Aufgabe 5) Geben Sie an, wie bei einer probabilistischen kontextfreien Grammatik folgende Werte definiert sind:

- die Wahrscheinlichkeit eines Parsebaumes
- die Wahrscheinlichkeit eines Satzes (= Menge von Parsebäumen)
- die Wahrscheinlichkeit eines Korpus (= Folge von Sätzen) (2 Punkte)

Aufgabe 5) Geben Sie an, wie bei einer probabilistischen kontextfreien Grammatik folgende Werte definiert sind:

- die Wahrscheinlichkeit eines Parsebaumes
- die Wahrscheinlichkeit eines Satzes (= Menge von Parsebäumen)
- die Wahrscheinlichkeit eines Korpus (= Folge von Sätzen) (2 Punkte)

Korpuswahrscheinlichkeit $p(C) = \prod_{s \in C} p(s)$

Satzwahrscheinlichkeit $p(s) = \sum_{t \in T(s)} p(t)$

Parsewahrscheinlichkeit $p(t) = p(r_1, ..., r_n) = \prod_{i=1}^{n} p(r_i)$

$T(s)$ sind die Analysen des Satzes s.

$r_1, \ldots, r_n$ sind die Regeln der Linksableitung von t.

PCFG-Zusammenfassung

- PCFG = CFG + Regelwahrscheinlichkeiten
- Die Wahrscheinlichkeiten aller Regeln mit derselben linken Seite summieren zu 1, d.h.

$$\sum_{\alpha: A \to \alpha \in P} p(A \to \alpha) = 1 \text{ für alle } A \in V$$

- Parsebaum-Wahrscheinlichkeit = Produkt der Regelwahrscheinlichkeiten
- syntaktische Desambiguierung durch Auswahl der wahrscheinlichsten Analyse
- Die Wahrscheinlichkeit eines Satzes S definieren wir als Summe der Wahrscheinlichkeiten all seiner Parsebäume.

Aufgabe 8) Wie wird bei PCFGs die Wahrscheinlichkeit eines Parsebaumes, die Wahrscheinlichkeit eines Satzes und die Wahrscheinlichkeit einer Folge von Sätzen (=Korpus) definiert? (3 Punkte)

Korpuswahrscheinlichkeit	$p(C) = \prod_{s \in C} p(s)$
Satzwahrscheinlichkeit	$p(s) = \sum_{t \in T(s)} p(t)$
Parsewahrscheinlichkeit	$p(t) = p(r_1, ..., r_n) = \prod_{i=1}^{n} p(r_i)$

$T(s)$ sind die Analysen des Satzes s.

$r_1, \ldots, r_n$ sind die Regeln der Linksableitung von t.

Aufgabe 6) Wie ist die Wahrscheinlichkeit eines Parsebaumes bei einer probabilistischen kontextfreien Grammatik (PCFG) definiert? (2 Punkte)

Parsewahrscheinlichkeit $p(t) = p(r_1, ..., r_n) = \prod_{i=1}^{n} p(r_i)$

$T(s)$ sind die Analysen des Satzes s.

$r_1, \ldots, r_n$ sind die Regeln der Linksableitung von t.

Aufgabe 7) Erläutern Sie den EM-Algorithmus am Beispiel des unüberwachten Trainings von PCFGs (also Training auf Rohtexten). Welche Daten benötigen Sie? Welche Berechnungsschritte führt der EM-Algorithmus aus? (4 Punkte)

- Training auf automatisch geparsten Texten (von einem Parser erzeugte Parsewälder)

s. 190,172

EM-Algorithmus: wiederholte Ausführung der beiden Schritte

1. E-Schritt
   - Jeder Satz des Trainingskorpus wird geparst und ein Parsewald ausgegeben.
   - Die erwartete Häufigkeit jeder Parsewaldregel wird berechnet.
   - Die erwarteten Häufigkeiten werden für jede CFG-Regel über alle Trainingssätze summiert.
2. M-Schritt
   Die Regel-Wahrscheinlichkeiten werden aus den erwarteten Häufigkeiten neu geschätzt:

$$p(A \to \delta) = \frac{\gamma(A \to \delta)}{\sum_{\delta'} \gamma(A \to \delta')}$$

Aufgabe 9) Wie berechnet der Viterbi-Algorithmus den besten Parse für einen gegebenen Parsewald? Wie werden die Viterbi-Wahrscheinlichkeiten berechnet (mit Formeln)?
(5 Punkte)

Viterbi-Algorithmus

$$
\begin{aligned}
\delta(a) &= 1 \quad \text{für jedes Terminalsymbol a} \\
\delta(A \to X_1 \ldots X_n) &= p(A \to X_1 \ldots X_n) \prod_{i=1}^{n} \delta(X_i) \quad \text{für Parsewald-Regeln} \\
\delta(A) &= \max_{\alpha} \delta(A \to \alpha) \quad \text{für Nichtterminale A} \\
\psi(A) &= \arg\max_{\alpha} \delta(A \to \alpha) \quad \text{beste Analyse von A}
\end{aligned}
$$

note: Nur die Formel reicht nicht. Man soll auch erklären wie der Algorithmus finktioniert

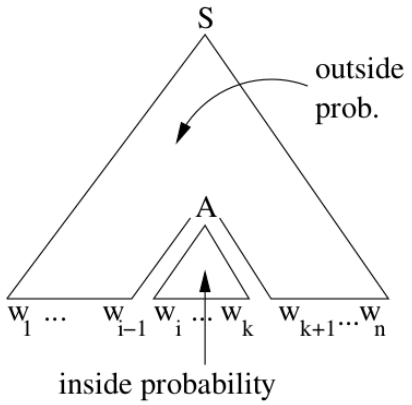
Aufgabe 11) Wofür wird der **Inside-Outside**-Algorithmus benutzt? (1 Punkt)

Der **Inside-Outside-Algorithmus**

- berechnet effizient die **erwarteten Regelhäufigkeiten** beim EM-Training

Aufgabe 10) Wie arbeitet der Inside-Outside-Algorithmus und wie trainiert man damit eine PCFG? (4 Punkte)

Angenommen der Parsewald für den Satz $w_1 ... w_n$ enthält eine Konstituente der Kategorie A, die zu $w_i ... w_k$ expandiert.



Inside-Wahrscheinlichkeit von A: Gesamtwahrscheinlichkeit aller Ableitungen

$$A \Rightarrow ... \Rightarrow w_i ... w_k$$

Outside-Wahrscheinlichkeit von A: Gesamtwahrscheinlichkeit aller Ableitungen

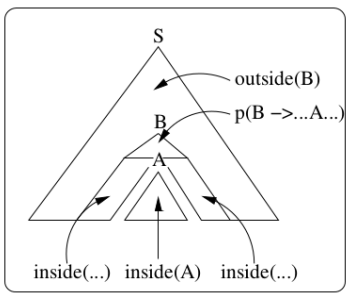
$$S \Rightarrow ... \Rightarrow w_1 ... w_{i-1} A w_{k+1} ... w_n$$

- Der **Inside**-Algorithmus berechnet die Gesamtwahrscheinlichkeit **aller Analysen** für jeden Parsewaldknoten.

Inside-Algorithmus

$$\begin{aligned} \alpha(a) &= 1 \quad \text{für Terminalsymbol a} \\ \alpha(A \rightarrow X_1 ... X_n) &= p(A \rightarrow X_1 ... X_n) \prod_{i=1}^{n} \alpha(X_i) \text{ für Parsewaldregeln} \\ \alpha(A) &= \sum_{A \rightarrow \gamma} \alpha(A \rightarrow \gamma) \quad \text{für Nichtterminale A} \end{aligned}$$

Outside-Algorithmus



$$\begin{aligned} \beta(S) &= 1 \quad \text{für Startsymbol S} \\ \beta(A) &= \sum_{B \rightarrow \gamma A \delta} \beta(B \rightarrow \gamma \underline{A} \delta) \end{aligned}$$

$$\beta(B \rightarrow X_1 ... X_m \underline{A} X_{m+1} ... X_n) = \beta(B) p(B \rightarrow X_1 ... X_m A X_{m+1} ... X_n) \prod_{i=1}^{n} \alpha(X_i)$$

$$\beta(B \rightarrow \gamma \underline{A} \delta) = \beta(B) \frac{\alpha(B \rightarrow \gamma A \delta)}{\alpha(A)} \quad \text{mit } \gamma = X_1 ... X_m \text{ und } \delta = X_{m+1}, , , X_n$$

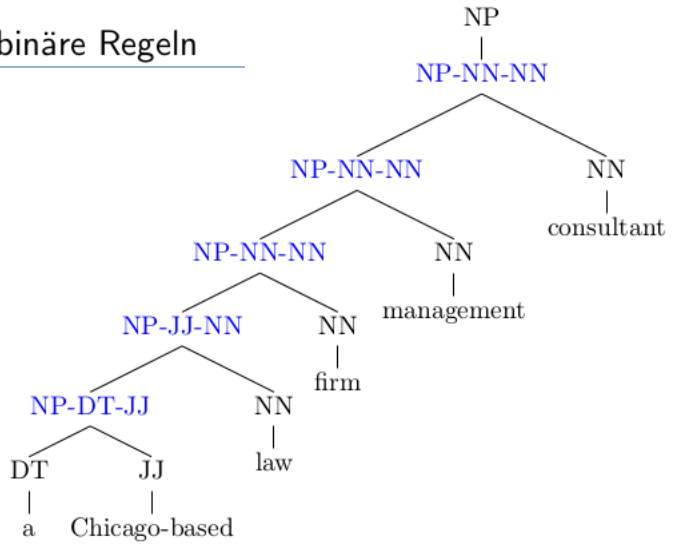
Aufgabe 11) Wie wird eine Grammatik **markowisiert**. Was ist der Vorteil der Markowisierung? (2 Punkte)

Markowisierung

- Aufspaltung von langen Regeln in binäre Regeln
- Neue Hilfskategorien mit z.B.
  - der Elternkategorie und
  - den Kategorien der beiden letzten Tochterknoten
- Das Entfernen der Hilfsknoten liefert wieder den Originalparse

⇒ Reduktion der Grammatikgröße

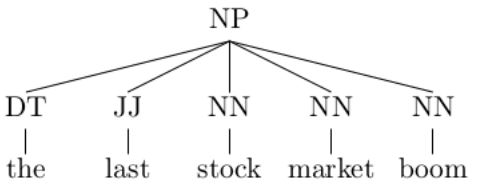
⇒ Verbesserung ihrer Abdeckung



Die gezeigte Markowisierung ist äquivalent dazu, dass wir die rechten Seiten (und Wahrscheinlichkeiten) von bspw. NP-Regeln mit einem Markow-Modell 2. Ordnung erzeugen.

Aufgabe 5) Warum wird beim PCFG-Parsen die Grammatik oft markowisiert? Welches Problem soll dadurch gelöst werden? (3 Punkte)

- Viele Baumbanken verwenden flache Strukturen.



- Die extrahierten Grammatiken enthalten viele Regeln mit langen rechten Seiten, die nur einmal auftauchen.

$$\text{NP} \rightarrow \text{DT JJ NN NN NN}$$

- Andere ähnliche Regeln fehlen, werden aber für das Parsen mancher Sätze benötigt.

$$\text{NP} \rightarrow \text{DT JJ NN NN NN NN}$$

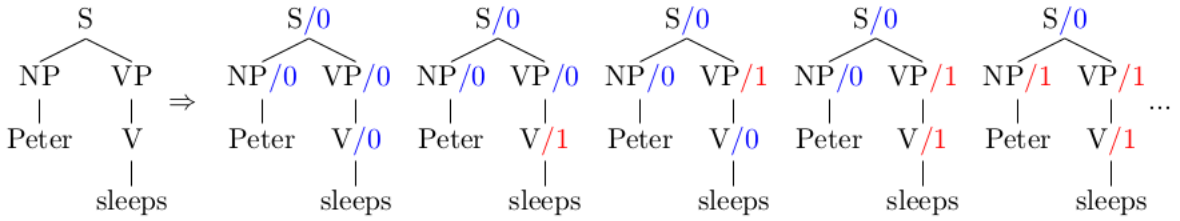
$\Rightarrow$ Reduktion der Grammatikgröße

$\Rightarrow$ Verbesserung ihrer Abdeckung

Aufgabe 9) Erklären Sie die Grundidee der Berkeley-Parsers. Mit welcher Methode wird er trainiert? (2 Punkte)

Grundidee (von Petrov/Klein)

- Alle Kategorien werden durch ein synthetisches Merkmal mit den Werten 0 bzw. 1 aufgespalten.
- Jeder Parse der Baumbank kann von der neuen Grammatik auf viele unterschiedliche Arten generiert werden.
- Durch EM-Training wird die neue Grammatik an die Baumbank angepasst.



Die modifizierte Grammatik liefert für den alten Parsebaum $2^4 = 16$ neue Parsebäume.

s.214-218

Aufgabe 6) Welche Maße werden üblicherweise verwendet bei der Evaluierung von

- Sprachmodellen — Crossentropie oder Perplexität
- Wortart-Taggern
- Parsern — Precision, Recall, F-Score auf Konstituenten

(2 Punkte)

Die Evaluierung eines Wortart-Taggers erfordert ein manuell annotiertes **Testkorpus**, das nicht Teil der Trainingsdaten war.

1. Training des Taggers auf den Trainingsdaten
2. Taggen der Wortfolge des Testkorpus mit dem Tagger
3. Vergleich der Taggerausgabe mit den **Goldstandard**-Tags
4. Genauigkeit = Anzahl korrekte Tags / Anzahl Wörter

s.127

Aufgabe 10) Wie wird üblicherweise die Genauigkeit von (Konstituenten-)Parsern gemessen? Erklären Sie die Methode und wie das Maß berechnet wird. (3 Punkte)

Es werden korrekte Konstituenten statt korrekter Parsebäume gezählt

Eine Konstituente ist korrekt, wenn der Goldstandard-Parse eine Konstituente mit derselben Start- und Endposition und derselben Kategorie enthält.

TP (True Positives): Zahl der ausgegebenen Konstituenten, die korrekt sind

FP (False Positives): Zahl der ausgegebenen Konstituenten, die falsch sind

FN (False Negatives): Zahl der Goldstandard-Konstituenten, die fehlten

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \qquad \text{Recall} = \frac{\text{TP}}{\text{TP+FN}}$$

F-Score: harmonisches Mittel aus Precision und Recall

$$F_1 = \frac{1}{\frac{1}{2}(\frac{1}{P} + \frac{1}{R})} = \frac{2}{\frac{R}{PR} + \frac{P}{PR}} = \frac{2PR}{P+R}$$

Aufgabe 6) Wie wird der F-Score zur Messung der Genauigkeit von Parsern berechnet? (3 Punkte)

F-Score: harmonisches Mittel aus Precision und Recall

$$F_1 = \frac{1}{\frac{1}{2}(\frac{1}{P} + \frac{1}{R})} = \frac{2}{\frac{R}{PR} + \frac{P}{PR}} = \frac{2PR}{P + R}$$

Der gewichtete F-Score gibt Precision β-mal mehr Gewicht als Recall:

$$F_\beta = \frac{(1 + \beta^2)PR}{\beta^2 P + R}$$

Aufgabe 3) Geben Sie die Formeln zur Berechnung des besten Parsebaumes aus einer Menge von Parsebäumen in Parsewaldrepräsentation an. Wie wird initialisiert? Wie wird des beste Parse am Ende ausgegeben?

Sie können davon ausgehen, dass $w_1, ..., w_n$ die Wörter des Eingabesatzes sind, N die Menge der nichtterminalen Knoten im Parsewald und P die Menge der Parsewaldregeln. Die Funktion $p(r)$ liefert die Wahrscheinlichkeit der PCFG-Regel, welche der Parsewaldregel $r \in P$ entspricht. (5 Punkte)

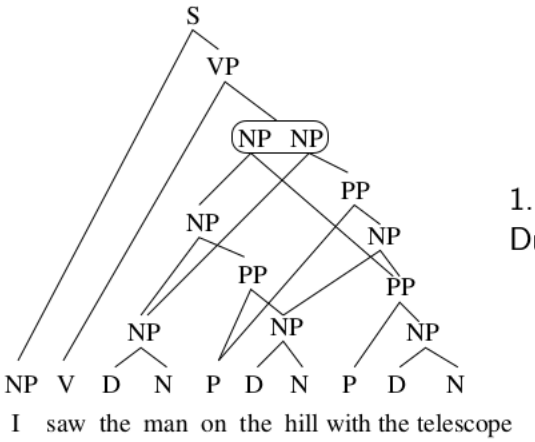
mit Viterbi

Viterbi-Algorithmus

$$\begin{aligned}
\delta(a) &= 1 \quad \text{für jedes Terminalsymbol a} \\
\delta(A \rightarrow X_1 ... X_n) &= p(A \rightarrow X_1 ... X_n) \prod_{i=1}^{n} \delta(X_i) \quad \text{für Parsewald-Regeln} \\
\delta(A) &= \max_{\alpha} \delta(A \rightarrow \alpha) \quad \text{für Nichtterminale A} \\
\psi(A) &= \arg\max_{\alpha} \delta(A \rightarrow \alpha) \quad \text{beste Analyse von A}
\end{aligned}$$

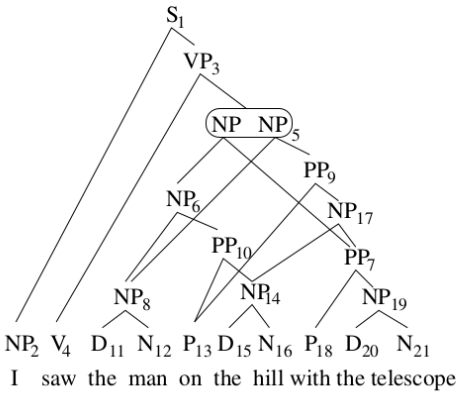
Der beste Parsebaum wird mit Hilfe der Werte der ψ-Variablen berechnet.
Man beginnt mit ψ(S1), der besten Analyse des Wurzelknotens S1.
Dann extrahiert man rekursiv die beste Analyse jedes Tochterknotens A von S1 mit Hilfe von ψ(A) und so weiter.

Grundidee des Viterbi-Algorithmus

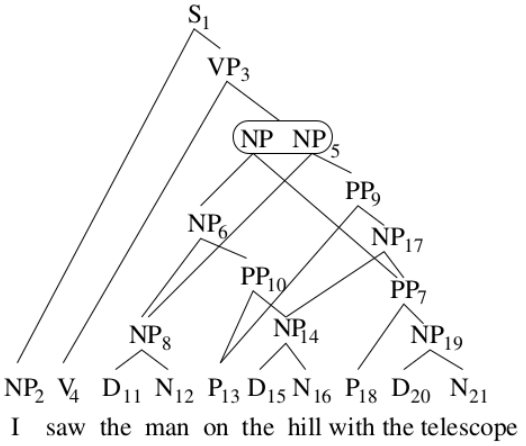


1. Schritt:
Durchnumerierung der Knoten

2. Schritt: Extraktion der Grammatikregeln



$S_1 \rightarrow NP_2\ VP_3$
$NP_2 \rightarrow I$
$VP_3 \rightarrow V_4\ NP_5$
$V_4 \rightarrow saw$
$NP_5 \rightarrow NP_6\ PP_7$
$NP_5 \rightarrow NP_8\ PP_9$
$NP_6 \rightarrow NP_8\ PP_{10}$
$NP_8 \rightarrow D_{11}\ N_{12}$
$D_{11} \rightarrow the$
$N_{12} \rightarrow man$
$PP_{10} \rightarrow P_{13}\ NP_{14}$
$NP_{14} \rightarrow \ldots$



- Die Konstituenten (Knoten) des Parsewaldes werden **bottom-up** durchwandert.
- Bei jeder Konstituente wird die **wahrscheinlichste Analyse** berechnet.
- Am Wurzelknoten wird die wahrscheinlichste **Satzanalyse** ausgegeben.

s.177

Aufgabe 4) Beim unüberwachten Training (Training ohne annotierte Daten) eines HMM-Taggers mit dem EM-Algorithmus wird der Forward-Backward-Algorithmus verwendet.

- Erklären Sie wozu der Forward-Backward-Algorithmus hier dient. Wird er im E-Schritt oder im M-Schritt benutzt?
- Begründen Sie, warum der Viterbi-Algorithmus hier nicht verwendet werden kann.
- Bei welchem Modell wird der Forward-Backward-Algorithmus beim *überwachten* Training eingesetzt? (Es handelt sich dabei nicht um EM-Training.)
- Welcher Algorithmus entspricht dem Forward-Backward-Algorithmus beim unüberwachten Training von PCFGs?
- Welcher Parser setzt diesen Algorithmus beim überwachten Training ein?

- Erklären Sie wozu der Forward-Backward-Algorithmus hier dient. Wird er im E-Schritt oder im M-Schritt benutzt?

EM-Training eines Wortart-Taggers

s.145

1. Uniforme Initialisierung aller $p(t|t')$
2. Uniforme Initialisierung aller $p(w|t)$, die im Lexikon auftauchen
3. Berechnung der erwarteten Wort-Tag- und Tag-Tag-Häufigkeiten mit dem Forward-Backward-Algorithmus (E-Schritt)
4. Neuschätzung der HMM-Wahrscheinlichkeiten aus den erwarteten Häufigkeiten (M-Schritt)
5. weiter mit Schritt 3 bis das Stoppkriterium erfüllt ist

- Begründen Sie, warum der Viterbi-Algorithmus hier nicht verwendet werden kann.

EM-Training

Variante 1: Wir benutzen den Viterbi-Algorithmus zum Taggen.

⇒ Nur die wahrscheinlichste Tagfolge wird berücksichtigt.
Alle anderen Tagfolgen werden ignoriert.

⇒ Am Anfang des Trainings gibt es aber noch keine eindeutige beste Tagfolge.

⇒ Das Training funktioniert deshalb so nicht.

Lösung

- Alle Tagfolgen bei der Extraktion der Taghäufigkeiten berücksichtigen.
- Jede Tagfolge wird dabei mit ihrer Wahrscheinlichkeit gewichtet, so dass doppelt so wahrscheinliche Tagfolgen doppelt so viel zu den extrahierten Häufigkeiten beitragen.

- Bei welchem Modell wird der Forward-Backward-Algorithmus beim *überwachten* Training eingesetzt? (Es handelt sich dabei nicht um EM-Training.)

Training von Conditional Random Fields

- Für das Training mit Gradientenanstieg werden die erwarteten Merkmalshäufigkeiten benötigt.
- Wie bei den HMMs können diese mit dem Forward-Backward-Algorithmus berechnet werden.
- Hier wird der FB-Algorithmus aber für überwachtes Training eingesetzt.

- Welcher Algorithmus entspricht dem Forward-Backward-Algorithmus beim unüberwachten Training von PCFGs?

Inside-Outside-Algorithmus

Der **Inside-Outside-Algorithmus**

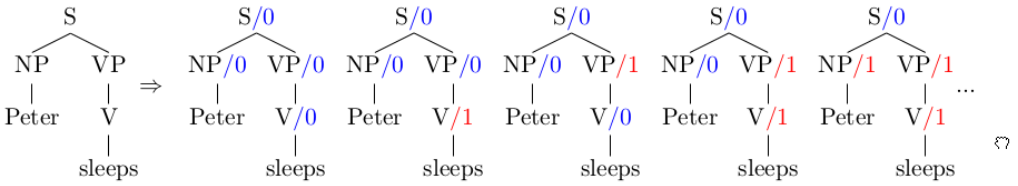
- berechnet effizient die **erwarteten Regelhäufigkeiten** beim EM-Training
- und entspricht damit dem Forward-Backward-Algorithmus bei den HMMs.
- Er berechnet bottom-up **Inside**-Wahrscheinlichkeiten (ähnlich dem Viterbi-Algorithmus)
- und top-down **Outside**-Wahrscheinlichkeiten.
- Aus den Inside- und Outside-Wahrscheinlichkeiten berechnet er die **erwarteten Häufigkeiten**.

- Welcher Algorithmus entspricht dem Forward-Backward-Algorithmus beim unüberwachten Training von PCFGs?
- Welcher Parser setzt diesen Algorithmus beim überwachten Training ein?

Synthetische Merkmale

Grundidee (von Petrov/Klein)

- Alle Kategorien werden durch ein synthetisches Merkmal mit den Werten 0 bzw. 1 aufgespalten.
- Jeder Parse der Baumbank kann von der neuen Grammatik auf viele unterschiedliche Arten generiert werden.
- Durch EM-Training wird die neue Grammatik an die Baumbank angepasst.



"Berkeley"-Parser von Slav Petrov und Dan Klein verwendet EM-Training um die Kategorien(Baumknoten) zu verfeinern.

Aufgabe 10) Wie werden bei einer PCFG die **Regel-Wahrscheinlichkeiten** $p(A \to \alpha)$ (ohne Glättung) aus den Regel-Häufigkeiten $f(A \to \alpha)$ **geschätzt**? (1 Punkt)

$$p(A \to \alpha) = \frac{f(A \to \alpha)}{\sum_{A \to \beta} f(A \to \beta)}$$