

Charts: Subsumption

Übersicht

- ◆ Komplexe Grammatik-Kategorien mit Variablen
 - ◆ sprengen den Rahmen der kontextfreien Sprachen!
 - ◆ bringen deshalb zusätzliche Komplexität ein!
- ◆ Fallunterscheidung
 - ◆ Wo bringen variable Argumente von komplexen Grammatik-Kategorien Probleme?
- ◆ Lösung: Term-Subsumption statt Term-Unifikation
- ◆ Term-Subsumption in Prolog: `subsumes_chk/2`
- ◆ Kanten-Subsumption
 - ◆ Top-Down-WFST-Parser
 - ◆ Earley-Parser

Charts: Subsumption – 1

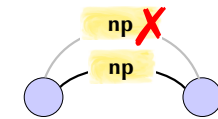
Fall I: Atomare Symbole

Einfügen von Kanten mit atomaren Symbolen

- ◆ Eine Kante mit Kategorie `np` sei in der Chart.
- ◆ Der Parser will eine Kante mit `np` einfügen.
- ▶ Kante wird **nicht** eingefügt, da beide Terme unifizierbar sind.

Das ist richtig so.

- ◆ Es ist bereits eine NP gefunden worden.



Charts: Subsumption – 2

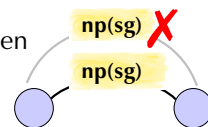
Fall II: Instantiiertes komplexer Term

Einfügen von Kanten mit komplexen Kategorien, aber ohne Variablen

- ◆ Eine Kante mit der komplexen Kategorie `np(singular)` sei in der Chart.
- ◆ Der Parser will eine Kante mit `np(singular)` einfügen.
- ▶ Kante wird **nicht** eingefügt, da beide Terme unifizierbar sind

Das ist richtig so.

- ◆ Es ist ja bereits eine Singular-NP gefunden worden



Charts: Subsumption – 3

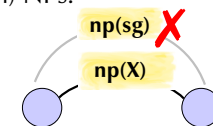
Fall III: Komplexer Term mit Variable

Einfügen von Kanten mit komplexen Kategorien und Variablen

- ◆ Eine Kante mit Kategorie `np(X)` sei in der Chart.
- ◆ Der Parser will eine Kante mit `np(singular)` einfügen.
- ▶ Kante wird **nicht** eingefügt, da beide Terme unifizierbar sind.

Das ist richtig so.

- ◆ Singular-NPs sind ein Spezialfall von (allgemeinen) NPs.
- ◆ Dass eine Singular-NP gefunden wurde, ist nichts Neues!



Charts: Subsumption – 4

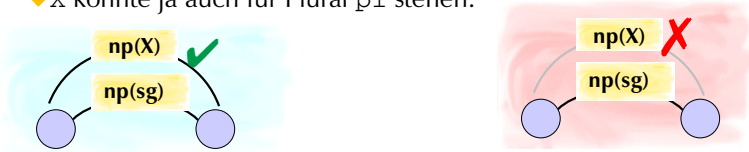
Fall IV: Komplexer Term mit Variable

Einfügen von Kanten mit komplexen Kategorien mit Variablen

- ◆ Eine Kante mit »Kategorie« $np(sg)$ sei in der Chart.
- ◆ Der Parser will eine Kante mit $np(X)$ einfügen.
- ▶ Kante wird **nicht** eingefügt, da beide Terme unifizierbar sind.

Das ist falsch!

- ◆ X könnte ja auch für Plural $p1$ stehen.



Charts: Subsumption – 5

Unifikation und Allgemeinheit

Eine neue Kante gilt als "bereits gefunden", wenn es eine Kante in der Chart gibt,

- ◆ die mit der neuen Kante unifizierbar ist und
- ◆ die **mindestens so allgemein** ist wie die neue Kante.

In Chart	neue Kante	bereits gefunden?
$f(X)$	$f(X)$	ja
$f(X)$	$f(a)$	ja
$f(a)$	$f(X)$	nein

Ein Term A ist mindestens so allgemein wie B , falls gilt:

- ◆ A subsumiert B .

Charts: Subsumption – 6

Term-Subsumption

Ein Term A **subsumiert** einen Term B gdw. gilt

- ◆ A und B sind unifizierbar und
- ◆ in B werden dabei keine Variablen instantiiert
 - ▶ Reines Umbenennen von Variablen ist erlaubt, solange nicht im Term bisher unterschiedliche Variablen dadurch gleich werden ("Einfangen von Variablen").

Mini-Quiz: Wahr oder falsch?

- | | | | |
|----------------------------|-----------|-------------------------------|---------------------------------|
| $f(X, b)$ subsumiert nicht | $f(a, b)$ | <input type="checkbox"/> wahr | <input type="checkbox"/> falsch |
| $f(X, X)$ subsumiert | $f(Y, Y)$ | <input type="checkbox"/> wahr | <input type="checkbox"/> falsch |
| $f(a, b)$ subsumiert nicht | $f(X, b)$ | <input type="checkbox"/> wahr | <input type="checkbox"/> falsch |
| $f(X, X)$ subsumiert | $f(X, Y)$ | <input type="checkbox"/> wahr | <input type="checkbox"/> falsch |
| $f(X, X)$ subsumiert nicht | $f(Y, b)$ | <input type="checkbox"/> wahr | <input type="checkbox"/> falsch |
| $f(X, Y)$ subsumiert | $f(b, b)$ | <input type="checkbox"/> wahr | <input type="checkbox"/> falsch |

Charts: Subsumption – 7

Term-Subsumption in Prolog

Das Prolog-Prädikat `subsumes_chk(A, B)` gelingt, falls Term A den Term B subsumiert.

- ◆ In SICStus Prolog in der Term-Bibliothek vordefiniert.
- ◆ Für beliebige Prologs siehe Covington (1994:174-5)
 - ▶ Wichtig: Es werden keine Variablen gebunden in den Termen!

```
:- use_module(library(terms)).

?- subsumes_chk(f(X), f(X)).
true ?

?- subsumes_chk(f(X), f(a)).
true ?

?- subsumes_chk(f(a), f(X)).
no
```

Charts: Subsumption – 8

WFST und Subsumptions-Check

Im Top-Down WFST-Parser (Folien "Charts")

- ◆ Nur 2. Klausel von complete/2 modifizieren.
- ◆ Nur die Kategorie muss auf Subsumption überprüft werden.
- ▶ Nur auf die Chart zurückgreifen, wenn dort eine vollständige Kategorie vorliegt, die die gesuchte Kategorie subsumiert!

```

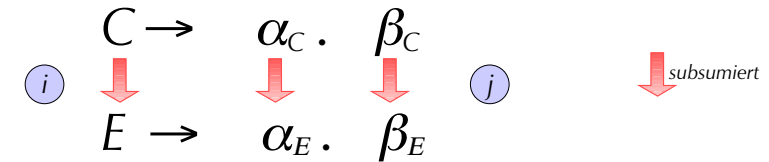
parse(C, S-Rest) :-
  complete(C0, S),
  subsumes_chk(C0, C),      % C0 subsumiert C
  !,
  C0 = C,                  % C0 unifiziert mit C
  edge(C, S, Rest).
  
```

Charts: Subsumption - 9

Aktive Chart und Kanten-Subsumption

Ignoriere neue Kante E , wenn eine andere Kante C bereits in der Chart ist, die folgende Bedingungen erfüllt:

- ◆ Startposition von C = Startposition von E
- ◆ Endposition von C = Endposition von E
- ◆ Die Kategorie von C subsumiert die Kategorie von E .
- ◆ Der Teil vor dem Punkt von C subsumiert den Teil vor dem Punkt von E .
- ◆ Der Teil nach dem Punkt von C subsumiert den Teil nach dem Punkt von E .



Charts: Subsumption - 10

Earley-Parser mit Subsumptions-Check

Im Earley-Parser ändert sich das Prädikat zur Kantenspeicherung.

- ◆ Ein potentieller Charteintrag muss mit allen bestehenden Einträge mit identischer Spannweite abgecheckt werden.
- ◆ Kategorie, Gefundenes und Gesuchtes müssen subsumieren!

```

store(edge(From, To, Cat, Found, ToFind)) :-
  \+ (edge(From, To, ChartCat, ChartFound, ChartToFind),
     subsumes_chk(ChartCat, Cat),
     subsumes_chk(ChartFound, Found),
     subsumes_chk(ChartToFind, ToFind)),
  asserta(edge(From, To, Cat, Found, ToFind)).
  
```

Charts: Subsumption - 11

Ausblick

Effizienz

- ◆ Die vorgestellten Verfahren von Chart-Parsing mit dynamischen Prädikaten für die Chart sind nicht effizient!
- ◆ Subsumptions-Checks sind aufwändig, aber oft unnötig!

Korrektheit

- ◆ Bei rekursiven Grammatiken kann Nicht-Termination entstehen, wenn ein variables Argument der Mutterkategorie in einer Unterstruktur einer Tochterkategorie vorkommt:

$$c(X) \rightarrow \alpha c(f(X)) \beta$$

- ◆ Ausweg beim Earley-Parser: Grammatikspezifische Restriktionen beim Vorhersagen von Kanten (Covington:1994:186ff.)

Charts: Subsumption - 12