

Übungen 9: Earley-Parsing

Programmiertechniken in der Computerlinguistik II · Sommersemester 2005

Programmtexte finden sich auf Homepage: <http://www.cl.unizh.ch/sidemati/lehre/ss05/pcl2/>

1. Aufrufdiagramm

Lade die Datei "earleyp.txt" von der Homepage herunter. Erstelle ein Aufrufdiagramm der Prädikate des Earley-Parsers. Versuche zusätzlich die Aufrufe via Metapredikat `foreach/2` mit gestrichelten Pfeilen einzuzichnen.

2. Mein Leben als Earley-Parser

a) Gegeben sei folgende Mini-Grammatik:

NP → Det N
 NP → Det Adj N
 Adj → leckeren
 Det → den
 N → Kuchen
 RelPron → den

Überlege dir die einzelnen Schritte, die du als Earley-Parser beim Analysieren der Eingabe "den leckeren Kuchen" als NP vornehmen würdest. **Zeichne** eine Chart und/oder mache eine Tabelle, in der du alle Kanten einfügst. **Versehe** alle Kanten mit einer fortlaufenden Nummer wie in den Folien, damit die Reihenfolge des Einfügens ersichtlich wird.

b) Ordne jede Kante (anhand der Kantenummer) der Komponente zu, die sie einfügt.

- Initialisierung:
- Predictor:
- Scanner:
- Completer:

Was fällt dir auf dabei?

c) Lade die Datei "verbose_earleyp.txt" von der Homepage herunter. Sie enthält das gleiche Earley-Programm, allerdings schreibt es für jede Verarbeitungsstufe Informationen auf den Bildschirm. Gib die obige Mini-Grammatik ein und überprüfe deine Resultate von b-c) durch die Anfrage

?- parse(mp, [den, Leckeren, kuchen]).

Hinweis: Im Unterschied zum Shift-Reduce-Parser und Left-Corner-Parser sind die beiden Argumente in word/2 vertauscht!

3. Tilgungsregeln

So wie wir den Earley-Parser kennen, kann er nicht mit Tilgungsregeln umgehen. Erweitere das Prädikat `predictor/1` in "verbose_earleyp.txt" zu folgendem Kode:

```

predictor(To) :-
    foreach(
        rule(LHS, []),
        ( store(edge(To, To, LHS, [], [])),
          complete(To, To, LHS) )
    ),
    fail.

predictor(To) :-
    foreach(
        edge(_, To, _, [Active|_]),
        predict(Active, To)
    ).
    
```

a) Formuliere die Predictor-Klausel für Tilgungsregeln als abstrakte Predictor-Regel wie auf Folie 12. Zeichne die graphische Form dieser Regel auf wie auf Folie 13.

b) Füge die Tilgungsregel "Det → ε" in die Grammatik ein und teste, ob jetzt auch die NP "leckeren Kuchen" erkannt wird.

c) Lösche alle bestehenden Regeln und füge nur die Regeln "VP → V Adv", " Adv → ε" und den Lexikoneintrag "V → isst" ein. Überlege dir, was der Parser mit folgender Anfrage anfangen sollte, und überprüfe dann, was effektiv passiert.

?- parse(vp, [isst]).

Durch welche Modifikation im Earley-Parser-Kode kannst du das Problem (einfach) beheben?

4. Once again

Irma L. hat die Definition von `foreach/2` vereinfacht, indem sie den Aufruf von `once/1` durch den Rumpf der Definition von `once/1` ersetzt hat. Wie findest du diese Idee?

Originaldefinition	Irmas Definition
<pre> foreach(B, A) :- call(B), once(A), fail. foreach(_, _). once(Goal) :- call(Goal), !. </pre>	<pre> foreach(B, A) :- call(B), call(A), !, fail. foreach(_, _). </pre>