

Merkmalsstrukturen Verarbeitung (Unifikation)

- Das Instanzieren einer neuen MS ist teuer. Es erfordert allocation und copy Operationen auf Seiten der Implementierung.
- Kosten beinhalten Zeit- und Speicherverbrauch.
- Die zeitlichen Kosten können bis zu 67% der gesamten Verarbeitungszeit beinhalten, Unifikation ca. 12% (s. auch [Tomabechi, 91] oder [Godden, 90]).
- Insbesondere im failure Fall ist der Aufwand „umsonst“. Es gibt einen Zusammenhang zwischen failure Wahrscheinlichkeit und Anwendungsparametern wie Grammatikkomplexität etc.
- Probleme und Vorgehen:
 - Evt. allocate und copy nur im Erfolgsfall ausführen durch Vermeidung von early-copying (s. etwa Wroblewskis Algorithmus [Wroblewski, 1987]).
 - Problem des over-copy: Wenn zwei MS kopiert werden, so wird allgemein die neue Struktur weniger Speicher benötigen als beide Ursprungs-MS zusammen. Over-copy ignoriert dies.
 - Verzögerung der Operationen bis zum Zeitpunkt der Resultatermittlung: lazy-evaluation [Godden, 90]
 - Kein allocate und copy sondern Veränderung der Ursprungs-MS (s. [Tomabechi, 91; Jurafsky und Martin, 2000]: destructive oder quasi-destructive unifikation.

138

Merkmalsstrukturen Verarbeitung (Unifikation)

Für destruktive Unifikation muss die MS angepasst werden indem alle MS einen content und einen pointer slot bekommen. Resultat ist eine erweiterte DAG Repräsentation mit folgenden Eigenschaften:

- content kann NULL oder eine einfache MS beinhalten.
- pointer kann NULL oder einen pointer auf eine MS beinhalten.
- Ist der pointer NULL, dann ist der content valide. Wenn nicht muss der pointer verfolgt werden.

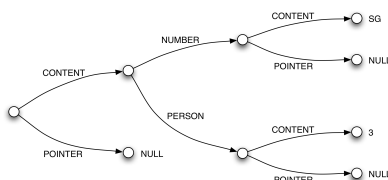
Ursprüngliche Matrix:

$$\begin{bmatrix} \text{number} : & \text{sg} \\ \text{person} : & 3 \end{bmatrix}$$

Matrix mit Anpassung:

$$\begin{bmatrix} \text{content} : & \begin{bmatrix} \text{number} : & \begin{bmatrix} \text{content} : & \text{sg} \\ \text{pointer} : & \text{NULL} \end{bmatrix} \\ \text{person} : & \begin{bmatrix} \text{content} : & 3 \\ \text{pointer} : & \text{NULL} \end{bmatrix} \end{bmatrix} \\ \text{pointer} : & \text{NULL} \end{bmatrix}$$

Erweiterte DAG Repräsentation:



- Erweiterte DAG Repräsentation beinhaltet content und pointer Felder sowohl für top-level features als auch für alle eingebetteten features bis zu den atomaren Werten

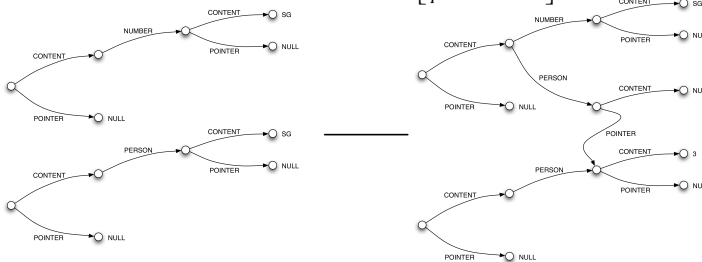
139

Merkmalsstrukturen Verarbeitung (Unifikation)

Beispiel-MS-Unifikation:

$$[number : sg] \sqcup [person : 3]$$

$$= \begin{bmatrix} number : sg \\ person : 3 \end{bmatrix}$$

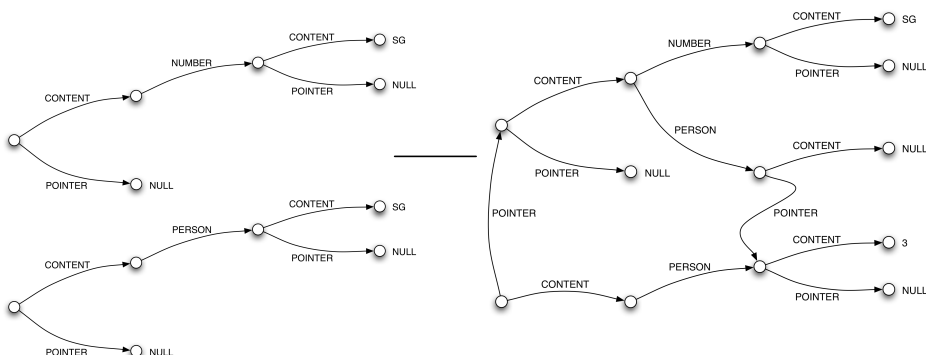


- Auf hoher Ebene resultiert die Unifikation in einer neuen MS.
- In der erweiterten Notation werden aber nur Pointerwerte verändert.
- Ist der Prozess in der erweiterten Version abgeschlossen?
- ✓ *Nein!* Die Unifikation ist hier nur einseitig umgesetzt, es ist keine Identität der beiden ursprünglichen DAG Repräsentationen hergestellt!
- Was muss gemacht werden, um die Unifikation zu vervollständigen?
- ✓ Entweder neues Attribut einführen (etwa number im unteren DAG) oder top-level pointer setzen! (s.n. Folie)

140

Merkmalsstrukturen Verarbeitung (Unifikation)

Unifikationsresultat für das Beispiel:



141

Merkmalsstrukturen Verarbeitung (Unifikation)

Unifikationsalgorithmus:

```

function Unify (f1,f2) returns fstructure or failure
    f1-real <- real content of f1 //dereferencing
    f2-real <- real content of f2
    if f1-real is NULL then
        f1.pointer <- f2
        return f2
    else if f2-real is NULL then
        f2.pointer <- f1
        return f1
    else if f1-real and f2-real are identical then
        f1.pointer <- f2
        return f2
    else if both f1-real and f2-real are complex fs then
        f2.pointer <- f1
        for each feature in f2.real do
            other-feature <- find or create a feature
                corresponding to feature in f1-real
            if Unify(feature.value, other-feature.value) is failure then
                return failure
        return f1
    else return failure
  
```

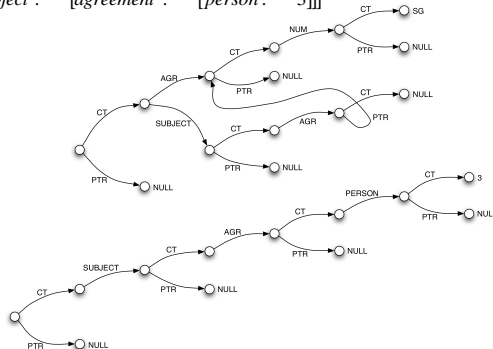
- Test auf die drei Basisfälle für die Rekursion:
 - Eins oder beide Argumente NULL
 - Argumente sind identisch
 - Argumente sind nicht komplex und nicht identisch
- Für identische Strukturen wird vollständige Unifikation durchgeführt (s. Beispiel vorher)
- Wenn Strukturen atomar und inkompatibel sind, wird failure zurückgegeben.
- Sonst wird über alle features in f2 iteriert und rekursiv unifiziert.
- > Kann es im gezeigten Alg. zu Problemen kommen?
 - ❖ Schleifenbildung bei Unifikation von fs1 mit fs2, wenn fs2 teil von fs1 ist.
- > Was ist NULL Wert in der Unifikationssemantik für Typenverbände wenn kein value vorhanden ist?
 - ❖ top

Merkmalsstrukturen Verarbeitung (Unifikation)

Beispiel-MS-Unifikation:

$$\left[\begin{array}{l} \text{agreement} : \langle 1 \rangle \text{number} : \text{sg} \\ \text{subject} \quad \text{agreement} : \langle 1 \rangle \end{array} \right] \sqcup \left[\begin{array}{l} \text{subject} : \text{agreement} : \text{person} : 3 \end{array} \right]$$

- Erweiterte DAG Repräsentation der Unifikationsargumente.
- Beachte die reentrante Struktur für die Koreferenz durch pointer.
- Die Startargumente sind weder identisch noch Null noch atomar, also wird die Hauptschleife aufgerufen.
- Die Iteration über die features von f2 ruft rekursiv den Unifikationsversuch für die korrespondierenden subject-features auf:



$$\left[\text{agreement} : \langle 1 \rangle \right] \sqcup \left[\text{agreement} : \text{person} : 3 \right]$$

Merkmalsstrukturen Verarbeitung (Unifikation)

$[agreement \ (1)] \sqcup [agreement \ [person: \ 3]]$

... sind ebenfalls nicht Null, nicht identisch und nicht atomar, also muss der value unifiziert werden:

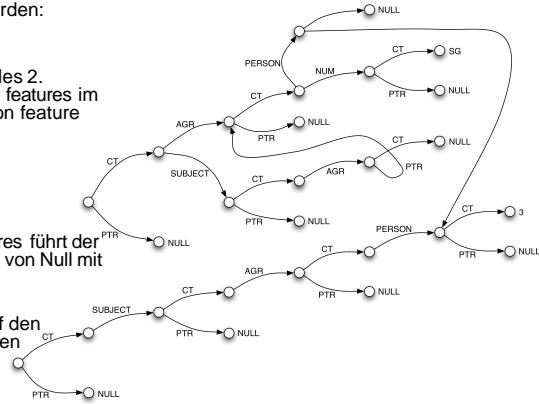
$[number: \ sg] \sqcup [person: \ 3]$

Während der Iteration über die features des 2. Arguments wird das Fehlen eines person features im 1. Argument entdeckt, also wird ein person feature erzeugt und mit Null initialisiert:

$\left[\begin{array}{l} number: \ sg \\ person: \ Null \end{array} \right] \sqcup [person: \ 3]$

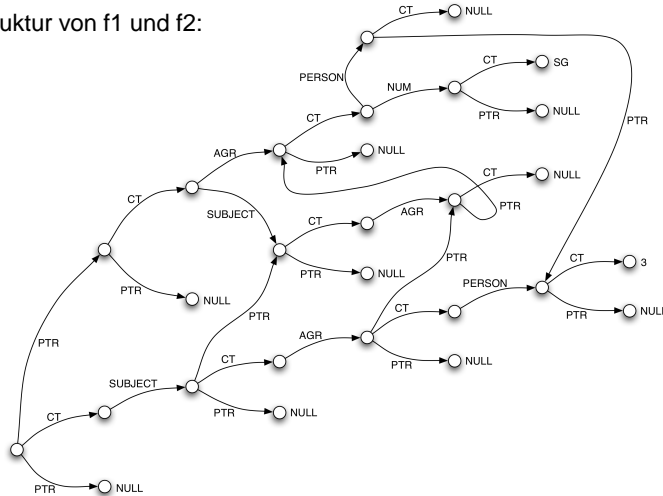
Nach der Instanziierung des neuen features führt der nächste Rekursionsschritt zur Unifikation von Null mit dem Wert 3.

Da keine weiteren features mehr in f2 auf den Rekursionsebenen zu iterieren sind, setzen diese die entsprechenden f2 pointer auf f1 und liefern diese zurück (s. n. Folie).



Merkmalsstrukturen Verarbeitung (Unifikation)

Finale Struktur von f1 und f2:



Merkmalsstrukturen Verarbeitung (Unifikation)

- Wie können Unifikation und MS in einem Parser eingesetzt werden?
- Bei Erhalt einer rudimentären PSG können alle Algorithmen, etwa chart-Parser um Unifikation erweitert werden.
- Unifikation löst aber die Ordnungsstruktur der Parsingalgorithmen auf, sie ist order-independent.
- Unifikation lässt viel mögliche Bearbeitung und Suchstrategien zu.
- ...

Semantik (continued)

- Aufgabe der semantischen Interpretation ist die **Bedeutungsrekonstruktion**.
- Was ist die *Bedeutung* der folgenden natürlich-sprachlichen Eingabekette?

„Er beginnt um zwei im Raum V2-122.“

- **Semantisches Potential:** Information, die sich ausschließlich aus linguistischem Wissen ermitteln lässt.

$$\left(\begin{array}{l} \text{BEGINN}_3(e : \text{event}, t : \text{time}, l : \text{location}) \\ \vee \text{BEGINN}_4(x : \text{male}, e : \text{event}, t : \text{time}, l : \text{location}) \end{array} \right)$$

$$\wedge \text{Uhrzeit}(t) \in \{2, 14\} \wedge \text{Name}(l) = \text{V2-122}$$

- **Aktueller semantischer Wert:** Kontextabhängige Bedeutung einer NL Eingabekette.

„Wann fängt der Vortrag von X an?“

$$\text{BEGINN}_3(e : \text{event}, t : \text{time}, l : \text{location})$$

$$\wedge e = \text{Vortrag von X}$$

$$\wedge t = 14.7.2004, 14\text{Uhr}$$

$$\wedge l = \text{Seminarraum V2-122 der Uni Bielefeld}$$

Semantik

- **Relevante Äußerungsinformation:** Für einen Kommunikationsteilnehmer relevante Kontextinformation im weitesten Sinne.
 - Beispiel: Das Wissen, dass Vorträge an deutschen Unis c.t. beginnen und in der Regel 90 Minuten dauern (-> Auswirkung auf zukünftige Planung).
 - *Relevant* ist relativ für verschiedene Beteiligte
- Aufgaben für die semantische Verarbeitung:
 - Semantikkonstruktion
 - Semantische Resolution
 - Semantische Auswertung
- Die Aufgabenstellungen führen nicht zwingend zu einer entsprechenden Modularisierung oder Sequentialisierung!
- Frage nach einem semantischen Repräsentationsformalismus.
 - Nicht nur als Notation sondern mit der Bereitstellung von Operationen (Bedeutungskomposition, Inferenz, etc.) über Datenstrukturen.
 - Verschiedene Formalismen oder einer?

Semantik

Semantikformalismus

- Neuere Systeme tendieren zu Basisformalismen, z.B. durch Merkmalsstrukturen (angestoßen durch PATRII [Shieber, 1983])
- Beispiel zeigt die Umsetzung der Semantik des NL Ausdrucks als MS und als prädikatenlogische Formel (syn: ist nur beispielhaft).
 - Gibt es Unterschiede in den beiden Repräsentationen?
 - Wo ist die Implikation in der MS?
- MS sind selber wohldefinierte Formalismen mit Deduktionssystem und denotationeller formaler Semantik.
 - Können diese direkt zur Semantikrepräsentation genutzt werden?
 - ✓ Nein, es gibt hier deutliche Grenzen obwohl in diese Richtung hin und wieder argumentiert wird.

„Jeder Student arbeitet.“

$$\left[\begin{array}{l} \text{syn:} \\ \text{sem:} \end{array} \left[\begin{array}{l} \left[\begin{array}{l} \text{cat:} \\ \text{agr:} \end{array} \left[\begin{array}{l} s \\ \left[\begin{array}{l} \text{num:} \\ \text{pers:} \end{array} \left[\begin{array}{l} sg \\ 3 \end{array} \right] \end{array} \right] \end{array} \right] \\ \left[\begin{array}{l} \text{quant:} \\ \text{var:} \\ \text{restr:} \\ \text{scope:} \end{array} \left[\begin{array}{l} \text{'jeder' } \\ \langle 1 \rangle \\ \left[\begin{array}{l} \text{pred:} \\ \text{agr:} \end{array} \left[\begin{array}{l} \text{'student' } \\ \langle 1 \rangle \end{array} \right] \\ \left[\begin{array}{l} \text{pred:} \\ \text{agr:} \end{array} \left[\begin{array}{l} \text{'arbeiten' } \\ \langle 1 \rangle \end{array} \right] \end{array} \right] \end{array} \right] \end{array} \right] \right]$$

$\forall x(\text{student}(x) \rightarrow \text{arbeiten}(x))$

Semantik

- Grammatikformalismen lassen sich für einen Teil der Semantikkonstruktion nutzbar machen.
- Für das Beispiel ergibt sich, dass das Pronomen in der 3. Person ist und das Verb im Plural und durch das grammatische Kongruenzmerkmal das Merkmal 3. Person Plural für das Verb zutrifft.

$$\langle SUBJ \ AGR \ PER \rangle = 3$$

$$\langle VERB \ AGR \ NUM \rangle = pl$$

$$\langle SUBJ \ AGR \rangle = \langle VERB \ AGR \rangle$$

$$\langle VERB \ AGR \rangle = \begin{bmatrix} num : pl \\ pers : 3 \end{bmatrix}$$
- Das Restriktorprädikat <sem restr pred> im Beispiel vorher kann z.B mit dem semantischen Wert des Gattungssubstantivs im Subjekt identifiziert werden. Für das Skopusprädikat und der VP gilt dies ebenso.
- Für eine semantische Auswertung werden aber weitere Operationen erforderlich. Aus dem vorherigen Beispiel und der Information, dass Peter Student ist sollte geschlossen werden, dass Peter arbeitet. Zusätzliche Regeln wie „wer arbeitet hat Erfolg“ erfolgen ähnliche Operationen.
- MS unterstützen diese nicht, wohl aber die PL1:
 - Universelle Instantiierung
 - Kettenschlussregel
- Es besteht ein grundsätzlicher Unterschied zwischen dem **Basisformalismus** und dem **Semantikformalismus**.

150

Semantik Repräsentation

- Bisher eingeführte Beispiele:
 - Theorie der konzeptuellen Dependenz (Conceptual Dependency, CD-Theorie [Schank, 1972].
 - Prädikatenlogik 1. Stufe [n. Frege].
- Denotationelle Semantik der PL erfüllt das so genannte Prinzip der Kompositionalität.
 - Bedeutung des Ganzen (Ausdrucks) ergibt sich aus der syntaktischen Struktur und den Teilbedeutungen der Konstituenten.
- Damit verbunden ist das Substitutionsprinzip:
 - Denotatgleiche Ausdrücke eines komplexen Ausdrucks dürfen ausgetauscht werden

Beispiele:

(Der Kanzler hat die Richtlinienkompetenz) und (Der Kanzler ist Schröder) -> (Schröder hat die Richtlinienkompetenz.)

(Der Kanzler hat immer die Richtlinienkompetenz) und (Der Kanzler ist Schröder) -> (Schröder hat immer die Richtlinienkompetenz.) ???

(Es ist nicht der Fall, dass Tom ununterbrochen arbeitet) und (Tom hat Erfolg g.d.w. er ununterbrochen arbeitet) -> (Es ist nicht der Fall, dass Tom Erfolg hat.)

(Es ist nicht schön, dass Tom ununterbrochen arbeitet) und (Tom hat Erfolg g.d.w. er ununterbrochen arbeitet) -> (Es ist nicht schön, dass Tom Erfolg hat.) ???

151

- Die Bedeutung bestimmter Satzkonstituenten ändert die Interpretation.
 - Bsp.: Temporaladverbien beschreiben den Weltzustand zu einer bestimmten Zeit.
 - Dieses ist i.d.R. nicht der aktuelle Zustand.
 - Dieser wird aber von der benutzten PL Modellstruktur zugrunde gelegt.
 - Ähnliches gilt für die Negation im 2. Beispiel.
- Denotate sind in Bezug auf eine bestimmte **Modellstruktur** (also einen Weltzustand) definiert.
- Man spricht hier von **extensionalen Modellen**. Die Denotate sind die entsprechenden **Extensionen** von Ausdrücken.
- Denotatgleichheit im Sinne von Extensionsgleichheit garantiert nur die Identität in einem bestimmten Weltzustand.
- Temporale Ausdrücke, Einstellungsprädikate (es ist schön, dass, zusammen mit glauben, wissen,...) und Modalausdrücke (notwendig, möglicherweise,...) sind **intensionale Ausdrücke**.
- Intensionale Ausdrücke beziehen sich auf die Intension ihrer Argumentausdrücke und erlauben daher Substitutionsschlüsse nur bei Intensionsgleichheit.
 - Die PL bräuchte also für die Beispielschlüsse eine intensionale Semantik, in der als formale Denotate Intensionen repräsentiert sind.

- Mögliche-Welten-Semantik nach [Kripke, 1971] in der Montague-Grammatik ist ein Versuch, Intension zu formalisieren:
 - Denotate werden als Abbildung von möglichen Weltzuständen in Extensionen spezifiziert.
 - Das Denotat von Bundeskanzler wäre nicht die Person Schröder, sondern eine Abbildung von Weltzuständen in das Universum U, bei der sich der Wert über die Zeit ändert.
 - Satzdenotate sind also Abbildungen von Weltzuständen zu Wahrheitswerten.
 - Wahrheitswertverläufe, Prädikatsdenotate Abbildungen von Weltzuständen in Objektmengen.
- Damit kann das Substitutionsprinzip angewendet werden, es ergeben sich aber dennoch Schwierigkeiten (s. Einstellungsprädikate)
- Ein weiterer Problemtyp für die PL:

Tom ist promovierter Chemiker :	Chemiker(Tom) & promoviert(Tom)
Tom ist leidenschaftlicher Chemiker:	Chemiker(Tom) & leidenschaftlich(Tom) ???
Dieses Auto rostet schnell:	schnell(dieses-Auto) & rostet(dieses -Auto)???
- Die Bedeutung obiger Prädikate ist es andere Prädikate zu modifizieren.
- Dies geht nicht in der PL, dort müssen alle nicht-logischen Ausdrücke entweder als Individuenkonstanten oder n-stellige Relationskonstanten kategorisiert werden.

Typtheorie:

- Einführung zusätzlicher nicht-logischer Ausdrücke unter Erweiterung der Syntax und der denotationellen Semantik. Bestandteile:
 - zwei elementare Typen zum Aufbau von Sätzen/Formeln und der semantischen Komposition:
 - e (entity) als Typ von Eigennamen und referentiellen Ausdrücken, die ein Individuum denotieren.
 - t (truth value) als Typ von Sätzen, die einen Wahrheitswert denotieren (bzw. einen Sachverhalt in der hier nicht näher betrachteten intensionalen Version der Typtheorie).
 - Die Menge der nicht -logischen Konstanten eines Typs s (Con_s).
 - Die Menge Variablen eines Typs s (Var_s).
 - Die aus der PL bekannten Symbole (Junktoren, Quantoren, Identität)
- Es ergibt sich die rekursive Bildungsmenge der komplexen Typen:

Wenn s und t Typen sind (elementar oder komplex), so ist auch $\langle s, t \rangle$ ein Typ.
- Also:

Wenn ein Ausdruck A den Typ $\langle s, t \rangle$ besitzt, bedeutet das:
 A nimmt ein ein s und gibt mit diesem ein t .