Training of Hidden Markov models as an instance of the expectation maximization algorithm

Stefan Majewsky

22. August 2017

Caveat emptor!

Die Definitionen in diesem Vortrag sind aus Zeitgründen stark verkürzt. In der eigentlichen Arbeit wird alles rigoros eingeführt und abgeleitet.

Gliederung

Sprachmodelle

- Bigramm-Modell
- Hidden-Markov-Modell

Gliederung

Sprachmodelle

- Bigramm-Modell
- Hidden-Markov-Modell

EM-Algorithmen

- Baum-Welch-Algorithmus

Gliederung

Sprachmodelle

- Bigramm-Modell
- Hidden-Markov-Modell

EM-Algorithmen

- Baum-Welch-Algorithmus
- Inside-Outside-EM-Algorithmus
- Instanziierung für Hidden-Markov-Modell

Sprachmodell

- Ziel: probabilistische Beschreibung einer Sprache

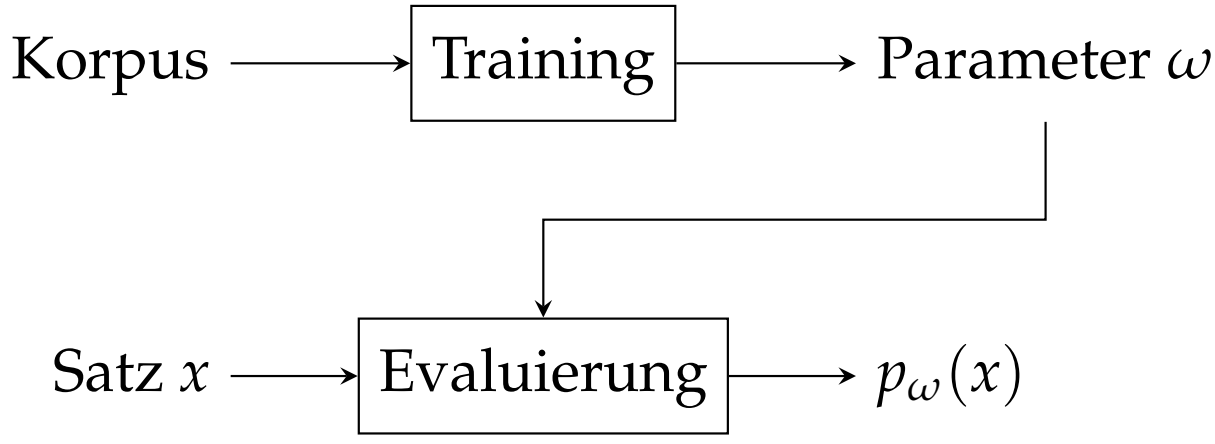
$$\text{Satz } x \mapsto \text{Wahrscheinlichkeit } p(x)$$

Sprachmodell

- Ziel: probabilistische Beschreibung einer Sprache

$$\text{Satz } x \mapsto \text{Wahrscheinlichkeit } p(x)$$

- beschrieben durch *Modellparameter* $\omega \in \Omega$



Grafik nach: H. Vogler. *Maschinelles Übersetzen natürlicher Sprachen.* Vorlesung, TU Dresden, Wintersemester 2015/2016.

Bigramm-Modell

Idee:

- jedes Wort individuell betrachten
- Wahrscheinlichkeit hängt nur vom vorherigen Wort ab

Bigramm-Modell

Idee:

- jedes Wort individuell betrachten
- Wahrscheinlichkeit hängt nur vom vorherigen Wort ab

$$\underbrace{\#}\quad\underbrace{\text{Alice}}\quad\underbrace{\text{sees}}\quad\underbrace{\text{Bob}}\quad\underbrace{\#}$$

$$p_b(x) = b(\text{Alice}|\#) \cdot b(\text{sees}|\text{Alice}) \cdot b(\text{Bob}|\text{sees}) \cdot b(\#|\text{Bob})$$

Bigramm-Modell: Training

- Likelihood eines Korpus...

$$p_b(c) = \prod_x p_b(x)^{c(x)}$$

- ...wird durch empirische Wahrscheinlichkeit maximiert

$$b(w'|w) = \frac{\text{count}(w\ w')}{\sum_{w''} \text{count}(w\ w'')}$$

Bigramm-Modell: Training

- Likelihood eines Korpus...

$$p_b(c) = \prod_x p_b(x)^{c(x)}$$

- ...wird durch empirische Wahrscheinlichkeit maximiert

$$b(w'|w) = \frac{\text{count}(w\ w')}{\sum_{w''} \text{count}(w\ w'')}$$

- z. B. für Korpus: „Alice sees Bob“, „Alice likes cake“

x		
Alice		
sees		
Bob		
likes		
cake		
#		

Bigramm-Modell: Training

- Likelihood eines Korpus...

$$p_b(c) = \prod_x p_b(x)^{c(x)}$$

- ...wird durch empirische Wahrscheinlichkeit maximiert

$$b(w'|w) = \frac{\text{count}(w\ w')}{\sum_{w''} \text{count}(w\ w'')}$$

- z. B. für Korpus: „Alice sees Bob“, „Alice likes cake“

x	count(Alice x)	
Alice	0	
sees	1	
Bob	0	
likes	1	
cake	0	
#	0	

Bigramm-Modell: Training

- Likelihood eines Korpus...

$$p_b(c) = \prod_x p_b(x)^{c(x)}$$

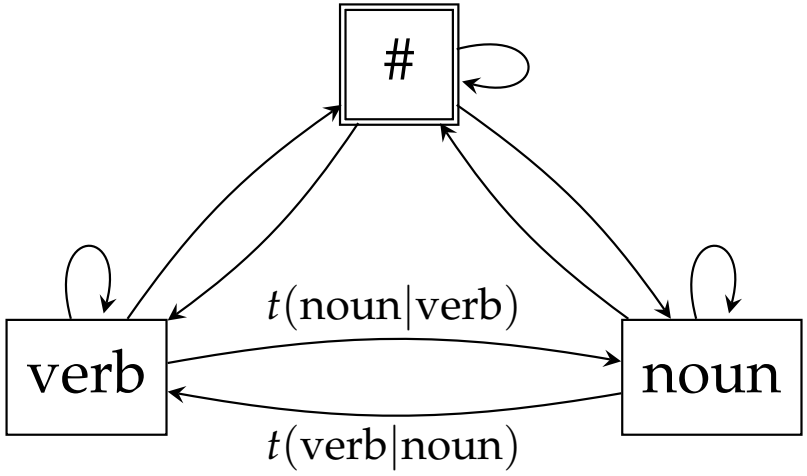
- ...wird durch empirische Wahrscheinlichkeit maximiert

$$b(w'|w) = \frac{\text{count}(w\ w')}{\sum_{w''} \text{count}(w\ w'')}$$

- z. B. für Korpus: „Alice sees Bob“, „Alice likes cake“

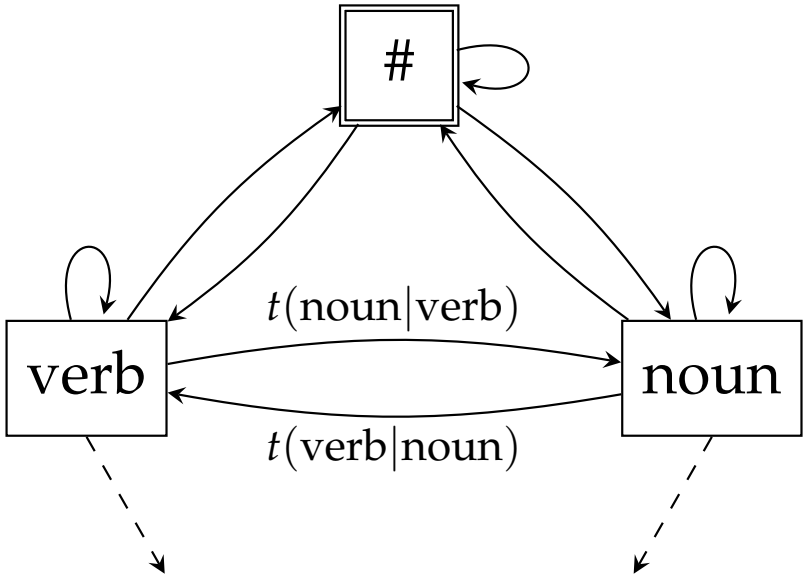
x	count(Alice x)	$b(x\|\text{Alice})$
Alice	0	0
sees	1	0,5
Bob	0	0
likes	1	0,5
cake	0	0
#	0	0

Hidden-Markov-Modell



- Idee: Wahrscheinlichkeit jedes Wortes hängt vom Verhalten eines Zustandsautomaten ab

Hidden-Markov-Modell

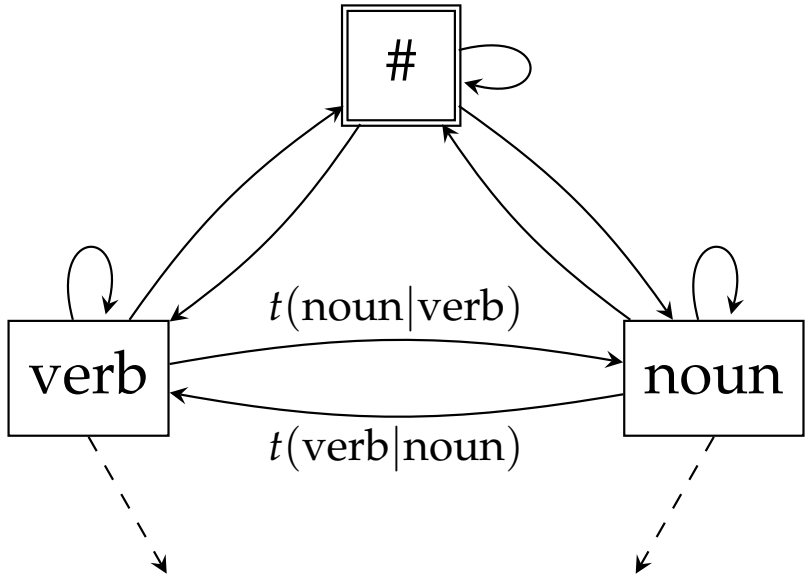


x	$e(x\|\text{verb})$
Alice	0
Bob	0
cake	0
likes	0,5
sees	0,5

x	$e(x\|\text{noun})$
Alice	0,5
Bob	0,25
cake	0,25
likes	0
sees	0

- Idee: Wahrscheinlichkeit jedes Wortes hängt vom Verhalten eines Zustandsautomaten ab

Hidden-Markov-Modell



x	$e(x\|\text{verb})$
Alice	0
Bob	0
cake	0
likes	0,5
sees	0,5

x	$e(x\|\text{noun})$
Alice	0,5
Bob	0,25
cake	0,25
likes	0
sees	0

- Idee: Wahrscheinlichkeit jedes Wortes hängt vom Verhalten eines Zustandsautomaten ab
- Modellparameter: $\omega = (t, e)$
  - Transitions-Wkt. t
  - Emissions-Wkt. e

$$p_\omega(x = x_1 \cdots x_n) = \sum_{q_1,\ldots,q_n} t(q_1|\#) \cdot e(x_1|q_1) \cdot \left[\prod_{i=2}^{n} t(q_i|q_{i-1}) \cdot e(x_i|q_i)\right] \cdot t(\#|q_n)$$

Hidden-Markov-Modell: Training

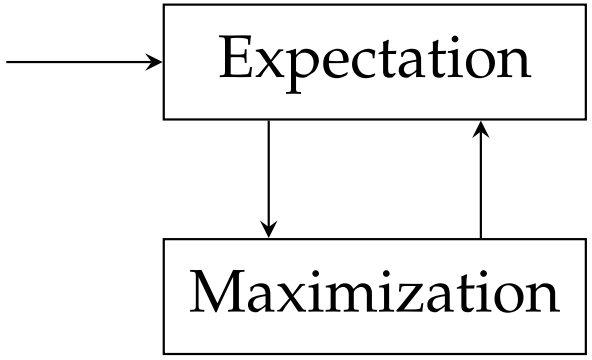
- Problem: Korpus enthält nur Sätze, keine Zustände
  - optimales $\omega = (t, e)$ nicht direkt ablesbar

Hidden-Markov-Modell: Training

- Problem: Korpus enthält nur Sätze, keine Zustände
  - optimales $\omega = (t, e)$ nicht direkt ablesbar
- Idee: alle möglichen Beiträge zum Ergebnis abschätzen
  - *Baum-Welch-Algorithmus*

Hidden-Markov-Modell: Training

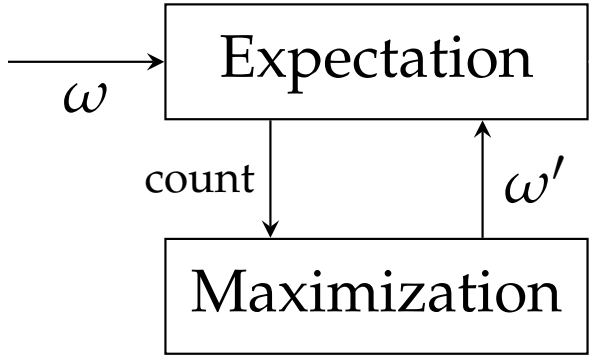
- Problem: Korpus enthält nur Sätze, keine Zustände
  - optimales $\omega = (t, e)$ nicht direkt ablesbar
- Idee: alle möglichen Beiträge zum Ergebnis abschätzen
  - *Baum-Welch-Algorithmus*
- z. B. für Transitionswahrscheinlichkeit von # nach q



$$\text{count}(q, \#) \leftarrow \sum_x c(x) \cdot P(q_1 = q|x)$$

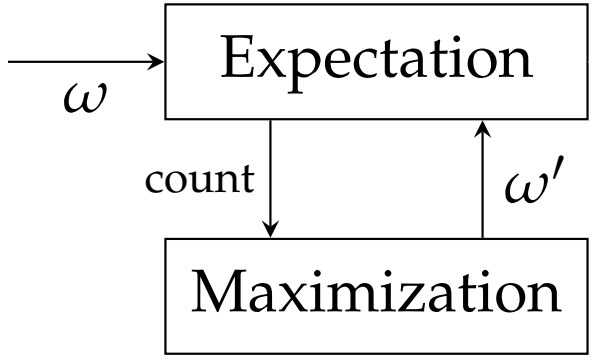
$$t(q|\#) \leftarrow \frac{\text{count}(q,\#)}{\sum_{q'} \text{count}(q',\#)}$$

Warum ein allgemeiner EM-Algorithmus?



- EM = iterative Anpassung des Modellparameters ω

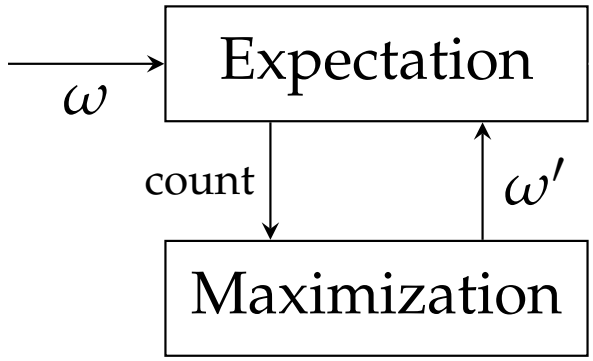
Warum ein allgemeiner EM-Algorithmus?



- EM = iterative Anpassung des Modellparameters ω
- Ist der neue Modellparameter wirklich *besser* als der alte?

$$p_{\omega'}(c) \geq p_{\omega}(c)$$

Warum ein allgemeiner EM-Algorithmus?



- EM = iterative Anpassung des Modellparameters ω
- Ist der neue Modellparameter wirklich *besser* als der alte?

$$p_{\omega'}(c) \geq p_{\omega}(c)$$

- EM-Algorithmen nach [BSV15] haben bewiesene Konvergenz-Eigenschaften

[BSV15] Matthias Büchse, Torsten Stüber und Heiko Vogler. *A generic inside-outside EM algorithm*. Unpublished Manuscript, 2015.

Gliederung

Sprachmodelle

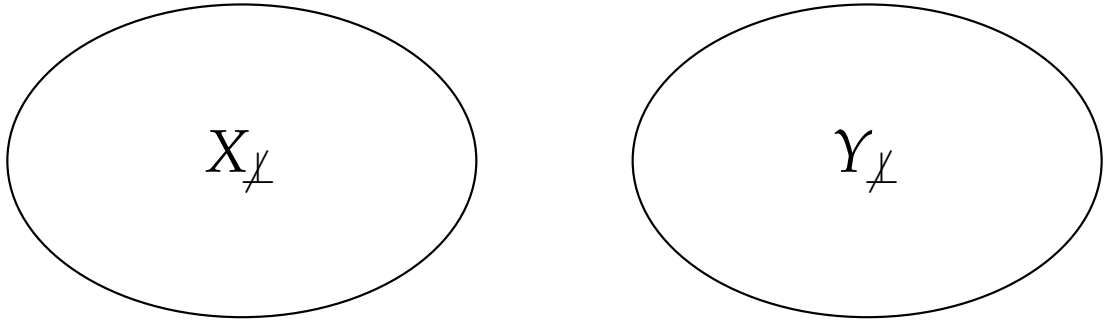
- Bigramm-Modell
- Hidden-Markov-Modell

EM-Algorithmen

- Baum-Welch-Algorithmus
- Inside-Outside-EM-Algorithmus
- Instanziierung für Hidden-Markov-Modell

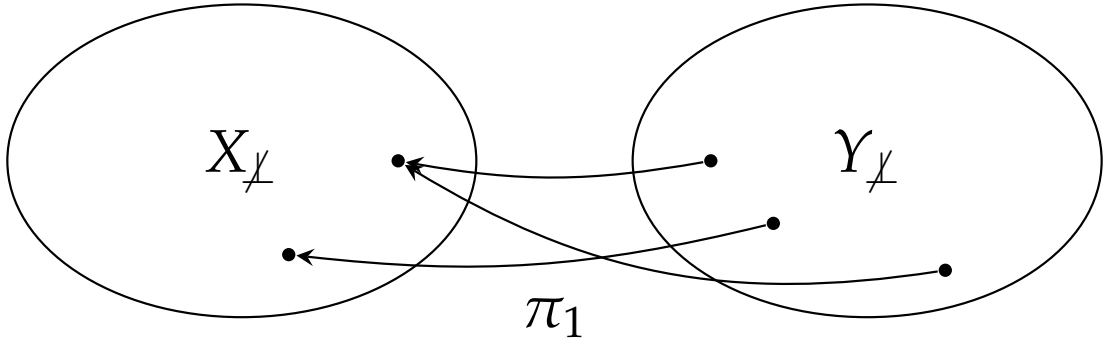
Inside-Outside-EM-Algorithmus nach [BSV15]

- $X_{\not\perp}$: abzählbare Menge der *Beobachtungen* (z. B. Sätze)
- $Y_{\not\perp}$: abzählbare Menge der *versteckten Informationen* (z. B. Syntaxbäume)

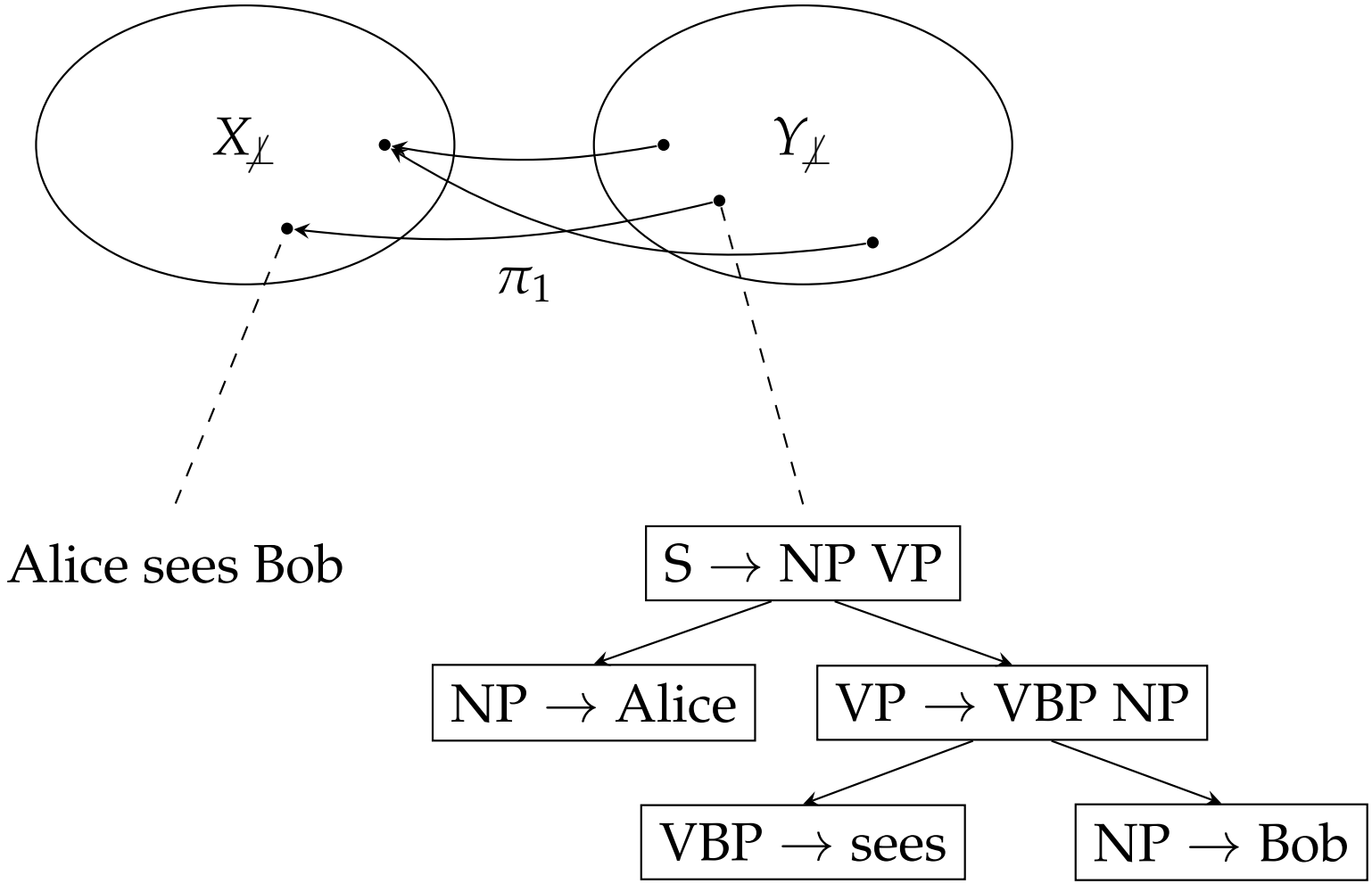


Inside-Outside-EM-Algorithmus nach [BSV15]

- $X_{\not\perp}$: abzählbare Menge der *Beobachtungen* (z. B. Sätze)
- $Y_{\not\perp}$: abzählbare Menge der *versteckten Informationen* (z. B. Syntaxbäume)
- $\pi_1 \colon Y_{\not\perp} \to X_{\not\perp}$ ordnet jedem y eine Beobachtung x zu



Bsp. Kontextfreie Grammatiken



Zählinformationen

Jedes $y \in Y_{\not\perp}$ ist ein Baum mit Rang, der mit Zählereignissen beschriftet ist.

Zählinformationen

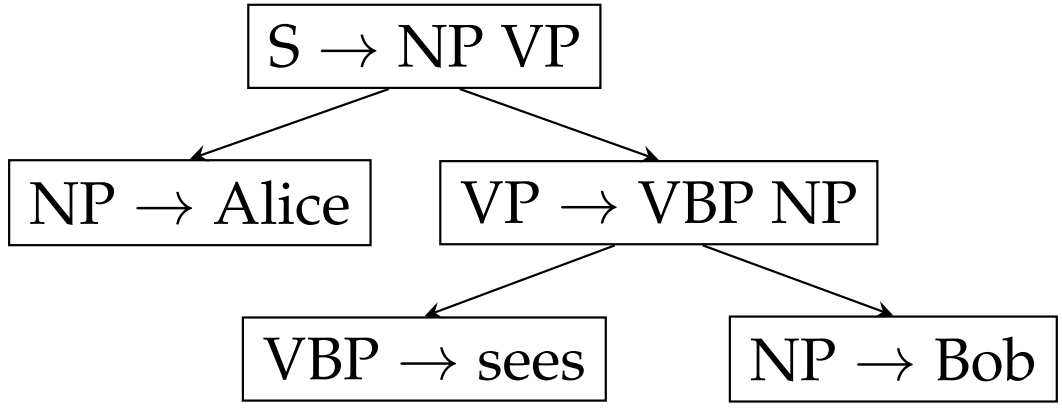
Jedes $y \in Y_{\not\perp}$ ist ein Baum mit Rang, der mit Zählereignissen beschriftet ist.

- $C \subseteq A \times B$: *Zählereignisse* (hierbei A, B abzählbare Mengen)
- *Baum mit Rang:* Anzahl der Nachfolger eines Knotens ist eine Funktion der Beschriftung des Knotens

Zählinformationen

Jedes $y \in Y_{\not\perp}$ ist ein Baum mit Rang, der mit Zählereignissen beschriftet ist.

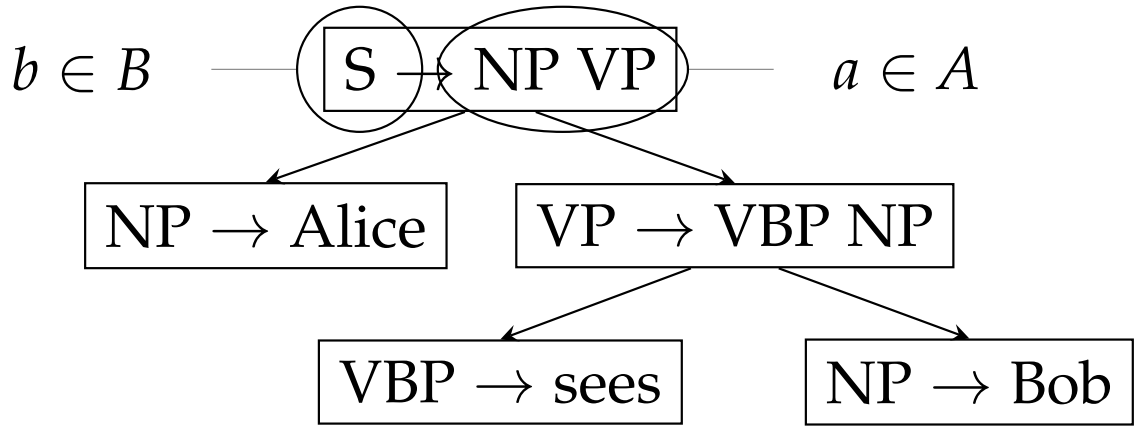
- $C \subseteq A \times B$: *Zählereignisse* (hierbei A, B abzählbare Mengen)
- *Baum mit Rang:* Anzahl der Nachfolger eines Knotens ist eine Funktion der Beschriftung des Knotens



Zählinformationen

Jedes $y \in Y_{\not\perp}$ ist ein Baum mit Rang, der mit Zählereignissen beschriftet ist.

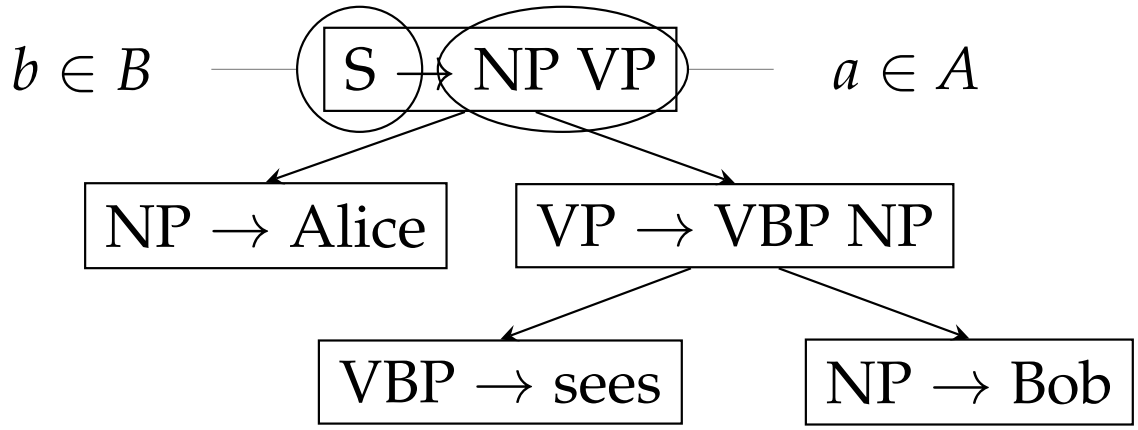
- $C \subseteq A \times B$: *Zählereignisse* (hierbei A, B abzählbare Mengen)
- *Baum mit Rang:* Anzahl der Nachfolger eines Knotens ist eine Funktion der Beschriftung des Knotens



Zählinformationen

Jedes $y \in Y_{\not\perp}$ ist ein Baum mit Rang, der mit Zählereignissen beschriftet ist.

- Ω: Menge von *Modellparametern* ω
- q: Ω-*Wahrscheinlichkeitsmodell* für A und B (eingeschränkt auf C)
- $q_\omega(a|b)$: Wahrscheinlichkeit des Zählereignisses $c = (a, b)$



$$p_\omega(y) = \prod_{\substack{w \in \mathrm{pos}(y) \\ (a,b) = y(w)}} q_\omega(a|b)$$

Inside-Outside-EM-Algorithmus (Forts.)

Bestandteile einer *Inside-Outside-Information:*

- X, Y, π_1, A, B, C, Ω, q: wie gesehen

Inside-Outside-EM-Algorithmus (Forts.)

Bestandteile einer *Inside-Outside-Information:*

- X, Y, π_1, A, B, C, Ω, q: wie gesehen
- K: reguläre Baumgrammatik (RTG), die alle $y \in Y_{\not\perp}$ erzeugt
- $H(x)$: RTG, die alle $y \in Y_{\not\perp}$ mit $\pi_1(y) = x$ erzeugt
- Eindeutigkeit: je RTG genau eine Ableitung für jedes y

Inside-Outside-EM-Algorithmus (Forts.)

Bestandteile einer *Inside-Outside-Information:*

- X, Y, π_1, A, B, C, Ω, q: wie gesehen
- K: reguläre Baumgrammatik (RTG), die alle $y \in Y_{\not\perp}$ erzeugt
- $H(x)$: RTG, die alle $y \in Y_{\not\perp}$ mit $\pi_1(y) = x$ erzeugt
- Eindeutigkeit: je RTG genau eine Ableitung für jedes y
- *reguläre Baumgrammatik:* erzeugt Bäumen aus Regeln wie

$$q_{\mathrm{VP}} \to \left(\begin{array}{c} \boxed{\mathrm{VP} \to \mathrm{VBP\ NP}} \\ \swarrow \qquad \searrow \\ q_{\mathrm{VBP}} \qquad q_{\mathrm{NP}} \end{array} \right)$$

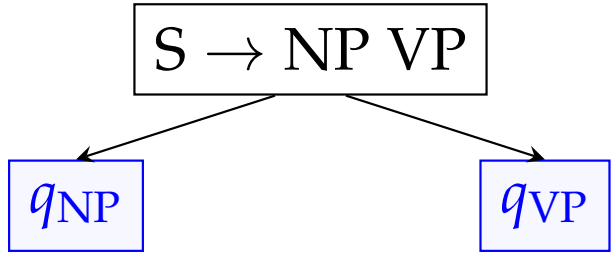
Bsp. Syntaxbaum einer CFG aus Baumgrammatik

$\boxed{q_{\mathrm{S}}}$

Beispiele für Regeln:

$$q_{\mathrm{S}} \to \left(\begin{array}{c} \boxed{\mathrm{S} \to \mathrm{NP\ VP}} \\ \swarrow \qquad \searrow \\ q_{\mathrm{NP}} \qquad\quad q_{\mathrm{VP}} \end{array} \right) \qquad q_{\mathrm{NP}} \to \boxed{\mathrm{NP} \to \mathrm{Alice}}$$

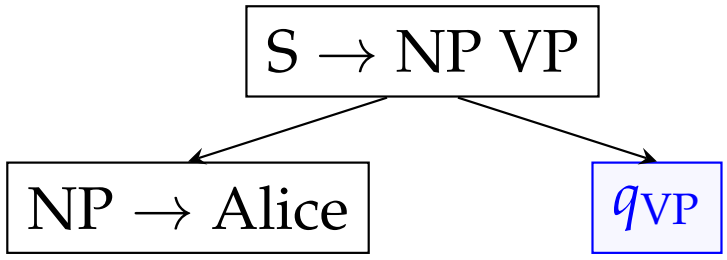
Bsp. Syntaxbaum einer CFG aus Baumgrammatik



Beispiele für Regeln:

$$q_{\text{S}} \to \left(\begin{array}{c} \boxed{\text{S} \to \text{NP VP}} \\ q_{\text{NP}} \qquad q_{\text{VP}} \end{array} \right) \qquad q_{\text{NP}} \to \boxed{\text{NP} \to \text{Alice}}$$

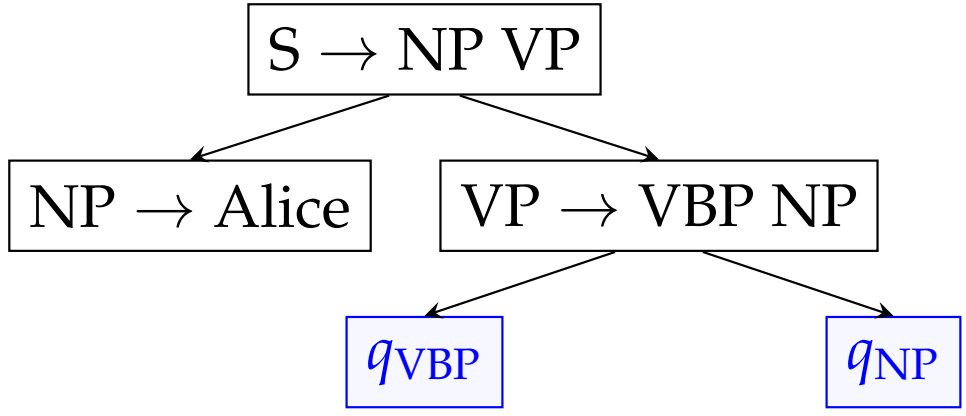
Bsp. Syntaxbaum einer CFG aus Baumgrammatik



Beispiele für Regeln:

$$q_{\text{S}} \to \left(\begin{array}{c} \boxed{\text{S} \to \text{NP VP}} \\ q_{\text{NP}} \qquad q_{\text{VP}} \end{array} \right) \qquad q_{\text{NP}} \to \boxed{\text{NP} \to \text{Alice}}$$

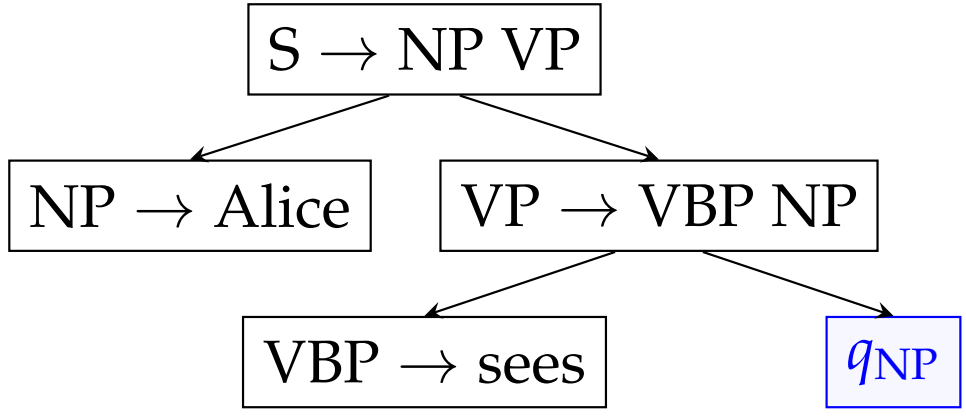
Bsp. Syntaxbaum einer CFG aus Baumgrammatik



Beispiele für Regeln:

$$q_{\text{S}} \to \left(\begin{array}{c} \boxed{\text{S} \to \text{NP VP}} \\ q_{\text{NP}} \qquad q_{\text{VP}} \end{array} \right) \qquad q_{\text{NP}} \to \boxed{\text{NP} \to \text{Alice}}$$

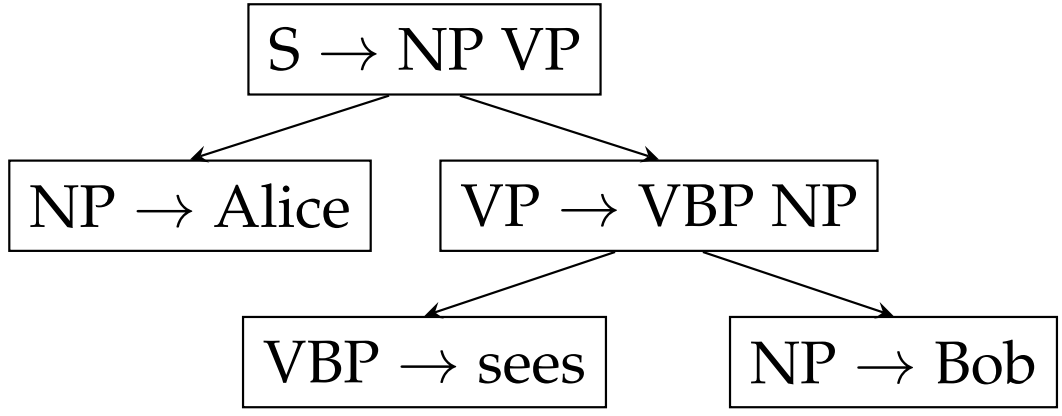
Bsp. Syntaxbaum einer CFG aus Baumgrammatik



Beispiele für Regeln:

$$q_{\text{S}} \to \left(\begin{array}{c} \boxed{\text{S} \to \text{NP VP}} \\ q_{\text{NP}} \qquad q_{\text{VP}} \end{array} \right) \qquad q_{\text{NP}} \to \boxed{\text{NP} \to \text{Alice}}$$

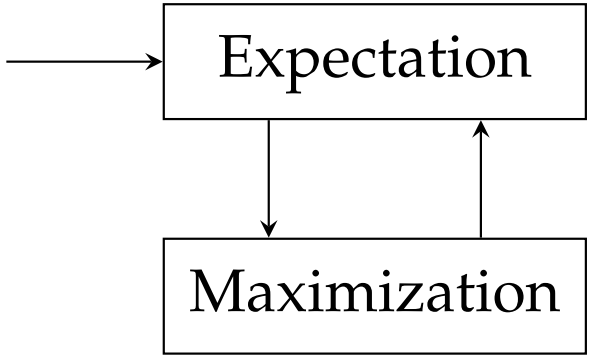
Bsp. Syntaxbaum einer CFG aus Baumgrammatik



Beispiele für Regeln:

$$q_{\mathrm{S}} \to \left(\begin{array}{c} \boxed{\mathrm{S} \to \mathrm{NP\ VP}} \\ q_{\mathrm{NP}} \qquad q_{\mathrm{VP}} \end{array} \right) \qquad q_{\mathrm{NP}} \to \boxed{\mathrm{NP} \to \mathrm{Alice}}$$

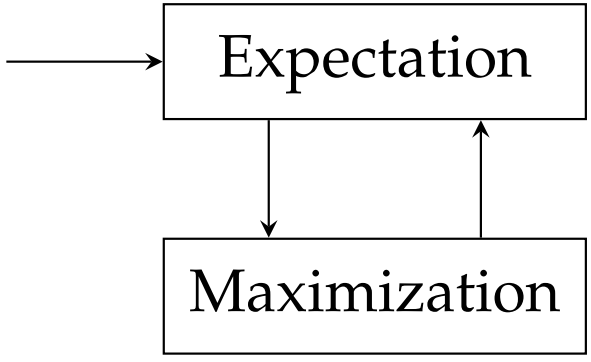
EM-Algorithmus für Inside-Outside-Informationen



$$\text{count}(q,\#) \leftarrow \sum_x c(x) \cdot P(q_1 = q|x)$$

$$t(q|\#) \leftarrow \frac{\text{count}(q,\#)}{\sum_{q'} \text{count}(q',\#)}$$

EM-Algorithmus für Inside-Outside-Informationen

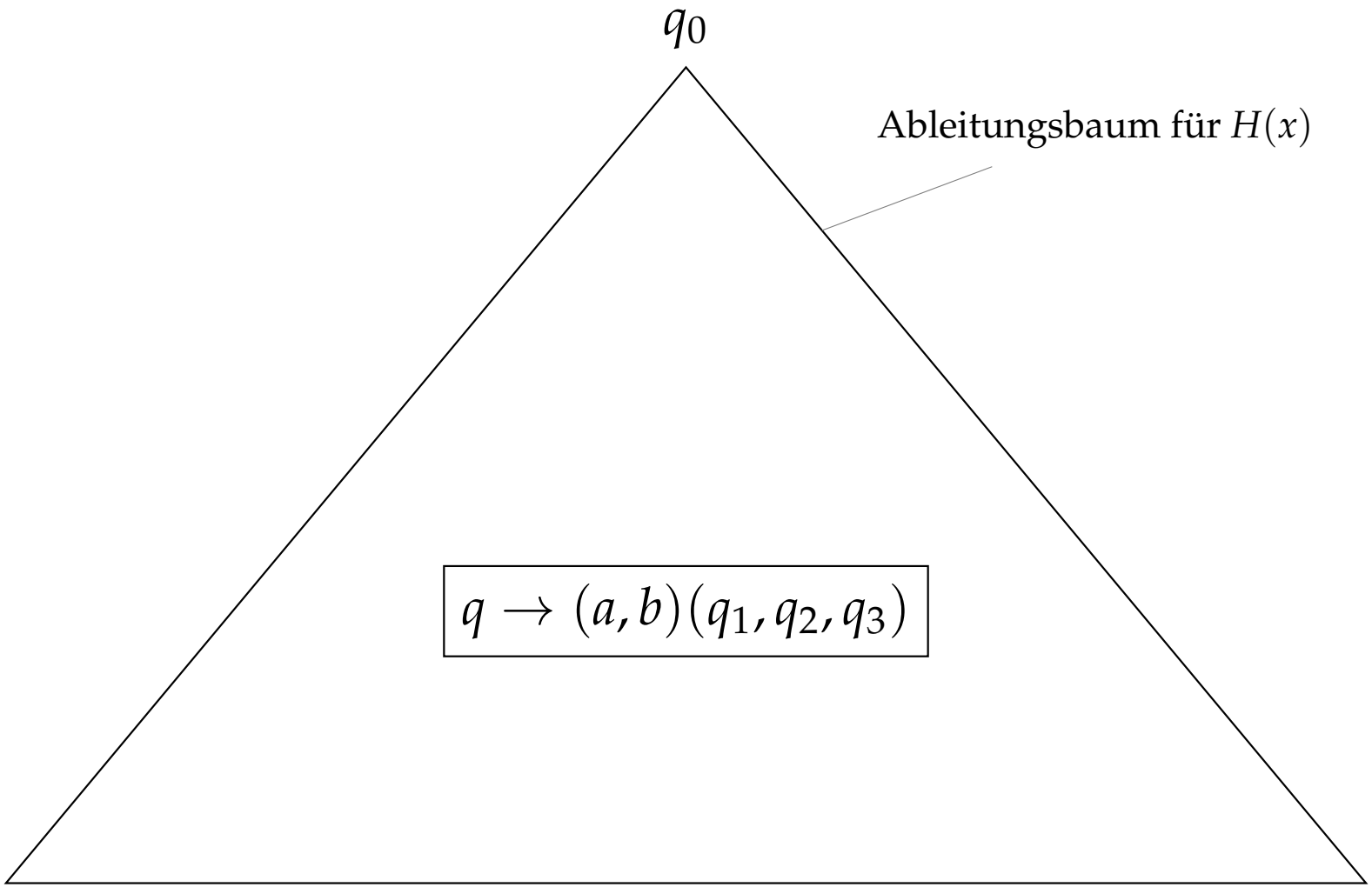


$$\text{count}(a,b) \leftarrow \sum_x c(x) \cdot \chi_{\omega,x}(a,b)$$

$$q_{\omega'}(a|b) \leftarrow \frac{\text{count}(a,b)}{\sum_{a'} \text{count}(a',b)}$$

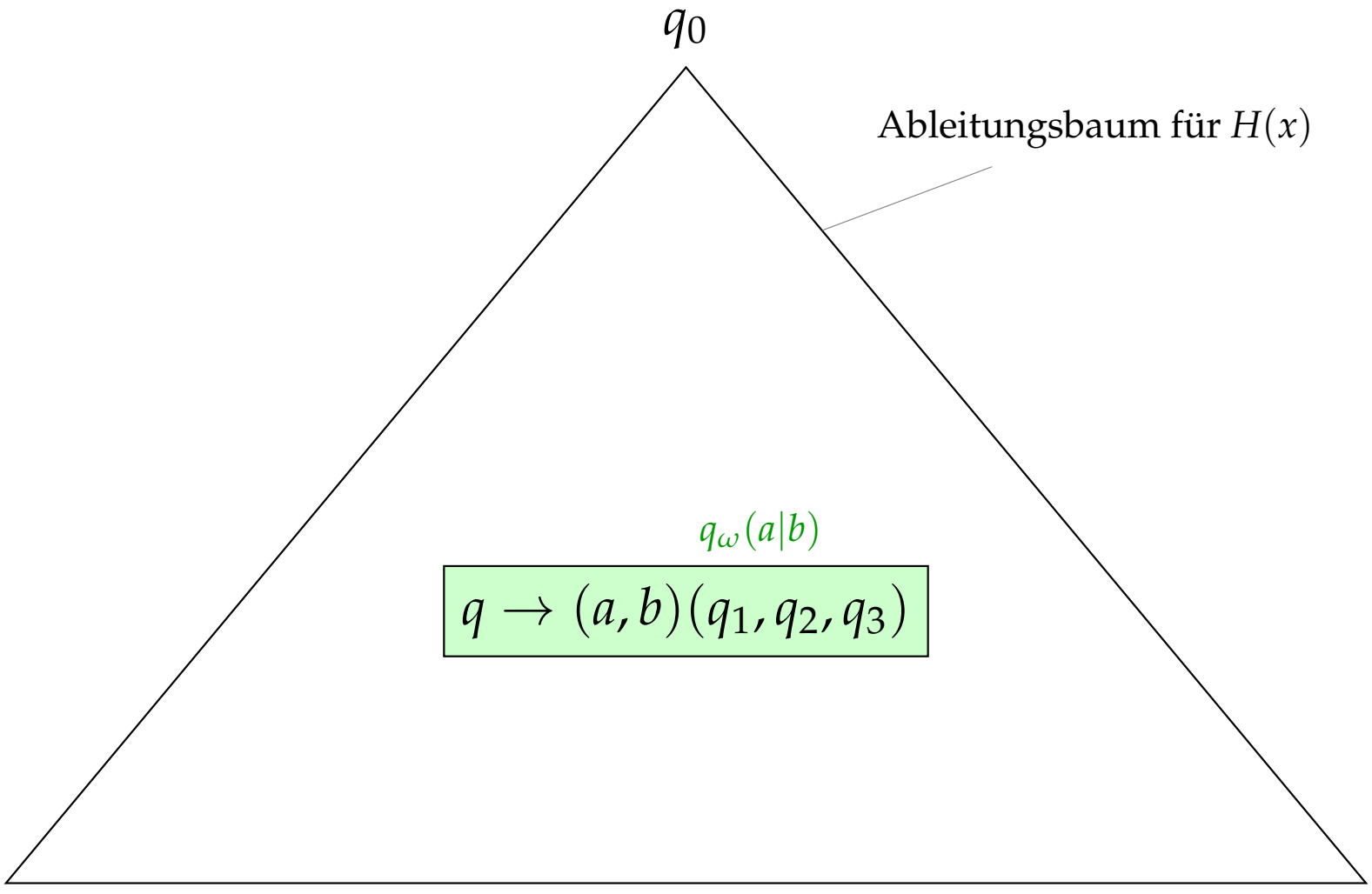
- $\chi_{\omega,x}(a,b)$ nutzt bekanntes q_ω, um Häufigkeit von (a,b) in der Beobachtung x abzuschätzen
- dann ω' so wählen, dass $q_{\omega'}$ der empirischen Wahrscheinlichkeitsverteilung von $\text{count}(a,b)$ entspricht

Zur Berechnung von $\chi_{\omega,x}(a,b)$



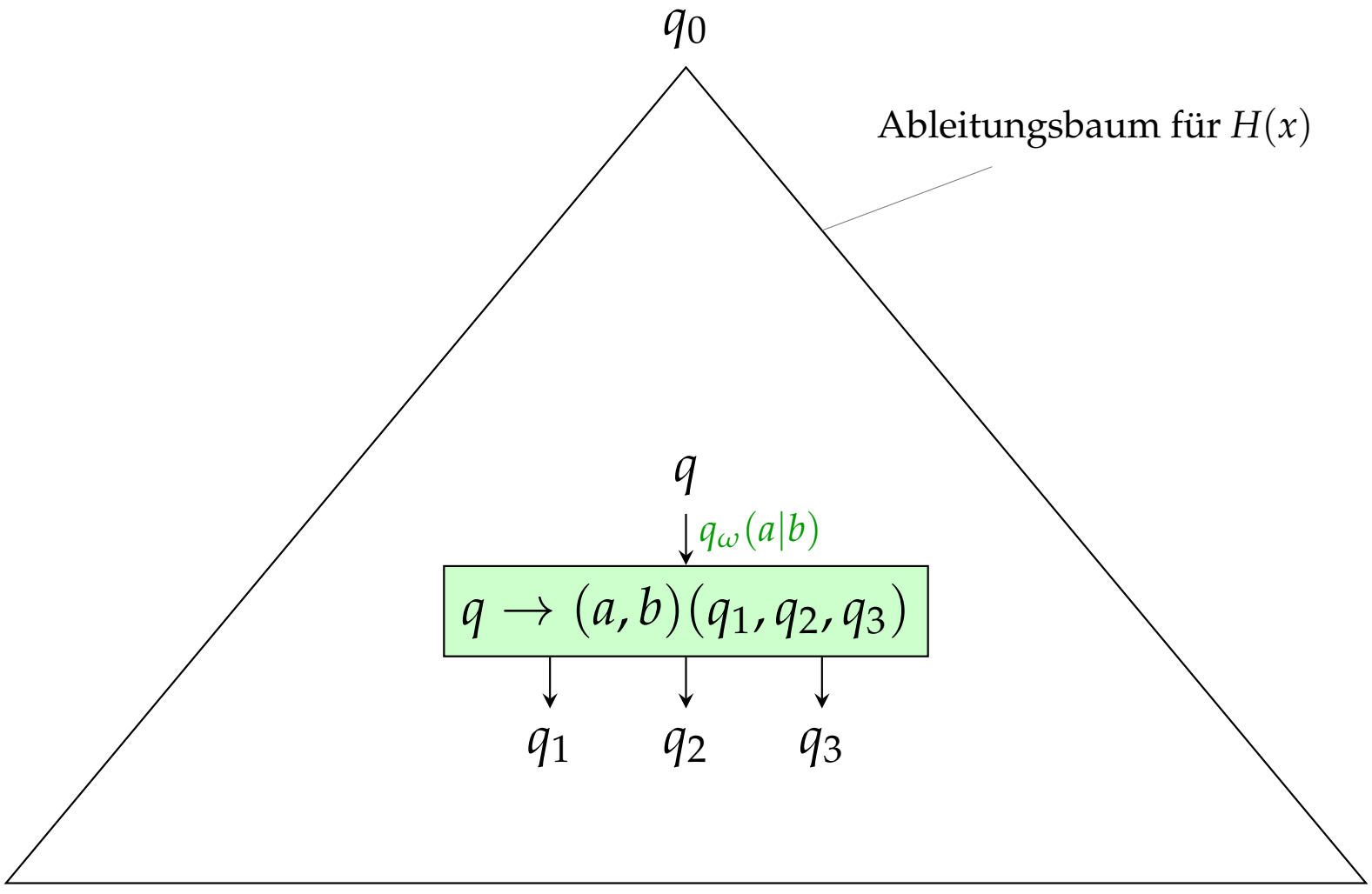
Grafik nach: H. Vogler. *Maschinelles Übersetzen natürlicher Sprachen.* Vorlesung, TU Dresden, Wintersemester 2015/2016.

Zur Berechnung von $\chi_{\omega,x}(a,b)$



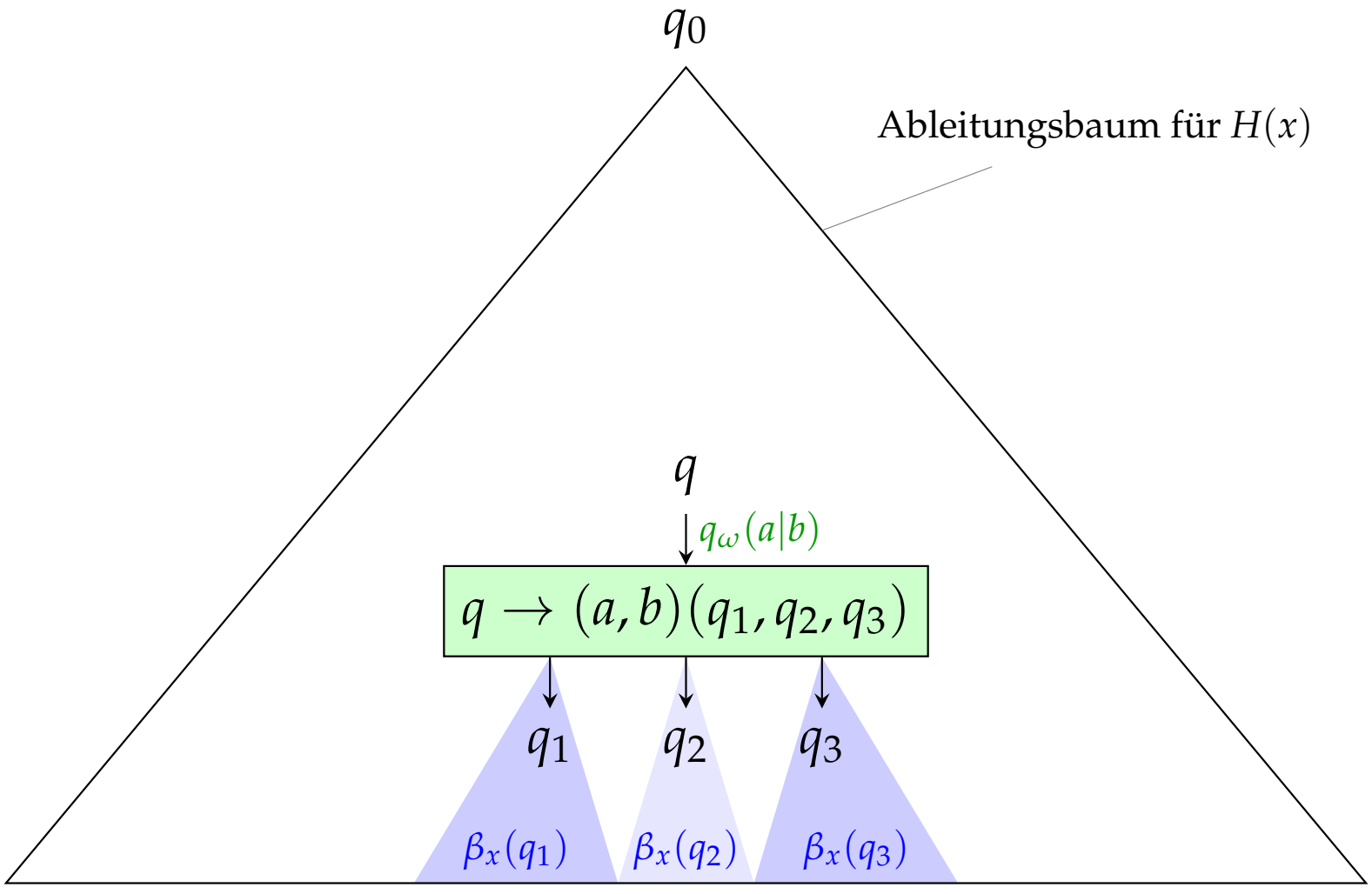
Grafik nach: H. Vogler. *Maschinelles Übersetzen natürlicher Sprachen.* Vorlesung, TU Dresden, Wintersemester 2015/2016.

Zur Berechnung von $\chi_{\omega,x}(a,b)$



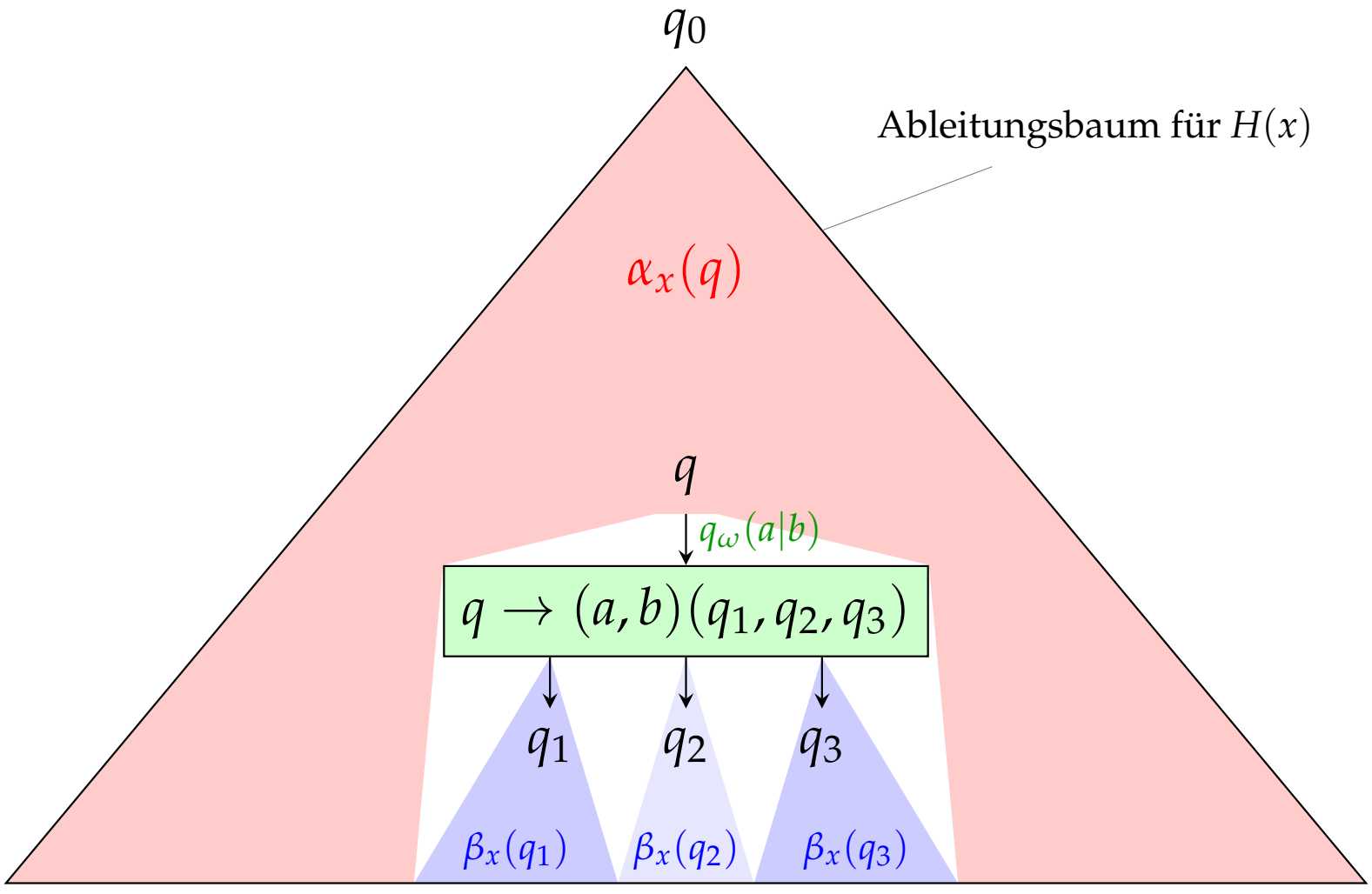
Grafik nach: H. Vogler. *Maschinelles Übersetzen natürlicher Sprachen.* Vorlesung, TU Dresden, Wintersemester 2015/2016.

Zur Berechnung von $\chi_{\omega,x}(a,b)$



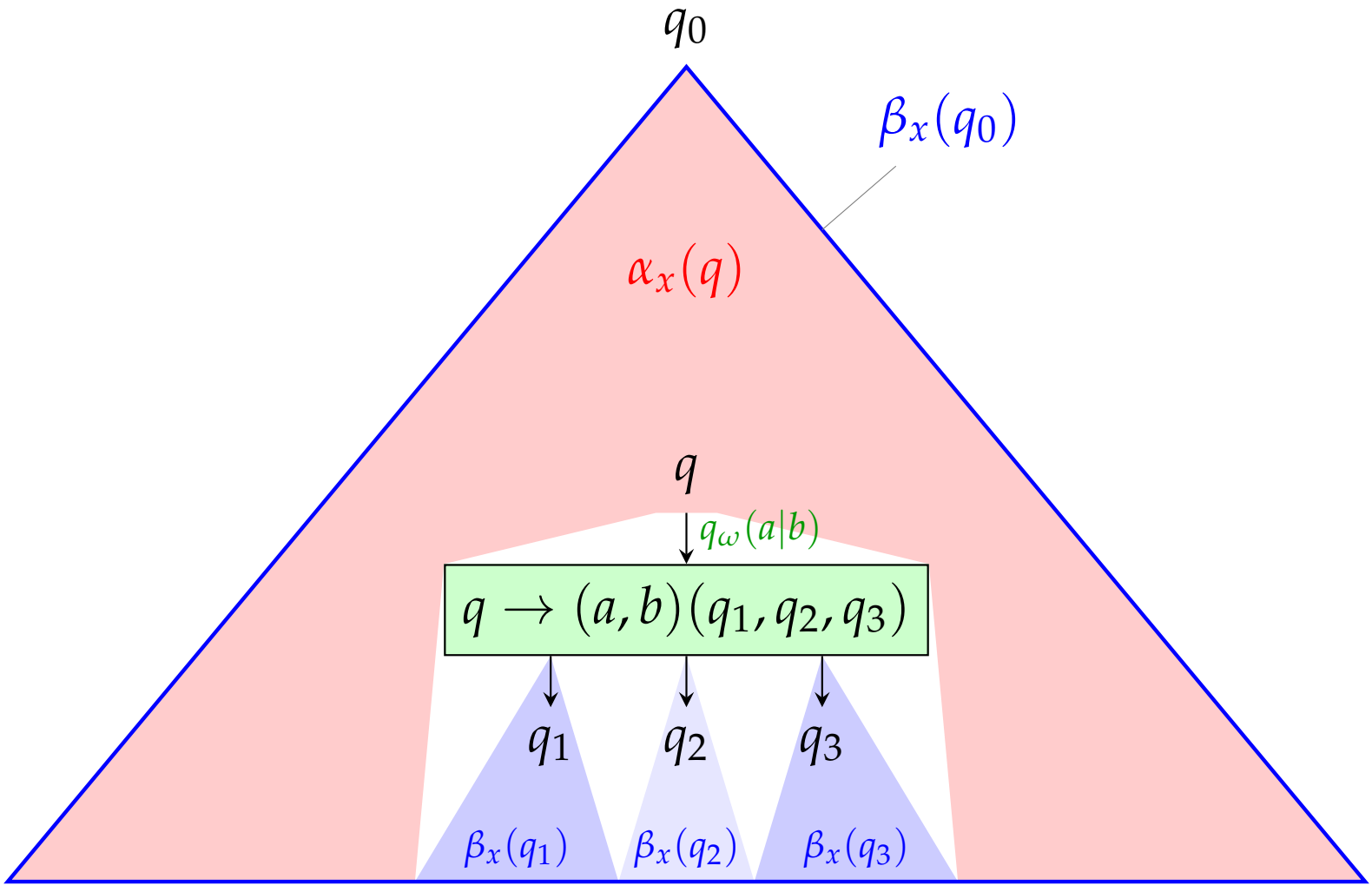
Grafik nach: H. Vogler. *Maschinelles Übersetzen natürlicher Sprachen.* Vorlesung, TU Dresden, Wintersemester 2015/2016.

Zur Berechnung von $\chi_{\omega,x}(a,b)$



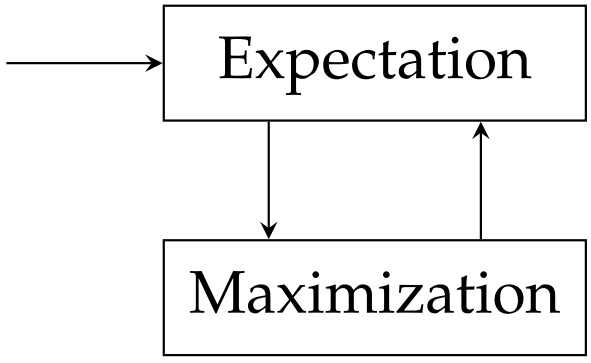
Grafik nach: H. Vogler. *Maschinelles Übersetzen natürlicher Sprachen.* Vorlesung, TU Dresden, Wintersemester 2015/2016.

Zur Berechnung von $\chi_{\omega,x}(a,b)$



Grafik nach: H. Vogler. *Maschinelles Übersetzen natürlicher Sprachen.* Vorlesung, TU Dresden, Wintersemester 2015/2016.

EM-Algorithmus für Inside-Outside-Informationen

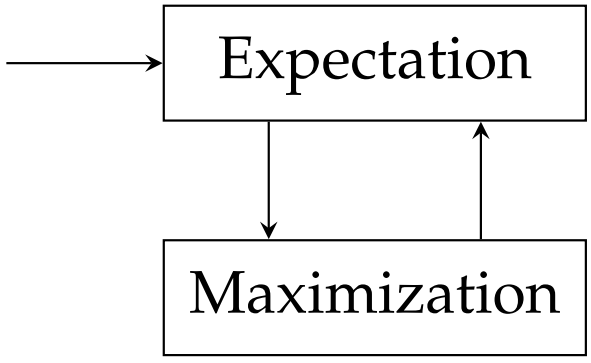


$$\text{count}(a,b) \leftarrow \sum_x c(x) \cdot \chi_{\omega,x}(a,b)$$

$$q_{\omega'}(a|b) \leftarrow \frac{\text{count}(a,b)}{\sum_{a'} \text{count}(a',b)}$$

$$\chi_{\omega,x}(a,b) = \frac{\sum_{q \to (a,b)(q_1,\ldots,q_k)} \alpha_x(q) \cdot q_\omega(a|b) \cdot \beta_x(q_1) \cdots \beta_x(q_k)}{\beta_x(q_0)}$$

EM-Algorithmus für Inside-Outside-Informationen



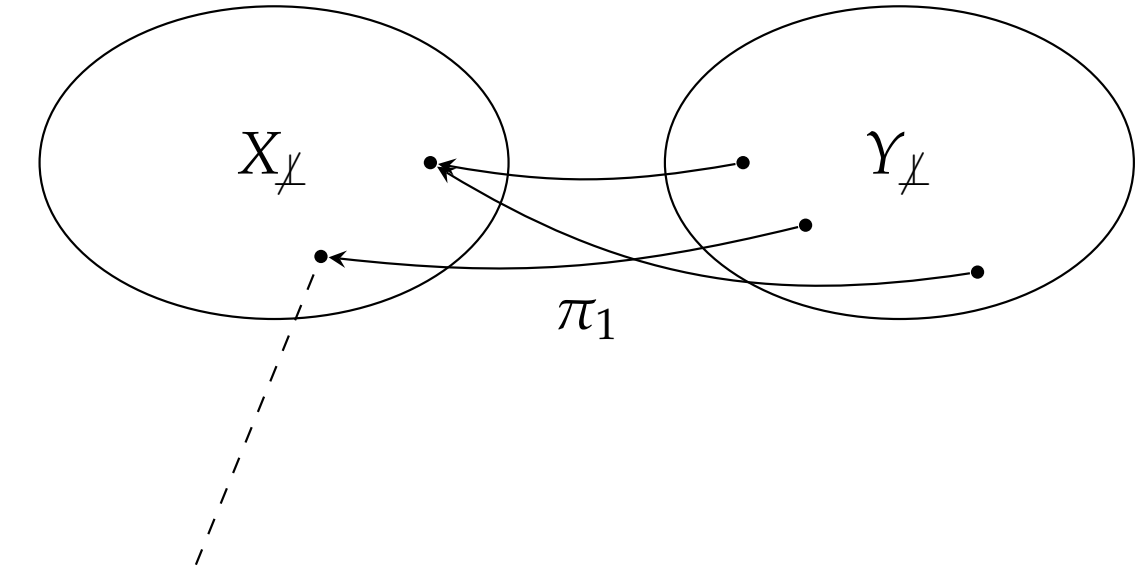
$$\text{count}(a,b) \leftarrow \sum_x c(x) \cdot \chi_{\omega,x}(a,b)$$

$$q_{\omega'}(a|b) \leftarrow \frac{\text{count}(a,b)}{\sum_{a'} \text{count}(a',b)}$$

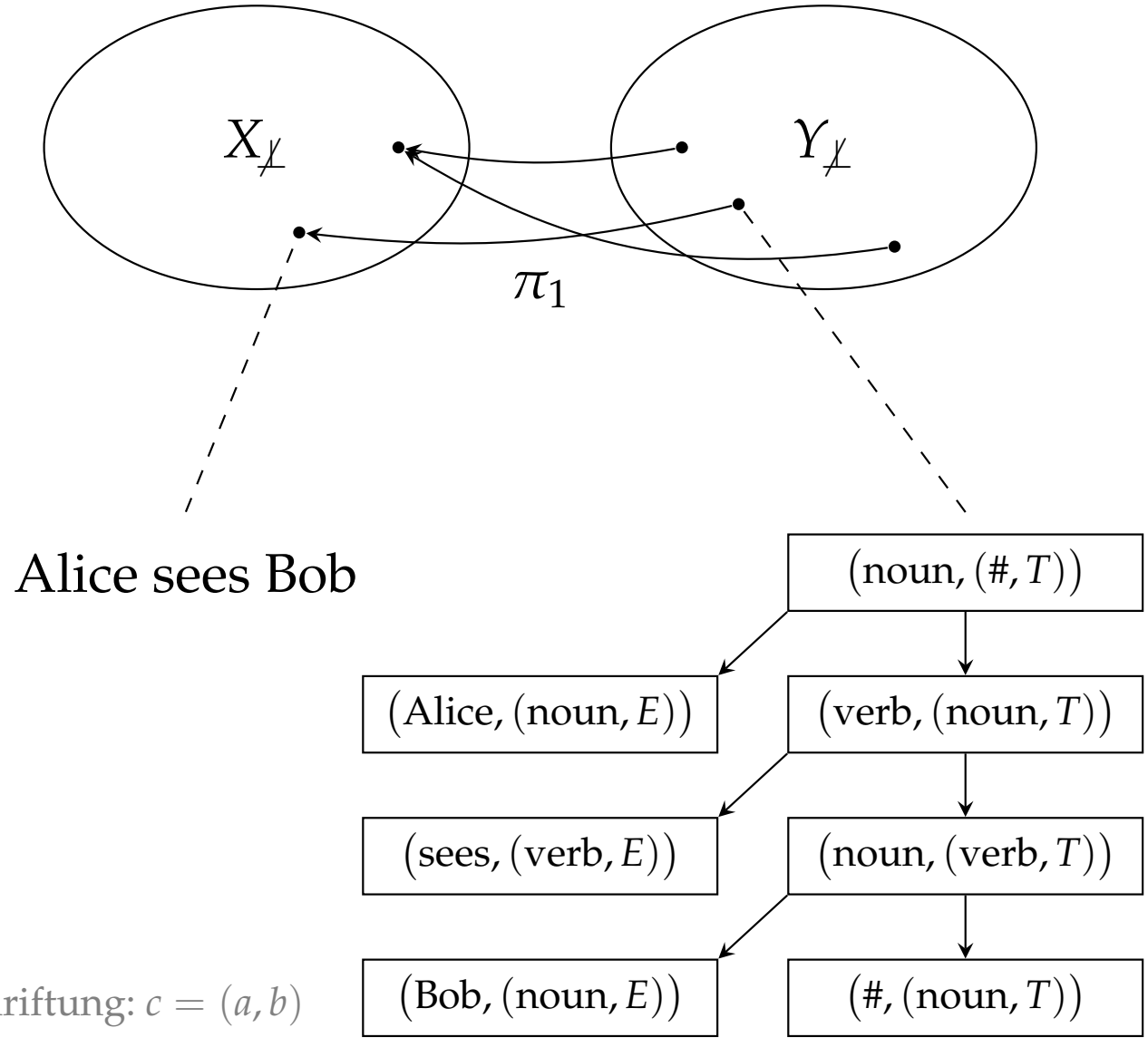
$$\chi_{\omega,x}(a,b) = \frac{\sum_{q \to (a,b)(q_1,\ldots,q_k)} \alpha_x(q) \cdot q_\omega(a|b) \cdot \beta_x(q_1) \cdots \beta_x(q_k)}{\beta_x(q_0)}$$

Ist der Baum-Welch-Algorithmus eine Instanz hiervon?

Versteckte Information für Hidden-Markov-Modell

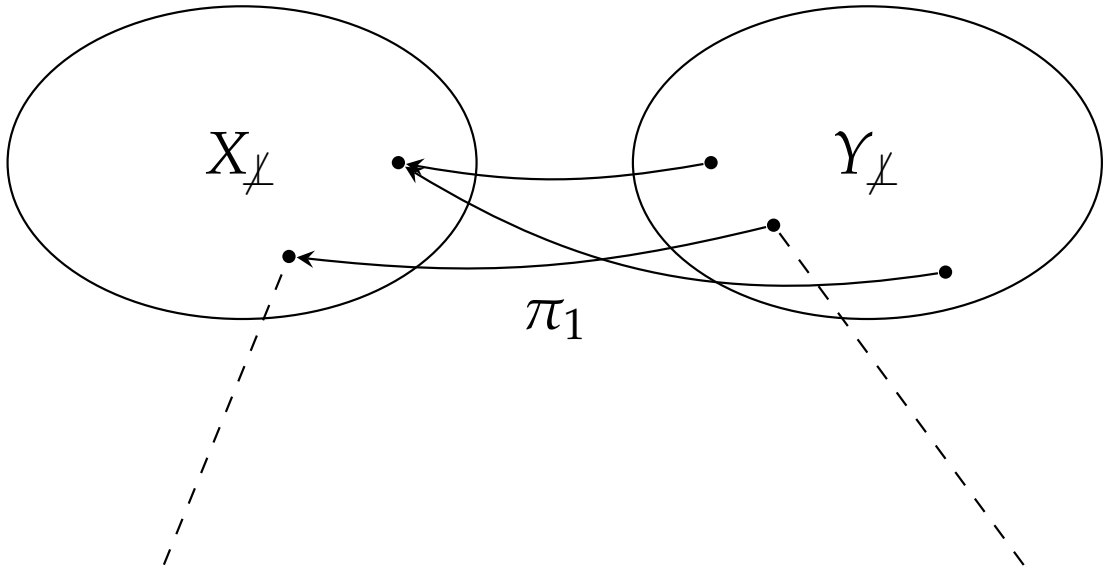


Versteckte Information für Hidden-Markov-Modell

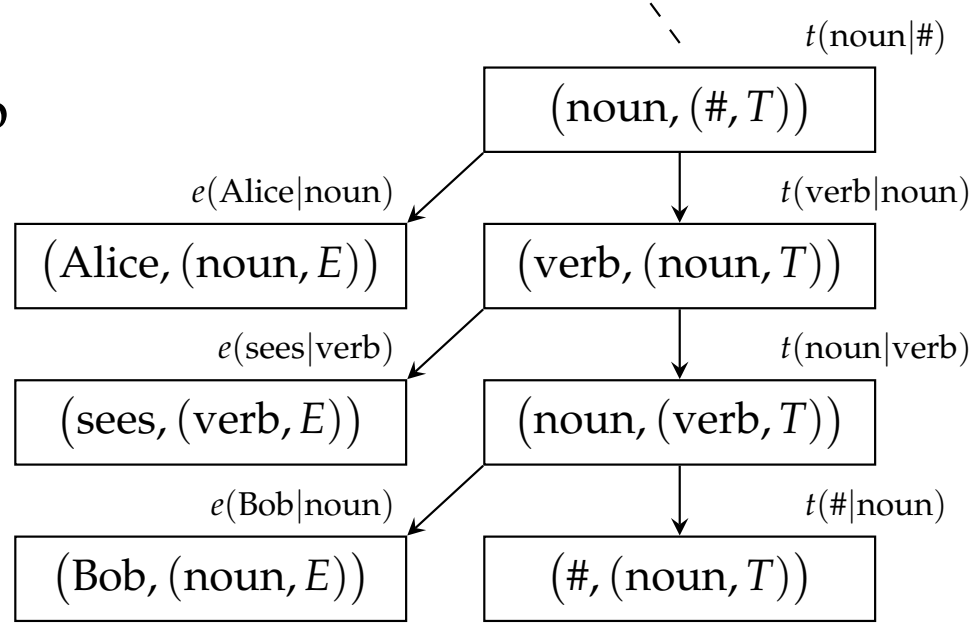


Knotenbeschriftung: $c = (a, b)$

Versteckte Information für Hidden-Markov-Modell



Alice sees Bob



Knotenbeschriftung: $c = (a, b)$
Annotation: $q_\omega(a|b)$

Beispiel: Ableitung aus der Baumgrammatik K

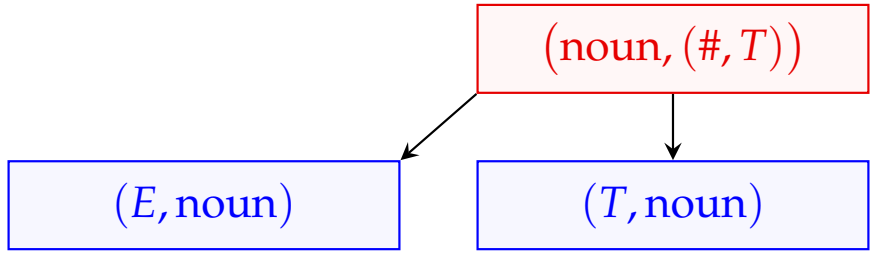
$(T, \#)$

- Zustände: (T, q) für $q \in Q \cup \{\#\}$ und (E, q) für $q \in Q$
- Startzustand: $(T, \#)$; Regeln:

$$
\begin{aligned}
(T, q) &\to \big(q', (q, T)\big)\,\big((E, q'), (T, q')\big) && \text{für } q \in Q \cup \{\#\}, q' \in Q \\
(T, q) &\to \big(\#, (q, T)\big) && \text{für } q \in Q \cup \{\#\} \\
(E, q) &\to \big(v, (q, E)\big) && \text{für } q \in Q, v \in V
\end{aligned}
$$

- $H(x)$: Zustände erweitert um noch nicht generierte Wörter

Beispiel: Ableitung aus der Baumgrammatik K

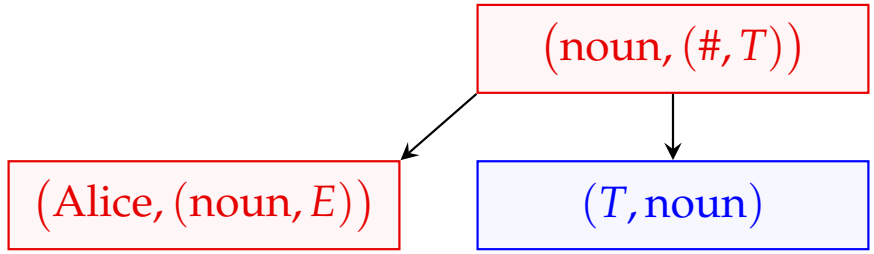


- Zustände: (T, q) für $q \in Q \cup \{\#\}$ und (E, q) für $q \in Q$
- Startzustand: $(T, \#)$; Regeln:

$$
\begin{aligned}
(T, q) &\to \big(q', (q, T)\big)\,\big((E, q'), (T, q')\big) && \text{für } q \in Q \cup \{\#\}, q' \in Q \\
(T, q) &\to \big(\#, (q, T)\big) && \text{für } q \in Q \cup \{\#\} \\
(E, q) &\to \big(v, (q, E)\big) && \text{für } q \in Q, v \in V
\end{aligned}
$$

- $H(x)$: Zustände erweitert um noch nicht generierte Wörter

Beispiel: Ableitung aus der Baumgrammatik K

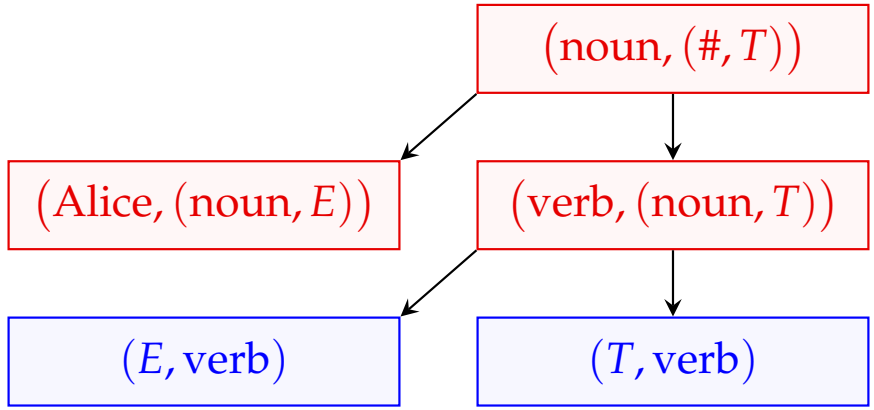


- Zustände: (T, q) für $q \in Q \cup \{\#\}$ und (E, q) für $q \in Q$
- Startzustand: $(T, \#)$; Regeln:

$$
\begin{aligned}
(T, q) &\to \big(q', (q, T)\big)\,\big((E, q'), (T, q')\big) && \text{für } q \in Q \cup \{\#\}, q' \in Q \\
(T, q) &\to \big(\#, (q, T)\big) && \text{für } q \in Q \cup \{\#\} \\
(E, q) &\to \big(v, (q, E)\big) && \text{für } q \in Q, v \in V
\end{aligned}
$$

- $H(x)$: Zustände erweitert um noch nicht generierte Wörter

Beispiel: Ableitung aus der Baumgrammatik K

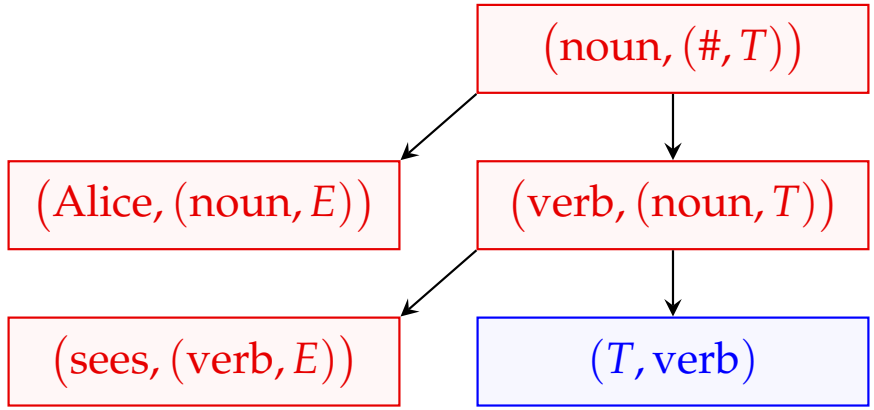


- Zustände: (T, q) für $q \in Q \cup \{\#\}$ und (E, q) für $q \in Q$
- Startzustand: $(T, \#)$; Regeln:

$$(T,q) \to \big(q', (q,T)\big)\big((E,q'),(T,q')\big) \quad \text{für } q \in Q \cup \{\#\}, q' \in Q$$
$$(T,q) \to \big(\#, (q,T)\big) \quad \text{für } q \in Q \cup \{\#\}$$
$$(E,q) \to \big(v, (q,E)\big) \quad \text{für } q \in Q, v \in V$$

- $H(x)$: Zustände erweitert um noch nicht generierte Wörter

Beispiel: Ableitung aus der Baumgrammatik K

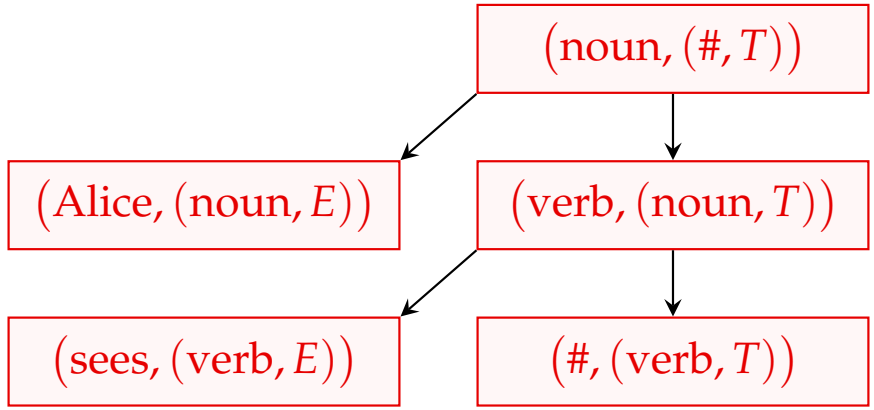


- Zustände: (T, q) für $q \in Q \cup \{\#\}$ und (E, q) für $q \in Q$
- Startzustand: $(T, \#)$; Regeln:

$$
\begin{aligned}
(T, q) &\to \big(q', (q, T)\big)\,\big((E, q'), (T, q')\big) && \text{für } q \in Q \cup \{\#\}, q' \in Q \\
(T, q) &\to \big(\#, (q, T)\big) && \text{für } q \in Q \cup \{\#\} \\
(E, q) &\to \big(v, (q, E)\big) && \text{für } q \in Q, v \in V
\end{aligned}
$$

- $H(x)$: Zustände erweitert um noch nicht generierte Wörter

Beispiel: Ableitung aus der Baumgrammatik K

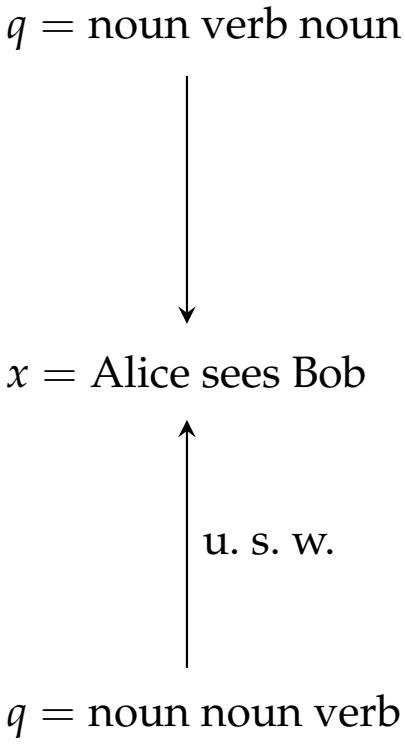


- Zustände: (T, q) für $q \in Q \cup \{\#\}$ und (E, q) für $q \in Q$
- Startzustand: $(T, \#)$; Regeln:

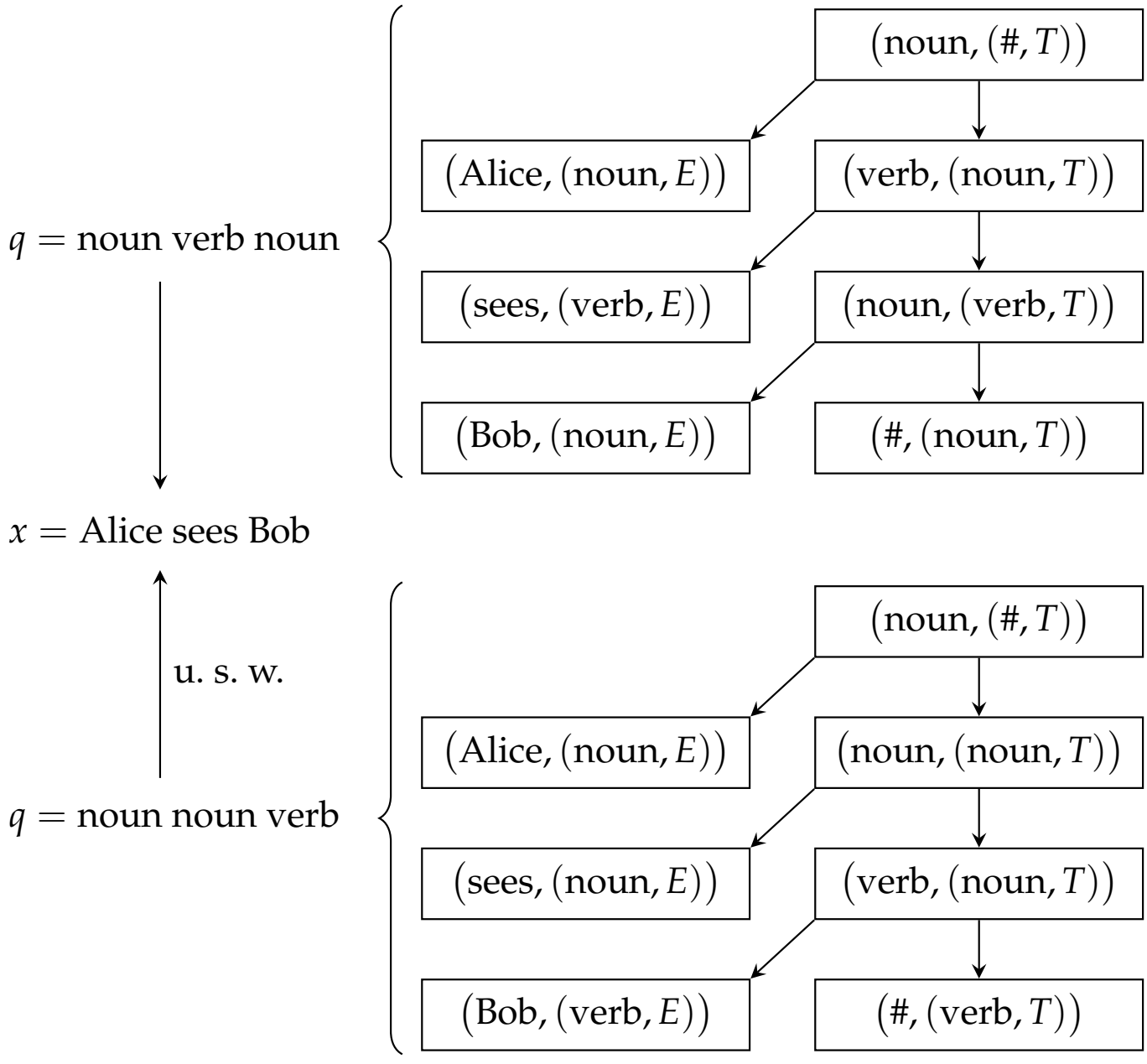
$$(T, q) \to \big(q', (q, T)\big)\,\big((E, q'), (T, q')\big) \quad \text{für } q \in Q \cup \{\#\}, q' \in Q$$
$$(T, q) \to \big(\#, (q, T)\big) \quad \text{für } q \in Q \cup \{\#\}$$
$$(E, q) \to \big(v, (q, E)\big) \quad \text{für } q \in Q, v \in V$$

- $H(x)$: Zustände erweitert um noch nicht generierte Wörter

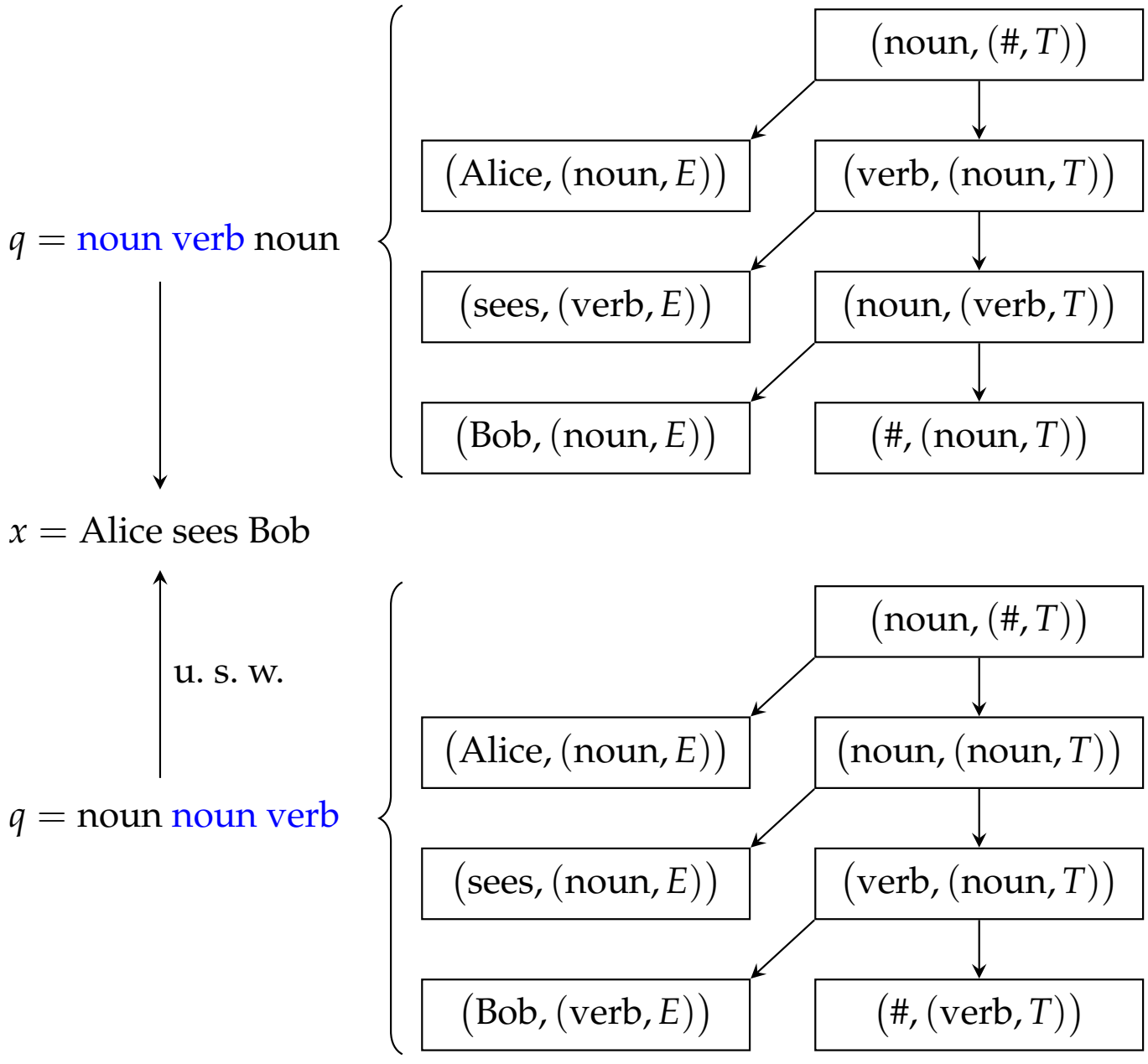
Zur Instanziierung des IO-EM-Algorithmus



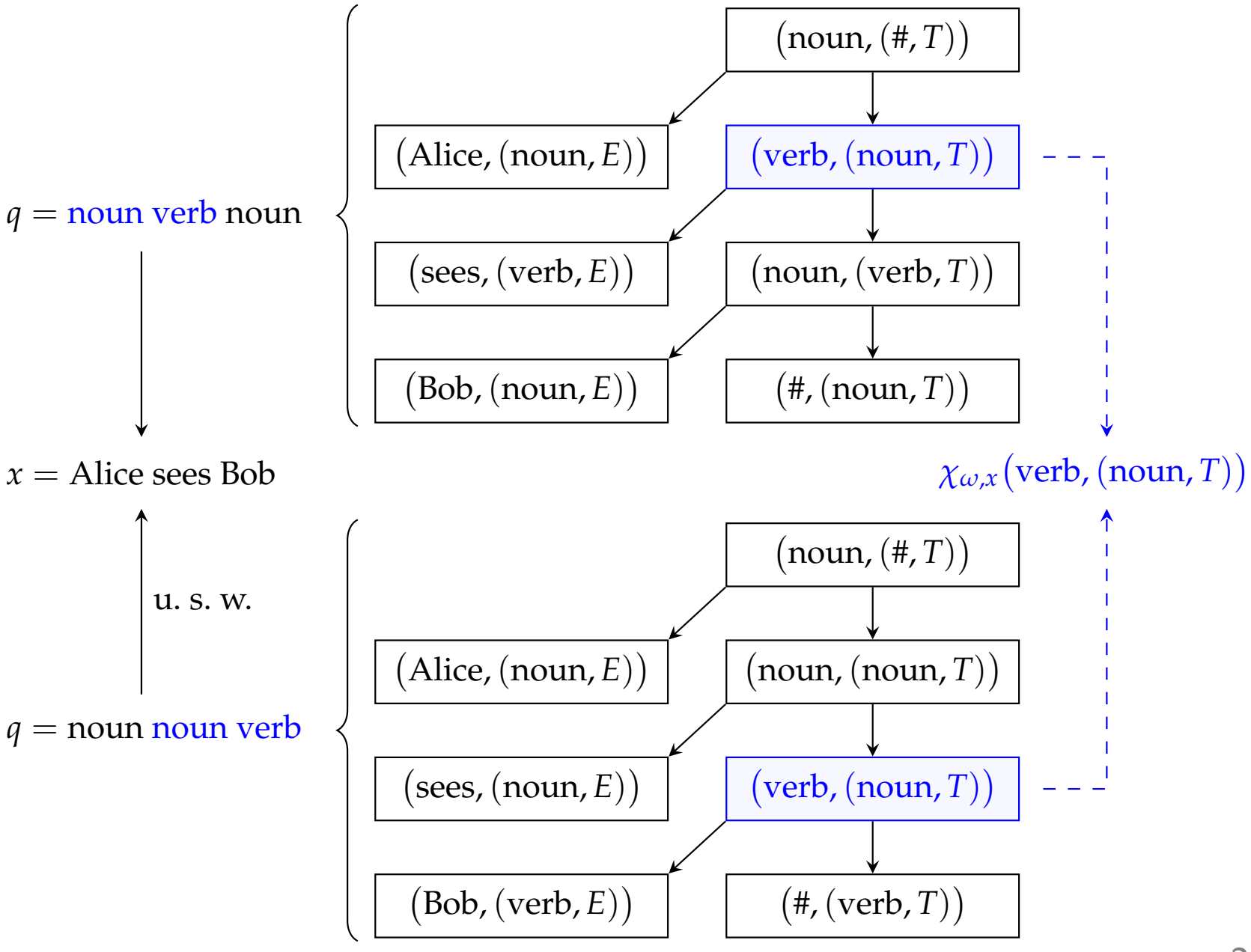
Zur Instanziierung des IO-EM-Algorithmus



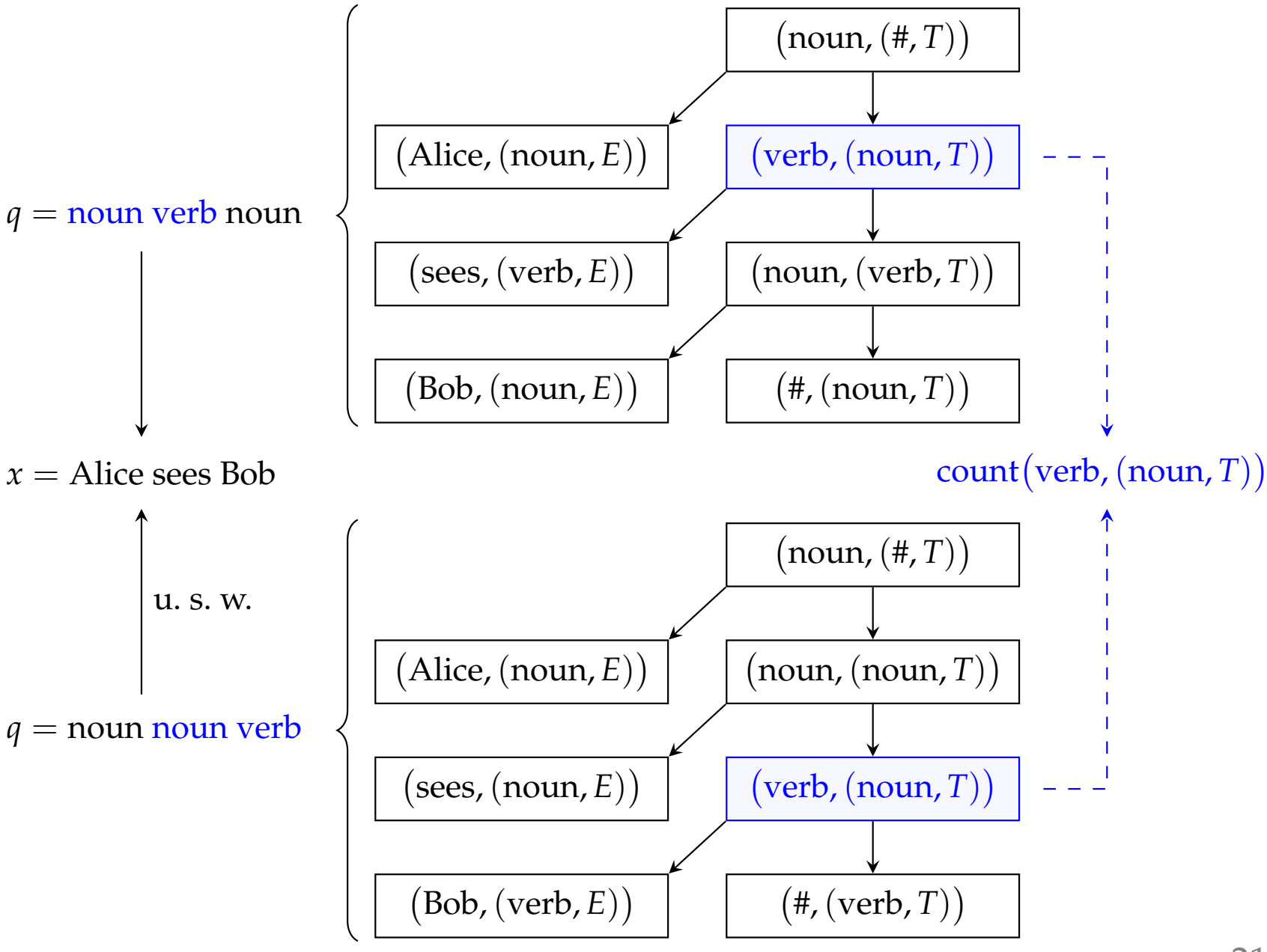
Zur Instanziierung des IO-EM-Algorithmus



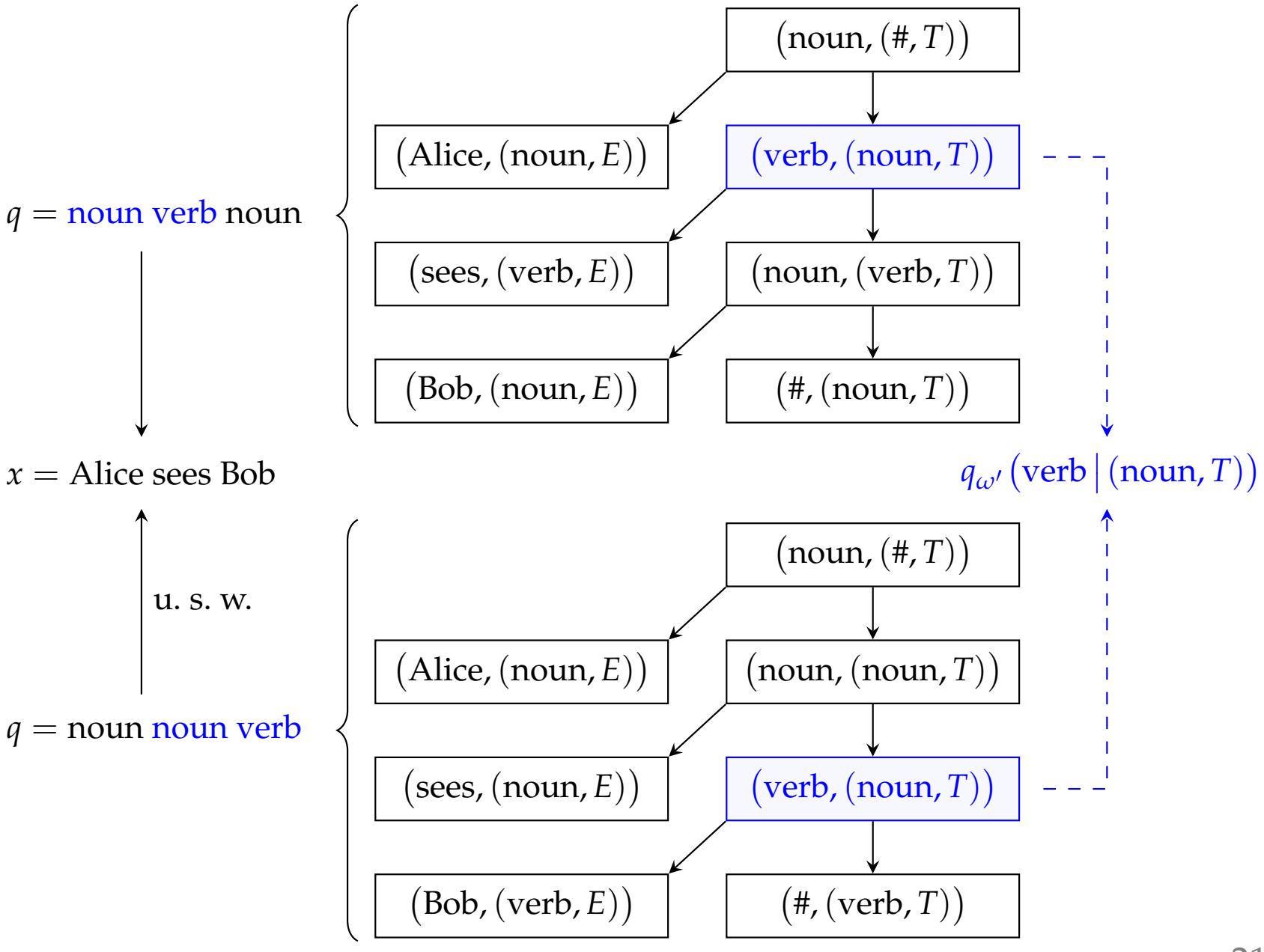
Zur Instanziierung des IO-EM-Algorithmus



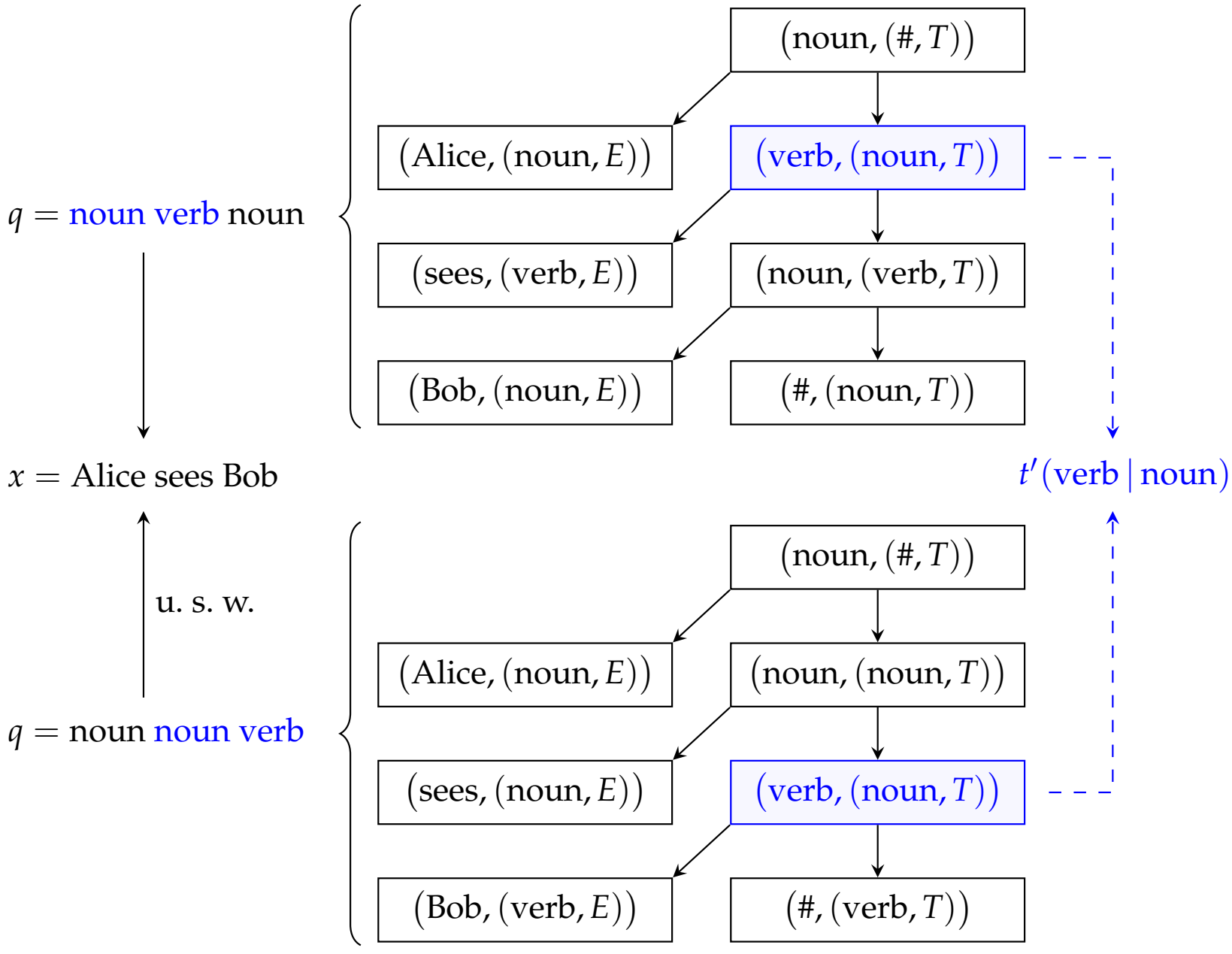
Zur Instanziierung des IO-EM-Algorithmus



Zur Instanziierung des IO-EM-Algorithmus



Zur Instanziierung des IO-EM-Algorithmus



Gliederung

Sprachmodelle

- Bigramm-Modell
- Hidden-Markov-Modell

EM-Algorithmen

- Baum-Welch-Algorithmus
- Inside-Outside-EM-Algorithmus
- Instanziierung für Hidden-Markov-Modell