bezeichnet, wenn er eine Variable bindet, deren Domäne in Schritt 3 des vorherigen Absatzes erweitert wurde. Ist ein Suchzustand offen, so füge einen Vermerk ein, um welche Werte an dieser Stelle der Suchraum zu erweitern ist. Stände an dieser Stelle des Suchraumes die Domäne schon vollständig zur Verfügung, so wäre entsprechend genügend verzweigt worden. Dies muß die Suche später nachholen.

Jetzt erst sind wir in der Lage, das veränderte Problem zu lösen, indem wir eine Suche anstoßen. Dabei verwenden wir wie in Abschnitt 2.2.3 eine Agenda, die alle Suchzustände der Güte nach sortiert beinhaltet. Auch hier wird in jedem Schritt ein Suchzustand aus der Agenda entnommen und expandiert, indem die nächste freie Variable instanziiert wird. Zusätzlich kann es nun sein, daß ein Suchzustand aus der Agenda entnommen wird, der auf seinem Pfad offene Suchzustände durchläuft. Ist dem so, so wurde in den entsprechenden Suchzuständen eine Menge von Wertebelegungen hinterlegt, die auszuprobieren sind, bevor mit der nächsten freien Variable fortgefahren werden kann. Sonst wird wie im normalen Branch & Bound weiterverfahren, indem das Tupel aller bis dahin belegten Variablen neu bewertet und der Agenda wieder zugeführt wird. Betrachten wir nun alle Schritte der Problembewältigung der Reihe nach:

1. Entnehme der Agenda den bisher besten Suchzustand.
2. Wenn der aktuelle Suchzustand schlechter als alle bisherigen gefundenen Lösungen ist, so füge ihn einer Ausschußliste hinzu. Fahre fort mit Schritt 1.
3. Wenn der aktuelle Suchzustand alle Variablen belegt und besser als alle bisherigen Lösungen ist, so kopiere alle bisherigen Lösungen in die Ausschußliste und eröffne eine neue Liste der besten Lösungen mit der aktuellen Belegung. Fahre fort mit Schritt 1.
4. Wenn der aktuelle Suchzustand genauso gut wie die bisherigen Lösungen ist, so füge ihn zur Liste der besten hinzu. Fahre fort mit Schritt 1.
5. Wenn auf dem Pfad zu diesem Suchzustand ein Suchzustand offen ist, so instanziiere die erste Variable, deren Domäne erweitert worden ist, mit den registrierten neuen Werten. Fahre fort mit Schritt 7. (Ergänzung des Standard Branch & Bound)
6. Wenn auf dem Pfad zu diesem Suchzustand kein offener Suchzustand verwendet wird, so ist die aktuell zu instanziierede Variable die nächste freie.
7. Erzeuge für jede mögliche Belegung der aktuellen Variable einen neuen Suchzustand und füge sie der Agenda sortiert hinzu.

8. Falls ein offener Suchzustand bearbeitet wird, so schließe diesen ab; seine erweiterte Domäne wurde soweit erschöpfend behandelt. (Ergänzung des Standard Branch & Bound)

9. Falls die Agenda nicht leer ist, fahre mit Schritt 1 fort.

10. Ist die Agenda leer, so füge die Ausschußliste und die Liste der besten Lösungen zu einer neuen Agenda zusammen, die beim nächsten Aufruf der Routine abgearbeitet wird. (Ergänzung des Standard Branch & Bound)

Nach Beendigung des Suchschrittes liegt nun eine Liste von möglichen Zwischenlösungen vor. Bevor diese ausgegeben und mit dem nächsten Segment der Eingabe fortgefahren wird, prüfen wir, ob ein mögliches Satzende vorliegt und berechnen das Workingset so, wie es im vorherigen Abschnitt bereits beschrieben wurde. Nach vollständiger Bearbeitung der Eingabe überprüfen wir ein letztes Mal die Lösung auf enthaltene *nonspec*-Bindungen, löschen diese gegebenenfalls und suchen ein letztes Mal eine Lösung, die frei ist von *nonspec*-Bindungen.

4.5 Satzende-Hypothesen

Bei der Analyse des in Abschnitt 4.2 betrachteten Satzbeispiels (4) sind wir die Antwort auf die Frage schuldig geblieben, durch welche Constraints erreicht werden kann, daß nach dem Parsen des letzten Wortes keine *nonspec*-Bindung mehr vorliegt.

(4) der große rote Ball gehört Peter.

Ohne weiteres ist nicht gewährleistet, daß „Peter“ am besten an ein rückwärtiges Verb angebunden wird. Denkbar wären ebenfalls die Bindungen

$$\langle \text{Peter}_{nsm}, syn, obj, nonspec \rangle \quad \text{und} \quad \langle \text{Peter}_{nsm}, sem, theme, nonspec \rangle \ . \quad (4.20)$$

Derartige Situationen lassen sich durch binäre Constraints vermeiden, die auf der Syntax- und der Hilfsebene für Objekte operieren und ausdrücken, daß der Forderung eines Verbs nach einem Objekt entsprochen werden sollte. Im Rahmen der in Kapitel 3 beschriebenen Grammatik ließe sich zum Beispiel folgendes Constraint formulieren:

$$\begin{aligned} C^{37}_{x,y} \quad := \quad & x.role = syn \ \wedge \ y.role = obj \\ & \ \wedge \ x.label = obj \\ & \ \wedge \ nonspec(x{\uparrow}id) \ \wedge \ nonspec(y{\uparrow}id) \\ & \Rightarrow \ false \\ & w(C^{37}_{x}, x, y) = 0.01 \end{aligned}$$

Bei diesem Constraint ist jedoch Vorsicht geboten, da es nur im inkrementellen Fall einschlägig ist und unter Umständen einer Zwischenlösung eine Strafe zuführt, die sie später wieder verliert. Diese Situation ist vergleichbar mit der durch Constraint $C^{36}_{x,y}$ formulierten Heuristik, daß unterspezifiziert angebundene Adjektive bereits kongruieren sollten. Constraints $C^{37}_{x,y}$ und $C^{36}_{x,y}$ dienen beide der Modellierung von Zwischenlösungen bzw. von Satzende-Prognosen, die als Hypothese des weiteren Verlaufs der Analyse interpretierbar sind. Wie bereits in Abschnitt 4.2 diskutiert, können derartige Heuristiken in Sätzen wie (5) fehlschlagen und dazu führen, daß Bindungen vorheriger Zwischenlösungen revidiert werden müssen. Tritt dieser Effekt mitten in einem Satz wie (5) auf, so ist eine erhöhte Problembewältigung im Verhältnis zur Problemveränderung zu verzeichnen.

(5) alte grünes Gras fressende Zebras jagen Löwen.

Beachtenswert ist, daß das obige Constraint $C^{37}_{x,y}$ mit einschlägigen Distanz-Constraints interagiert, die zum Beispiel die Länge der Objekt-Anbindung bestrafen. Zum Einen bieten Distanz-Constraints eine gute Heuristik zur Optimierung des Analyseaufwandes, da deren Gewicht mit der Länge der Dependenz steigt, zum Anderen haben sie den ungewünschten Nebeneffekt, daß Dependenzen wie in 4.20 bevorzugt werden, dem nun mit Constraint $C^{37}_{x,y}$ entgegen zu wirken ist. Erst so sind korrekte Satzenden von (4) und vergleichbare Sätze prognostizierbar.

Betrachten wir nun aber Analysen, die aufgrund des verwendeten Verfahrens und der formulierten Constraints fälschlicherweise ein Satzende prognostizieren. Da nach einer Satzende-Erkennung vorher gefundene Dependenzen nicht mehr revidiert werden können, ist das Satzfragment zwischen prognostiziertem Satzende und tatsächlichem Satzende entweder nach links anzubinden oder, falls dies nicht mehr möglich ist, als eigenständiger Dependenzbaum zu analysieren. Letztere Alternative kann durchaus als Erkennung eines eigenständigen Satzes interpretiert werden. Auf diese Weise erhalten wir also eine Analyse, die sich von einer Analyse unterscheidet, die mit einem nicht-inkrementellen vollständigen Verfahren gefunden wird. Dies ist jedoch nicht unbedingt eine Fehlleistung des Parsers, die es zu vermeiden gilt, sondern gerade diese Verhalten zu modellieren kann Ziel eines Grammatikentwurfs sein. Sie können als Vorhersage interpretiert werden, daß auch der Mensch unter Umständen Schwierigkeiten mit derartigen Konstrukten hat.

Sätze, die dieses Verhalten nahelegen sind (6) und (7).

(6) die Frau sieht die Mäuse tötende Katze.

(7) der Mann gibt das Buch der Frau dem Kind.

In Satz (6) vermuten wir eine Satzendeannahme nach dem Wort „Mäuse", und folglich wird der vorherige Artikel dem Wort „Mäusen" zugeordnet. Nach Anwendung der Satzende-Regel ist es nicht mehr möglich, den Artikel die^{2}_{ap} statt an „Mäuse" an „Katze" anzubinden sowie die Hypothese, daß „Mäuse" direktes Objekt des Verbs ist, zu revidieren. Kurz vor Ende der Eingabe wird das verbleibende Segment noch als Subjekt bzw. Agens an *nonspec* angenommen. Nach Löschung dieser nicht gewünschten *nonspec*-Bindungen wird dieses Segment als Genitiv-Attribut des direkten Objekts analysiert und entsprechend bestraft. Dies entspricht der Analyse eines Satzes wie (8), wobei Kongruenzverletzungen in Kauf genommen werden.

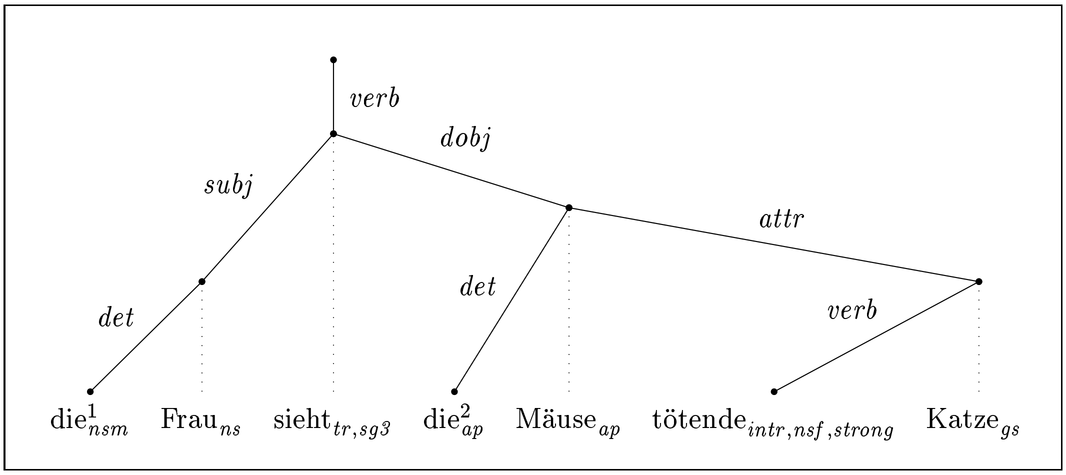


Abbildung 4.5: Inkorrekte Lösung von Satz (6)

(8) die Frau sieht die Mäuse tötender Katzen.

In Anhang A.1 kann der genaue Verlauf der Analyse nachvollzogen werden, deren Lösung in Abbildung 4.5 dargestellt ist.

Im Vergleich dazu schlägt die Analyse von (9) nicht fehl, da an der entsprechenden Stelle kein Satzende aufgrund der fehlenden Artikelanbindung angenommen wird.

(9) die Frau sieht die die Mäuse tötende Katze.

Statt dessen wird der näher liegende dritte Artikel „die“ korrekt dem Wort „Mäuse“ zugeordnet und der zweite Artikel bis zum Ende an *nonspec* gebunden. Nachdem „Katze“ gelesen wird, kann diese Bindung durch das konkrete Lexem ausspezifiziert werden, und das Satzende wird erkannt, da keine weiteren *nonspec*-Bindungen verblieben sind.

In ähnlicher Weise schlägt die Analyse von Satz (7) fehl. Hier wird eine Sättigung des Kasusrahmens gegenüber einer Genitiv-Modifikation präferiert. Allerdings kann das Segment „dem Kind“ nicht mehr integriert werden, und es wird als eigenständiger Baum analysiert. Diese Präferenz ergibt sich aus der Wirkung des Objekt-Distanz-Constraints, das die Streckung der Objekt-Bindung von „dem Kind“ bestraft. Auch das oben eingeführte Constraint $C^{37}_{x,y}$ liefert eine Heuristik, die eine Konstellation bestraft, in der der Forderung des Verbs nach einem indirekten Objekt nicht Folge geleistet wird, so daß erst das Fragment „dem Kind“ in dieser Argumentposition des Verbs eingegliedert wird. In den Abbildungen 4.7 und 4.8 ist die falsche der eigentlich gewünschten Analyse gegenübergestellt.

4.6 Gartenpfad-Effekte

Gartenpfad-Effekte werden hier derart verstanden, daß in der Abfolge der Zwischenlösungen Bindungshypothesen revidiert werden, und daß anzunehmen ist, daß dadurch ein höherer Berechnungsaufwand zu leisten ist, wenn nicht gar die Analyse

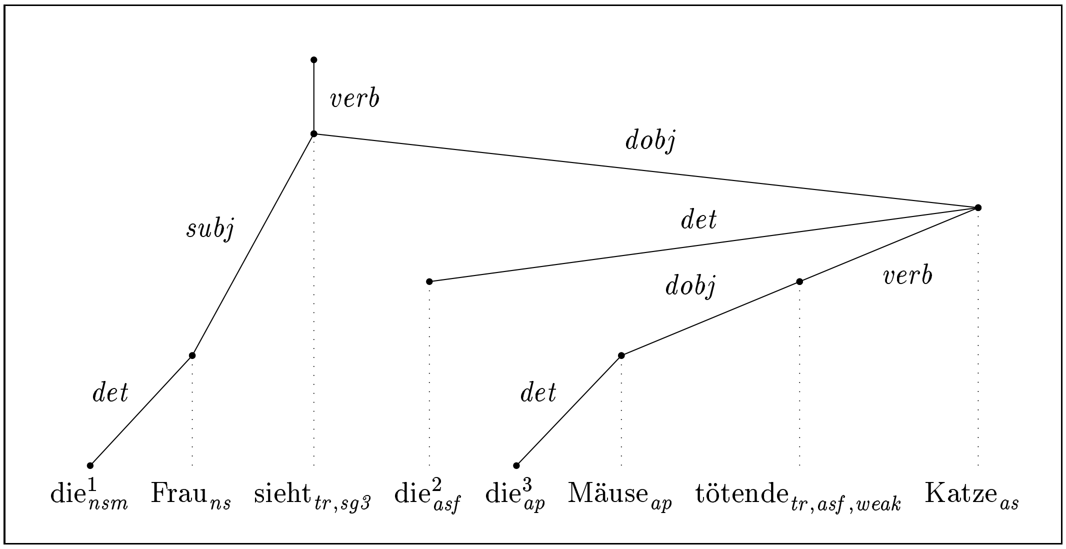


Abbildung 4.6: Korrekte Lösung von Satz (9)

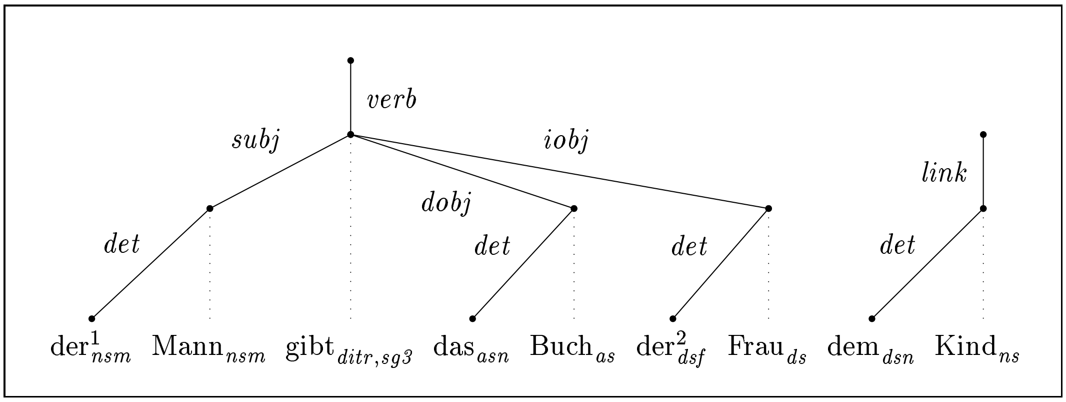


Abbildung 4.7: Inkorrekte Lösung von Satz (7)

gänzlich fehl schlägt. So ist es zum Beispiel nicht mehr möglich, Satz (7) richtig zu analysieren, nachdem einmal festgelegt wurde, daß das Fragment „der Frau“ indirektes Objekt des Verbs „gibt“ ist. Allerdings funktioniert durchaus die Reparatur der Subjekt-Hypothese in Satz (10), nachdem sortale Restriktionen des Verbs einschlägig werden.

(10) die Flöte hört die Frau.

Das Verhalten, daß vorerst dispräferierte Lösungen sich im weiteren Verlauf der Analyse durchsetzen, stellt einen grundlegenden Mechanismus des eliminativen Parsings dar. Allerdings wird erst durch die Präsentation der Zwischenlösungen durch einen inkrementellen Parser diese Eigenschaft sichtbar. Im üblichen Sinne schreibt man dem Satz (10) noch nicht einen Gartenpfad-Effekt zu, obwohl sein Analyseverhalten dem bekannter Gartenpfad-Sätze entspricht. Im Unterschied zu Satz (7) kann in Satz (10) eine fehlgerichtete Analyse noch korrigiert werden. Ein Beispiel für einen tatsächlichen deutschen Gartenpfad-Satz ist (11):

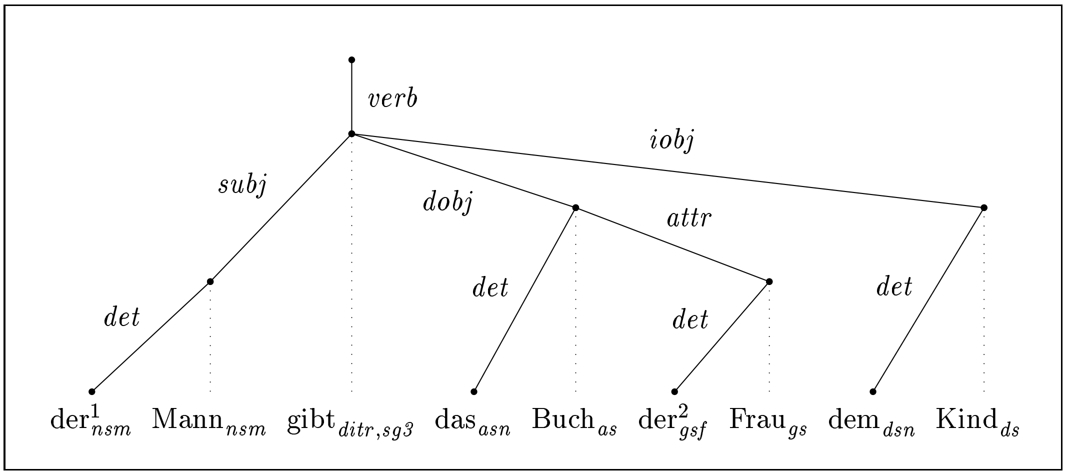


Abbildung 4.8: Korrekte Lösung von Satz (7)

(11) Es erfuhr der Entdecker von Amerika erst im letzten Jahrhundert.

Obwohl Satz (11) nicht von der hiesigen Grammatik abgedeckt wird, ist eine Analyse vorstellbar, in der der Phrase „der Entdecker von Amerika“ eine hohe Präferenz zugeschrieben wird und nach Ende der Eingabe der Kasusrahmen des Verbs unerfüllt bleibt.

Kapitel 5

Abschlußbemerkungen

Die vorliegende Arbeit kann als erster Versuch gewertet werden, constraint-basiertes und inkrementelles Parsing in Einklang zu bringen. Dazu wurde ein bestehendes System untersucht und um eine inkrementelle Verarbeitungskomponente erweitert. Dieser praktische Versuch ermöglichte es, grundlegende Schwierigkeiten des inkrementellen Constraint-Satisfaction auszumachen, um diese anschließend geeignet zu formalisieren. Es konnte eine Beschreibung gefunden werden, die in gleicher Weise, wie nicht-inkrementelles Parsing als *CSP* beschreibbar ist, inkrementelles constraint-basiertes Parsing als *DCSPs* auffaßt.

Es wurden im hiesigen Rahmen notwendige Bedingungen diskutiert, welche ein inkrementelles Verfahren erfüllen muß, um ein beschränktes Laufzeitverhalten zu ermöglichen. Dabei wurden unter Berücksichtigung der Domäne der Sprachverarbeitung zwei neue Konzepte eingeführt: das der unterspezifizierten Variablenbelegung, sowie die Modellierung von Workingsets zur Markierung der Aktivitätsbereiche eines *DCSPs*.

Unterspezifizierte Variablenbelegungen gestalten sich hier derart, daß Dependenzbäume aufgebaut werden, in denen einige Lexeme das maximal unterspezifizierte Lexem *nonspec* modifizieren. Mit Hilfe dieser unterspezifizierten Dependenzen wurde versucht, eine Erwartungsstrategie im vorliegenden Parsing-Paradigma zu modellieren. Dabei stellt der verfolgte Ansatz nur den einfachsten Fall der maximalen Unterspezifizierung dar, da das Lexem *nonspec* keinerlei weitere Eigenschaften besitzt. Es bedarf der weitergehenden Untersuchung, ob in das vorliegende Verfahren ein Unifikationsmechanismus integrierbar ist, mit dessen Hilfe die Eigenschaften der Lexeme im Dependenzbaum propagiert werden können. Auf diese Weise ist es möglich, unterspezifizierte Dependenzen mit mehr Informationen anzureichern und dadurch eine restriktivere Erwartungsstrategie zu modellieren. Damit ergibt sich allerdings eine Hierarchie verschieden spezifizierter Variablenbelegungen. Ob eine Constraint-Logik, die derartige Strategien implementiert, noch mit vertretbarem Aufwand zu verarbeiten ist, bleibt zu untersuchen.

Grundsätzlich ist jedoch festzuhalten, daß eine Modellierung menschlicher Sprachperformanz durch ein inkrementelles Parsingverfahren nicht ohne eine Erwartungsstrategie auskommt, denn Erwartungen stellen eine grundlegende Strategie der menschlichen Wahrnehmung dar. Sie erlauben jedoch nicht nur, Wahrnehmungen effizienter und adäquater zu verarbeiten, sondern können ebenfalls in die Irre leiten, wie man am Beispiel der Gartenpfadsätze sieht. Sprachliche Erwartungen stellen also zum einen eine Strategie zur Bewältigung des Parsingproblems dar, und zum anderen sind sie Grund einschlägiger Performanzprobleme.

Die zweite hier untersuchte Strategie des Workingsets ist eher theoretisch motiviert und entspringt der einfachen Überlegung, daß das Laufzeitverhalten eines inkrementellen Verfahrens konvergieren muß, auch wenn Eingaben stark anwachsen oder gar unendlich groß sind. Dem gegenüber steht die Klassifizierung des Constraint-Satisfaction als NP-vollständiges Problem. Die einzige Möglichkeit, diese beiden Aspekte zu vereinbaren, besteht darin, die Eingangsgrößen des Problems zu verringern, womit sich die Idee des Workingsets motiviert, dessen Zweck es ist, veränderbare Aspekte des *DCSPs* zu beschränken. Da die Größe eines Workingsets unmittelbar mit der Laufzeit des Verfahrens korelliert, ist es folglich notwendige Bedingung für dessen Beschränktheit. Obwohl diese Argumentationen einleuchtend erscheinen, steht eine weitergehende theoretische Betrachtung von *DCSPs* noch aus. (Ramalingam und Reps, 1991, Ramalingam, 1996, Reps und Turnidge, 1996, van Leeuwen et al., 1994).

Unter der Berücksichtigung der Domäne der Sprachverarbeitung wurden hier mehrere Varianten der Definition von Workingsets vorgestellt, und in einem ersten Schritt wurde eine dieser Varianten implementiert. Sie beschränkt allerdings vorerst nur Teile der aktiven Bereiche eines *DCSPs*, nämlich den der veränderbaren Domänen einzelner Variablen. Nicht beschränkt ist die Verwendung vorhandener Lexeme zur Berechnung der Domänen neu gelesener Lexeme. Eine Weiterführung der Idee der Workingsets ist die Verwendung einer ganzen Reihe von Workingsets zu verschiedenen Zwecken, denn auch eine endliche Anzahl beschränkter Workingsets kann ein *DCSP* geeignet beschränken.

Als Begleiterscheinung der skizzierten Strategien ergibt sich eine einfache Heuristik zur Prognostizierung von Satzenden in dem Fall, daß mit einem *DCSP* eine Reihe von Sätzen analysiert wird: stehen in den gefundenen Zwischenlösungen des *DCSPs* keinerlei unerfüllte Erwartungen mehr aus, so liegt ein stabiles Fragment vor, dessen Variablenbelegungen eingefroren werden können. Hier zeigt sich also, daß ein direkter Einfluß von Workingsets und Erwartungsstrategien besteht, da im Falle eines Satzendes Elemente der Workingsets entfernt werden können. An einigen einfachen Beispielen hat sich gezeigt, daß auf diese Weise Textfragmente analysiert werden konnten, die auf herkömmliche, nicht-inkrementelle Weise nicht berechenbar waren.

Die Analyse mehrerer Sätze in einem geschlossenen Verfahren gestattet es, satzübergreifende Phänomene zu modellieren, sodaß es durchaus nicht abwegig ist, anstelle der Berechnung einzelne *DCSPs* pro Satz zu berechnen, eine Reihe von Sätzen in einem einzigen *DCSP* zu analysieren.

Schließlich bleibt noch zu untersuchen, ob auch ein größerer Sprachausschnitt mit Dependenzgrammatiken derart modellierbar ist, daß tatsächlich einschlägige Gartenpfadsätze untersucht werden können.

Abgesehen davon scheint das jetzige Verfahren wenig geeignet, tatsächlich menschliche Performanzprobleme derart vorherzusagen, sodaß es an den selben Stellen erhöhte Berechnungsaufwände zeigt, wie der Mensch beim Verständnis sprachlicher Äußerungen. Dies ist unter anderem in der grundlegenden Eigenschaft des agendabasierten Branch & Bound-Verfahrens begründet, daß der Prozeß der Spezifizierung von Lösungen, also der Überprüfung unerfüllter Erwartungen, nur darin besteht, sie im Vergleich zu anderen Lösungskandidaten schlechter zu bewerten und Lösungen nicht selbst entsprechend umzuwandeln. Hier bleibt zu untersuchen, inwieweit andere Verfahren adäquater Performanzeffekte simulieren können.

Anhang A

Beispielsitzungen

A.1 „Die Frau sieht die Mäuse tötende Katze.“

```
################################################################################
INFO:    reading word 'die' ...
INFO:    solution with better score 1.000e+00 found
INFO:    4 nodes in workingset
INFO:    agenda size: 8/8/1000, 1 solution with score 1.000e+00:
INFO:    #evaluations this loop
         966 unary, 0 statistics, 76 binary

INFO: net: net0, wordgraph: parti4a
INFO: 1 solution(s) with score 1.000e+00:
--------------------------------------------------
   +--- modifier `-' lexical entry, `*' word form
   |+-- label
   ||+- modifiee `-' lexical entry, `*' word form
   |||
00     1.00 (00-01)              IOBJ: die_nsf --LINK--> NIL
01     1.00 (00-01)              DOBJ: die_nsf --LINK--> NIL
02     1.00 (00-01)              SUBJ: die_nsf --LINK--> NIL
03     1.00 (00-01)               DET: die_nsf --LINK--> NIL
04     1.00 (00-01)               SEM: die_nsf --LINK--> NIL
05     1.00 (00-01)               SYN: die_nsf --DET--> NONSPEC

INFO: #unary best: 6/6 6 0 0 0 0 0 0 0 0 0
--------------------------------------------------

INFO: net: id net0, wordgraph parti4a
      #nodes 11, #edges 0
      #evaluations: 966 unary, 0 statistics, 76 binary
      #values: min 1, max 2, total 14, average  1.27
      cache: size 211 #values 53, #hits 4, 0.1 on average, usage 25.12%

################################################################################
INFO:    reading word 'Frau' ...
INFO:    solution with better score 1.000e+00 found
INFO:    8 nodes in workingset
INFO:    agenda size: 35/35/1000, 1 solution with score 1.000e+00:
INFO:    #evaluations this loop
         3702 unary, 0 statistics, 446 binary
```

```
INFO: net: net0, wordgraph: parti4a
INFO: 1 solution(s) with score 1.000e+00:
--------------------------------------------------
   +--- modifier '-' lexical entry, '*' word form
   |+-- label
   ||+- modifiee '-' lexical entry, '*' word form
   |||
00     1.00 (00-01)              IOBJ: die_nsf --LINK--> NIL
01     1.00 (00-01)              DOBJ: die_nsf --LINK--> NIL
02     1.00 (00-01)              SUBJ: die_nsf --LINK--> NIL
03     1.00 (00-01)               DET: die_nsf --LINK--> NIL
04     1.00 (00-01)               SEM: die_nsf --LINK--> NIL
05     1.00 (00-01) (01-02)       SYN: die_nsf --DET--> Frau_nomsg
06     1.00 (01-02)              IOBJ: Frau_nomsg --LINK--> NIL
07     1.00 (01-02)              DOBJ: Frau_nomsg --LINK--> NIL
08     1.00 (01-02)              SUBJ: Frau_nomsg --LINK--> NIL
09     1.00 (01-02) (00-01)       DET: Frau_nomsg --LINK--> die_nsf
10     1.00 (01-02)               SEM: Frau_nomsg --AGENS--> NONSPEC
11     1.00 (01-02)               SYN: Frau_nomsg --SUBJ--> NONSPEC

INFO: #unary best: 12/12 12 0 0 0 0 0 0 0 0 0
--------------------------------------------------

INFO: net: id net0, wordgraph parti4a
      #nodes 20, #edges 0
      #evaluations: 4668 unary, 0 statistics, 522 binary
      #values: min 1, max 9, total 64, average  3.20
      cache: size 431 #values 309, #hits 19, 0.1 on average, usage 71.69%

################################################################################
INFO:    reading word 'sieht' ...
INFO:    solution with better score 1.000e+00 found
INFO:    9 nodes in workingset
INFO:    agenda size: 43/43/1000, 1 solution with score 1.000e+00:
INFO:    #evaluations this loop
         2521 unary, 0 statistics, 417 binary

INFO: net: net0, wordgraph: parti4a
INFO: 1 solution(s) with score 1.000e+00:
--------------------------------------------------
   +--- modifier '-' lexical entry, '*' word form
   |+-- label
   ||+- modifiee '-' lexical entry, '*' word form
   |||
00     1.00 (00-01)              IOBJ: die_nsf --LINK--> NIL
01     1.00 (00-01)              DOBJ: die_nsf --LINK--> NIL
02     1.00 (00-01)              SUBJ: die_nsf --LINK--> NIL
03     1.00 (00-01)               DET: die_nsf --LINK--> NIL
04     1.00 (00-01)               SEM: die_nsf --LINK--> NIL
05     1.00 (00-01) (01-02)       SYN: die_nsf --DET--> Frau_nomsg
06     1.00 (01-02)              IOBJ: Frau_nomsg --LINK--> NIL
07     1.00 (01-02)              DOBJ: Frau_nomsg --LINK--> NIL
08     1.00 (01-02)              SUBJ: Frau_nomsg --LINK--> NIL
09     1.00 (01-02) (00-01)       DET: Frau_nomsg --LINK--> die_nsf
10     1.00 (01-02) (02-03)       SEM: Frau_nomsg --AGENS--> sieht_tr_sg3
11     1.00 (01-02) (02-03)       SYN: Frau_nomsg --SUBJ--> sieht_tr_sg3
12     1.00 (02-03)              IOBJ: sieht_tr_sg3 --LINK--> NIL
```

```
13     1.00 (02-03)               DOBJ: sieht_tr_sg3 --LINK--> NONSPEC
14     1.00 (02-03) (01-02)       SUBJ: sieht_tr_sg3 --LINK--> Frau_nomsg
15     1.00 (02-03)                DET: sieht_tr_sg3 --LINK--> NIL
16     1.00 (02-03)                SEM: sieht_tr_sg3 --LINK--> NIL
17     1.00 (02-03)                SYN: sieht_tr_sg3 --V--> NIL

INFO: #unary best: 18/18 18 0 0 0 0 0 0 0 0 0
--------------------------------------------------

INFO: net: id net0, wordgraph parti4a
      #nodes 26, #edges 0
      #evaluations: 7189 unary, 0 statistics, 939 binary
      #values: min 1, max 12, total 108, average  4.15
      cache: size 863 #values 564, #hits 73, 0.1 on average, usage 65.35%

################################################################################
INFO:    reading word 'die' ...
INFO:    solution with better score 9.167e-01 found
INFO:    13 nodes in workingset
INFO:    agenda size: 86/86/1000, 1 solution with score 9.167e-01:
INFO:    #evaluations this loop
         6589 unary, 0 statistics, 2618 binary

INFO: net: net0, wordgraph: parti4a
INFO: 1 solution(s) with score 9.167e-01:
--------------------------------------------------
   +--- modifier `-' lexical entry, `*' word form
   |+-- label
   ||+- modifiee `-' lexical entry, `*' word form
   |||
00     1.00 (00-01)               IOBJ: die_nsf --LINK--> NIL
01     1.00 (00-01)               DOBJ: die_nsf --LINK--> NIL
02     1.00 (00-01)               SUBJ: die_nsf --LINK--> NIL
03     1.00 (00-01)                DET: die_nsf --LINK--> NIL
04     1.00 (00-01)                SEM: die_nsf --LINK--> NIL
05     1.00 (00-01) (01-02)        SYN: die_nsf --DET--> Frau_nomsg
06     1.00 (01-02)               IOBJ: Frau_nomsg --LINK--> NIL
07     1.00 (01-02)               DOBJ: Frau_nomsg --LINK--> NIL
08     1.00 (01-02)               SUBJ: Frau_nomsg --LINK--> NIL
09     1.00 (01-02) (00-01)        DET: Frau_nomsg --LINK--> die_nsf
10     1.00 (01-02) (02-03)        SEM: Frau_nomsg --AGENS--> sieht_tr_sg3
11     1.00 (01-02) (02-03)        SYN: Frau_nomsg --SUBJ--> sieht_tr_sg3
12     1.00 (02-03)               IOBJ: sieht_tr_sg3 --LINK--> NIL
13     1.00 (02-03) (01-02)       SUBJ: sieht_tr_sg3 --LINK--> Frau_nomsg
14     1.00 (02-03)                DET: sieht_tr_sg3 --LINK--> NIL
15     1.00 (02-03)                SEM: sieht_tr_sg3 --LINK--> NIL
16     1.00 (02-03)                SYN: sieht_tr_sg3 --V--> NIL
17     0.92 (02-03)               DOBJ: sieht_tr_sg3 --LINK--> NONSPEC
18     1.00 (03-04)               IOBJ: die_nsf --LINK--> NIL
19     1.00 (03-04)               DOBJ: die_nsf --LINK--> NIL
20     1.00 (03-04)               SUBJ: die_nsf --LINK--> NIL
21     1.00 (03-04)                DET: die_nsf --LINK--> NIL
22     1.00 (03-04)                SEM: die_nsf --LINK--> NIL
23     1.00 (03-04)                SYN: die_nsf --DET--> NONSPEC

INFO: #unary best: 23/24 23 0 1 0 0 0 0 0 0 0
```

```
       obj_distance2(9.167e-01): 17
--------------------------------------------------

INFO: net: id net0, wordgraph parti4a
      #nodes 37, #edges 0
      #evaluations: 13778 unary, 0 statistics, 3557 binary
      #values: min 1, max 16, total 163, average  4.41
      cache: size 1733 #values 1448, #hits 156, 0.1 on average, usage 83.55%

################################################################################
INFO:    reading word 'Mäuse' ...
INFO:    solution with better score 9.130e-01 found
WARNING: possible end of scentence!!!
INFO:    0 nodes in workingset
INFO:    agenda size: 1/169/1000, 1 solution with score 9.130e-01:
INFO:    #evaluations this loop
         10050 unary, 0 statistics, 4417 binary

INFO: net: net0, wordgraph: parti4a
INFO: 1 solution(s) with score 9.130e-01:
--------------------------------------------------
   +--- modifier '-' lexical entry, '*' word form
   |+-- label
   ||+- modifiee '-' lexical entry, '*' word form
   |||
00     1.00 (00-01)               IOBJ: die_nsf --LINK--> NIL
01     1.00 (00-01)               DOBJ: die_nsf --LINK--> NIL
02     1.00 (00-01)               SUBJ: die_nsf --LINK--> NIL
03     1.00 (00-01)                DET: die_nsf --LINK--> NIL
04     1.00 (00-01)                SEM: die_nsf --LINK--> NIL
05     1.00 (00-01) (01-02)        SYN: die_nsf --DET--> Frau_nomsg
06     1.00 (01-02)               IOBJ: Frau_nomsg --LINK--> NIL
07     1.00 (01-02)               DOBJ: Frau_nomsg --LINK--> NIL
08     1.00 (01-02)               SUBJ: Frau_nomsg --LINK--> NIL
09     1.00 (01-02) (00-01)        DET: Frau_nomsg --LINK--> die_nsf
10     1.00 (01-02) (02-03)        SEM: Frau_nomsg --AGENS--> sieht_tr_sg3
11     1.00 (01-02) (02-03)        SYN: Frau_nomsg --SUBJ--> sieht_tr_sg3
12     1.00 (02-03)               IOBJ: sieht_tr_sg3 --LINK--> NIL
13     1.00 (02-03) (01-02)       SUBJ: sieht_tr_sg3 --LINK--> Frau_nomsg
14     1.00 (02-03)                DET: sieht_tr_sg3 --LINK--> NIL
15     1.00 (02-03)                SEM: sieht_tr_sg3 --LINK--> NIL
16     1.00 (02-03)                SYN: sieht_tr_sg3 --V--> NIL
17     1.00 (03-04)               IOBJ: die_ap --LINK--> NIL
18     1.00 (03-04)               DOBJ: die_ap --LINK--> NIL
19     1.00 (03-04)               SUBJ: die_ap --LINK--> NIL
20     1.00 (03-04)                DET: die_ap --LINK--> NIL
21     1.00 (03-04)                SEM: die_ap --LINK--> NIL
22     0.92 (02-03) (04-05)       DOBJ: sieht_tr_sg3 --LINK--> Mäuse_accpl
23     1.00 (03-04) (04-05)        SYN: die_ap --DET--> Mäuse_accpl
24     1.00 (04-05)               IOBJ: Mäuse_accpl --LINK--> NIL
25     1.00 (04-05)               DOBJ: Mäuse_accpl --LINK--> NIL
26     1.00 (04-05)               SUBJ: Mäuse_accpl --LINK--> NIL
27     1.00 (04-05) (03-04)        DET: Mäuse_accpl --LINK--> die_ap
28     1.00 (04-05) (02-03)        SEM: Mäuse_accpl --THEME--> sieht_tr_sg3
29     1.00 (04-05) (02-03)        SYN: Mäuse_accpl --DOBJ--> sieht_tr_sg3

INFO: #unary best: 29/30 29 1 0 0 0 0 0 0 0 0
```

```
       obj_distance2(9.167e-01): 22
       pref_case_acc(9.990e-01): 23
         pref_num_pl(9.990e-01): 23
       pref_case_acc(9.990e-01): 29
         pref_num_pl(9.990e-01): 29
--------------------------------------------------

INFO: net: id net0, wordgraph parti4a
      #nodes 46, #edges 0
      #evaluations: 23828 unary, 0 statistics, 7974 binary
      #values: min 0, max 1, total 30, average  0.65
      cache: size 3467 #values 2734, #hits 242, 0.1 on average, usage 78.86%

################################################################################
INFO:    reading word 'tötende' ...
INFO:    solution with better score 9.130e-01 found
INFO:    another solution found
INFO:    30 nodes in workingset
INFO:    agenda size: 37/169/1000, 2 solutions with score 9.130e-01:
INFO:    #evaluations this loop
         3000 unary, 0 statistics, 7588 binary

INFO: net: net0, wordgraph: parti4a
INFO: 2 solution(s) with score 9.130e-01:
--------------------------------------------------
   +--- modifier '-' lexical entry, '*' word form
   |+-- label
   ||+- modifiee '-' lexical entry, '*' word form
   |||
00     1.00 (00-01)               IOBJ: die_nsf --LINK--> NIL
01     1.00 (00-01)               DOBJ: die_nsf --LINK--> NIL
02     1.00 (00-01)               SUBJ: die_nsf --LINK--> NIL
03     1.00 (00-01)                DET: die_nsf --LINK--> NIL
04     1.00 (00-01)                SEM: die_nsf --LINK--> NIL
05     1.00 (00-01) (01-02)        SYN: die_nsf --DET--> Frau_nomsg
06     1.00 (01-02)               IOBJ: Frau_nomsg --LINK--> NIL
07     1.00 (01-02)               DOBJ: Frau_nomsg --LINK--> NIL
08     1.00 (01-02)               SUBJ: Frau_nomsg --LINK--> NIL
09     1.00 (01-02) (00-01)        DET: Frau_nomsg --LINK--> die_nsf
10     1.00 (01-02) (02-03)        SEM: Frau_nomsg --AGENS--> sieht_tr_sg3
11     1.00 (01-02) (02-03)        SYN: Frau_nomsg --SUBJ--> sieht_tr_sg3
12     1.00 (02-03)               IOBJ: sieht_tr_sg3 --LINK--> NIL
13     1.00 (02-03) (01-02)       SUBJ: sieht_tr_sg3 --LINK--> Frau_nomsg
14     1.00 (02-03)                DET: sieht_tr_sg3 --LINK--> NIL
15     1.00 (02-03)                SEM: sieht_tr_sg3 --LINK--> NIL
16     1.00 (02-03)                SYN: sieht_tr_sg3 --V--> NIL
17     1.00 (03-04)               IOBJ: die_ap --LINK--> NIL
18     1.00 (03-04)               DOBJ: die_ap --LINK--> NIL
19     1.00 (03-04)               SUBJ: die_ap --LINK--> NIL
20     1.00 (03-04)                DET: die_ap --LINK--> NIL
21     1.00 (03-04)                SEM: die_ap --LINK--> NIL
22     0.92 (02-03) (04-05)       DOBJ: sieht_tr_sg3 --LINK--> Mäuse_accpl
23     1.00 (03-04) (04-05)        SYN: die_ap --DET--> Mäuse_accpl
24     1.00 (04-05)               IOBJ: Mäuse_accpl --LINK--> NIL
25     1.00 (04-05)               DOBJ: Mäuse_accpl --LINK--> NIL
26     1.00 (04-05)               SUBJ: Mäuse_accpl --LINK--> NIL
```

```
27     1.00 (04-05) (03-04)        DET: Mäuse_accpl --LINK--> die_ap
28     1.00 (04-05) (02-03)        SEM: Mäuse_accpl --THEME--> sieht_tr_sg3
29     1.00 (04-05) (02-03)        SYN: Mäuse_accpl --DOBJ--> sieht_tr_sg3
30     1.00 (05-06)               IOBJ: tötende_tr_nomsgfem_strong --LINK--> NIL
31     1.00 (05-06)               DOBJ: tötende_tr_nomsgfem_strong --LINK--> NONSPEC
32     1.00 (05-06)               SUBJ: tötende_tr_nomsgfem_strong --LINK--> NIL
33     1.00 (05-06)                DET: tötende_tr_nomsgfem_strong --LINK--> NIL
34     1.00 (05-06)                SEM: tötende_tr_nomsgfem_strong --AGENS_REV--> NONSPEC
35     1.00 (05-06)                SYN: tötende_tr_nomsgfem_strong --V--> NONSPEC

INFO: #unary best: 35/36 35 1 0 0 0 0 0 0 0 0

       obj_distance2(9.167e-01): 22
       pref_case_acc(9.990e-01): 23
         pref_num_pl(9.990e-01): 23
       pref_case_acc(9.990e-01): 29
         pref_num_pl(9.990e-01): 29
--------------------------------------------------
   +--- modifier ‘-’ lexical entry, ‘*’ word form
   |+-- label
   ||+- modifiee ‘-’ lexical entry, ‘*’ word form
   |||
00     1.00 (00-01)               IOBJ: die_nsf --LINK--> NIL
01     1.00 (00-01)               DOBJ: die_nsf --LINK--> NIL
02     1.00 (00-01)               SUBJ: die_nsf --LINK--> NIL
03     1.00 (00-01)                DET: die_nsf --LINK--> NIL
04     1.00 (00-01)                SEM: die_nsf --LINK--> NIL
05     1.00 (00-01) (01-02)        SYN: die_nsf --DET--> Frau_nomsg
06     1.00 (01-02)               IOBJ: Frau_nomsg --LINK--> NIL
07     1.00 (01-02)               DOBJ: Frau_nomsg --LINK--> NIL
08     1.00 (01-02)               SUBJ: Frau_nomsg --LINK--> NIL
09     1.00 (01-02) (00-01)        DET: Frau_nomsg --LINK--> die_nsf
10     1.00 (01-02) (02-03)        SEM: Frau_nomsg --AGENS--> sieht_tr_sg3
11     1.00 (01-02) (02-03)        SYN: Frau_nomsg --SUBJ--> sieht_tr_sg3
12     1.00 (02-03)               IOBJ: sieht_tr_sg3 --LINK--> NIL
13     1.00 (02-03) (01-02)       SUBJ: sieht_tr_sg3 --LINK--> Frau_nomsg
14     1.00 (02-03)                DET: sieht_tr_sg3 --LINK--> NIL
15     1.00 (02-03)                SEM: sieht_tr_sg3 --LINK--> NIL
16     1.00 (02-03)                SYN: sieht_tr_sg3 --V--> NIL
17     1.00 (03-04)               IOBJ: die_ap --LINK--> NIL
18     1.00 (03-04)               DOBJ: die_ap --LINK--> NIL
19     1.00 (03-04)               SUBJ: die_ap --LINK--> NIL
20     1.00 (03-04)                DET: die_ap --LINK--> NIL
21     1.00 (03-04)                SEM: die_ap --LINK--> NIL
22     0.92 (02-03) (04-05)       DOBJ: sieht_tr_sg3 --LINK--> Mäuse_accpl
23     1.00 (03-04) (04-05)        SYN: die_ap --DET--> Mäuse_accpl
24     1.00 (04-05)               IOBJ: Mäuse_accpl --LINK--> NIL
25     1.00 (04-05)               DOBJ: Mäuse_accpl --LINK--> NIL
26     1.00 (04-05)               SUBJ: Mäuse_accpl --LINK--> NIL
27     1.00 (04-05) (03-04)        DET: Mäuse_accpl --LINK--> die_ap
28     1.00 (04-05) (02-03)        SEM: Mäuse_accpl --THEME--> sieht_tr_sg3
29     1.00 (04-05) (02-03)        SYN: Mäuse_accpl --DOBJ--> sieht_tr_sg3
30     1.00 (05-06)               IOBJ: tötende_intr_nomsgfem_strong --LINK--> NIL
31 - * 1.00 (05-06)               DOBJ: tötende_intr_nomsgfem_strong --LINK--> NIL
32     1.00 (05-06)               SUBJ: tötende_intr_nomsgfem_strong --LINK--> NIL
33     1.00 (05-06)                DET: tötende_intr_nomsgfem_strong --LINK--> NIL
34 -   1.00 (05-06)                SEM: tötende_intr_nomsgfem_strong --AGENS_REV--> NONSPEC
```

```
35     1.00 (05-06)                SYN: tötende_intr_nomsgfem_strong --V--> NONSPEC

INFO: #unary best: 35/36 35 1 0 0 0 0 0 0 0 0

       obj_distance2(9.167e-01): 22
       pref_case_acc(9.990e-01): 23
         pref_num_pl(9.990e-01): 23
       pref_case_acc(9.990e-01): 29
         pref_num_pl(9.990e-01): 29
--------------------------------------------------

INFO: net: id net0, wordgraph parti4a
      #nodes 70, #edges 0
      #evaluations: 26828 unary, 0 statistics, 15562 binary
      #values: min 0, max 3, total 61, average  0.87
      cache: size 6947 #values 3853, #hits 242, 0.1 on average, usage 55.46%

################################################################################
INFO:    reading word 'Katze' ...
INFO:    solution with better score 1.187e-01 found
INFO:    34 nodes in workingset
INFO:    agenda size: 379/379/1000, 1 solution with score 1.187e-01:
INFO:    #evaluations this loop
         19472 unary, 0 statistics, 64353 binary

INFO: net: net0, wordgraph: parti4a
INFO: 1 solution(s) with score 1.187e-01:
--------------------------------------------------
   +--- modifier '-' lexical entry, '*' word form
   |+-- label
   ||+- modifiee '-' lexical entry, '*' word form
   |||
00     1.00 (00-01)               IOBJ: die_nsf --LINK--> NIL
01     1.00 (00-01)               DOBJ: die_nsf --LINK--> NIL
02     1.00 (00-01)               SUBJ: die_nsf --LINK--> NIL
03     1.00 (00-01)                DET: die_nsf --LINK--> NIL
04     1.00 (00-01)                SEM: die_nsf --LINK--> NIL
05     1.00 (00-01) (01-02)        SYN: die_nsf --DET--> Frau_nomsg
06     1.00 (01-02)               IOBJ: Frau_nomsg --LINK--> NIL
07     1.00 (01-02)               DOBJ: Frau_nomsg --LINK--> NIL
08     1.00 (01-02)               SUBJ: Frau_nomsg --LINK--> NIL
09     1.00 (01-02) (00-01)        DET: Frau_nomsg --LINK--> die_nsf
10     1.00 (01-02) (02-03)        SEM: Frau_nomsg --AGENS--> sieht_tr_sg3
11     1.00 (01-02) (02-03)        SYN: Frau_nomsg --SUBJ--> sieht_tr_sg3
12     1.00 (02-03)               IOBJ: sieht_tr_sg3 --LINK--> NIL
13     1.00 (02-03) (01-02)       SUBJ: sieht_tr_sg3 --LINK--> Frau_nomsg
14     1.00 (02-03)                DET: sieht_tr_sg3 --LINK--> NIL
15     1.00 (02-03)                SEM: sieht_tr_sg3 --LINK--> NIL
16     1.00 (02-03)                SYN: sieht_tr_sg3 --V--> NIL
17     1.00 (03-04)               IOBJ: die_ap --LINK--> NIL
18     1.00 (03-04)               DOBJ: die_ap --LINK--> NIL
19     1.00 (03-04)               SUBJ: die_ap --LINK--> NIL
20     1.00 (03-04)                DET: die_ap --LINK--> NIL
21     1.00 (03-04)                SEM: die_ap --LINK--> NIL
22     0.92 (02-03) (04-05)       DOBJ: sieht_tr_sg3 --LINK--> Mäuse_accpl
23     1.00 (03-04) (04-05)        SYN: die_ap --DET--> Mäuse_accpl
24     1.00 (04-05)               IOBJ: Mäuse_accpl --LINK--> NIL
```

```
25     1.00 (04-05)               DOBJ: Mäuse_accpl --LINK--> NIL
26     1.00 (04-05)               SUBJ: Mäuse_accpl --LINK--> NIL
27     1.00 (04-05) (03-04)        DET: Mäuse_accpl --LINK--> die_ap
28     1.00 (04-05) (02-03)        SEM: Mäuse_accpl --THEME--> sieht_tr_sg3
29     1.00 (04-05) (02-03)        SYN: Mäuse_accpl --DOBJ--> sieht_tr_sg3
30     1.00 (05-06)               IOBJ: tötende_intr_nomsgfem_strong --LINK--> NIL
31     1.00 (05-06)               DOBJ: tötende_intr_nomsgfem_strong --LINK--> NIL
32     1.00 (05-06)               SUBJ: tötende_intr_nomsgfem_strong --LINK--> NIL
33     1.00 (05-06)                DET: tötende_intr_nomsgfem_strong --LINK--> NIL
34     1.00 (05-06) (06-07)        SEM: tötende_intr_nomsgfem_strong --AGENS_REV--> Katze_nomsg
35     1.00 (05-06) (06-07)        SYN: tötende_intr_nomsgfem_strong --V--> Katze_nomsg
36     1.00 (06-07)               IOBJ: Katze_nomsg --LINK--> NIL
37     1.00 (06-07)               DOBJ: Katze_nomsg --LINK--> NIL
38     1.00 (06-07)               SUBJ: Katze_nomsg --LINK--> NIL
39     0.13 (06-07)                DET: Katze_nomsg --LINK--> NONSPEC
40     1.00 (06-07)                SEM: Katze_nomsg --AGENS--> NONSPEC
41     1.00 (06-07)                SYN: Katze_nomsg --SUBJ--> NONSPEC

INFO: #unary best: 41/42 41 1 0 0 0 0 0 0 0 0

       obj_distance2(9.167e-01): 22
       pref_case_acc(9.990e-01): 23
         pref_num_pl(9.990e-01): 23
       pref_case_acc(9.990e-01): 29
         pref_num_pl(9.990e-01): 29
             det_pos(1.300e-01): 39
--------------------------------------------------

INFO: net: id net0, wordgraph parti4a
      #nodes 79, #edges 0
      #evaluations: 46300 unary, 0 statistics, 79915 binary
      #values: min 0, max 50, total 306, average  3.87
      cache: size 13901 #values 8799, #hits 378, 0.0 on average, usage 63.30%

################################################################################
INFO:    end of scentence!!!
INFO:    solution with better score 9.130e-09 found
INFO:    #evaluations: 19472 unary, 0 statistics, 243659 binary

INFO: net: net0, wordgraph: parti4a
INFO: 1 solution(s) with score 9.130e-09:
--------------------------------------------------
   +--- modifier '-' lexical entry, '*' word form
   |+-- label
   ||+- modifiee '-' lexical entry, '*' word form
   |||
00     1.00 (00-01)               IOBJ: die_nsf --LINK--> NIL
01     1.00 (00-01)               DOBJ: die_nsf --LINK--> NIL
02     1.00 (00-01)               SUBJ: die_nsf --LINK--> NIL
03     1.00 (00-01)                DET: die_nsf --LINK--> NIL
04     1.00 (00-01)                SEM: die_nsf --LINK--> NIL
05     1.00 (00-01) (01-02)        SYN: die_nsf --DET--> Frau_nomsg
06     1.00 (01-02)               IOBJ: Frau_nomsg --LINK--> NIL
07     1.00 (01-02)               DOBJ: Frau_nomsg --LINK--> NIL
08     1.00 (01-02)               SUBJ: Frau_nomsg --LINK--> NIL
09     1.00 (01-02) (00-01)        DET: Frau_nomsg --LINK--> die_nsf
10     1.00 (01-02) (02-03)        SEM: Frau_nomsg --AGENS--> sieht_tr_sg3
```

```
11     1.00 (01-02) (02-03)        SYN: Frau_nomsg --SUBJ--> sieht_tr_sg3
12     1.00 (02-03)               IOBJ: sieht_tr_sg3 --LINK--> NIL
13     1.00 (02-03) (01-02)       SUBJ: sieht_tr_sg3 --LINK--> Frau_nomsg
14     1.00 (02-03)                DET: sieht_tr_sg3 --LINK--> NIL
15     1.00 (02-03)                SEM: sieht_tr_sg3 --LINK--> NIL
16     1.00 (02-03)                SYN: sieht_tr_sg3 --V--> NIL
17     1.00 (03-04)               IOBJ: die_ap --LINK--> NIL
18     1.00 (03-04)               DOBJ: die_ap --LINK--> NIL
19     1.00 (03-04)               SUBJ: die_ap --LINK--> NIL
20     1.00 (03-04)                DET: die_ap --LINK--> NIL
21     1.00 (03-04)                SEM: die_ap --LINK--> NIL
22     0.92 (02-03) (04-05)       DOBJ: sieht_tr_sg3 --LINK--> Mäuse_accpl
23     1.00 (03-04) (04-05)        SYN: die_ap --DET--> Mäuse_accpl
24     1.00 (04-05)               IOBJ: Mäuse_accpl --LINK--> NIL
25     1.00 (04-05)               DOBJ: Mäuse_accpl --LINK--> NIL
26     1.00 (04-05)               SUBJ: Mäuse_accpl --LINK--> NIL
27     1.00 (04-05) (03-04)        DET: Mäuse_accpl --LINK--> die_ap
28     1.00 (04-05) (02-03)        SEM: Mäuse_accpl --THEME--> sieht_tr_sg3
29     1.00 (04-05) (02-03)        SYN: Mäuse_accpl --DOBJ--> sieht_tr_sg3
30     1.00 (05-06)               IOBJ: tötende_intr_nomsgfem_strong --LINK--> NIL
31     1.00 (05-06)               DOBJ: tötende_intr_nomsgfem_strong --LINK--> NIL
32     1.00 (05-06)               SUBJ: tötende_intr_nomsgfem_strong --LINK--> NIL
33     1.00 (05-06)                DET: tötende_intr_nomsgfem_strong --LINK--> NIL
34     1.00 (05-06) (06-07)        SEM: tötende_intr_nomsgfem_strong --AGENS_REV--> Katze_nomsg
35     1.00 (05-06) (06-07)        SYN: tötende_intr_nomsgfem_strong --V--> Katze_nomsg
36     1.00 (06-07)               IOBJ: Katze_nomsg --LINK--> NIL
37     1.00 (06-07)               DOBJ: Katze_nomsg --LINK--> NIL
38     1.00 (06-07)               SUBJ: Katze_nomsg --LINK--> NIL
39     0.01 (06-07)                DET: Katze_nomsg --LINK--> NIL
40     1.00 (06-07)                SEM: Katze_nomsg --LINK--> NIL
41     0.00 (06-07)                SYN: Katze_nomsg --LINK--> NIL

INFO: #unary best: 41/42 41 1 0 0 0 0 0 0 0 0

       obj_distance2(9.167e-01): 22
       pref_case_acc(9.990e-01): 23
         pref_num_pl(9.990e-01): 23
       pref_case_acc(9.990e-01): 29
         pref_num_pl(9.990e-01): 29
           det_exist(1.000e-02): 39
    syn_link_penalty(1.000e-06): 41
--------------------------------------------------

INFO: net: id net0, wordgraph parti4a
      #nodes 79, #edges 0
      #evaluations: 46300 unary, 0 statistics, 259221 binary
      #values: min 0, max 46, total 268, average  3.39
      cache: size 13901 #values 10307, #hits 378, 0.0 on average, usage 74.15%
```

A.2 „Der Mann gibt das Buch der Frau dem Kind.“

```
################################################################################
INFO:    reading word 'der' ...
INFO:    solution with better score 1.000e+00 found
INFO:    4 nodes in workingset
INFO:    agenda size: 7/7/1000, 1 solution with score 1.000e+00:
```

```
INFO:    #evaluations this loop
         944 unary, 0 statistics, 64 binary

INFO: net: net0, wordgraph: attr2
INFO: 1 solution(s) with score 1.000e+00:
--------------------------------------------------
   +--- modifier `-' lexical entry, `*' word form
   |+-- label
   ||+- modifiee `-' lexical entry, `*' word form
   |||
00     1.00 (00-01)               IOBJ: der_nsm --LINK--> NIL
01     1.00 (00-01)               DOBJ: der_nsm --LINK--> NIL
02     1.00 (00-01)               SUBJ: der_nsm --LINK--> NIL
03     1.00 (00-01)                DET: der_nsm --LINK--> NIL
04     1.00 (00-01)                SEM: der_nsm --LINK--> NIL
05     1.00 (00-01)                SYN: der_nsm --DET--> NONSPEC

INFO: #unary best: 6/6 6 0 0 0 0 0 0 0 0 0
--------------------------------------------------

INFO: net: id net0, wordgraph attr2
      #nodes 11, #edges 0
      #evaluations: 944 unary, 0 statistics, 64 binary
      #values: min 1, max 2, total 13, average  1.18
      cache: size 211 #values 48, #hits 4, 0.1 on average, usage 22.75%

#################################################################################
INFO:    reading word 'Mann' ...
INFO:    solution with better score 1.000e+00 found
INFO:    7 nodes in workingset
INFO:    agenda size: 28/28/1000, 1 solution with score 1.000e+00:
INFO:    #evaluations this loop
         2863 unary, 0 statistics, 400 binary

INFO: net: net0, wordgraph: attr2
INFO: 1 solution(s) with score 1.000e+00:
--------------------------------------------------
   +--- modifier `-' lexical entry, `*' word form
   |+-- label
   ||+- modifiee `-' lexical entry, `*' word form
   |||
00     1.00 (00-01)               IOBJ: der_nsm --LINK--> NIL
01     1.00 (00-01)               DOBJ: der_nsm --LINK--> NIL
02     1.00 (00-01)               SUBJ: der_nsm --LINK--> NIL
03     1.00 (00-01)                DET: der_nsm --LINK--> NIL
04     1.00 (00-01)                SEM: der_nsm --LINK--> NIL
05     1.00 (00-01) (01-02)        SYN: der_nsm --DET--> Mann_nomsg
06     1.00 (01-02)               IOBJ: Mann_nomsg --LINK--> NIL
07     1.00 (01-02)               DOBJ: Mann_nomsg --LINK--> NIL
08     1.00 (01-02)               SUBJ: Mann_nomsg --LINK--> NIL
09     1.00 (01-02) (00-01)        DET: Mann_nomsg --LINK--> der_nsm
10     1.00 (01-02)                SEM: Mann_nomsg --AGENS--> NONSPEC
11     1.00 (01-02)                SYN: Mann_nomsg --SUBJ--> NONSPEC

INFO: #unary best: 12/12 12 0 0 0 0 0 0 0 0 0
--------------------------------------------------
```

```
INFO: net: id net0, wordgraph attr2
      #nodes 19, #edges 0
      #evaluations: 3807 unary, 0 statistics, 464 binary
      #values: min 1, max 9, total 54, average  2.84
      cache: size 431 #values 278, #hits 16, 0.1 on average, usage 64.50%

################################################################################
INFO:    reading word 'gibt' ...
INFO:    solution with better score 1.000e+00 found
INFO:    8 nodes in workingset
INFO:    agenda size: 36/36/1000, 1 solution with score 1.000e+00:
INFO:    #evaluations this loop
         2123 unary, 0 statistics, 417 binary

INFO: net: net0, wordgraph: attr2
INFO: 1 solution(s) with score 1.000e+00:
-------------------------------------------------
   +--- modifier '-' lexical entry, '*' word form
   |+-- label
   ||+- modifiee '-' lexical entry, '*' word form
   |||
00     1.00 (00-01)              IOBJ: der_nsm --LINK--> NIL
01     1.00 (00-01)              DOBJ: der_nsm --LINK--> NIL
02     1.00 (00-01)              SUBJ: der_nsm --LINK--> NIL
03     1.00 (00-01)               DET: der_nsm --LINK--> NIL
04     1.00 (00-01)               SEM: der_nsm --LINK--> NIL
05     1.00 (00-01) (01-02)       SYN: der_nsm --DET--> Mann_nomsg
06     1.00 (01-02)              IOBJ: Mann_nomsg --LINK--> NIL
07     1.00 (01-02)              DOBJ: Mann_nomsg --LINK--> NIL
08     1.00 (01-02)              SUBJ: Mann_nomsg --LINK--> NIL
09     1.00 (01-02) (00-01)       DET: Mann_nomsg --LINK--> der_nsm
10     1.00 (01-02) (02-03)       SEM: Mann_nomsg --AGENS--> gibt_ditr_sg3
11     1.00 (01-02) (02-03)       SYN: Mann_nomsg --SUBJ--> gibt_ditr_sg3
12     1.00 (02-03)              IOBJ: gibt_ditr_sg3 --LINK--> NONSPEC
13     1.00 (02-03)              DOBJ: gibt_ditr_sg3 --LINK--> NONSPEC
14     1.00 (02-03) (01-02)      SUBJ: gibt_ditr_sg3 --LINK--> Mann_nomsg
15     1.00 (02-03)               DET: gibt_ditr_sg3 --LINK--> NIL
16     1.00 (02-03)               SEM: gibt_ditr_sg3 --LINK--> NIL
17     1.00 (02-03)               SYN: gibt_ditr_sg3 --V--> NIL

INFO: #unary best: 18/18 18 0 0 0 0 0 0 0 0 0
-------------------------------------------------

INFO: net: id net0, wordgraph attr2
      #nodes 25, #edges 0
      #evaluations: 5930 unary, 0 statistics, 881 binary
      #values: min 1, max 13, total 92, average  3.68
      cache: size 863 #values 533, #hits 70, 0.1 on average, usage 61.76%

################################################################################
INFO:    reading word 'das' ...
INFO:    solution with better score 8.403e-01 found
INFO:    10 nodes in workingset
INFO:    agenda size: 49/49/1000, 1 solution with score 8.403e-01:
INFO:    #evaluations this loop
         3349 unary, 0 statistics, 1159 binary
```

```
INFO: net: net0, wordgraph: attr2
INFO: 1 solution(s) with score 8.403e-01:
------------------------------------------------
   +--- modifier '-' lexical entry, '*' word form
   |+-- label
   ||+- modifiee '-' lexical entry, '*' word form
   |||
00     1.00 (00-01)              IOBJ: der_nsm --LINK--> NIL
01     1.00 (00-01)              DOBJ: der_nsm --LINK--> NIL
02     1.00 (00-01)              SUBJ: der_nsm --LINK--> NIL
03     1.00 (00-01)               DET: der_nsm --LINK--> NIL
04     1.00 (00-01)               SEM: der_nsm --LINK--> NIL
05     1.00 (00-01) (01-02)       SYN: der_nsm --DET--> Mann_nomsg
06     1.00 (01-02)              IOBJ: Mann_nomsg --LINK--> NIL
07     1.00 (01-02)              DOBJ: Mann_nomsg --LINK--> NIL
08     1.00 (01-02)              SUBJ: Mann_nomsg --LINK--> NIL
09     1.00 (01-02) (00-01)       DET: Mann_nomsg --LINK--> der_nsm
10     1.00 (01-02) (02-03)       SEM: Mann_nomsg --AGENS--> gibt_ditr_sg3
11     1.00 (01-02) (02-03)       SYN: Mann_nomsg --SUBJ--> gibt_ditr_sg3
12     1.00 (02-03) (01-02)      SUBJ: gibt_ditr_sg3 --LINK--> Mann_nomsg
13     1.00 (02-03)               DET: gibt_ditr_sg3 --LINK--> NIL
14     1.00 (02-03)               SEM: gibt_ditr_sg3 --LINK--> NIL
15     1.00 (02-03)               SYN: gibt_ditr_sg3 --V--> NIL
16     0.92 (02-03)              IOBJ: gibt_ditr_sg3 --LINK--> NONSPEC
17     0.92 (02-03)              DOBJ: gibt_ditr_sg3 --LINK--> NONSPEC
18     1.00 (03-04)              IOBJ: das_nsn --LINK--> NIL
19     1.00 (03-04)              DOBJ: das_nsn --LINK--> NIL
20     1.00 (03-04)              SUBJ: das_nsn --LINK--> NIL
21     1.00 (03-04)               DET: das_nsn --LINK--> NIL
22     1.00 (03-04)               SEM: das_nsn --LINK--> NIL
23     1.00 (03-04)               SYN: das_nsn --DET--> NONSPEC

INFO: #unary best: 22/24 22 0 0 2 0 0 0 0 0 0

       obj_distance3(9.167e-01): 16
       obj_distance2(9.167e-01): 17
------------------------------------------------

INFO: net: id net0, wordgraph attr2
      #nodes 32, #edges 0
      #evaluations: 9279 unary, 0 statistics, 2040 binary
      #values: min 1, max 15, total 123, average  3.84
      cache: size 1733 #values 971, #hits 124, 0.1 on average, usage 56.03%

################################################################################
INFO:    reading word 'Buch' ...
INFO:    solution with better score 7.741e-01 found
INFO:    13 nodes in workingset
INFO:    agenda size: 109/109/1000, 1 solution with score 7.741e-01:
INFO:    #evaluations this loop
         6153 unary, 0 statistics, 3833 binary

INFO: net: net0, wordgraph: attr2
INFO: 1 solution(s) with score 7.741e-01:
------------------------------------------------
   +--- modifier '-' lexical entry, '*' word form
   |+-- label
```

```
   ||+- modifiee ‘-’ lexical entry, ‘*’ word form
   |||
00     1.00 (00-01)              IOBJ: der_nsm --LINK--> NIL
01     1.00 (00-01)              DOBJ: der_nsm --LINK--> NIL
02     1.00 (00-01)              SUBJ: der_nsm --LINK--> NIL
03     1.00 (00-01)               DET: der_nsm --LINK--> NIL
04     1.00 (00-01)               SEM: der_nsm --LINK--> NIL
05     1.00 (00-01) (01-02)       SYN: der_nsm --DET--> Mann_nomsg
06     1.00 (01-02)              IOBJ: Mann_nomsg --LINK--> NIL
07     1.00 (01-02)              DOBJ: Mann_nomsg --LINK--> NIL
08     1.00 (01-02)              SUBJ: Mann_nomsg --LINK--> NIL
09     1.00 (01-02) (00-01)       DET: Mann_nomsg --LINK--> der_nsm
10     1.00 (01-02) (02-03)       SEM: Mann_nomsg --AGENS--> gibt_ditr_sg3
11     1.00 (01-02) (02-03)       SYN: Mann_nomsg --SUBJ--> gibt_ditr_sg3
12     1.00 (02-03) (01-02)      SUBJ: gibt_ditr_sg3 --LINK--> Mann_nomsg
13     1.00 (02-03)               DET: gibt_ditr_sg3 --LINK--> NIL
14     1.00 (02-03)               SEM: gibt_ditr_sg3 --LINK--> NIL
15     1.00 (02-03)               SYN: gibt_ditr_sg3 --V--> NIL
16     1.00 (03-04)              IOBJ: das_asn --LINK--> NIL
17     1.00 (03-04)              DOBJ: das_asn --LINK--> NIL
18     1.00 (03-04)              SUBJ: das_asn --LINK--> NIL
19     1.00 (03-04)               DET: das_asn --LINK--> NIL
20     1.00 (03-04)               SEM: das_asn --LINK--> NIL
21     0.85 (02-03)              IOBJ: gibt_ditr_sg3 --LINK--> NONSPEC
22     1.00 (03-04) (04-05)       SYN: das_asn --DET--> Buch_accsg
23     0.92 (02-03) (04-05)      DOBJ: gibt_ditr_sg3 --LINK--> Buch_accsg
24     1.00 (04-05)              IOBJ: Buch_accsg --LINK--> NIL
25     1.00 (04-05)              DOBJ: Buch_accsg --LINK--> NIL
26     1.00 (04-05)              SUBJ: Buch_accsg --LINK--> NIL
27     1.00 (04-05) (03-04)       DET: Buch_accsg --LINK--> das_asn
28     1.00 (04-05) (02-03)       SEM: Buch_accsg --THEME--> gibt_ditr_sg3
29     1.00 (04-05) (02-03)       SYN: Buch_accsg --DOBJ--> gibt_ditr_sg3

INFO: #unary best: 28/30 28 0 1 0 1 0 0 0 0 0

       obj_distance3(8.462e-01): 21
       pref_case_acc(9.990e-01): 22
       obj_distance2(9.167e-01): 23
       pref_case_acc(9.990e-01): 29
-------------------------------------------------

INFO: net: id net0, wordgraph attr2
      #nodes 40, #edges 0
      #evaluations: 15432 unary, 0 statistics, 5873 binary
      #values: min 1, max 18, total 206, average  5.15
      cache: size 3467 #values 2145, #hits 178, 0.1 on average, usage 61.87%

################################################################################
INFO:    reading word 'der' ...
INFO:    solution with better score 7.188e-01 found
INFO:    17 nodes in workingset
INFO:    agenda size: 169/169/1000, 1 solution with score 7.188e-01:
INFO:    #evaluations this loop
         9054 unary, 0 statistics, 6662 binary

INFO: net: net0, wordgraph: attr2
INFO: 1 solution(s) with score 7.188e-01:
```

```
--------------------------------------------------
   +--- modifier '-' lexical entry, '*' word form
   |+-- label
   ||+- modifiee '-' lexical entry, '*' word form
   |||
00     1.00 (00-01)              IOBJ: der_nsm --LINK--> NIL
01     1.00 (00-01)              DOBJ: der_nsm --LINK--> NIL
02     1.00 (00-01)              SUBJ: der_nsm --LINK--> NIL
03     1.00 (00-01)               DET: der_nsm --LINK--> NIL
04     1.00 (00-01)               SEM: der_nsm --LINK--> NIL
05     1.00 (00-01) (01-02)       SYN: der_nsm --DET--> Mann_nomsg
06     1.00 (01-02)              IOBJ: Mann_nomsg --LINK--> NIL
07     1.00 (01-02)              DOBJ: Mann_nomsg --LINK--> NIL
08     1.00 (01-02)              SUBJ: Mann_nomsg --LINK--> NIL
09     1.00 (01-02) (00-01)       DET: Mann_nomsg --LINK--> der_nsm
10     1.00 (01-02) (02-03)       SEM: Mann_nomsg --AGENS--> gibt_ditr_sg3
11     1.00 (01-02) (02-03)       SYN: Mann_nomsg --SUBJ--> gibt_ditr_sg3
12     1.00 (02-03) (01-02)      SUBJ: gibt_ditr_sg3 --LINK--> Mann_nomsg
13     1.00 (02-03)               DET: gibt_ditr_sg3 --LINK--> NIL
14     1.00 (02-03)               SEM: gibt_ditr_sg3 --LINK--> NIL
15     1.00 (02-03)               SYN: gibt_ditr_sg3 --V--> NIL
16     1.00 (03-04)              IOBJ: das_asn --LINK--> NIL
17     1.00 (03-04)              DOBJ: das_asn --LINK--> NIL
18     1.00 (03-04)              SUBJ: das_asn --LINK--> NIL
19     1.00 (03-04)               DET: das_asn --LINK--> NIL
20     1.00 (03-04)               SEM: das_asn --LINK--> NIL
21     1.00 (03-04) (04-05)       SYN: das_asn --DET--> Buch_accsg
22     0.92 (02-03) (04-05)      DOBJ: gibt_ditr_sg3 --LINK--> Buch_accsg
23     1.00 (04-05)              IOBJ: Buch_accsg --LINK--> NIL
24     1.00 (04-05)              DOBJ: Buch_accsg --LINK--> NIL
25     1.00 (04-05)              SUBJ: Buch_accsg --LINK--> NIL
26     1.00 (04-05) (03-04)       DET: Buch_accsg --LINK--> das_asn
27     1.00 (04-05) (02-03)       SEM: Buch_accsg --THEME--> gibt_ditr_sg3
28     1.00 (04-05) (02-03)       SYN: Buch_accsg --DOBJ--> gibt_ditr_sg3
29     0.79 (02-03)              IOBJ: gibt_ditr_sg3 --LINK--> NONSPEC
30     1.00 (05-06)              IOBJ: der_nsm --LINK--> NIL
31     1.00 (05-06)              DOBJ: der_nsm --LINK--> NIL
32     1.00 (05-06)              SUBJ: der_nsm --LINK--> NIL
33     1.00 (05-06)               DET: der_nsm --LINK--> NIL
34     1.00 (05-06)               SEM: der_nsm --LINK--> NIL
35     1.00 (05-06)               SYN: der_nsm --DET--> NONSPEC

INFO: #unary best: 34/36 34 1 0 0 1 0 0 0 0 0

       pref_case_acc(9.990e-01): 21
       obj_distance2(9.167e-01): 22
       pref_case_acc(9.990e-01): 28
       obj_distance3(7.857e-01): 29
--------------------------------------------------

INFO: net: id net0, wordgraph attr2
      #nodes 51, #edges 0
      #evaluations: 24486 unary, 0 statistics, 12535 binary
      #values: min 1, max 22, total 284, average  5.57
      cache: size 3467 #values 3446, #hits 318, 0.1 on average, usage 99.39%

################################################################################
```

```
INFO:    reading word 'Frau' ...
INFO:    solution with better score 7.159e-01 found
WARNING: possible end of scentence!!!
INFO:    0 nodes in workingset
INFO:    agenda size: 1/336/1000, 1 solution with score 7.159e-01:
INFO:    #evaluations this loop
         12582 unary, 0 statistics, 13011 binary

INFO: net: net0, wordgraph: attr2
INFO: 1 solution(s) with score 7.159e-01:
-------------------------------------------------
   +--- modifier `-' lexical entry, `*' word form
   |+-- label
   ||+- modifiee `-' lexical entry, `*' word form
   |||
00     1.00 (00-01)               IOBJ: der_nsm --LINK--> NIL
01     1.00 (00-01)               DOBJ: der_nsm --LINK--> NIL
02     1.00 (00-01)               SUBJ: der_nsm --LINK--> NIL
03     1.00 (00-01)                DET: der_nsm --LINK--> NIL
04     1.00 (00-01)                SEM: der_nsm --LINK--> NIL
05     1.00 (00-01) (01-02)        SYN: der_nsm --DET--> Mann_nomsg
06     1.00 (01-02)               IOBJ: Mann_nomsg --LINK--> NIL
07     1.00 (01-02)               DOBJ: Mann_nomsg --LINK--> NIL
08     1.00 (01-02)               SUBJ: Mann_nomsg --LINK--> NIL
09     1.00 (01-02) (00-01)        DET: Mann_nomsg --LINK--> der_nsm
10     1.00 (01-02) (02-03)        SEM: Mann_nomsg --AGENS--> gibt_ditr_sg3
11     1.00 (01-02) (02-03)        SYN: Mann_nomsg --SUBJ--> gibt_ditr_sg3
12     1.00 (02-03) (01-02)       SUBJ: gibt_ditr_sg3 --LINK--> Mann_nomsg
13     1.00 (02-03)                DET: gibt_ditr_sg3 --LINK--> NIL
14     1.00 (02-03)                SEM: gibt_ditr_sg3 --LINK--> NIL
15     1.00 (02-03)                SYN: gibt_ditr_sg3 --V--> NIL
16     1.00 (03-04)               IOBJ: das_asn --LINK--> NIL
17     1.00 (03-04)               DOBJ: das_asn --LINK--> NIL
18     1.00 (03-04)               SUBJ: das_asn --LINK--> NIL
19     1.00 (03-04)                DET: das_asn --LINK--> NIL
20     1.00 (03-04)                SEM: das_asn --LINK--> NIL
21     1.00 (03-04) (04-05)        SYN: das_asn --DET--> Buch_accsg
22     0.92 (02-03) (04-05)       DOBJ: gibt_ditr_sg3 --LINK--> Buch_accsg
23     1.00 (04-05)               IOBJ: Buch_accsg --LINK--> NIL
24     1.00 (04-05)               DOBJ: Buch_accsg --LINK--> NIL
25     1.00 (04-05)               SUBJ: Buch_accsg --LINK--> NIL
26     1.00 (04-05) (03-04)        DET: Buch_accsg --LINK--> das_asn
27     1.00 (04-05) (02-03)        SEM: Buch_accsg --THEME--> gibt_ditr_sg3
28     1.00 (04-05) (02-03)        SYN: Buch_accsg --DOBJ--> gibt_ditr_sg3
29     1.00 (05-06)               IOBJ: der_dsf --LINK--> NIL
30     1.00 (05-06)               DOBJ: der_dsf --LINK--> NIL
31     1.00 (05-06)               SUBJ: der_dsf --LINK--> NIL
32     1.00 (05-06)                DET: der_dsf --LINK--> NIL
33     1.00 (05-06)                SEM: der_dsf --LINK--> NIL
34     0.79 (02-03) (06-07)       IOBJ: gibt_ditr_sg3 --LINK--> Frau_datsg
35     1.00 (05-06) (06-07)        SYN: der_dsf --DET--> Frau_datsg
36     1.00 (06-07)               IOBJ: Frau_datsg --LINK--> NIL
37     1.00 (06-07)               DOBJ: Frau_datsg --LINK--> NIL
38     1.00 (06-07)               SUBJ: Frau_datsg --LINK--> NIL
39     1.00 (06-07) (05-06)        DET: Frau_datsg --LINK--> der_dsf
40     1.00 (06-07) (02-03)        SEM: Frau_datsg --PATIENCE--> gibt_ditr_sg3
41     1.00 (06-07) (02-03)        SYN: Frau_datsg --IOBJ--> gibt_ditr_sg3
```

```
INFO: #unary best: 40/42 40 1 0 1 0 0 0 0 0 0

       pref_case_acc(9.990e-01): 21
       obj_distance2(9.167e-01): 22
       pref_case_acc(9.990e-01): 28
       obj_distance3(7.857e-01): 34
       pref_case_dat(9.980e-01): 35
       pref_case_dat(9.980e-01): 41
--------------------------------------------------

INFO: net: id net0, wordgraph attr2
      #nodes 60, #edges 0
      #evaluations: 37068 unary, 0 statistics, 25546 binary
      #values: min 0, max 1, total 42, average  0.70
      cache: size 6947 #values 5523, #hits 553, 0.1 on average, usage 79.50%

################################################################################
INFO:    reading word 'dem' ...
INFO:    solution with better score 7.145e-01 found
INFO:    another solution found
INFO:    2 nodes in workingset
INFO:    agenda size: 3/336/1000, 2 solutions with score 7.145e-01:
INFO:    #evaluations this loop
         502 unary, 0 statistics, 852 binary

INFO: net: net0, wordgraph: attr2
INFO: 2 solution(s) with score 7.145e-01:
--------------------------------------------------
   +--- modifier '-' lexical entry, '*' word form
   |+-- label
   ||+- modifiee '-' lexical entry, '*' word form
   |||
00     1.00 (00-01)              IOBJ: der_nsm --LINK--> NIL
01     1.00 (00-01)              DOBJ: der_nsm --LINK--> NIL
02     1.00 (00-01)              SUBJ: der_nsm --LINK--> NIL
03     1.00 (00-01)               DET: der_nsm --LINK--> NIL
04     1.00 (00-01)               SEM: der_nsm --LINK--> NIL
05     1.00 (00-01) (01-02)       SYN: der_nsm --DET--> Mann_nomsg
06     1.00 (01-02)              IOBJ: Mann_nomsg --LINK--> NIL
07     1.00 (01-02)              DOBJ: Mann_nomsg --LINK--> NIL
08     1.00 (01-02)              SUBJ: Mann_nomsg --LINK--> NIL
09     1.00 (01-02) (00-01)       DET: Mann_nomsg --LINK--> der_nsm
10     1.00 (01-02) (02-03)       SEM: Mann_nomsg --AGENS--> gibt_ditr_sg3
11     1.00 (01-02) (02-03)       SYN: Mann_nomsg --SUBJ--> gibt_ditr_sg3
12     1.00 (02-03) (01-02)      SUBJ: gibt_ditr_sg3 --LINK--> Mann_nomsg
13     1.00 (02-03)               DET: gibt_ditr_sg3 --LINK--> NIL
14     1.00 (02-03)               SEM: gibt_ditr_sg3 --LINK--> NIL
15     1.00 (02-03)               SYN: gibt_ditr_sg3 --V--> NIL
16     1.00 (03-04)              IOBJ: das_asn --LINK--> NIL
17     1.00 (03-04)              DOBJ: das_asn --LINK--> NIL
18     1.00 (03-04)              SUBJ: das_asn --LINK--> NIL
19     1.00 (03-04)               DET: das_asn --LINK--> NIL
20     1.00 (03-04)               SEM: das_asn --LINK--> NIL
21     1.00 (03-04) (04-05)       SYN: das_asn --DET--> Buch_accsg
22     0.92 (02-03) (04-05)      DOBJ: gibt_ditr_sg3 --LINK--> Buch_accsg
23     1.00 (04-05)              IOBJ: Buch_accsg --LINK--> NIL
```

```
24     1.00 (04-05)              DOBJ: Buch_accsg --LINK--> NIL
25     1.00 (04-05)              SUBJ: Buch_accsg --LINK--> NIL
26     1.00 (04-05) (03-04)       DET: Buch_accsg --LINK--> das_asn
27     1.00 (04-05) (02-03)       SEM: Buch_accsg --THEME--> gibt_ditr_sg3
28     1.00 (04-05) (02-03)       SYN: Buch_accsg --DOBJ--> gibt_ditr_sg3
29     1.00 (05-06)              IOBJ: der_dsf --LINK--> NIL
30     1.00 (05-06)              DOBJ: der_dsf --LINK--> NIL
31     1.00 (05-06)              SUBJ: der_dsf --LINK--> NIL
32     1.00 (05-06)               DET: der_dsf --LINK--> NIL
33     1.00 (05-06)               SEM: der_dsf --LINK--> NIL
34     0.79 (02-03) (06-07)      IOBJ: gibt_ditr_sg3 --LINK--> Frau_datsg
35     1.00 (05-06) (06-07)       SYN: der_dsf --DET--> Frau_datsg
36     1.00 (06-07)              IOBJ: Frau_datsg --LINK--> NIL
37     1.00 (06-07)              DOBJ: Frau_datsg --LINK--> NIL
38     1.00 (06-07)              SUBJ: Frau_datsg --LINK--> NIL
39     1.00 (06-07) (05-06)       DET: Frau_datsg --LINK--> der_dsf
40     1.00 (06-07) (02-03)       SEM: Frau_datsg --PATIENCE--> gibt_ditr_sg3
41     1.00 (06-07) (02-03)       SYN: Frau_datsg --IOBJ--> gibt_ditr_sg3
42     1.00 (07-08)              IOBJ: dem_dsn --LINK--> NIL
43     1.00 (07-08)              DOBJ: dem_dsn --LINK--> NIL
44     1.00 (07-08)              SUBJ: dem_dsn --LINK--> NIL
45     1.00 (07-08)               DET: dem_dsn --LINK--> NIL
46     1.00 (07-08)               SEM: dem_dsn --LINK--> NIL
47     1.00 (07-08)               SYN: dem_dsn --DET--> NONSPEC

INFO: #unary best: 46/48 46 1 0 1 0 0 0 0 0 0

       pref_case_acc(9.990e-01): 21
       obj_distance2(9.167e-01): 22
       pref_case_acc(9.990e-01): 28
       obj_distance3(7.857e-01): 34
       pref_case_dat(9.980e-01): 35
       pref_case_dat(9.980e-01): 41
       pref_case_dat(9.980e-01): 47
--------------------------------------------------
   +--- modifier '-' lexical entry, '*' word form
   |+-- label
   ||+- modifiee '-' lexical entry, '*' word form
   |||
00     1.00 (00-01)              IOBJ: der_nsm --LINK--> NIL
01     1.00 (00-01)              DOBJ: der_nsm --LINK--> NIL
02     1.00 (00-01)              SUBJ: der_nsm --LINK--> NIL
03     1.00 (00-01)               DET: der_nsm --LINK--> NIL
04     1.00 (00-01)               SEM: der_nsm --LINK--> NIL
05     1.00 (00-01) (01-02)       SYN: der_nsm --DET--> Mann_nomsg
06     1.00 (01-02)              IOBJ: Mann_nomsg --LINK--> NIL
07     1.00 (01-02)              DOBJ: Mann_nomsg --LINK--> NIL
08     1.00 (01-02)              SUBJ: Mann_nomsg --LINK--> NIL
09     1.00 (01-02) (00-01)       DET: Mann_nomsg --LINK--> der_nsm
10     1.00 (01-02) (02-03)       SEM: Mann_nomsg --AGENS--> gibt_ditr_sg3
11     1.00 (01-02) (02-03)       SYN: Mann_nomsg --SUBJ--> gibt_ditr_sg3
12     1.00 (02-03) (01-02)      SUBJ: gibt_ditr_sg3 --LINK--> Mann_nomsg
13     1.00 (02-03)               DET: gibt_ditr_sg3 --LINK--> NIL
14     1.00 (02-03)               SEM: gibt_ditr_sg3 --LINK--> NIL
15     1.00 (02-03)               SYN: gibt_ditr_sg3 --V--> NIL
16     1.00 (03-04)              IOBJ: das_asn --LINK--> NIL
17     1.00 (03-04)              DOBJ: das_asn --LINK--> NIL
```

```
18     1.00 (03-04)              SUBJ: das_asn --LINK--> NIL
19     1.00 (03-04)               DET: das_asn --LINK--> NIL
20     1.00 (03-04)               SEM: das_asn --LINK--> NIL
21     1.00 (03-04) (04-05)       SYN: das_asn --DET--> Buch_accsg
22     0.92 (02-03) (04-05)      DOBJ: gibt_ditr_sg3 --LINK--> Buch_accsg
23     1.00 (04-05)              IOBJ: Buch_accsg --LINK--> NIL
24     1.00 (04-05)              DOBJ: Buch_accsg --LINK--> NIL
25     1.00 (04-05)              SUBJ: Buch_accsg --LINK--> NIL
26     1.00 (04-05) (03-04)       DET: Buch_accsg --LINK--> das_asn
27     1.00 (04-05) (02-03)       SEM: Buch_accsg --THEME--> gibt_ditr_sg3
28     1.00 (04-05) (02-03)       SYN: Buch_accsg --DOBJ--> gibt_ditr_sg3
29     1.00 (05-06)              IOBJ: der_dsf --LINK--> NIL
30     1.00 (05-06)              DOBJ: der_dsf --LINK--> NIL
31     1.00 (05-06)              SUBJ: der_dsf --LINK--> NIL
32     1.00 (05-06)               DET: der_dsf --LINK--> NIL
33     1.00 (05-06)               SEM: der_dsf --LINK--> NIL
34     0.79 (02-03) (06-07)      IOBJ: gibt_ditr_sg3 --LINK--> Frau_datsg
35     1.00 (05-06) (06-07)       SYN: der_dsf --DET--> Frau_datsg
36     1.00 (06-07)              IOBJ: Frau_datsg --LINK--> NIL
37     1.00 (06-07)              DOBJ: Frau_datsg --LINK--> NIL
38     1.00 (06-07)              SUBJ: Frau_datsg --LINK--> NIL
39     1.00 (06-07) (05-06)       DET: Frau_datsg --LINK--> der_dsf
40     1.00 (06-07) (02-03)       SEM: Frau_datsg --PATIENCE--> gibt_ditr_sg3
41     1.00 (06-07) (02-03)       SYN: Frau_datsg --IOBJ--> gibt_ditr_sg3
42     1.00 (07-08)              IOBJ: dem_dsm --LINK--> NIL
43     1.00 (07-08)              DOBJ: dem_dsm --LINK--> NIL
44     1.00 (07-08)              SUBJ: dem_dsm --LINK--> NIL
45     1.00 (07-08)               DET: dem_dsm --LINK--> NIL
46     1.00 (07-08)               SEM: dem_dsm --LINK--> NIL
47 -   1.00 (07-08)               SYN: dem_dsm --DET--> NONSPEC

INFO: #unary best: 46/48 46 1 0 1 0 0 0 0 0 0

       pref_case_acc(9.990e-01): 21
       obj_distance2(9.167e-01): 22
       pref_case_acc(9.990e-01): 28
       obj_distance3(7.857e-01): 34
       pref_case_dat(9.980e-01): 35
       pref_case_dat(9.980e-01): 41
       pref_case_dat(9.980e-01): 47
--------------------------------------------------

INFO: net: id net0, wordgraph attr2
      #nodes 67, #edges 0
      #evaluations: 37570 unary, 0 statistics, 26398 binary
      #values: min 0, max 2, total 50, average  0.75
      cache: size 6947 #values 5883, #hits 553, 0.1 on average, usage 84.68%

################################################################################
INFO:    reading word 'Kind' ...
INFO:    solution with better score 7.124e-01 found
INFO:    another solution found
INFO:    another solution found
INFO:    another solution found
INFO:    5 nodes in workingset
INFO:    agenda size: 18/336/1000, 4 solutions with score 7.124e-01:
INFO:    #evaluations this loop
```

```
         1907 unary, 0 statistics, 6446 binary

INFO: net: net0, wordgraph: attr2
INFO: 4 solution(s) with score 7.124e-01:
------------------------------------------------
   +--- modifier ‘-’ lexical entry, ‘*’ word form
   |+-- label
   ||+- modifiee ‘-’ lexical entry, ‘*’ word form
   |||
00      1.00 (00-01)              IOBJ: der_nsm --LINK--> NIL
01      1.00 (00-01)              DOBJ: der_nsm --LINK--> NIL
02      1.00 (00-01)              SUBJ: der_nsm --LINK--> NIL
03      1.00 (00-01)               DET: der_nsm --LINK--> NIL
04      1.00 (00-01)               SEM: der_nsm --LINK--> NIL
05      1.00 (00-01) (01-02)       SYN: der_nsm --DET--> Mann_nomsg
06      1.00 (01-02)              IOBJ: Mann_nomsg --LINK--> NIL
07      1.00 (01-02)              DOBJ: Mann_nomsg --LINK--> NIL
08      1.00 (01-02)              SUBJ: Mann_nomsg --LINK--> NIL
09      1.00 (01-02) (00-01)       DET: Mann_nomsg --LINK--> der_nsm
10      1.00 (01-02) (02-03)       SEM: Mann_nomsg --AGENS--> gibt_ditr_sg3
11      1.00 (01-02) (02-03)       SYN: Mann_nomsg --SUBJ--> gibt_ditr_sg3
12      1.00 (02-03) (01-02)      SUBJ: gibt_ditr_sg3 --LINK--> Mann_nomsg
13      1.00 (02-03)               DET: gibt_ditr_sg3 --LINK--> NIL
14      1.00 (02-03)               SEM: gibt_ditr_sg3 --LINK--> NIL
15      1.00 (02-03)               SYN: gibt_ditr_sg3 --V--> NIL
16      1.00 (03-04)              IOBJ: das_asn --LINK--> NIL
17      1.00 (03-04)              DOBJ: das_asn --LINK--> NIL
18      1.00 (03-04)              SUBJ: das_asn --LINK--> NIL
19      1.00 (03-04)               DET: das_asn --LINK--> NIL
20      1.00 (03-04)               SEM: das_asn --LINK--> NIL
21      1.00 (03-04) (04-05)       SYN: das_asn --DET--> Buch_accsg
22      0.92 (02-03) (04-05)      DOBJ: gibt_ditr_sg3 --LINK--> Buch_accsg
23      1.00 (04-05)              IOBJ: Buch_accsg --LINK--> NIL
24      1.00 (04-05)              DOBJ: Buch_accsg --LINK--> NIL
25      1.00 (04-05)              SUBJ: Buch_accsg --LINK--> NIL
26      1.00 (04-05) (03-04)       DET: Buch_accsg --LINK--> das_asn
27      1.00 (04-05) (02-03)       SEM: Buch_accsg --THEME--> gibt_ditr_sg3
28      1.00 (04-05) (02-03)       SYN: Buch_accsg --DOBJ--> gibt_ditr_sg3
29      1.00 (05-06)              IOBJ: der_dsf --LINK--> NIL
30      1.00 (05-06)              DOBJ: der_dsf --LINK--> NIL
31      1.00 (05-06)              SUBJ: der_dsf --LINK--> NIL
32      1.00 (05-06)               DET: der_dsf --LINK--> NIL
33      1.00 (05-06)               SEM: der_dsf --LINK--> NIL
34      0.79 (02-03) (06-07)      IOBJ: gibt_ditr_sg3 --LINK--> Frau_datsg
35      1.00 (05-06) (06-07)       SYN: der_dsf --DET--> Frau_datsg
36      1.00 (06-07)              IOBJ: Frau_datsg --LINK--> NIL
37      1.00 (06-07)              DOBJ: Frau_datsg --LINK--> NIL
38      1.00 (06-07)              SUBJ: Frau_datsg --LINK--> NIL
39      1.00 (06-07) (05-06)       DET: Frau_datsg --LINK--> der_dsf
40      1.00 (06-07) (02-03)       SEM: Frau_datsg --PATIENCE--> gibt_ditr_sg3
41      1.00 (06-07) (02-03)       SYN: Frau_datsg --IOBJ--> gibt_ditr_sg3
42      1.00 (07-08)              IOBJ: dem_dsn --LINK--> NIL
43      1.00 (07-08)              DOBJ: dem_dsn --LINK--> NIL
44      1.00 (07-08)              SUBJ: dem_dsn --LINK--> NIL
45      1.00 (07-08)               DET: dem_dsn --LINK--> NIL
46      1.00 (07-08)               SEM: dem_dsn --LINK--> NIL
47      1.00 (07-08) (08-09)       SYN: dem_dsn --DET--> Kind_datsg
```

```
48     1.00 (08-09)              IOBJ: Kind_datsg --LINK--> NIL
49     1.00 (08-09)              DOBJ: Kind_datsg --LINK--> NIL
50     1.00 (08-09)              SUBJ: Kind_datsg --LINK--> NIL
51     1.00 (08-09) (07-08)       DET: Kind_datsg --LINK--> dem_dsn
52     1.00 (08-09)               SEM: Kind_datsg --THEME--> NONSPEC
53     1.00 (08-09)               SYN: Kind_datsg --IOBJ--> NONSPEC

INFO: #unary best: 52/54 52 1 0 1 0 0 0 0 0 0

       pref_case_acc(9.990e-01): 21
       obj_distance2(9.167e-01): 22
       pref_case_acc(9.990e-01): 28
       obj_distance3(7.857e-01): 34
       pref_case_dat(9.980e-01): 35
       pref_case_dat(9.980e-01): 41
       pref_case_dat(9.980e-01): 47
       pref_case_dat(9.980e-01): 53
        pref_obj_pos(9.990e-01): 53
--------------------------------------------------
   +--- modifier '-' lexical entry, '*' word form
   |+-- label
   ||+- modifiee '-' lexical entry, '*' word form
   |||
00     1.00 (00-01)              IOBJ: der_nsm --LINK--> NIL
01     1.00 (00-01)              DOBJ: der_nsm --LINK--> NIL
02     1.00 (00-01)              SUBJ: der_nsm --LINK--> NIL
03     1.00 (00-01)               DET: der_nsm --LINK--> NIL
04     1.00 (00-01)               SEM: der_nsm --LINK--> NIL
05     1.00 (00-01) (01-02)       SYN: der_nsm --DET--> Mann_nomsg
06     1.00 (01-02)              IOBJ: Mann_nomsg --LINK--> NIL
07     1.00 (01-02)              DOBJ: Mann_nomsg --LINK--> NIL
08     1.00 (01-02)              SUBJ: Mann_nomsg --LINK--> NIL
09     1.00 (01-02) (00-01)       DET: Mann_nomsg --LINK--> der_nsm
10     1.00 (01-02) (02-03)       SEM: Mann_nomsg --AGENS--> gibt_ditr_sg3
11     1.00 (01-02) (02-03)       SYN: Mann_nomsg --SUBJ--> gibt_ditr_sg3
12     1.00 (02-03) (01-02)      SUBJ: gibt_ditr_sg3 --LINK--> Mann_nomsg
13     1.00 (02-03)               DET: gibt_ditr_sg3 --LINK--> NIL
14     1.00 (02-03)               SEM: gibt_ditr_sg3 --LINK--> NIL
15     1.00 (02-03)               SYN: gibt_ditr_sg3 --V--> NIL
16     1.00 (03-04)              IOBJ: das_asn --LINK--> NIL
17     1.00 (03-04)              DOBJ: das_asn --LINK--> NIL
18     1.00 (03-04)              SUBJ: das_asn --LINK--> NIL
19     1.00 (03-04)               DET: das_asn --LINK--> NIL
20     1.00 (03-04)               SEM: das_asn --LINK--> NIL
21     1.00 (03-04) (04-05)       SYN: das_asn --DET--> Buch_accsg
22     0.92 (02-03) (04-05)      DOBJ: gibt_ditr_sg3 --LINK--> Buch_accsg
23     1.00 (04-05)              IOBJ: Buch_accsg --LINK--> NIL
24     1.00 (04-05)              DOBJ: Buch_accsg --LINK--> NIL
25     1.00 (04-05)              SUBJ: Buch_accsg --LINK--> NIL
26     1.00 (04-05) (03-04)       DET: Buch_accsg --LINK--> das_asn
27     1.00 (04-05) (02-03)       SEM: Buch_accsg --THEME--> gibt_ditr_sg3
28     1.00 (04-05) (02-03)       SYN: Buch_accsg --DOBJ--> gibt_ditr_sg3
29     1.00 (05-06)              IOBJ: der_dsf --LINK--> NIL
30     1.00 (05-06)              DOBJ: der_dsf --LINK--> NIL
31     1.00 (05-06)              SUBJ: der_dsf --LINK--> NIL
32     1.00 (05-06)               DET: der_dsf --LINK--> NIL
33     1.00 (05-06)               SEM: der_dsf --LINK--> NIL
```

```
34     0.79 (02-03) (06-07)       IOBJ: gibt_ditr_sg3 --LINK--> Frau_datsg
35     1.00 (05-06) (06-07)        SYN: der_dsf --DET--> Frau_datsg
36     1.00 (06-07)               IOBJ: Frau_datsg --LINK--> NIL
37     1.00 (06-07)               DOBJ: Frau_datsg --LINK--> NIL
38     1.00 (06-07)               SUBJ: Frau_datsg --LINK--> NIL
39     1.00 (06-07) (05-06)        DET: Frau_datsg --LINK--> der_dsf
40     1.00 (06-07) (02-03)        SEM: Frau_datsg --PATIENCE--> gibt_ditr_sg3
41     1.00 (06-07) (02-03)        SYN: Frau_datsg --IOBJ--> gibt_ditr_sg3
42     1.00 (07-08)               IOBJ: dem_dsn --LINK--> NIL
43     1.00 (07-08)               DOBJ: dem_dsn --LINK--> NIL
44     1.00 (07-08)               SUBJ: dem_dsn --LINK--> NIL
45     1.00 (07-08)                DET: dem_dsn --LINK--> NIL
46     1.00 (07-08)                SEM: dem_dsn --LINK--> NIL
47     1.00 (07-08) (08-09)        SYN: dem_dsn --DET--> Kind_datsg
48     1.00 (08-09)               IOBJ: Kind_datsg --LINK--> NIL
49     1.00 (08-09)               DOBJ: Kind_datsg --LINK--> NIL
50     1.00 (08-09)               SUBJ: Kind_datsg --LINK--> NIL
51     1.00 (08-09) (07-08)        DET: Kind_datsg --LINK--> dem_dsn
52     1.00 (08-09)                SEM: Kind_datsg --THEME--> NONSPEC
53  1  1.00 (08-09)                SYN: Kind_datsg --DOBJ--> NONSPEC

INFO: #unary best: 52/54 52 1 0 1 0 0 0 0 0 0

       pref_case_acc(9.990e-01): 21
       obj_distance2(9.167e-01): 22
       pref_case_acc(9.990e-01): 28
       obj_distance3(7.857e-01): 34
       pref_case_dat(9.980e-01): 35
       pref_case_dat(9.980e-01): 41
       pref_case_dat(9.980e-01): 47
       pref_case_dat(9.980e-01): 53
        pref_obj_pos(9.990e-01): 53
--------------------------------------------------
   +--- modifier '-' lexical entry, '*' word form
   |+-- label
   ||+- modifiee '-' lexical entry, '*' word form
   |||
00     1.00 (00-01)               IOBJ: der_nsm --LINK--> NIL
01     1.00 (00-01)               DOBJ: der_nsm --LINK--> NIL
02     1.00 (00-01)               SUBJ: der_nsm --LINK--> NIL
03     1.00 (00-01)                DET: der_nsm --LINK--> NIL
04     1.00 (00-01)                SEM: der_nsm --LINK--> NIL
05     1.00 (00-01) (01-02)        SYN: der_nsm --DET--> Mann_nomsg
06     1.00 (01-02)               IOBJ: Mann_nomsg --LINK--> NIL
07     1.00 (01-02)               DOBJ: Mann_nomsg --LINK--> NIL
08     1.00 (01-02)               SUBJ: Mann_nomsg --LINK--> NIL
09     1.00 (01-02) (00-01)        DET: Mann_nomsg --LINK--> der_nsm
10     1.00 (01-02) (02-03)        SEM: Mann_nomsg --AGENS--> gibt_ditr_sg3
11     1.00 (01-02) (02-03)        SYN: Mann_nomsg --SUBJ--> gibt_ditr_sg3
12     1.00 (02-03) (01-02)       SUBJ: gibt_ditr_sg3 --LINK--> Mann_nomsg
13     1.00 (02-03)                DET: gibt_ditr_sg3 --LINK--> NIL
14     1.00 (02-03)                SEM: gibt_ditr_sg3 --LINK--> NIL
15     1.00 (02-03)                SYN: gibt_ditr_sg3 --V--> NIL
16     1.00 (03-04)               IOBJ: das_asn --LINK--> NIL
17     1.00 (03-04)               DOBJ: das_asn --LINK--> NIL
18     1.00 (03-04)               SUBJ: das_asn --LINK--> NIL
19     1.00 (03-04)                DET: das_asn --LINK--> NIL
```

```
20     1.00 (03-04)               SEM: das_asn --LINK--> NIL
21     1.00 (03-04) (04-05)       SYN: das_asn --DET--> Buch_accsg
22     0.92 (02-03) (04-05)      DOBJ: gibt_ditr_sg3 --LINK--> Buch_accsg
23     1.00 (04-05)              IOBJ: Buch_accsg --LINK--> NIL
24     1.00 (04-05)              DOBJ: Buch_accsg --LINK--> NIL
25     1.00 (04-05)              SUBJ: Buch_accsg --LINK--> NIL
26     1.00 (04-05) (03-04)       DET: Buch_accsg --LINK--> das_asn
27     1.00 (04-05) (02-03)       SEM: Buch_accsg --THEME--> gibt_ditr_sg3
28     1.00 (04-05) (02-03)       SYN: Buch_accsg --DOBJ--> gibt_ditr_sg3
29     1.00 (05-06)              IOBJ: der_dsf --LINK--> NIL
30     1.00 (05-06)              DOBJ: der_dsf --LINK--> NIL
31     1.00 (05-06)              SUBJ: der_dsf --LINK--> NIL
32     1.00 (05-06)               DET: der_dsf --LINK--> NIL
33     1.00 (05-06)               SEM: der_dsf --LINK--> NIL
34     0.79 (02-03) (06-07)      IOBJ: gibt_ditr_sg3 --LINK--> Frau_datsg
35     1.00 (05-06) (06-07)       SYN: der_dsf --DET--> Frau_datsg
36     1.00 (06-07)              IOBJ: Frau_datsg --LINK--> NIL
37     1.00 (06-07)              DOBJ: Frau_datsg --LINK--> NIL
38     1.00 (06-07)              SUBJ: Frau_datsg --LINK--> NIL
39     1.00 (06-07) (05-06)       DET: Frau_datsg --LINK--> der_dsf
40     1.00 (06-07) (02-03)       SEM: Frau_datsg --PATIENCE--> gibt_ditr_sg3
41     1.00 (06-07) (02-03)       SYN: Frau_datsg --IOBJ--> gibt_ditr_sg3
42     1.00 (07-08)              IOBJ: dem_dsn --LINK--> NIL
43     1.00 (07-08)              DOBJ: dem_dsn --LINK--> NIL
44     1.00 (07-08)              SUBJ: dem_dsn --LINK--> NIL
45     1.00 (07-08)               DET: dem_dsn --LINK--> NIL
46     1.00 (07-08)               SEM: dem_dsn --LINK--> NIL
47     1.00 (07-08) (08-09)       SYN: dem_dsn --DET--> Kind_datsg
48     1.00 (08-09)              IOBJ: Kind_datsg --LINK--> NIL
49     1.00 (08-09)              DOBJ: Kind_datsg --LINK--> NIL
50     1.00 (08-09)              SUBJ: Kind_datsg --LINK--> NIL
51     1.00 (08-09) (07-08)       DET: Kind_datsg --LINK--> dem_dsn
52  l  1.00 (08-09)               SEM: Kind_datsg --PATIENCE--> NONSPEC
53  l  1.00 (08-09)               SYN: Kind_datsg --IOBJ--> NONSPEC

INFO: #unary best: 52/54 52 1 0 1 0 0 0 0 0 0

       pref_case_acc(9.990e-01): 21
       obj_distance2(9.167e-01): 22
       pref_case_acc(9.990e-01): 28
       obj_distance3(7.857e-01): 34
       pref_case_dat(9.980e-01): 35
       pref_case_dat(9.980e-01): 41
       pref_case_dat(9.980e-01): 47
       pref_case_dat(9.980e-01): 53
        pref_obj_pos(9.990e-01): 53
--------------------------------------------------
   +--- modifier '-' lexical entry, '*' word form
   |+-- label
   ||+- modifiee '-' lexical entry, '*' word form
   |||
00     1.00 (00-01)              IOBJ: der_nsm --LINK--> NIL
01     1.00 (00-01)              DOBJ: der_nsm --LINK--> NIL
02     1.00 (00-01)              SUBJ: der_nsm --LINK--> NIL
03     1.00 (00-01)               DET: der_nsm --LINK--> NIL
04     1.00 (00-01)               SEM: der_nsm --LINK--> NIL
05     1.00 (00-01) (01-02)       SYN: der_nsm --DET--> Mann_nomsg
```

```
06     1.00 (01-02)              IOBJ: Mann_nomsg --LINK--> NIL
07     1.00 (01-02)              DOBJ: Mann_nomsg --LINK--> NIL
08     1.00 (01-02)              SUBJ: Mann_nomsg --LINK--> NIL
09     1.00 (01-02) (00-01)       DET: Mann_nomsg --LINK--> der_nsm
10     1.00 (01-02) (02-03)       SEM: Mann_nomsg --AGENS--> gibt_ditr_sg3
11     1.00 (01-02) (02-03)       SYN: Mann_nomsg --SUBJ--> gibt_ditr_sg3
12     1.00 (02-03) (01-02)      SUBJ: gibt_ditr_sg3 --LINK--> Mann_nomsg
13     1.00 (02-03)               DET: gibt_ditr_sg3 --LINK--> NIL
14     1.00 (02-03)               SEM: gibt_ditr_sg3 --LINK--> NIL
15     1.00 (02-03)               SYN: gibt_ditr_sg3 --V--> NIL
16     1.00 (03-04)              IOBJ: das_asn --LINK--> NIL
17     1.00 (03-04)              DOBJ: das_asn --LINK--> NIL
18     1.00 (03-04)              SUBJ: das_asn --LINK--> NIL
19     1.00 (03-04)               DET: das_asn --LINK--> NIL
20     1.00 (03-04)               SEM: das_asn --LINK--> NIL
21     1.00 (03-04) (04-05)       SYN: das_asn --DET--> Buch_accsg
22     0.92 (02-03) (04-05)      DOBJ: gibt_ditr_sg3 --LINK--> Buch_accsg
23     1.00 (04-05)              IOBJ: Buch_accsg --LINK--> NIL
24     1.00 (04-05)              DOBJ: Buch_accsg --LINK--> NIL
25     1.00 (04-05)              SUBJ: Buch_accsg --LINK--> NIL
26     1.00 (04-05) (03-04)       DET: Buch_accsg --LINK--> das_asn
27     1.00 (04-05) (02-03)       SEM: Buch_accsg --THEME--> gibt_ditr_sg3
28     1.00 (04-05) (02-03)       SYN: Buch_accsg --DOBJ--> gibt_ditr_sg3
29     1.00 (05-06)              IOBJ: der_dsf --LINK--> NIL
30     1.00 (05-06)              DOBJ: der_dsf --LINK--> NIL
31     1.00 (05-06)              SUBJ: der_dsf --LINK--> NIL
32     1.00 (05-06)               DET: der_dsf --LINK--> NIL
33     1.00 (05-06)               SEM: der_dsf --LINK--> NIL
34     0.79 (02-03) (06-07)      IOBJ: gibt_ditr_sg3 --LINK--> Frau_datsg
35     1.00 (05-06) (06-07)       SYN: der_dsf --DET--> Frau_datsg
36     1.00 (06-07)              IOBJ: Frau_datsg --LINK--> NIL
37     1.00 (06-07)              DOBJ: Frau_datsg --LINK--> NIL
38     1.00 (06-07)              SUBJ: Frau_datsg --LINK--> NIL
39     1.00 (06-07) (05-06)       DET: Frau_datsg --LINK--> der_dsf
40     1.00 (06-07) (02-03)       SEM: Frau_datsg --PATIENCE--> gibt_ditr_sg3
41     1.00 (06-07) (02-03)       SYN: Frau_datsg --IOBJ--> gibt_ditr_sg3
42     1.00 (07-08)              IOBJ: dem_dsn --LINK--> NIL
43     1.00 (07-08)              DOBJ: dem_dsn --LINK--> NIL
44     1.00 (07-08)              SUBJ: dem_dsn --LINK--> NIL
45     1.00 (07-08)               DET: dem_dsn --LINK--> NIL
46     1.00 (07-08)               SEM: dem_dsn --LINK--> NIL
47     1.00 (07-08) (08-09)       SYN: dem_dsn --DET--> Kind_datsg
48     1.00 (08-09)              IOBJ: Kind_datsg --LINK--> NIL
49     1.00 (08-09)              DOBJ: Kind_datsg --LINK--> NIL
50     1.00 (08-09)              SUBJ: Kind_datsg --LINK--> NIL
51     1.00 (08-09) (07-08)       DET: Kind_datsg --LINK--> dem_dsn
52     1.00 (08-09)               SEM: Kind_datsg --PATIENCE--> NONSPEC
53  1  1.00 (08-09)               SYN: Kind_datsg --DOBJ--> NONSPEC

INFO: #unary best: 52/54 52 1 0 1 0 0 0 0 0 0

       pref_case_acc(9.990e-01): 21
       obj_distance2(9.167e-01): 22
       pref_case_acc(9.990e-01): 28
       obj_distance3(7.857e-01): 34
       pref_case_dat(9.980e-01): 35
       pref_case_dat(9.980e-01): 41
```

```
      pref_case_dat(9.980e-01): 47
      pref_case_dat(9.980e-01): 53
       pref_obj_pos(9.990e-01): 53
--------------------------------------------------

INFO: net: id net0, wordgraph attr2
      #nodes 75, #edges 0
      #evaluations: 39477 unary, 0 statistics, 32844 binary
      #values: min 0, max 7, total 81, average  1.08
      cache: size 13901 #values 7129, #hits 763, 0.1 on average, usage 51.28%

################################################################################
INFO:    end of scentence!!!
INFO:    solution with better score 1.286e-07 found
INFO:    #evaluations: 1907 unary, 0 statistics, 9213 binary

INFO: net: net0, wordgraph: attr2
INFO: 1 solution(s) with score 1.286e-07:
--------------------------------------------------
   +--- modifier '-' lexical entry, '*' word form
   |+-- label
   ||+- modifiee '-' lexical entry, '*' word form
   |||
00     1.00 (00-01)              IOBJ: der_nsm --LINK--> NIL
01     1.00 (00-01)              DOBJ: der_nsm --LINK--> NIL
02     1.00 (00-01)              SUBJ: der_nsm --LINK--> NIL
03     1.00 (00-01)               DET: der_nsm --LINK--> NIL
04     1.00 (00-01)               SEM: der_nsm --LINK--> NIL
05     1.00 (00-01) (01-02)       SYN: der_nsm --DET--> Mann_nomsg
06     1.00 (01-02)              IOBJ: Mann_nomsg --LINK--> NIL
07     1.00 (01-02)              DOBJ: Mann_nomsg --LINK--> NIL
08     1.00 (01-02)              SUBJ: Mann_nomsg --LINK--> NIL
09     1.00 (01-02) (00-01)       DET: Mann_nomsg --LINK--> der_nsm
10     1.00 (01-02) (02-03)       SEM: Mann_nomsg --AGENS--> gibt_ditr_sg3
11     1.00 (01-02) (02-03)       SYN: Mann_nomsg --SUBJ--> gibt_ditr_sg3
12     1.00 (02-03) (01-02)      SUBJ: gibt_ditr_sg3 --LINK--> Mann_nomsg
13     1.00 (02-03)               DET: gibt_ditr_sg3 --LINK--> NIL
14     1.00 (02-03)               SEM: gibt_ditr_sg3 --LINK--> NIL
15     1.00 (02-03)               SYN: gibt_ditr_sg3 --V--> NIL
16     1.00 (03-04)              IOBJ: das_asn --LINK--> NIL
17     1.00 (03-04)              DOBJ: das_asn --LINK--> NIL
18     1.00 (03-04)              SUBJ: das_asn --LINK--> NIL
19     1.00 (03-04)               DET: das_asn --LINK--> NIL
20     1.00 (03-04)               SEM: das_asn --LINK--> NIL
21     1.00 (03-04) (04-05)       SYN: das_asn --DET--> Buch_accsg
22     0.92 (02-03) (04-05)      DOBJ: gibt_ditr_sg3 --LINK--> Buch_accsg
23     1.00 (04-05)              IOBJ: Buch_accsg --LINK--> NIL
24     1.00 (04-05)              DOBJ: Buch_accsg --LINK--> NIL
25     1.00 (04-05)              SUBJ: Buch_accsg --LINK--> NIL
26     1.00 (04-05) (03-04)       DET: Buch_accsg --LINK--> das_asn
27     1.00 (04-05) (02-03)       SEM: Buch_accsg --THEME--> gibt_ditr_sg3
28     1.00 (04-05) (02-03)       SYN: Buch_accsg --DOBJ--> gibt_ditr_sg3
29     1.00 (05-06)              IOBJ: der_dsf --LINK--> NIL
30     1.00 (05-06)              DOBJ: der_dsf --LINK--> NIL
31     1.00 (05-06)              SUBJ: der_dsf --LINK--> NIL
32     1.00 (05-06)               DET: der_dsf --LINK--> NIL
33     1.00 (05-06)               SEM: der_dsf --LINK--> NIL
```

```
34     0.79 (02-03) (06-07)       IOBJ: gibt_ditr_sg3 --LINK--> Frau_datsg
35     1.00 (05-06) (06-07)        SYN: der_dsf --DET--> Frau_datsg
36     1.00 (06-07)               IOBJ: Frau_datsg --LINK--> NIL
37     1.00 (06-07)               DOBJ: Frau_datsg --LINK--> NIL
38     1.00 (06-07)               SUBJ: Frau_datsg --LINK--> NIL
39     1.00 (06-07) (05-06)        DET: Frau_datsg --LINK--> der_dsf
40     1.00 (06-07) (02-03)        SEM: Frau_datsg --PATIENCE--> gibt_ditr_sg3
41     1.00 (06-07) (02-03)        SYN: Frau_datsg --IOBJ--> gibt_ditr_sg3
42     1.00 (07-08)               IOBJ: dem_dsn --LINK--> NIL
43     1.00 (07-08)               DOBJ: dem_dsn --LINK--> NIL
44     1.00 (07-08)               SUBJ: dem_dsn --LINK--> NIL
45     1.00 (07-08)                DET: dem_dsn --LINK--> NIL
46     1.00 (07-08)                SEM: dem_dsn --LINK--> NIL
47     0.18 (07-08) (08-09)        SYN: dem_dsn --DET--> Kind_nomsg
48     1.00 (08-09)               IOBJ: Kind_nomsg --LINK--> NIL
49     1.00 (08-09)               DOBJ: Kind_nomsg --LINK--> NIL
50     1.00 (08-09)               SUBJ: Kind_nomsg --LINK--> NIL
51     1.00 (08-09) (07-08)        DET: Kind_nomsg --LINK--> dem_dsn
52     1.00 (08-09)                SEM: Kind_nomsg --LINK--> NIL
53     0.00 (08-09)                SYN: Kind_nomsg --LINK--> NIL

INFO: #unary best: 51/54 51 1 0 1 1 0 0 0 0 0

       pref_case_acc(9.990e-01): 21
       obj_distance2(9.167e-01): 22
       pref_case_acc(9.990e-01): 28
       obj_distance3(7.857e-01): 34
       pref_case_dat(9.980e-01): 35
       pref_case_dat(9.980e-01): 41
       pref_case_dat(9.980e-01): 47
            syn_case(1.800e-01): 47
    syn_link_penalty(1.000e-06): 53
--------------------------------------------------

INFO: net: id net0, wordgraph attr2
      #nodes 75, #edges 0
      #evaluations: 39477 unary, 0 statistics, 35611 binary
      #values: min 0, max 4, total 66, average  0.88
      cache: size 13901 #values 7323, #hits 763, 0.1 on average, usage 52.68%
```

Abbildungsverzeichnis

Literaturverzeichnis

Aho, A.V., John.E. Hopcroft and Jeffrey D.Ullman. *Data structures and algorithms.* Addison–Wesley, 1983.

Alpern, Bowen, Roger Hoover, Barry K. Rosen, Peter F. Sweeney and F. Kenneth Zadeck. Incremental Evaluation of Computational Circuits. In *First Annual ACM–SIAM Synposium on Discret Algorithms*, pages 32–42, San Francisco/California/USA, 1990.

Bessière, Christian. Arc-Consistency in Dynamic Constraint Satisfaction Problems. In *Proceeding of the Ninth National Conference on Artificial Intelligence*, pages 221–226, 1991.

Bessière, Christian. Using CSPs to encode TMSs. Technical report, LIRMM, 1992.

Bitner, J. R. and E. M. Reingold. *Backtrack Programming Techniques.* Comm. ACM, 1975. 18:651–655.

Cohen, P.R. and E.A. Feigenbaum. *The handbook of artificial intelligence*, volume 3. Morgan Kaufmann, 1982.

Dechter, R. Enhancement Schemes for Constraint Processing: Backjumping, Learning, and Cutset Decomposition. *Artificial Intelligence*, 1990. 41:273-312.

Dechter, Rina and Avi Dechter. Belief Maintenance in Dynamic Constraint Networks. In *7th Annual Conference of the American Association of Artificial Intelligence*, pages 37–42, 1988.

Foth, Kilian A. Disjunktive Lexikoninformation im eliminativen Parsing. Studienarbeit, Universität Hamburg, Fachbereich Informatik, 1999.

Foth, Kilian A. and Michael Schulz. Handbuch der libcdg Bibliothek. Technical Report Dawai-HH-98-002, Universität Hamburg, 1999.

Freuder, E.C. Partial constraint–satisfaction. In *11th International Joint Conference on Artificial Intelligence*, pages 278–283, 1989.

Gaschnig, J. A General Backtracking Algorithm that Eliminates Most redundant tests. In *5th International Joint Conference on Artificial Intelligence*, page 457, Cambridge, Mass., 1977.

Gaschnig, J. Experimental Case Studies of Backtrack vs. Waltz-type vs. New Algorithms for Satisficing Assignment Problems. In *Second Canadian Conference on Artificial Intelligence*, pages 268–277, Toronto, Ont., 1978.

Gaschnig, J. *Performance Measurement and Analysis of Certain Search Algorithms.* PhD thesis, Dept. of Computer Science, Carnegie Mellon University, 1979.

Golomb, S. and L. Baumert. *Backtrack programming.* J. ACM, 1965. 12:516-524.

Hall, P.A.V. Branch and Bound and beyond. In *International Joint Conference on Artificial Intelligence*, pages 641–650, 1971.

Han, P.A.V. and C. Lee. Comments on Mohr and Henderson's path consistency Algorithms. *Artificial Intelligence*, 36:125–130, 1988.

Haralick, R. M. and G. L. Elliott. Increasing tree search efficiency for constraint satisfaction problems. *Artificial Intelligence*, 1980. 14:263-313.

Hentenryck, P. Van. *Constraint Satisfaction in Logic Programming.* MIT Press, 1989.

Lawler, E.W. and D.E. Wood. Branch and bound methods: a survey. In *Operations Research*, volume 14, pages 699–719, 1966.

Mackworth, A. K. Consistency in networks of relations. *Artificial Intelligence*, 1977. 8:99-118.

Maruyama, Hiroshi. Structure diambiguation with constraint propagation. In *28th Annual Meeting of the ACL*, pages 31–38, 1990.

Menzel, Wolfgang. Parsing of spoken language under time constraints. In *11th European Conference on Artificial Intelligence*, pages 560–564, Amsterdam, 1994.

Menzel, Wolfgang. Robust parsing of natural language. In *19th German Annual Conference on Artificial Intelligence*, pages 19–34, Berlin, 1995. cmp-lg/9507003.

Menzel, Wolfgang. Constraint satisfaction for robust parsing of natural language. *Theoretical and Experimental Artificial Intelligence*, 1997.

Menzel, Wolfgang and Ingo Schröder. Decision procedures for dependency parsing using graded constraints. In Kahane, Sylvain and Alain Polguère, editors, *Proc. of the Joint Conference COLING/ACL Workshop: Processing of Dependency-based Grammars*, Montréal, Canada, 1998.

Minton, S., M. Johnston, A.B. Philips and P. Laird. Solving large–scale constraint–satisfaction and scheduling problems using a heuristic repair method. In *National Conference on Artificial Intelligence*, pages 17–24, 1990.

Minton, S., M. Johnston, A.B. Philips and P. Laird. Minimizing conflicts: a heuristic repair method for constraint–satisfaction and scheduling problems. In *Artificial Intelligence*, volume 58, pages 215–219, 1992.

Mohr, R. and T.C. Henderson. Arc and path consistency revised. *Artificial Intelligence*, 28:225–233, 1986.

Montanari, U. Networks of constraints: Fundamental properties and applications to picture processing, 1974. Inform. Sci., 7:95-132.

Morris, P. On the density of solutions in equilibrium points for the queens problem. In *National Conference on Artificial Intelligence*, pages 428–433, 1992.

Nadel, B. A. Constraint satisfaction algorithms. *Computational Intelligence*, 1989. 5:188-224.

Nudel, Bernhard. Consitent–labeling problems and their algorithms: Expected complexities and theory-based heuristics. *Artificial Intelligence*, 1983.

Prosser, P. Hybrid algorithms for the constraint satisfaction problem. *Computational Intelligence*, 1993. 9:268-299.

Ramalingam, G. *Bounded Incremental Computation*, volume 1089. Springer-Verlag, 1996.

Ramalingam, G. and Thomas W. Reps. On the computational complexity of incremental algorithms. Technical Report 1033, Computer Science Department, University of Wisconsin, Madison/Wisconsin/USA, 1991.

Reingold, E.M., J. Nievergelt and N. Deo. *Combinatorial algorithms: theory and practice.* Prentice Hall, 1977.

Reps, Thomas W. and T. Turnidge. Program specialization via program slicing. In *Dagstuhl Seminar on Partial Evaluation*, Schloss Dagstuhl, Wadern, Germany, 1996. Lecture Notes in Computer Science, Vol. 1110, O. Danvy, R. Glueck, and P. Thiemann (eds.), Springer-Verlag, New York, NY, 1996, pp. 409-429.

Schröder, Ingo. Anaylse natürlicher sprache durch beschränkungserfüllung. Studienarbeit, Universität Hamburg, Fachbereich Informatik, 1995.

Schröder, Ingo. Integration statistischer Methoden in eliminative Verfahren zur Analyse natürlicher Sprache. Diplomarbeit, Universität Hamburg, Fachbereich Informatik, 1996.

Schröder, Ingo, Kilian A. Foth and Michael Schulz. [X]cdg Benutzerhandbuch. Technical Report Dawai-HH-13, Universität Hamburg, 1999.

Tsang, Edward P.K. *Foundation of Constraint Satisfaction.* Academic Press, Harcourt Brace & Company, London, 1994.

Leeuwen, J.van, K. Mehlhorn and Thomas W. Reps. Incremental computation and dynamic algorithms. In *Dagstuhl Seminar Report 88*, Schloss Dagstuhl, Wadern, Germany, 1994.

Waltz, David. Understanding line drawings of scenes with shadows. In Winston, P. H., editor, *The Psychology of Computer Vision.* McGraw–Hill, New York, 1975.

Wang, C.J. and Edward P.K. Tsang. Solving constraint–satisfaction problems using neural networks. In *IEEE Second International Conference on Artificial Neural Networks*, pages 295–299, 1991.

Wang, C.J. and Edward P.K. Tsang. A sascadable VLSI design for GENET. In *International Workshop on VLSI for Neural Networks and Artificial Intelligence*, Oxford, 1992.

Wirén, Mats. Bounded incremental parsing. In *6th Twente Workshop on Language Technology*, pages 145–156, Enschede/Netherlands, 1993.

Wirén, Mats. *Minimal Change and Bounded Incremental Parsing.* Saarbrücken, 1994. cmp-lg/9410028.