



Transformationen von Parsing Schematas

Seminar “Formal Grammars”

23.3.2007

Jochen Setz (info@jochensetz.de)

Einleitung

- Wir kennen verschiedene kontextfreie Parsingalgorithmen:
 - Early-Algorithmus
 - Cocke-Younger-Kasami (CYK)
- Diese wurden durch Inferenzsysteme formalisiert.
- Etliche Tricks und Kniffe sind nötig, um eine akzeptable Laufzeit zu erhalten.

Einleitung

- Wir kennen verschiedene kontextfreie Parsingalgorithmen:
 - Early-Algorithmus
 - Cocke-Younger-Kasami (CYK)
- Diese wurden durch Inferenzsysteme formalisiert.
- Etliche Tricks und Kniffe sind nötig, um eine akzeptable Laufzeit zu erhalten.
- Diese Tricks und Kniffe kann man verallgemeinern
⇒ Transformationen.

Gliederung

Folgende Transformationen wollen wir heute betrachten:

- Folding
- Unfolding
- Speculation
- Semantik

Einleitung - Notation

Die bisher bekannten Inferenzregeln der Form:

$$\frac{(Y, I, J) \quad (Z, J, K)}{(X, I, K)} X \rightarrow Y Z$$

Schreiben wir nun so:

$$\begin{aligned} \text{constit}(X, I, K) &: - \text{rewrite}(X, Y, Z), \\ &\quad \text{constit}(Y, I, J), \text{constit}(Z, J, K) \end{aligned}$$

Einleitung - CYK

$constit(X, I, K) : - \text{rewrite}(X, W), \text{word}(W, I, K)$

$constit(X, I, K) : - \text{rewrite}(X, Y, Z),$
 $\text{constit}(Y, I, J), \text{constit}(Z, J, K)$

$goal : - \text{constit}(s, 0, N), \text{length}(N)$

$\text{rewrite}(s, np, vp)$

$\text{rewrite}(np, det, n)$

$\text{rewrite}(np, "Dumbo")$

$\text{rewrite}(vp, "flies")$

$\text{word}("Dumbo", 0, 1)$

$\text{word}("flies", 1, 2)$

$\text{length}(2)$



Folding

Folding

- Folding ist eine einfache, aber sehr effektive Methode, die asymptotische Laufzeit zu senken.
- Sie basiert darauf, unnötiges Iterieren zu verhindern.
- Dafür werden die Prämissen einer Regel, die als einzigste eine freie Variable enthalten, in eine neue Regel ausgelagert.

Folding - Beispiel

- Betrachten wir die teuerste Regel des CKY-Algorithmus:

$$\begin{aligned} \text{constit}(X, I, K) \quad : - \quad & \text{rewrite}(X, \underline{Y}, \underline{Z}), \\ & \text{constit}(\underline{Y}, I, \underline{J}), \text{constit}(\underline{Z}, \underline{J}, K) \end{aligned}$$

Folding - Beispiel

- Betrachten wir die teuerste Regel des CKY-Algorithmus:

$$\begin{aligned} \text{constit}(X, I, K) \quad : - \quad & \text{rewrite}(X, \underline{Y}, \underline{Z}), \\ & \text{constit}(\underline{Y}, I, \underline{J}), \text{constit}(\underline{Z}, \underline{J}, K) \end{aligned}$$

- Eine Möglichkeit des Foldings:

$$\begin{aligned} \text{temp}(X, Y, Z, I, J) \quad : - \quad & \text{rewrite}(X, Y, Z), \text{constit}(Y, I, J) \\ \text{constit}(X, I, K) \quad : - \quad & \text{temp}(X, \underline{Y}, \underline{Z}, I, \underline{J}), \\ & \text{constit}(\underline{Z}, \underline{J}, K) \end{aligned}$$

Folding - Beispiel

- Eine Möglichkeit des Foldings:

$$\begin{aligned} temp(X, Y, Z, I, J) & : - \quad rewrite(X, Y, Z), constit(Y, I, J) \\ constit(X, I, K) & : - \quad temp(X, \underline{Y}, \underline{Z}, I, \underline{J}), \\ & \quad \quad \quad constit(\underline{Z}, \underline{J}, K) \end{aligned}$$

- Besser:

$$\begin{aligned} temp2(X, Z, I, J) & : - \quad rewrite(X, \underline{Y}, Z), constit(\underline{Y}, I, J) \\ constit(X, I, K) & : - \quad temp2(X, \underline{Z}, I, \underline{J}), \\ & \quad \quad \quad constit(\underline{Z}, \underline{J}, K) \end{aligned}$$

Folding - Komplexität

- Teuereste Regel des CKY-Algorithmus:

$$\begin{aligned} \text{constit}(X, I, K) \quad : - \quad & \text{rewrite}(X, \underline{Y}, \underline{Z}), \\ & \text{constit}(\underline{Y}, I, \underline{J}), \text{constit}(\underline{Z}, \underline{J}, K) \end{aligned}$$

Iterieren über: X, Y, Z und $I, J, K \Rightarrow O(N^3 \cdot n^3)$

Folding - Komplexität

- Teuereste Regel des CKY-Algorithmus:

$$\begin{aligned} \text{constit}(X, I, K) \quad : - \quad & \text{rewrite}(X, \underline{Y}, \underline{Z}), \\ & \text{constit}(\underline{Y}, I, \underline{J}), \text{constit}(\underline{Z}, \underline{J}, K) \end{aligned}$$

Iterieren über: X, Y, Z und $I, J, K \Rightarrow O(N^3 \cdot n^3)$

- 2. Folding:

$$\text{temp2}(X, Z, I, J) \quad : - \quad \text{rewrite}(X, \underline{Y}, \underline{Z}), \text{constit}(\underline{Y}, I, \underline{J})$$

Iterieren über X, Y, Z und $I, J \Rightarrow O(N^3 \cdot n^2)$

$$\text{constit}(X, I, K) \quad : - \quad \text{temp2}(X, \underline{Z}, I, \underline{J}), \text{constit}(\underline{Z}, \underline{J}, K)$$

Iterieren über X, Z und $I, J, K \Rightarrow O(N^2 \cdot n^3)$
 $\Rightarrow O(N^3 \cdot n^2 + N^2 \cdot n^3)$

Folding - Beispiel (Fortsetzung)

- Betrachten wir CKY ohne NF-Einschränkung:

$$\begin{aligned} \text{constit}(X, I, L) \quad : - \quad & ((\text{rewrite}(X, \underline{Y1}, \underline{Y2}, \underline{Y3}), \\ & \text{constit}(\underline{Y1}, I, \underline{J})), \\ & \text{constit}(\underline{Y2}, \underline{J}, \underline{K}), \text{constit}(\underline{Y3}, \underline{K}, L)) \end{aligned}$$

Folding - Beispiel (Fortsetzung)

- Betrachten wir CKY ohne NF-Einschränkung:

$$\begin{aligned} \text{constit}(X, I, L) \quad : - \quad & ((\text{rewrite}(X, \underline{Y1}, \underline{Y2}, \underline{Y3}), \\ & \text{constit}(\underline{Y1}, I, \underline{J})), \\ & \text{constit}(\underline{Y2}, \underline{J}, \underline{K}), \text{constit}(\underline{Y3}, \underline{K}, L)) \end{aligned}$$

- Mit Folding:

$$\begin{aligned} \text{temp3}(X, Y2, Y3, I, J) \quad : - \quad & \text{rewrite}(X, \underline{Y1}, \underline{Y2}, \underline{Y3}), \\ & \text{constit}(\underline{Y1}, I, J) \end{aligned}$$

$$\begin{aligned} \text{temp4}(X, Y3, I, K) \quad : - \quad & \text{temp3}(X, \underline{Y2}, \underline{Y3}, I, \underline{J}), \\ & \text{constit}(\underline{Y2}, \underline{J}, K) \end{aligned}$$

$$\begin{aligned} \text{constit}(X, I, L) \quad : - \quad & \text{temp4}(X, \underline{Y3}, I, \underline{K}), \\ & \text{constit}(\underline{Y3}, \underline{K}, L) \end{aligned}$$

Folding - Beispiel (Fortsetzung)

- Analyse des CKY ohne NF-Einschränkung:

$$\begin{aligned} \text{constit}(X, I, L) \quad : - \quad & ((\text{rewrite}(X, \underline{Y1}, \underline{Y2}, \underline{Y3}), \\ & \text{constit}(\underline{Y1}, I, \underline{J})), \\ & \text{constit}(\underline{Y2}, \underline{J}, \underline{K}), \text{constit}(\underline{Y3}, \underline{K}, L)) \end{aligned}$$

Iterieren über: $X, Y1, Y2, Y3$ und $I, J, K, L \Rightarrow$
 $O(N^4 \cdot n^4)$

Folding - Beispiel (Fortsetzung)

● Analyse des Folding:

$temp3(X, Y2, Y3, I, J) : - \text{rewrite}(X, \underline{Y1}, \underline{Y2}, \underline{Y3}),$
 $\text{constit}(\underline{Y1}, I, J)$

Iterieren über: X, Y2, Y3, Y1 und I, J $\Rightarrow O(N^4 \cdot n^2)$

$temp4(X, Y3, I, K) : - \text{temp3}(X, \underline{Y2}, \underline{Y3}, I, \underline{J}),$
 $\text{constit}(\underline{Y2}, \underline{J}, K)$

Iterieren über: X, Y2, Y3 und I, J, K $\Rightarrow O(N^3 \cdot n^3)$

$\text{constit}(X, I, L) : - \text{temp4}(X, \underline{Y3}, I, \underline{K}),$
 $\text{constit}(\underline{Y3}, \underline{K}, L)$

Iterieren über: X, Y3 und I, K, L $\Rightarrow O(N^2 \cdot n^3)$
 $\Rightarrow O(N^4 \cdot n^2 + N^3 \cdot n^3 + N^2 \cdot n^3)$



Unfolding

Unfolding

- Unfolding ist die Umkehrung des Folding.
- Eine Regel wird durch eine Menge von Regeln ersetzt.
- Pro Regel, deren Konklusion mit der Unfold-Prämisse unifiziert, kommt eine spezialisierte Regel hinzu.
- Dabei entstehen spezialisierte Parser.

Unfolding - Beispiel

- Betrachten wir wieder den CKY-Algorithmus:

$$\begin{aligned} \text{constit}(X, I, K) \quad : - \quad & \text{rewrite}(X, \underline{Y}, \underline{Z}), \\ & \text{constit}(\underline{Y}, I, \underline{J}), \text{constit}(\underline{Z}, \underline{J}, K) \end{aligned}$$

$\text{rewrite}(s, np, vp)$

$\text{rewrite}(np, det, n)$

Unfolding - Beispiel

- Betrachten wir wieder den CKY-Algorithmus:

$$\begin{aligned} \text{constit}(X, I, K) &: - \text{rewrite}(X, \underline{Y}, \underline{Z}), \\ &\quad \text{constit}(\underline{Y}, I, \underline{J}), \text{constit}(\underline{Z}, \underline{J}, K) \end{aligned}$$

$\text{rewrite}(s, np, vp)$

$\text{rewrite}(np, det, n)$

- Transformiert durch Unfolding in:

$$\text{constit}(s, I, K) : - \text{constit}(np, I, J), \text{constit}(vp, J, K)$$

$$\text{constit}(np, I, K) : - \text{constit}(det, I, J), \text{constit}(n, J, K)$$

$\text{rewrite}(s, np, vp)$

$\text{rewrite}(np, det, n)$

Unfolding - Komplexität

- Die asymptotische Laufzeit bleibt erhalten oder wird schlechter

Unfolding - Komplexität

- Die asymptotische Laufzeit bleibt erhalten oder wird schlechter
- Allerdings kann die Laufzeit um einen konstanten Faktor reduziert werden.

Unfolding - Komplexität

- Die asymptotische Laufzeit bleibt erhalten oder wird schlechter
- Allerdings kann die Laufzeit um einen konstanten Faktor reduziert werden.
- Im Beispiel: Laufzeit wird um konstanten Faktor schneller, da die verschiedenen *rewrite*-Regeln nicht angeschaut werden.

Unfolding - Komplexität

- Die asymptotische Laufzeit bleibt erhalten oder wird schlechter
- Allerdings kann die Laufzeit um einen konstanten Faktor reduziert werden.
- Im Beispiel: Laufzeit wird um konstanten Faktor schneller, da die verschiedenen *rewrite*-Regeln nicht angeschaut werden.
- Unfolding kann benutzt werden um “schlechtes” Folding rückgängig zu machen, um dann besser zu folden. Damit kann eine echte Laufzeitverbesserung erreicht werden.



Speculation

Speculation

- Durch Speculation kann die Verarbeitung von sehr langen Ketten und Zyklen (beim parsen von Semiringen) verbessert werden.

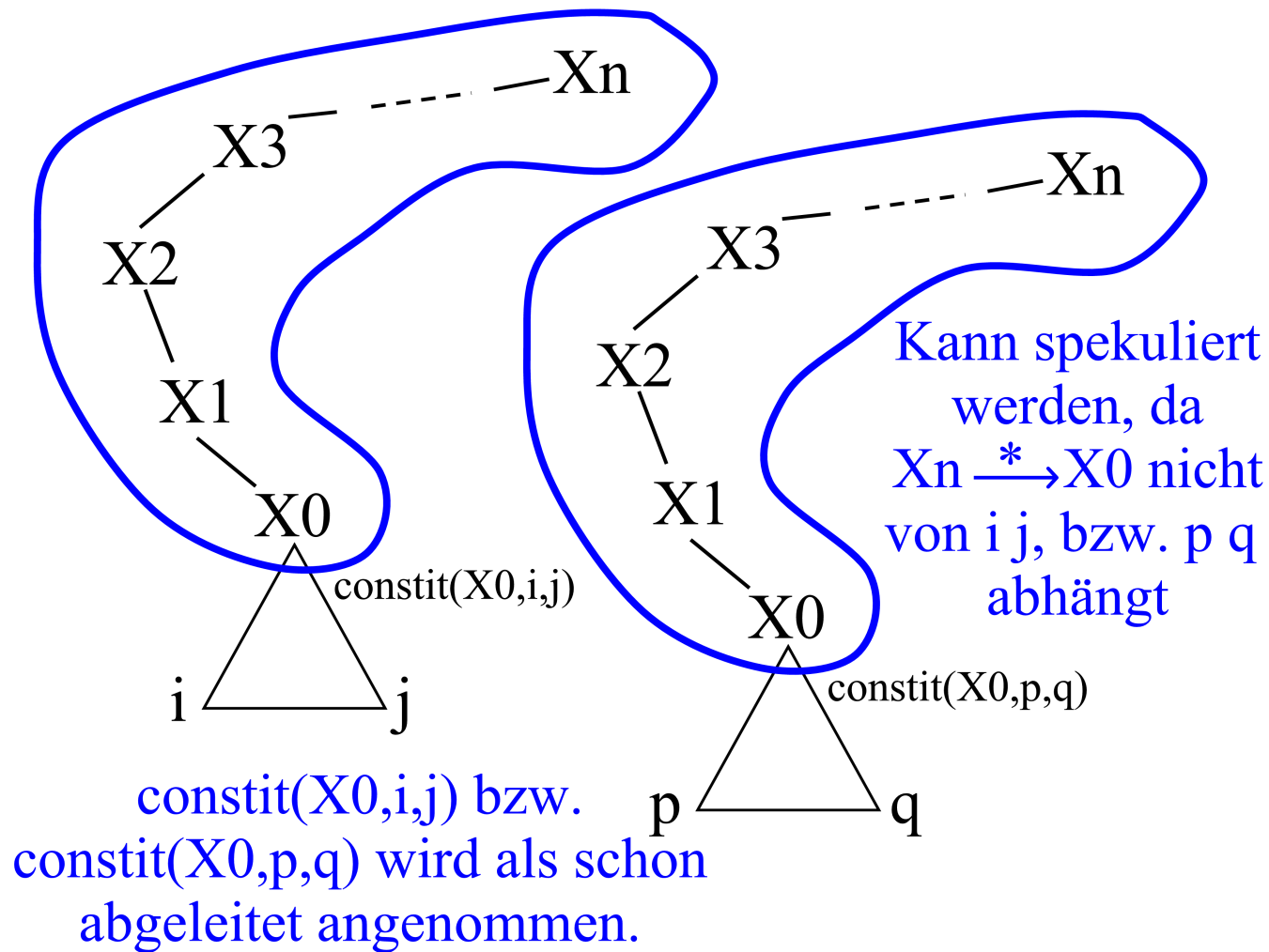
Speculation

- Durch Speculation kann die Verarbeitung von sehr langen Ketten und Zyklen (beim parsen von Semiringen) verbessert werden.
- Die Idee ist, Ableitungsketten, die nicht von konkreter Instanz einer Prämisse abhängig sind, einmalig im Vorfeld zu berechnen.

Speculation

- Durch Speculation kann die Verarbeitung von sehr langen Ketten und Zyklen (beim parsen von Semiringen) verbessert werden.
- Die Idee ist, Ableitungsketten, die nicht von konkreter Instanz einer Prämisse abhängig sind, einmalig im Vorfeld zu berechnen.
- Dabei muss darauf geachtet werden, die spekulierten Teile später nicht doch nochmals zu berechnen.

Speculation - Idee



Speculation - Beispiel

- Erweiterung des CKY-Algorithmus:

$constit(X, I, K) : - \text{rewrite}(X, \underline{W}), \text{word}(\underline{W}, J, K)$

$constit(X, I, K) : - \text{rewrite}(X, \underline{Y}), constit(\underline{Y}, I, K)$

$constit(X, I, K) : - \text{rewrite}(X, \underline{Y}, \underline{Z}),$
 $constit(\underline{Y}, I, \underline{J}), constit(\underline{Z}, \underline{J}, K)$

Speculation - Beispiel

- Erweiterung des CKY-Algorithmus:

$constit(X, I, K) : - \text{rewrite}(X, \underline{W}), \text{word}(\underline{W}, J, K)$

$constit(X, I, K) : - \text{rewrite}(X, \underline{Y}), constit(\underline{Y}, I, K)$

$constit(X, I, K) : - \text{rewrite}(X, \underline{Y}, \underline{Z}),$
 $constit(\underline{Y}, I, \underline{J}), constit(\underline{Z}, \underline{J}, K)$

- Mit Regeln 1 und 2 sind jetzt auch unäre Regeln der Form $X \rightarrow Y$ erlaubt.

Speculation - Beispiel

- Mit Speculation:

$temp(\underline{X0}, \underline{X0})$

$temp(X, X0) \quad : - \quad rewrite(X, \underline{Y}), temp(\underline{Y}, X0)$

$other(constit(X, I, K)) \quad : - \quad rewrite(X, \underline{W}), word(\underline{W}, J, K)$

$other(constit(X, I, K)) \quad : - \quad rewrite(X, \underline{Y}, \underline{Z})$

, $constit(\underline{Y}, I, J), constit(\underline{Z}, \underline{J}, K)$

$constit(X, I, K) \quad : - \quad temp(X, \underline{X0}),$

$other(constit(\underline{X0}, I, K))$

Speculation - Beispiel

- Mit Speculation:

$temp(\underline{X0}, \underline{X0})$

$temp(X, X0) \quad : - \quad rewrite(X, \underline{Y}), temp(\underline{Y}, X0)$

$other(constit(X, I, K)) \quad : - \quad rewrite(X, \underline{W}), word(\underline{W}, J, K)$

$other(constit(X, I, K)) \quad : - \quad rewrite(X, \underline{Y}, \underline{Z})$

$, \quad constit(\underline{Y}, I, J), constit(\underline{Z}, \underline{J}, K)$

$constit(X, I, K) \quad : - \quad temp(X, \underline{X0}),$

$other(constit(\underline{X0}, I, K))$

- Das *other* verhindert, dass das vorher spekulierte nochmal berechnet wird.

Speculation - Analyse

- Speculation kann die Laufzeit enorm reduzieren, abhängig von der Grammatik und des Satzes.

Speculation - Analyse

- Speculation kann die Laufzeit enorm reduzieren, abhängig von der Grammatik und des Satzes.
- Im Worst-Case müsste eine Kette für ein $\text{constit}(X, I, J)$ n^2 -mal berechnet werden.

Speculation - Analyse

- Speculation kann die Laufzeit enorm reduzieren, abhängig von der Grammatik und des Satzes.
- Im Worst-Case müsste eine Kette für ein $\text{constit}(X, I, J)$ n^2 -mal berechnet werden.
- Durch Speculation nur einmal.

Speculation - Analyse

- Speculation kann die Laufzeit enorm reduzieren, abhängig von der Grammatik und des Satzes.
- Im Worst-Case müsste eine Kette für ein $\text{constit}(X, I, J)$ n^2 -mal berechnet werden.
- Durch Speculation nur einmal.
- Beim Semiringparsen kann das wiederholte berechnen von Zyklen verhindert werden, und so die asymptotische Laufzeit verbessert werden.

Speculation - Beispiel (Fortsetzung)

- Bisher sah die Speculation so aus:

$temp(\underline{X0}, \underline{X0})$

$temp(X, X0) : - \text{rewrite}(X, \underline{Y}), temp(\underline{Y}, X0)$

Speculation - Beispiel (Fortsetzung)

- Bisher sah die Speculation so aus:

$$temp(\underline{X0}, \underline{X0})$$
$$temp(X, X0) : - \text{rewrite}(X, \underline{Y}), temp(\underline{Y}, X0)$$

- Mit sogenannten Filter Clauses:

$$temp(\underline{X0}, \underline{X0}) : - \text{needed_only_ifconstit}(X0, I0, \underline{K0})$$
$$temp(X, X0) : - \text{rewrite}(X, \underline{Y}), temp(\underline{Y}, X0)$$
$$\text{needed_only_ifconstit}(X0, I0, \underline{K0})$$

Speculation - Beispiel (Fortsetzung)

- Bisher sah die Speculation so aus:

$$temp(\underline{X0}, \underline{X0})$$
$$temp(X, X0) : - \text{rewrite}(X, \underline{Y}), temp(\underline{Y}, X0)$$

- Mit sogenannten Filter Clauses:

$$temp(\underline{X0}, \underline{X0}) : - \text{needed_only_ifconstit}(X0, I0, \underline{K0})$$
$$temp(X, X0) : - \text{rewrite}(X, \underline{Y}), temp(\underline{Y}, X0)$$
$$\text{needed_only_ifconstit}(X0, I0, \underline{K0})$$

- Dadurch wird eine Kette von $X \xrightarrow{*} Y$ nur berechnet, wenn man schon ein Y abgeleitet hat.

Semantik

- Alle Transformationen erhalten die Semantik.

- Alle Transformationen erhalten die Semantik.
- D.h. alle Items, die vor einer Transformation ableitbar waren, sind danach immernoch ableitbar.

- Alle Transformationen erhalten die Semantik.
- D.h. alle Items, die vor einer Transformation ableitbar waren, sind danach immernoch ableitbar.
- Und, nach der Transformation können nicht mehr Items abgeleitet werden, als vorher...

- Alle Transformationen erhalten die Semantik.
- D.h. alle Items, die vor einer Transformation ableitbar waren, sind danach immernoch ableitbar.
- Und, nach der Transformation können nicht mehr Items abgeleitet werden, als vorher...
- ... ausser den neu hinzugefügten Items!

Zusammenfassung

- Wir haben gesehen, wie Transformationen die Laufzeiten verbessern können.
- Asymptotische Verbesserungen erzielen:
 - Folding
 - Speculation
- Konstante Verbesserungen:
 - Unfolding
- Am Beispiel CKY haben wir gesehen, wie die Transformationen funktionieren, und ohne sie das Parsen viel langsamer wäre.



Verwendete Literatur:

Eisner, Jason and John Blatz (2007). Program transformations for optimization of parsing algorithms and other weighted logic programs. To appear in the Post-proceedings of the 11th Conference on Formal Grammar (FG-2006). CSLI Publications.



Verwendete Literatur:

Eisner, Jason and John Blatz (2007). Program transformations for optimization of parsing algorithms and other weighted logic programs. To appear in the Post-proceedings of the 11th Conference on Formal Grammar (FG-2006). CSLI Publications.

Danke