Kapitel L:II

II. Aussagenlogik

- Syntax der Aussagenlogik
- Semantik der Aussagenlogik
- Eigenschaften des Folgerungsbegriffs
- Äquivalenz
- Formeltransformation
- Normalformen

- Bedeutung der Folgerung
- Erfüllbarkeitsalgorithmen
- Semantische Bäume
- Weiterentwicklung semantischer Bäume
- Syntaktische Schlussfolgerungsverfahren
- Erfüllbarkeitsprobleme

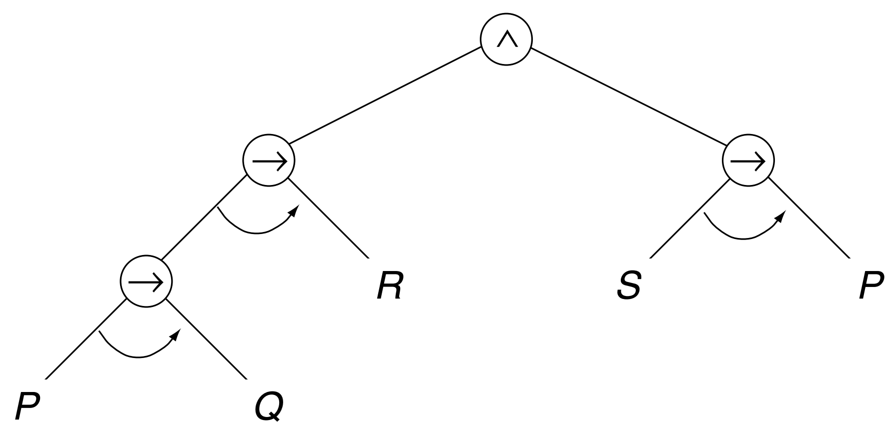
Formeltransformation

Jede Formel kann als Datenstruktur in der Form eines Baumes interpretiert werden:

- Junktoren entsprechen den inneren Knoten
- Atome entsprechen den Blättern
- (Teil-)Formeln entsprechen (Teil-)Bäumen
- Bei nicht-assoziativen Junktoren ist die Reihenfolge der Nachfolger eines Knotens zu beachten.

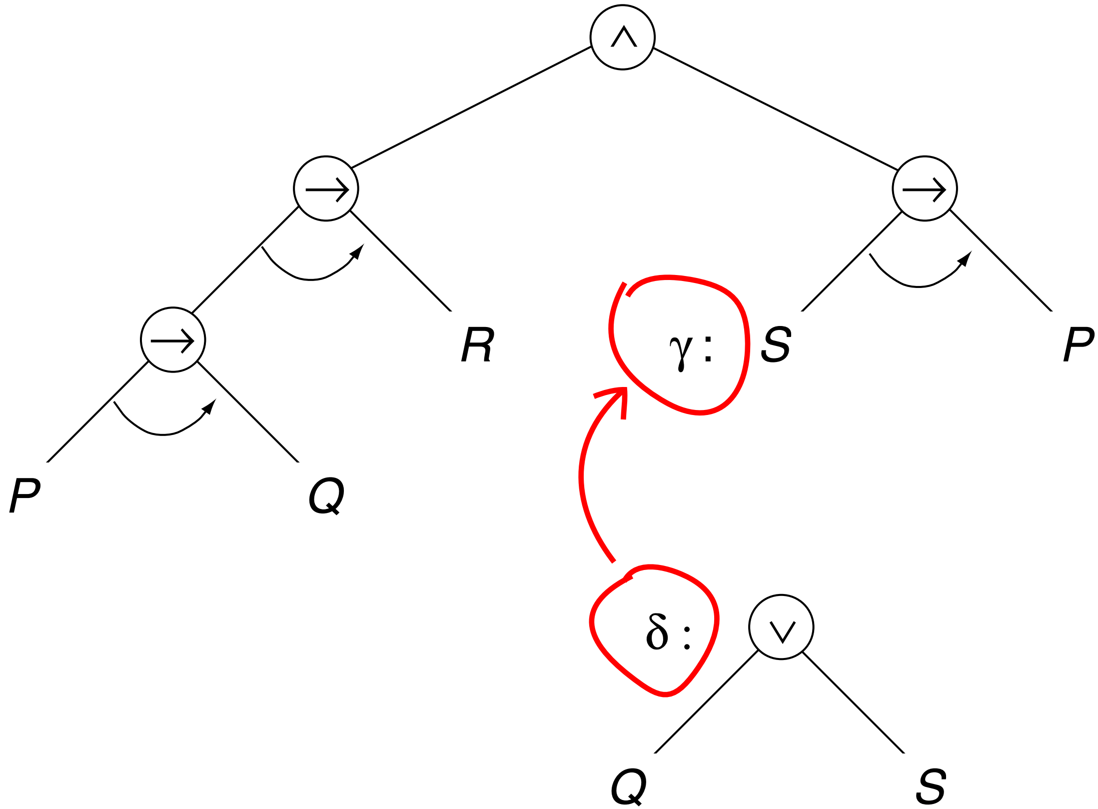
Beispiel:

$\alpha = ((P \to Q) \to R) \land (S \to P)$



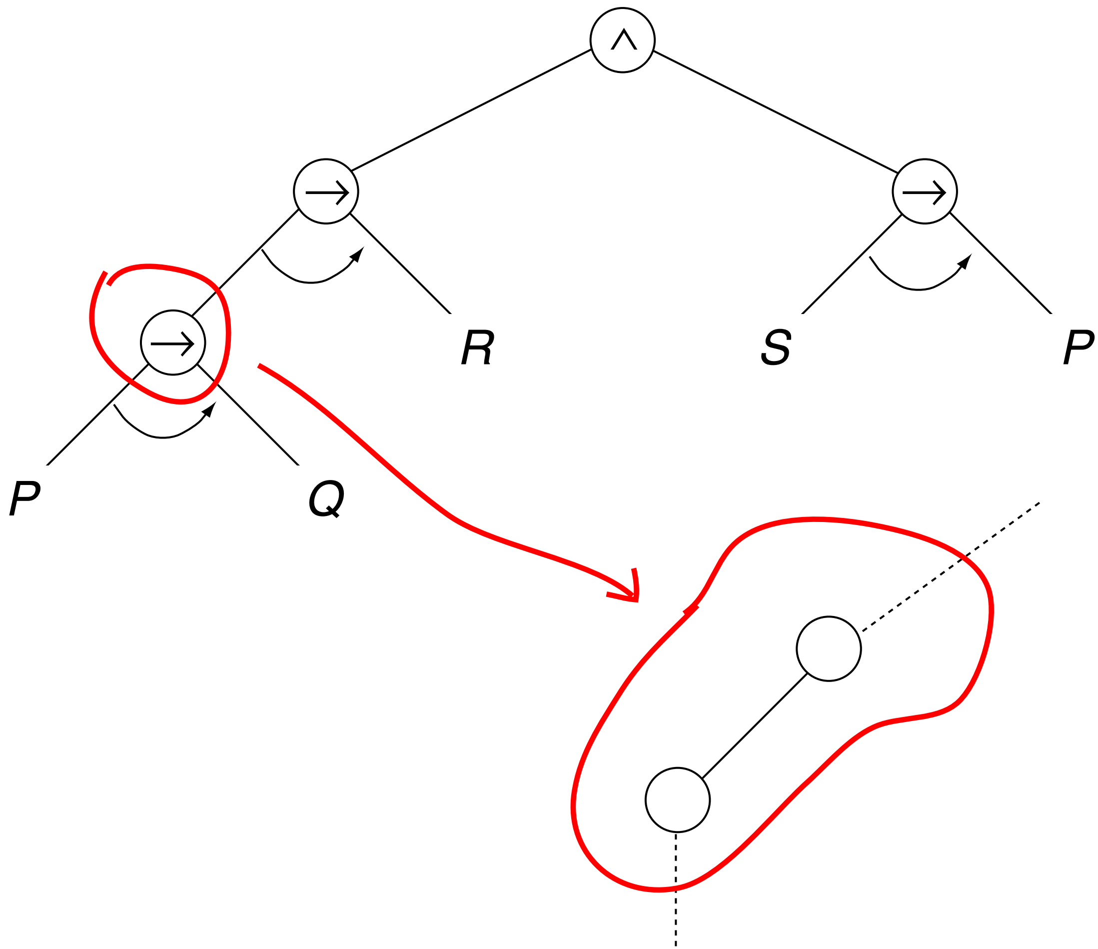
Formeltransformation

Die Ersetzung eines Vorkommens von γ in α durch δ entspricht der Ersetzung eines Blattes (oder Teilbaums) durch einen anderen Baum.



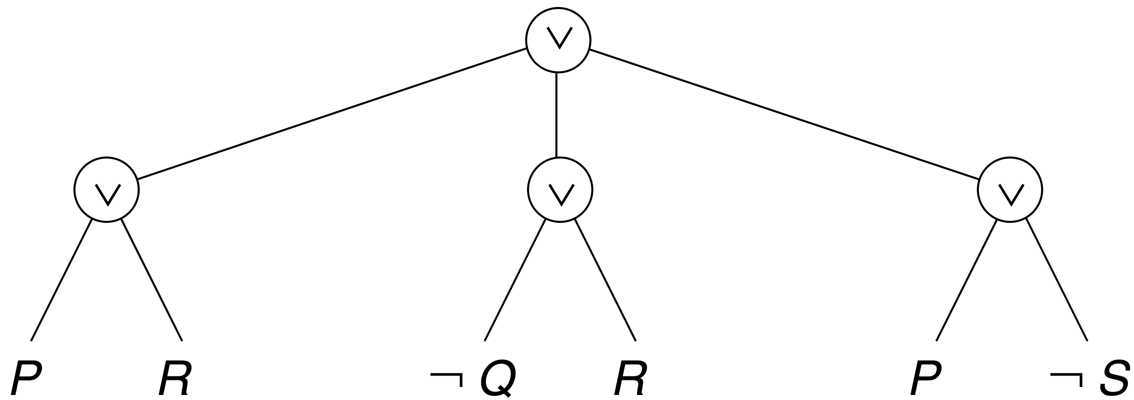
Formeltransformation

Aus Sicht der maschinellen Verarbeitung hätte man gerne kanonische Formeln bzw. Bäume:



Formeltransformation

Aus Sicht der maschinellen Verarbeitung hätte man gerne kanonische Formeln bzw. Bäume:



Bemerkung: $\wedge$ und $\vee$ können als n-äre Knoten aufgefasst werden.

Fragen:

- Was an Kanonisierung ist möglich?
  (unter den verschiedenen Äquivalenzbegriffen)
- Wie operationalisiert (Algorithmus) man Kanonisierung?

Normalformen

Erste Stufe einer Normalisierung:

- Reduzierung der Junktorenmenge.

- Ersetzung von $\rightarrow$, $\leftrightarrow$ entsprechend den Äquivalenzen.

„$\rightarrow$“-Ersetzung: Länge der entstehenden Formel ist . . .

„$\leftrightarrow$“-Ersetzung: Länge der entstehenden Formel ist . . .

Normalformen

Erste Stufe einer Normalisierung:

- Reduzierung der Junktorenmenge.

- Ersetzung von $\rightarrow$, $\leftrightarrow$ entsprechend den Äquivalenzen.

„$\rightarrow$"-Ersetzung: Länge der entstehenden Formel ist … linear in der Ausgangslänge

„$\leftrightarrow$"-Ersetzung: Länge der entstehenden Formel ist … exponentiell in der Ausgangslänge

Normalformen

Definition 19 (Negationsnormalform)

α ist in Negationsnormalform (NNF) genau dann, wenn jedes Negationszeichen in α unmittelbar vor einem Atom steht und α weder den Junktor $\rightarrow$ noch den Junktor $\leftrightarrow$ enthält:

1. Jede Primformel $\alpha = A,\ A \in \Sigma$ ist in NNF.
2. Jede negierte Primformel $\alpha = \neg A,\ A \in \Sigma$ ist in NNF.
3. Sind α, β in NNF, so sind es auch $(\alpha \land \beta)$ und $(\alpha \lor \beta)$.

Lemma 20

Zu jeder Formel α gibt es eine logisch äquivalente Formel β in NNF.

Beweis (Skizze)

Induktion mit Anwendung folgender Regeln:

Negation	$\neg\neg\alpha \approx \alpha$
De Morgan	$\neg(\alpha \land \beta) \approx \neg\alpha \lor \neg\beta$
	$\neg(\alpha \lor \beta) \approx \neg\alpha \land \neg\beta$

Normalformen

Algorithmus: NNF

Input: α. A formula in tree representation w/o $\rightarrow$ and $\leftrightarrow$.
neg. A flag which is initially 'FALSE'.

Output: A formula equivalent to α in NNF.

```
NNF (α, neg)
  IF α = ¬β
  THEN
    IF neg
    THEN RETURN(NNF(β, 'FALSE'))
    ELSE RETURN(NNF(β, 'TRUE'))
  ELSE_IF α = β ∧ γ
    IF neg
    THEN RETURN(NNF(β, 'TRUE') ∨ NNF(γ, 'TRUE'))
    ELSE RETURN(NNF(β, 'FALSE') ∧ NNF(γ, 'FALSE'))
  ELSE_IF α = β ∨ γ
    IF neg
    THEN RETURN(NNF(β, 'TRUE') ∧ NNF(γ, 'TRUE'))
    ELSE RETURN(NNF(β, 'FALSE') ∨ NNF(γ, 'FALSE'))
  ELSE   // alpha is atom.
    IF neg
    THEN RETURN(¬α)
    ELSE RETURN(α)
```

Frage: Wie ist die Laufzeit von `NNF` im $\mathcal{O}$-Kalkül?

Normalformen

Definition 21 (Literal)

Sei $A \in \Sigma$. A, $\neg A$ werden als Literale bezeichnet. Insbesondere heißt A positives Literal und $\neg A$ negatives Literal.

Definition 22 (Klausel)

Seien $L_1, \ldots, L_n$ Literale. Dann heißt $\alpha = L_1 \vee \ldots \vee L_n$ Klausel. Insbesondere heißt α

1. positive Klausel, falls $L_1, \ldots, L_n$ positive Literale sind,
2. negative Klausel, falls $L_1, \ldots, L_n$ negative Literale sind,
3. gemischte Klausel, falls nicht 1. und nicht 2. gilt,
4. Unit-Klausel, falls $n = 1$,
5. k-Klausel, falls $n \leq k$,
6. Krom-Klausel, falls $n \leq 2$,
7. Horn-Klausel, falls maximal ein Literal positiv ist,
8. definite Horn-Klausel, falls genau ein Literal positiv ist.

Normalformen

Definition 23 (Konjunktive Normalform)

Seien $\alpha_1, \ldots, \alpha_n$ Klauseln. Dann heißt $\alpha = \alpha_1 \wedge \ldots \wedge \alpha_n$ Formel in konjunktiver Normalform.

$$\text{KNF} = \{\alpha \mid \alpha \text{ ist in konjunktiver Normalform}\}$$

Weiterhin seien folgende Formelklassen vereinbart:

1. k-KNF. Alle Klauseln sind k-Klauseln (k beliebig aber fest).
2. HORN. Alle Klauseln sind Horn-Klauseln.
3. DHORN. Alle Klauseln sind definite Horn-Klauseln.

Normalformen

Lemma 24 (logisch äquivalente KNF)

Zu jeder Formel α gibt es eine logisch äquivalente Formel $\beta \in$ KNF – in Worten: „β in KNF“.

Beweis (Skizze)

1. Transformation in NNF.
2. Induktion mit Anwendung des Distributiv-Gesetzes:

$$(\alpha \land \beta) \lor \sigma \approx (\alpha \lor \sigma) \land (\beta \lor \sigma)$$

Normalformen

Algorithmus: EQ-CNF
Input: α. A formula in tree representation in NNF.
Output: A formula equivalent to α in CNF.

```
EQ-CNF(α)
  DO
    α_org = α
    IF α = β ∧ γ
    THEN α = EQ-CNF(β) ∧ EQ-CNF(γ)

    ELSE_IF α = (δ ∧ ϵ) ∨ γ
      α = (δ ∨ γ) ∧ (ϵ ∨ γ)
    ELSE_IF α = β ∨ (δ ∧ ϵ)
      α = (β ∨ δ) ∧ (β ∨ ϵ)
    ELSE_IF α = β ∨ γ
      α = EQ-CNF(β) ∨ EQ-CNF(γ)
    ELSE   // α is literal; do nothing.

  WHILE (α_org ≠ α)
  RETURN(α)
```

Bemerkung: Laufzeit und Platzbedarf von EQ-CNF im $\mathcal{O}$-Kalkül sind exponentiell in $|\alpha|$.

Normalformen

Beispiel zum exponentiellen Platzbedarf.

Formel α in NNF:

$$\alpha = \bigvee_{1 \leq i \leq n} (A_{i,1} \wedge A_{i,2})$$

$$\leadsto |\alpha| = 2n$$

Normalformen

Beispiel zum exponentiellen Platzbedarf.

Formel α in NNF:

$$\alpha = \bigvee_{1 \leq i \leq n} (A_{i,1} \wedge A_{i,2})$$

$$\leadsto |\alpha| = 2n$$

KNF zu α:

$$\beta = \bigwedge_{j_1,\ldots,j_n \in \{1,2\}} (A_{1,j_1} \vee \ldots \vee A_{n,j_n})$$

$$\leadsto |\beta| = 2^n \cdot n$$

Normalformen

Beispiel zum exponentiellen Platzbedarf.

Formel α in NNF:

$$\alpha = \bigvee_{1 \leq i \leq n} (A_{i,1} \wedge A_{i,2})$$

$\leadsto |\alpha| = 2n$

KNF zu α:

$$\beta = \bigwedge_{j_1,\ldots,j_n \in \{1,2\}} (A_{1,j_1} \vee \ldots \vee A_{n,j_n})$$

$\leadsto |\beta| = 2^n \cdot n$

Konkret für $n = 3$:

$\alpha = (A_{1,1} \wedge A_{1,2}) \vee (A_{2,1} \wedge A_{2,2}) \vee (A_{3,1} \wedge A_{3,2})$ bzw.

$\alpha = (A_1 \wedge A_2) \vee (A_3 \wedge A_4) \vee (A_5 \wedge A_6)$

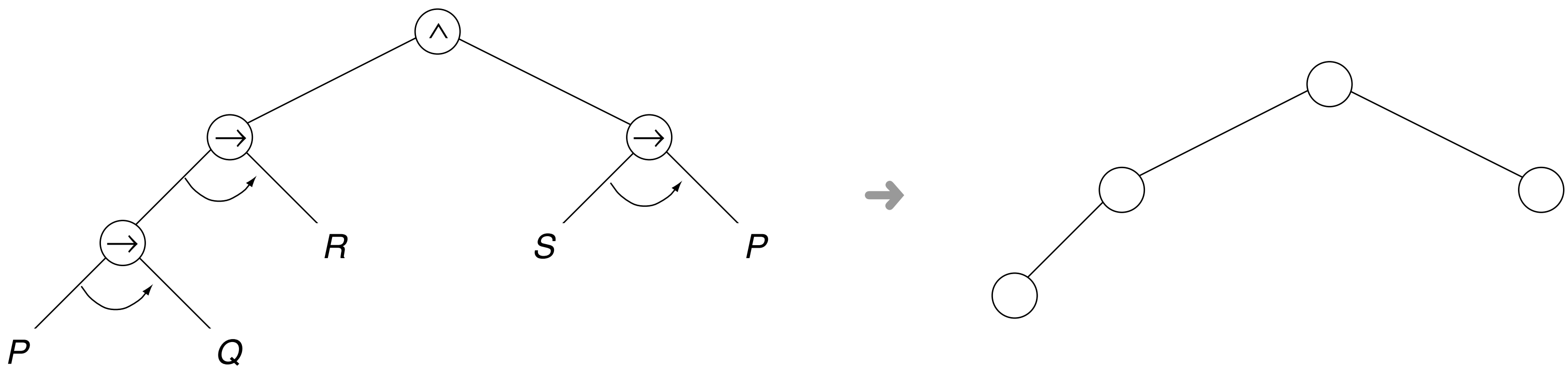
$\leadsto \beta = (A_{1,1} \vee A_{2,1} \vee A_{3,1}) \wedge \ldots \wedge (A_{1,2} \vee A_{2,2} \vee A_{3,2})$ mit $|\beta| = 8 \cdot 3$

Formeltransformation nach Tseitin

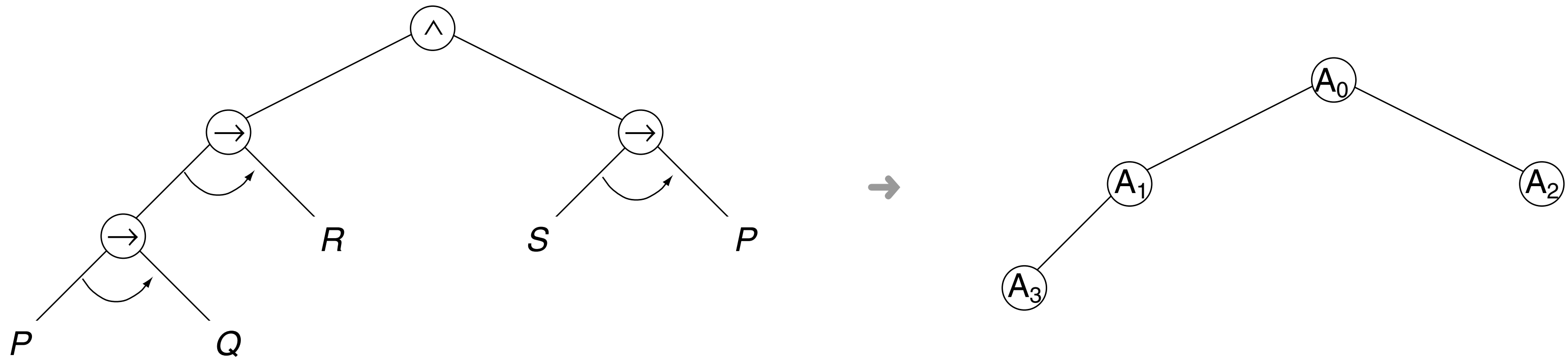
Andere Idee zur Erzeugung einer KNF aus α:

1. Beschreibung der *Formelstruktur* von α mit Hilfe einer neuen Formel, die erfüllbarkeitsäquivalent – aber nicht logisch äquivalent ist.
2. Umwandlung der neuen Formel in eine KNF in $\mathcal{O}(|\alpha|)$.

Beispiel: $((P \to Q) \to R) \land (S \to P)$



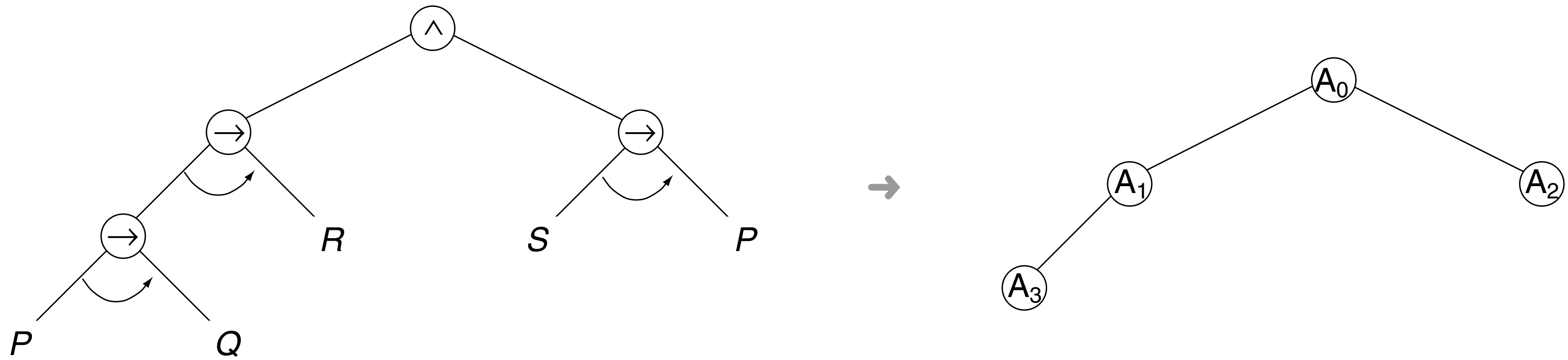
Formeltransformation nach Tseitin



Schritte:

1. Ersetzung der inneren Knoten durch neue Atome A_0, A_1, A_2, A_3.

Formeltransformation nach Tseitin



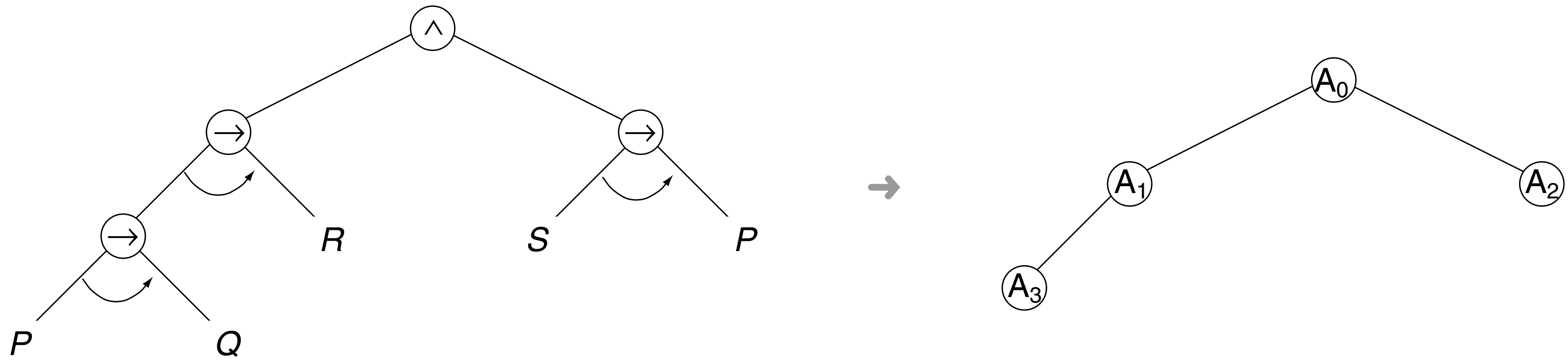
Schritte:

1. Ersetzung der inneren Knoten durch neue Atome A_0, A_1, A_2, A_3.

2. Einführung von Äquivalenzen für Zusammenhang:

$$\begin{array}{lcl} A_0 & \leftrightarrow & (A_1 \wedge A_2) \\ A_1 & \leftrightarrow & (A_3 \rightarrow R) \\ A_2 & \leftrightarrow & (S \rightarrow P) \\ A_3 & \leftrightarrow & (P \rightarrow Q) \end{array}$$

Formeltransformation nach Tseitin



Schritte:

1. Ersetzung der inneren Knoten durch neue Atome A_0, A_1, A_2, A_3.

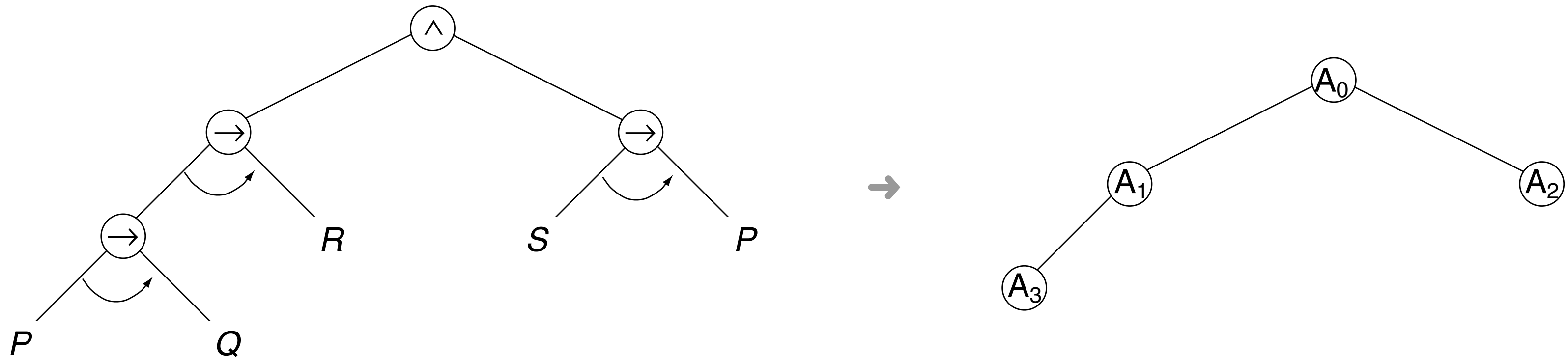
2. Einführung von Äquivalenzen für Zusammenhang:

$$\begin{array}{lcl} A_0 & \leftrightarrow & (A_1 \wedge A_2) \\ A_1 & \leftrightarrow & (A_3 \rightarrow R) \\ A_2 & \leftrightarrow & (S \rightarrow P) \\ A_3 & \leftrightarrow & (P \rightarrow Q) \end{array}$$

3. Konjunktive Verknüpfung von Äquivalenzen und Gesamtformelrepräsentant:

$$(A_0 \leftrightarrow (A_1 \wedge A_2)) \quad \wedge \quad (A_1 \leftrightarrow (A_3 \rightarrow R)) \quad \wedge$$
$$(A_2 \leftrightarrow (S \rightarrow P)) \quad \wedge \quad (A_3 \leftrightarrow (P \rightarrow Q)) \quad \wedge \quad A_0$$

Formeltransformation nach Tseitin



Schritte:

1. Ersetzung der inneren Knoten durch neue Atome A_0, A_1, A_2, A_3.

2. Einführung von Äquivalenzen für Zusammenhang:

$$\begin{array}{lll} A_0 & \leftrightarrow & (A_1 \land A_2) \\ A_1 & \leftrightarrow & (A_3 \rightarrow R) \\ A_2 & \leftrightarrow & (S \rightarrow P) \\ A_3 & \leftrightarrow & (P \rightarrow Q) \end{array}$$

3. Konjunktive Verknüpfung von Äquivalenzen und Gesamtformelrepräsentant:
$$(A_0 \leftrightarrow (A_1 \land A_2)) \quad \land \quad (A_1 \leftrightarrow (A_3 \rightarrow R)) \quad \land$$
$$(A_2 \leftrightarrow (S \rightarrow P)) \quad \land \quad (A_3 \leftrightarrow (P \rightarrow Q)) \quad \land \quad A_0$$

4. Expansion (Transformation in KNF) der Äquivalenzen:
$$\alpha \leftrightarrow \beta \approx (\alpha \rightarrow \beta) \land (\beta \rightarrow \alpha) \approx (\neg\alpha \lor \beta) \land (\neg\beta \lor \alpha)$$

Bemerkungen:

- Wenn die Namenslänge der Atome durch eine Konstante beschränkt ist, so ist Schritt 4 (Expansion) pro Äquivalenz in konstanter Zeit und konstantem Platz möglich, da $|\beta| = 2$.

Formeltransformation nach Tseitin

Lemma 25 (erfüllbarkeitsäquivalente KNF)

Zu jeder aussagenlogischen Formel α gibt es eine erfüllbarkeitsäquivalente Formel $\beta \in$ KNF, wobei gilt:

1. $|\beta| \in \mathcal{O}(|\alpha|)$ Platz.

2. β ist in $\mathcal{O}(|\alpha|)$ Schritten aus der Baumdarstellung von α berechenbar. Generierung neuer Namen zählt als ein Schritt.

Bemerkungen:

- Das Lemma geht hinsichtlich der Namensgenerierung von einer konstanten Namenslänge $|A_i| = c$ aus. Jedoch – wenn n verschiedene Namen zu generieren sind, müßte genau genommen $|A_i| = \text{log}(n)$ unterstellt werden. Mit $n \in \mathcal{O}(|\alpha|)$ würde die Berechnung von β also den Aufwand $\mathcal{O}(|\alpha| \cdot \text{log}(|\alpha|))$ erfordern.

- Warum ist die im Lemma gemachte Vereinfachung bedenkenlos akzeptierbar?

Normalformen

Frage: Gegeben eine Formel $\in$ KNF. Inwieweit lassen sich die Klausellängen unter Beibehaltung der logischen Äquivalenz reduzieren?

Lemma 26 (Klausellänge $\approx$)

Für alle $k \geq 1$ gibt es ein $\alpha \in (k+1)$-KNF, so dass kein $\beta \in k$-KNF existiert mit

$$\alpha \approx \beta$$

Beweis

Wähle $\alpha = A_1 \vee \ldots \vee A_{k+1}$

$\beta = \beta_1 \wedge \ldots \wedge \beta_m$, sei $\beta \in k$-KNF; d. h., die β_i sind k-Klauseln

o.b.d.A. gelte: $A_{k+1} \notin$ *atoms*(β_1)

Setze $\mathcal{I}$ so, dass $\mathcal{I}(\beta_1) = 0$ und $\mathcal{I}(A_{k+1}) = 1$

$\Rightarrow \mathcal{I}(\alpha) = 1 \neq 0 = \mathcal{I}(\beta)$

Normalformen

Frage: Gegeben eine Formel $\in$ KNF. Inwieweit lassen sich die Klausellängen unter Beibehaltung der Erfüllbarkeitsäquivalenz reduzieren?

Lemma 27 (Klausellänge $\approx_{\text{sat}}$)

Für jede Formel $\alpha \in$ KNF existiert eine erfüllbarkeitsäquivalente Formel $\beta \in 3$-KNF.

Beweis (Skizze)

Sei $\alpha_i = L_1 \vee \ldots \vee L_n$ eine Klausel aus α mit n Literalen, $n \geq 4$, und seien $A_0, \ldots, A_{n-4} \notin$ *atoms*(α). Dann setze:

$$\begin{aligned}
\beta_0 &= L_1 \vee L_2 \vee A_0 \\
\beta_1 &= \neg A_0 \vee L_3 \vee A_1 \\
&\vdots \\
\beta_{n-4} &= \neg A_{n-5} \vee L_{n-2} \vee A_{n-4} \\
\beta_{n-3} &= \neg A_{n-4} \vee L_{n-1} \vee L_n
\end{aligned}$$

$\Rightarrow \quad \alpha_i \approx_{\text{sat}} \beta_0 \wedge \ldots \wedge \beta_{n-3}$

Bemerkungen:

- Keine Reduzierung auf Klausellängen unter 3 ist möglich bei einer Ausgangslänge der Klauseln von $k \geq 3$.

Normalformen

Definition 28 (disjunktive Normalform)

Sei $\alpha = \alpha_1 \vee \ldots \vee \alpha_n$ eine Disjunktion von Klauseln der Form $\alpha_i = L_{i,1} \wedge \ldots \wedge L_{i,m}$ mit den Literalen $L_{i,1}, \ldots, L_{i,m}$. Dann heißt α Formel in disjunktiver Normalform.

$$\text{DNF} = \{\alpha \mid \alpha \text{ ist in disjunktiver Normalform}\}$$

Lemma 29 (NNF einer Formel in KNF)

Sei $\alpha \in$ KNF, dann ist NNF($\neg\alpha$) $\in$ DNF.

Beweis (Skizze)

Induktion mit Anwendung folgender Regeln:

Negation	$\neg\neg\alpha \approx \alpha$
De Morgan	$\neg(\alpha \wedge \beta) \approx \neg\alpha \vee \neg\beta$
	$\neg(\alpha \vee \beta) \approx \neg\alpha \wedge \neg\beta$

Bemerkungen:

- Bei Formeln in disjunktiver Normalform ist der Klauselbegriff eher selten. Manche Autoren verwenden den Begriff „Monom“ in diesem Zusammenhang.
- Betrachten Sie die Normalformen KNF und DNF. Welche Normalform würden Sie wählen, um zu überprüfen, ob eine Formel tautologisch bzw. widerspruchsvoll ist?

Normalformen

Definition 30 (duale Formel)

Sei $\alpha \in$ NNF. Dann ist die duale Formel zu α, *dual*(α), wie folgt definiert:

1. *dual*$(A) := A$
2. *dual*$(\neg A) := \neg A$
3. *dual*$(\alpha \vee \beta) :=$ *dual*$(\alpha) \wedge$ *dual*(β)
4. *dual*$(\alpha \wedge \beta) :=$ *dual*$(\alpha) \vee$ *dual*(β)

Beispiel:

$\alpha = (A \wedge B) \vee (A \wedge (B \vee C))$

dual$(\alpha) = (A \vee B) \wedge (A \vee (B \wedge C))$

Normalformen

Lemma 31 (duale Formeln und Erfüllbarkeit)

Sei $\alpha \in$ KNF. Dann gilt:

1. *dual*$(\alpha) \in$ DNF
2. α tautologisch $\Leftrightarrow$ *dual*(α) widerspruchsvoll
3. α erfüllbar $\Leftrightarrow$ *dual*(α) falsifizierbar

Beweis (Skizze)

Sei $\mathcal{I}$ eine Bewertung.

Definiere: $\mathcal{I}_{dual}(A) := 1 - \mathcal{I}(A)$, für alle $A \in \Sigma$

Es folgt mit Induktion über den Aufbau von α in NNF (hier ohne Erläuterung):

$$\mathcal{I}_{dual}(\textit{dual}(\alpha)) = 1 - \mathcal{I}(\alpha)$$

Normalformen

Definition 32 (Mengenschreibweise für Formeln in KNF)

Für eine Klausel $L_1 \vee \ldots \vee L_m$ sei folgende Mengenschreibweise vereinbart

$$\{L_1, \ldots, L_m\}$$

Für eine Formel $\alpha_1 \wedge \ldots \wedge \alpha_n$ sei folgende Mengenschreibweise vereinbart

$$\{\alpha_1, \ldots, \alpha_n\}$$

Beispiel:

$\alpha = A \wedge (B \vee \neg C) \wedge (\neg A \vee C \vee D)$

$\leadsto \alpha = \{\{A\}, \{B, \neg C\}, \{\neg A, C, D\}\}$

Bemerkungen:

- Bei der Mengenschreibweise wird implizit eine Reduktion der Formel auf Basis folgender Äquivalenzen vorgenommen:
  1. $\alpha \approx \alpha \land \alpha$
  2. $\alpha \approx \alpha \lor \alpha$
  3. $\alpha \land \beta \approx \beta \land \alpha$
  4. $\alpha \lor \beta \approx \beta \lor \alpha$
  5. $(\alpha \land \beta) \land \gamma \approx \alpha \land (\beta \land \gamma)$
  6. $(\alpha \lor \beta) \lor \gamma \approx \alpha \lor (\beta \lor \gamma)$
- Mit Hilfe von Mengenoperatoren kann man Begiffe wie *Teilklausel* sehr einfach auf der Mengendarstellung definieren.
- In welcher Zeit ist die Transformation einer KNF als Menge durchführbar?