Eliminationsmethoden auf dem dualen Berechnungsgraphen

Marc Ensenbach

15.08.2006

Sommerschule des Graduiertenkollegs
„Hierarchie und Symmetrie in mathematischen Modellen“
Universitätsakademie Bommerholz

Übersicht

Motivation

Beispiel in diesem Abschnitt:

verschiedene Berechnungsverfahren für die Ableitung, größtenteils auf Basis der automatischen Differentiation

zum Vergleich benötigt: Komplexitätsmaß

hier sinnvoll: Anzahl der Berechnungen der Form

$$a + bc,$$

der *affin-linearen Transformationen* (ALTs)

Motivation

Ziel: Berechnung der Ableitungsmatrix der Funktion $F : \mathbb{R}^2 \to \mathbb{R}^2$,

$$F((x_1, x_2)) = (\cos(x_1 x_2 \sin(x_1 x_2)), \exp(x_1 x_2 \sin(x_1 x_2)))$$

in einem Punkt $x_0 \in \mathbb{R}^2$.

für alle AD-Ansätze benötigt: Kodeliste

Motivation

Ausgangsfunktion:

$$F((x_1, x_2)) = (\cos(x_1 x_2 \sin(x_1 x_2)), \exp(x_1 x_2 \sin(x_1 x_2)))$$

Kodeliste: (die „Eingabevariablen“ x_1 und x_2 werden aus Konsistenzgründen als v_{-1} und v_0 geschrieben)

$$\begin{aligned}
v_1 &= v_{-1} v_0 \\
v_2 &= \sin v_1 \\
v_3 &= v_1 v_2 \\
v_4 &= \cos v_3 \\
v_5 &= \exp v_3
\end{aligned}$$

Motivation

Annahme für Komplexitätsvergleich: partielle Ableitungen der Terme der Kodeliste in x_0 bereits bekannt $(c_1, \ldots, c_7)$

Vergleich einiger Verfahren:

- direkte Berechnung von DF: Man erhält (bis auf Umnumerierungen der Indizes)

$$\mathrm{D}F = \begin{pmatrix} c_1 c_3 c_6 + c_1 c_4 c_5 c_6 & c_2 c_3 c_6 + c_2 c_4 c_5 c_6 \\ c_1 c_3 c_7 + c_1 c_4 c_5 c_7 & c_2 c_3 c_7 + c_2 c_4 c_5 c_7 \end{pmatrix}$$

  Aufwand: 20 ALTs
- AD im Vorwärts- oder Rückwärts-Modus: 14 ALTs
- AD mit Ausnutzung der Dünnbesetztheit: 12 ALTs

Motivation

Vergleich einiger Verfahren (Fortsetzung):

- Umformung von DF:

$$\mathrm{D}F = \begin{pmatrix} c_1c_3c_6 + c_1c_4c_5c_6 & c_2c_3c_6 + c_2c_4c_5c_6 \\ c_1c_3c_7 + c_1c_4c_5c_7 & c_2c_3c_7 + c_2c_4c_5c_7 \end{pmatrix}$$

$$= \begin{pmatrix} c_6 s & c_6 t \\ c_7 s & c_7 t \end{pmatrix}$$

mit $r = c_3 + c_5c_4$ sowie $s = rc_1$ und $t = rc_2$.

Aufwand: 7 ALTs

Motivation

Ziel

Automatisierung dieser Art von Umformung

Hilfsmittel: der Berechnungsgraph und der dazu duale Graph

Übersicht

Definition: Berechnungsgraph

Definition

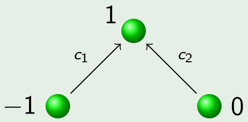
Sei $v_j = \varphi_j((v_i)_{i \prec j})$, $1 \leqslant j \leqslant l$ eine Kodeliste (dabei bedeute $i \prec j$, daß v_j direkt von v_i abhängt). Der *Berechnungsgraph* dieser Kodeliste ist der bogenbewertete gerichtete kreisfreie Graph (GKG) $G = (V, K, b)$ mit

- $V = \{-n+1, \ldots, 0, 1, \ldots, l\}$,
- $K = \{(i,j) \mid i \prec j\}$,
- $b : K \to \mathbb{R}$, $(i,j) \mapsto \dfrac{\partial}{\partial v_i}\varphi_j((v_k)_{k \prec j})$.

Berechnungsgraph – Beispiel

Beispiel

$$v_1 = v_{-1} v_0$$

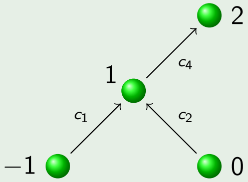


Berechnungsgraph – Beispiel

Beispiel

$$v_1 = v_{-1} v_0$$

$$v_2 = \sin v_1$$



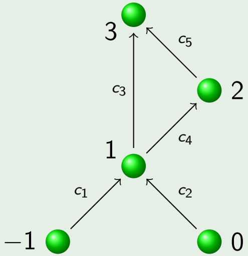
Berechnungsgraph – Beispiel

Beispiel

$$v_1 = v_{-1} v_0$$

$$v_2 = \sin v_1$$

$$v_3 = v_1 v_2$$



Berechnungsgraph – Beispiel

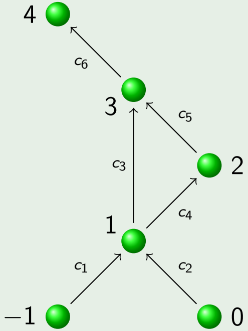
Beispiel

$$v_1 = v_{-1} v_0$$

$$v_2 = \sin v_1$$

$$v_3 = v_1 v_2$$

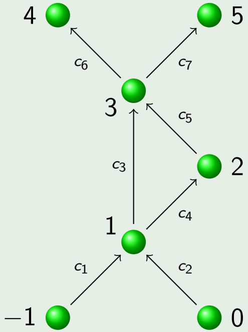
$$v_4 = \cos v_3$$



Berechnungsgraph – Beispiel

Beispiel

$$v_1 = v_{-1} v_0$$
$$v_2 = \sin v_1$$
$$v_3 = v_1 v_2$$
$$v_4 = \cos v_3$$
$$v_5 = \exp v_3$$



Berechnungsgraph – Eigenschaften

Der Berechnungsgraph bietet die Möglichkeit der Berechnung von partiellen Ableitungen der Form

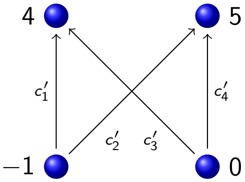
$$\frac{\partial v_j}{\partial v_i},$$

nämlich gemäß Kettenregel wie folgt:

1. Bilde für jeden Weg von i nach j das Produkt über die Bogenbewertungen.
2. Summiere diese Produkte auf.

Ziel

Ein Berechnungsgraph der Form



hat als Bogenbewertungen die Einträge der Ableitungsmatrix.

Gesucht:

Folge von Transformationen, die den Berechnungsgraph in die obige Form überführt

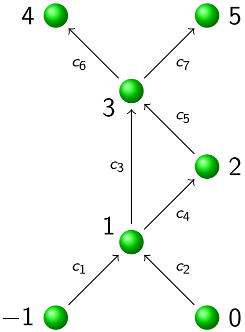
Dualer Berechnungsgraph – Einführung

„Transformationen“: Eliminationen

Eliminationen teilweise besser beschreibbar im *dualen Berechnungsgraphen (DBG)*

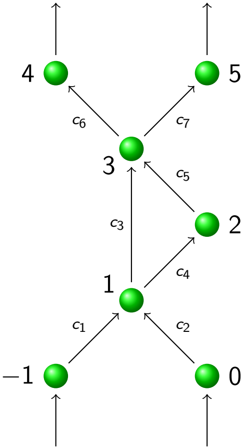
hier keine formale Definition, sondern Konstruktion am Beispiel

Dualer Berechnungsgraph – Konstruktion



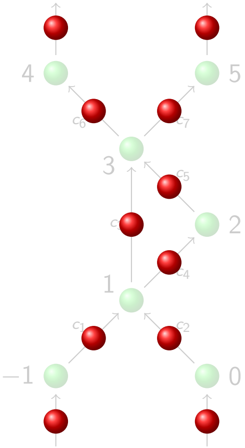
Dualer Berechnungsgraph – Konstruktion

1. Ergänze Eingangs- und Ausgangspfeile.



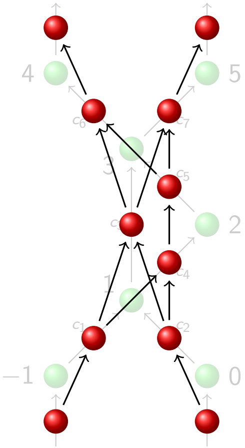
Dualer Berechnungsgraph – Konstruktion

1. Ergänze Eingangs- und Ausgangspfeile.
2. Ersetze Pfeile durch Knoten.



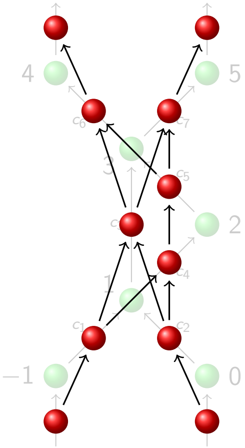
Dualer Berechnungsgraph – Konstruktion

1. Ergänze Eingangs- und Ausgangspfeile.
2. Ersetze Pfeile durch Knoten.
3. Verbinde neue Knoten durch Bogen, wenn entsprechende Pfeile direkt aufeinander folgen.



Dualer Berechnungsgraph – Konstruktion

1. Ergänze Eingangs- und Ausgangspfeile.
2. Ersetze Pfeile durch Knoten.
3. Verbinde neue Knoten durch Bogen, wenn entsprechende Pfeile direkt aufeinander folgen.
4. Bogenbewertungen werden zu Knotenbewertungen (Anfangs- und Endknoten erhalten Bewertung 1).



Dualer Berechnungsgraph – Eigenschaften

Abbildung von Berechnungsgraph auf dualen Berechnungsgraph ist injektiv (dazu benötigt: Eingangs- und Ausgangspfeile)

Ansatz für Rekonstruktion des Berechnungsgraphen aus dem DBG:

1. Ersetze innere Knoten im DBG durch zwei Knoten, die durch einen Bogen verbunden sind.
2. Zwei neue Knoten $\tilde{v}$ und $\tilde{w}$ seien äquivalent, wenn
   - $\tilde{v}$ Anfangspunkt eines neuen Bogens und $\tilde{w}$ Endpunkt eines weiteren neuen Bogens ist und
   - die den neuen Bögen entsprechenden Knoten im DBG durch einen Bogen verbunden sind.
3. Äquivalenzklassen entsprechen Knoten im Berechnungsgraphen. Streiche gegebenenfalls parallele Bögen.

Dualer Berechnungsgraph – Eigenschaften

Berechnung partieller Ableitungen: Summe von Produkten über *Knoten*bewertungen von Wegen

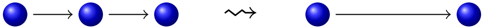
(Dies folgt aus der entsprechenden Eigenschaft des Berechnungsgraphen und der Konstruktion des DBG.)

Übersicht

Elimination – Vorüberlegungen

Ziel war: „Vereinfachung“ des Berechnungsgraphen

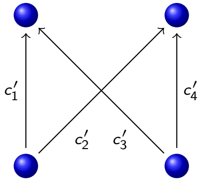
Methode: Elimination von *transitiven Abhängigkeiten:*



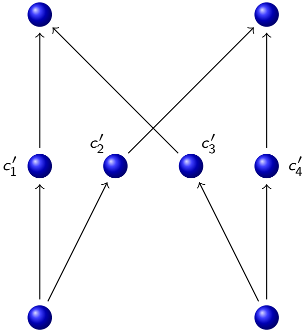
transitive Abhängigkeiten im Berechnungsgraphen $\widehat{=}$ Bögen im DBG

Elimination im DBG – Vorüberlegungen

Zielform Berechnungsgraph



Zielform DBG



Elimination im DBG – Vorüberlegungen/1. Schritt

Aufgabe

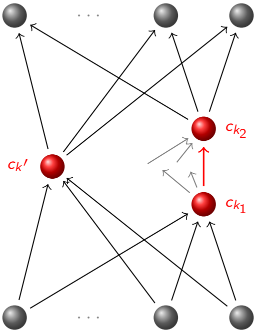
Eliminiere im DBG alle *inneren* Bögen.

für zu eliminierenden Bogen (k_1, k_2) im DBG zwei Fälle möglich:

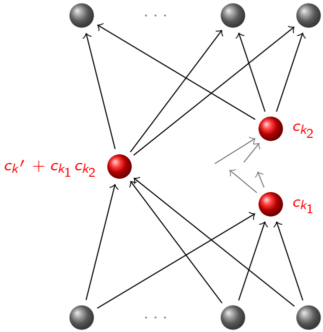
1. Es gibt einen Knoten k' im DBG, der die gleiche *Vorgängermenge* wie k_1 und die gleiche *Nachfolgermenge* wie k_2 hat.
2. Es gibt keinen solchen Knoten.

Elimination im DBG – 1. Schritt, 1. Fall

Elimination des Bogens (k_1, k_2):



$\rightsquigarrow$



Elimination im DBG – 1. Schritt, 1. Fall

weiterhin korrekte Berechnung partieller Ableitungen:

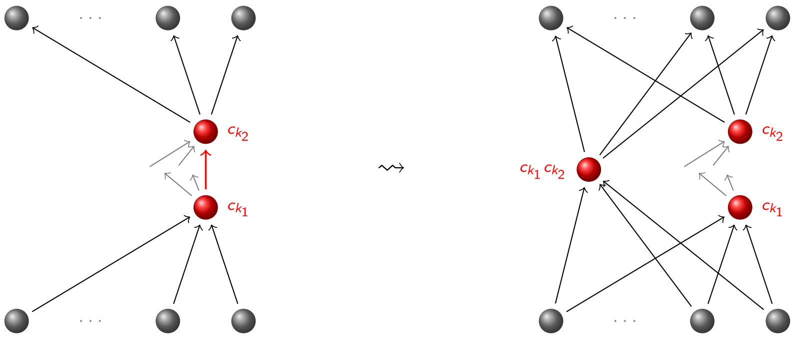
- Bei Wegen, die weder durch k' noch durch k_1 und k_2 verliefen, ändert sich das Produkt über den Weg nicht.
- Im DBG gibt es zu Wegen der Form $d_1, \ldots, d_i, k', d_{i+1}, \ldots, d_l$ immer einen Weg der Form $d_1, \ldots, d_i, k_1, k_2, d_{i+1}, \ldots, d_l$ und umgekehrt, also hat man in der Summe über alle Produkte entlang solcher Wege nach Zusammenfassen stets Ausdrücke der Form

$$(b(k') + b(k_1)b(k_2)) \cdot b(d_1) \cdots b(d_l)$$

stehen. Nach Streichen von (k_1, k_2) und Anpassen der Bewertung von k' ändert sich der Wert hier also auch nicht.

Elimination im DBG – 1. Schritt, 2. Fall

Einfügen eines neuen Knotens und Elimination des Bogens (k_1, k_2):



Elimination im DBG – 1. Schritt, 2. Fall

weiterhin korrekte Berechnung partieller Ableitungen:

- Bei Wegen, die nicht durch k_1 und k_2 verliefen, ändert sich das Produkt über den Weg nicht.
- Der neue Knoten sei k'. Zu Wegen der Form $d_1, \ldots, d_i, k_1, k_2, d_{i+1}, \ldots, d_l$ im DBG gibt es immer einen Weg der Form $d_1, \ldots, d_i, k', d_{i+1}, \ldots, d_l$ im neuen Graphen und umgekehrt. Nach Streichen von (k_1, k_2) und Setzen der Bewertung von k' ändert sich die Summe der Produkte hier also auch nicht.

Elimination im DBG – 2. Schritt

für Knoten k_1 drei Fälle möglich:

1. k_1 ist *isoliert*: k_1 hat keine eingehenden oder keine ausgehenden Bögen.
2. Es gibt einen Knoten k_1' im DBG, der gleiche Vorgänger- und gleiche Nachfolgermenge hat wie k_1.
3. Keiner der obigen Fälle tritt auf. (Hier ist nichts weiter zu tun.)

(Verfahre im nächsten Schritt analog mit k_2.)

Elimination im DBG – 2. Schritt, 1. Fall

k_1 ist isoliert.

Entferne k_1 und alle anliegenden Bögen aus dem Graphen.

Die Berechnung partieller Ableitungen ist weiterhin korrekt, da es keine Wege mit k_1 als innerem Punkt geben konnte.

Elimination im DBG – 2. Schritt, 2. Fall

Es gibt einen Knoten k_1' im DBG, der gleiche Vorgänger- und gleiche Nachfolgermenge hat wie k_1.

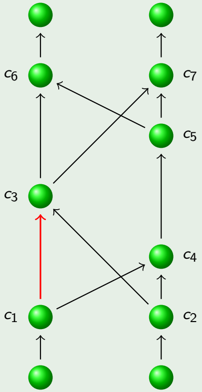
Verschmelze k_1 und k_1':

- Erhöhe Bewertung von k_1 um die Bewertung von k_1'.
- Entferne k_1' und alle anliegenden Bögen aus dem Graphen.

Die Berechnung partieller Ableitungen ist weiterhin korrekt: Es gibt Bijektion zwischen den Wegen durch k_1 und den Wegen durch k_1'; Begründung analog zur Bogen-Knoten-Verschmelzung im ersten Schritt.

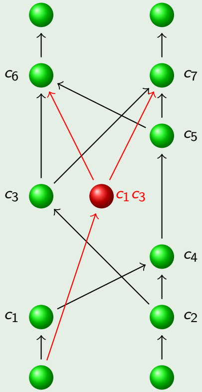
Elimination im DBG – Beispiel

Beispiel



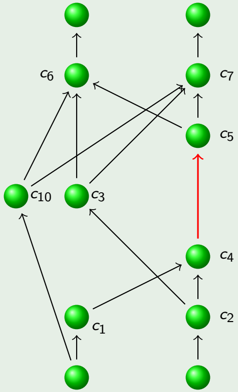
1. Schritt
2. Fall

$\rightsquigarrow$

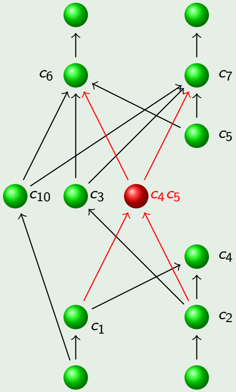


Elimination im DBG – Beispiel

Beispiel

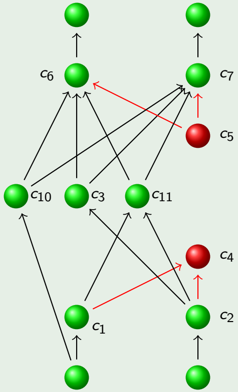


1. Schritt
2. Fall
$\rightsquigarrow$

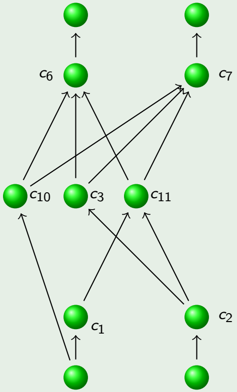


Elimination im DBG – Beispiel

Beispiel

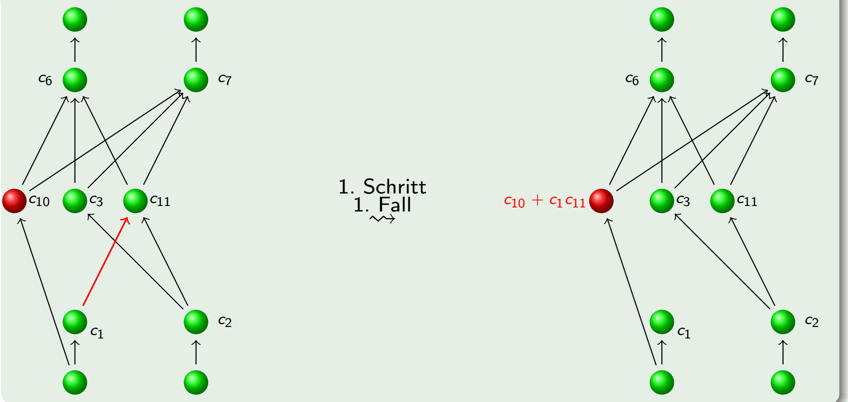


2. Schritt/
3. Schritt $\rightsquigarrow$



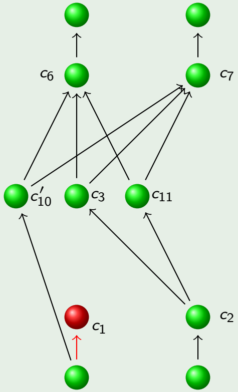
Elimination im DBG – Beispiel

Beispiel

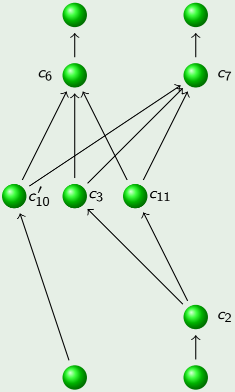


Elimination im DBG – Beispiel

Beispiel

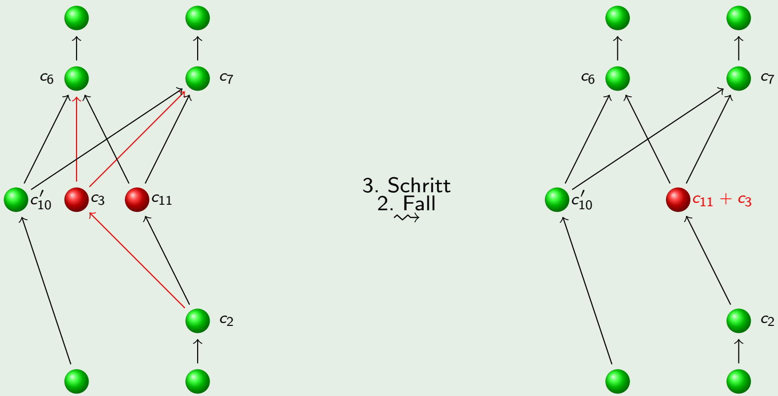


2. Schritt
1. Fall
$\rightsquigarrow$



Elimination im DBG – Beispiel

Beispiel



Elimination im DBG – Analyse

Satz

Jede Folge von solchen Transformationen ist endlich.

Beweisansatz:

Nach jeder Transformation wird die Summe der Längen aller Wege zwischen Anfangs- und Endknoten echt kleiner:

- klar bei Verschmelzung von Bogen mit Knoten, da Bogen ersatzlos gelöscht wird
- bei Einfügen eines neuen Knotens und Löschung des Bogens: Wege durch k_1 und k_2 werden durch Wege durch k' ersetzt $\Rightarrow$ Weglänge sinkt jeweils um Eins.

Elimination im DBG – Analyse

Satz

Jede Folge von Transformationen läßt sich so fortsetzen, daß der erhaltene Graph die gewünschte DBG-Zielform hat.

Beweisidee:

Die folgenden Eigenschaften sind invariant unter den Transformationen:

- Es gibt keine direkten Verbindungen von Anfangs- und Endknoten.
- Nachfolger von Anfangsknoten haben Eingangsgrad 1.
- Vorgänger von Endknoten haben Ausgangsgrad 1.

Solange es noch innere Bögen gibt, läßt sich stets noch ein Eliminationsschritt durchführen.

Elimination im DBG – Analyse

Satz

Die Knotenbewertungen im DBG in der Zielform sind genau die Einträge der Ableitungsmatrix.

Beweis:

bei der Konstruktionsvorschrift bereits gezeigt:
Berechnungsvorschrift für partielle Ableitungen ist invariant unter den Transformationen. Die Aussage folgt dann aus der Struktur der Zielform des DBG.

Elimination im Berechnungsgraphen – Vorüberlegungen

Bogenelimination im DBG muß nicht wieder einen DBG als Resultat haben.

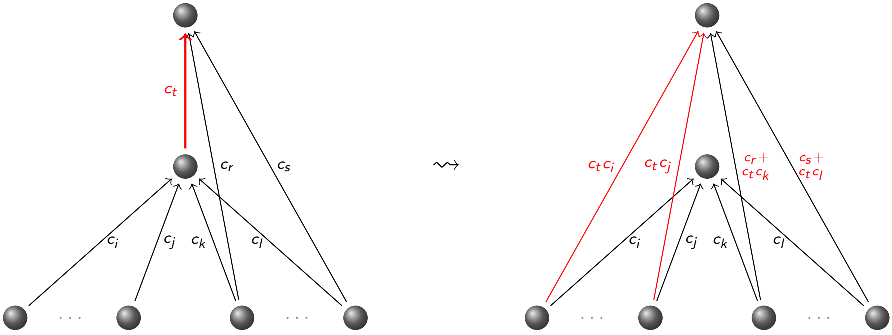
aber:

- Werden im DBG alle Ausgangsbögen eines Knotens eliminiert, erhält man wieder einen DBG.
- Werden im DBG alle Eingangsbögen eines Knotens eliminiert, erhält man wieder einen DBG.

Diese *Front-* bzw. *Rückelimination* des entsprechenden Bogens im Berechnungsgraphen kann dort direkt beschrieben werden.

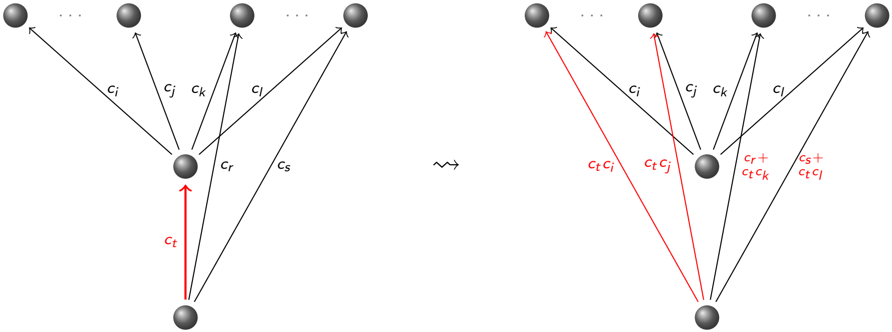
Elimination im Berechnungsgraphen – Rückelimination

Rückelimination des Bogens (v, w):



Elimination im Berechnungsgraphen – Frontelimination

Frontelimination des Bogens (v, w):



Elimination im Berechnungsgraphen – Knotenelimination

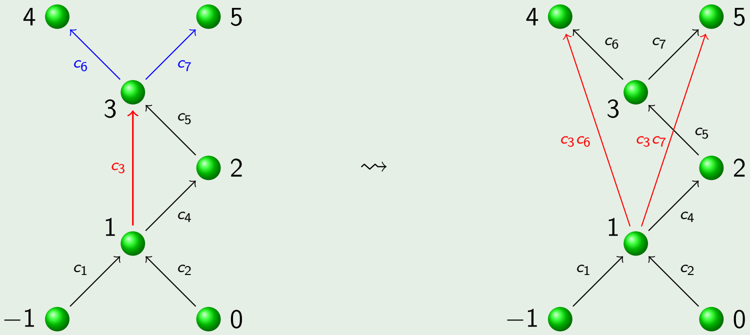
Gegebenenfalls dadurch entstehende isolierte Knoten werden mit allen anliegenden Bögen aus dem Graphen entfernt.

dadurch zwei Knoteneliminationsverfahren:

- Führe für einen Knoten Frontelimination für alle Eingangsbögen aus.
- Führe für einen Knoten Rückelimination für alle Ausgangsbögen aus.

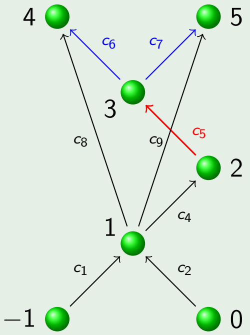
Elimination im Berechnungsgraphen – Beispiel

Beispiel

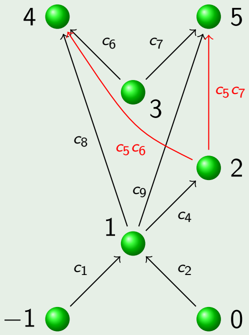


Elimination im Berechnungsgraphen – Beispiel

Beispiel

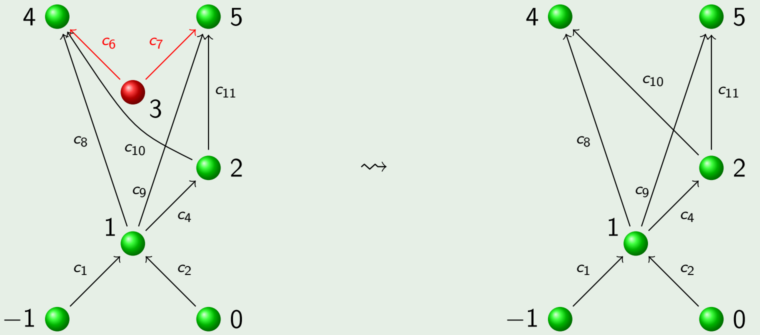


$\rightsquigarrow$



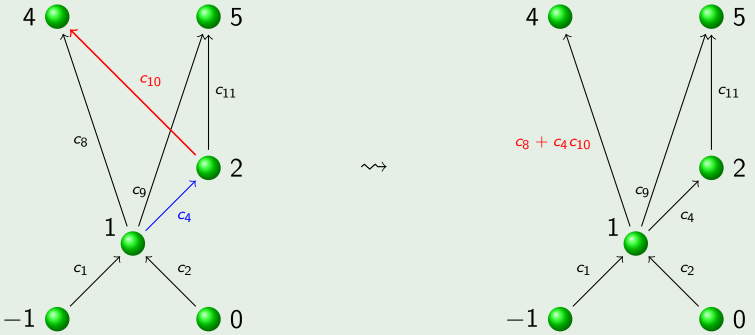
Elimination im Berechnungsgraphen – Beispiel

Beispiel



Elimination im Berechnungsgraphen – Beispiel

Beispiel



Übersicht

Fragestellungen

Verschiedene Eliminationsverfahren zur Berechnung der Ableitungsmatrix sind bekannt:

- Bogen- und Knotenelimination im DBG
- Bogenelimination im Berechnungsgraphen
- Knotenelimination im Berechnungsgraphen

Fragen:

- Findet man stets mit einem Verfahren die optimale Berechnungsmethode für die Ableitungsmatrix? (offen)
- Was ist die Komplexität des Problems des Findens einer optimalen Berechnungsmethode? (offen)
- Wie effizient können die einzelnen Verfahren sein?

Komplexitätsbetrachtung

bekannt: Das Problem, bei Knotenelimination im Berechnungsgraphen die Anzahl der neu hinzuzufügenden Bögen zu minimieren, ist NP-vollständig.

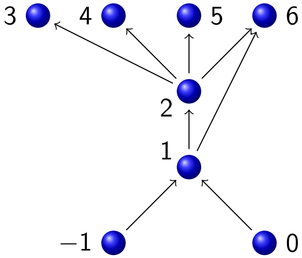
aber: Minimierung der Anzahl der neu hinzuzufügenden Bögen minimiert nicht notwendigerweise die Anzahl der ALTs und umgekehrt.

Bislang ist nicht bekannt, ob das Problem des Findens einer optimalen Berechnungsmethode NP-vollständig ist.

Effizienzvergleich

Satz

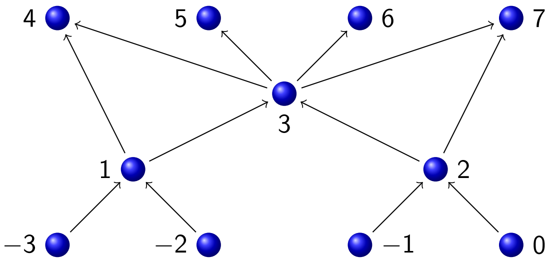
Es gibt einen Berechnungsgraphen, so daß eine optimale Folge von Bogeneliminationen im Berechnungsgraphen weniger ALTs benötigt als eine optimale Folge von Knoteneliminationen.



Effizienzvergleich

Satz

Es gibt einen Berechnungsgraphen, so daß eine optimale Folge von Bogen- und Knoteneliminationen im DBG weniger ALTs benötigt als eine optimale Folge von Bogeneliminationen im BG.



Algorithmen

Suchen der optimalen Eliminationsfolge in den einzelnen Verfahren ist im allgemeinen aufwendig.

Bei Umsetzung in Algorithmen werden Heuristiken verwendet:

- gierige Heuristik: in jedem Schritt nur lokale Optimierung (verschiedene lokale Optimalitätskriterien möglich)
- simulierte Abkühlung

Übersicht

Zusammenfassung

Ziel: optimierte Berechnung der Ableitungsmatrix

behandelte Hilfsmittel:

- Elimination im dualen Berechnungsgraphen
- Bogenelimination im Berechnungsgraphen
- Knotenelimination im Berechnungsgraphen

Zusammenfassung

Ziel: optimierte Berechnung der Ableitungsmatrix

behandelte Hilfsmittel:

- Elimination im dualen Berechnungsgraphen
- Bogenelimination im Berechnungsgraphen
- Knotenelimination im Berechnungsgraphen

— Ende —