Lineare Gleichungssysteme R

Signatur:
Menge Q der rationalen (Bruch-) Zahlen oder die Menge $\Re$ der reellen (Real-, Fließkomma-) Zahlen. Die zweistelligen Funktionssymbole sind $+$ und $*$, die Constraintsymbole sind $=_R, <, \leq, >, \geq$ und $\neq_R$.

Constrainttheorie:
Beschreibt die entsprechenden arithmetischen Funktionen und Relationen.

Erlaubte Constraints:
Lineare Gleichungen und Ungleichungen:

$$\begin{array}{lll} C & ::= & true \mid false \mid C \wedge C \\ & \mid & a_1 * X_1 + \ldots + a_n * X_n \odot b \end{array},$$

wobei $n > 0$ ist, die *Koeffizienten* a_i Zahlen ungleich 0 sind, die Variablen X_i paarweise voneinander verschieden sind, die *Konstante* b eine Zahl ist, und $\odot$ eines der Constraintsymbole $=_R, <, \leq, >, \geq, \neq_R$ ist.

Anmerkung

Der arithmetische Ausdruck $a_1 * X_1 + \ldots + a_n * X_n$ wird als *lineares Polynom* bezeichnet.

Eine Konjunktion von Gleichungen heißt *Gleichungssystem.*

Ein Gleichungssystem heißt *gelöst (oder in Normalform)*, wenn die erste Variable jeder Gleichung in keiner anderen Gleichung vorkommt.

Variablenelimination in R

Die einzelnen Gleichungen

$$a_1 * X_1 + \ldots + a_n * X_n = b,$$

löst man nach der ersten Variable auf

$$X_1 = (b - (a_2 * X_2 \ldots + a_n * X_n))/a_1,$$

ersetzt Vorkommen der Variable X_1 in den anderen Gleichungen durch den erhaltenen Ausdruck $(b - (a_2 * X_2 \ldots + a_n * X_n))/a_1$ und vereinfacht die Gleichungen, sodaß sie wieder erlaubte Constraints sind.

Es ist leicht zu sehen, daß wiederholte Variablenelimination zu einem gelösten Gleichungssystem führt.

Variablenelimination: Implementierung (I)

Eine Gleichung $a_1 * X_1 + \ldots + a_n * X_n + b = 0$ wird durch ein Constraint `[a1*X1,..., an*Xn, b]` $\doteq_R$ `0` dargestellt.

```
eliminiere @ [A1*X|P1]=R0, P=R0 <=>
   kommt_vor_in(X,P) |
   entferne(A2*X,P,P2),
   berechne(P2+P1*A2/A1,P3),
   [A1*X|P1]=R0,
   P3=R0.         % P ohne X
```

Variablenelimination: Implementierung (II)

Eine Gleichung $a_1 * X_1 + \ldots + a_n * X_n = b$ wird durch ein Constraint `[a1*X1,..., an*Xn]` $\doteq_R$ `b` dargestellt.

```
eliminiere @ [A1*X|P1] =R B1, P =R B2 <=>
    % Kommt Variable X in P vor?
    entferne(A2*X,P,P2) |
    % loese lokal nach erster Variable X auf
    multipliziere_poly_konst(P1-B1,A2/A1,P3-B3),
    % ersetze P3-B3 fuer X in zweiter Gleichung
    subtrahiere_poly_poly(P2-B2,P3-B3,P4-B4),
    % lasse die erste Gleichung unveraendert
    [A1*X|P1] =R B1,
    % neue zweite Gleichung ohne Variable X
    P4 =R B4.

leer @ P =R B <=> P=[] | B=0.
```

Anmerkung

Die Prolog-Variablen `P, P1, P2, P3, P4` stehen für Polynome, `B, B1, B2, B3, B4` für Konstanten, und `A1, A2` für Koeffizienten. Das Hilfsprädikat `entferne(E,L1,L2)` ist genau dann erfolgreich, wenn die Liste `L1` das Element `E` enthält und die Liste `L2` die Liste `L1` ist, aus der das Element `E` entfernt wurde.

Im Gegensatz zum oben beschrieben Prinzip eliminiert die CHR-Regel `eliminiere` mit Hilfe einer Gleichung das Vorkommen einer Variable in nur einer anderen Gleichung, wobei die erste Gleichung unverändert bleibt, d. h. es wird nicht explizit nach dieser Variable aufgelöst. Wiederholte Anwendung von `eliminiere` entfernt wieder jedes Vorkommen einer Variable bis auf eines und führt so zur gewünschten Normalform.

Die Regel `leer` behandelt einen Spezialfall: Wenn die Liste `P`, die ein Polynom repräsentiert, leer ist (d. h. von der Form `[]`), dann muß die Konstante identisch Null sein. Eine leere Liste erhält man z. B. wenn die Regel `eliminiere` zwei Gleichungen mit identischen Polynomen vereinfacht.

Lineare Gleichungssysteme R

Beispiel 1 Gegeben sei das Gleichungssystem $X + Y = 7, X - Y = 3$. Es kann repräsentiert werden als Ziel `[1*X,1*Y]` $=_R$ `7, [1*X,-1*Y]` $=_R$ `3` mit den Reduktionen:

$$
\begin{array}{ll}
 & \texttt{[1*X,1*Y]} =_R \texttt{7, [1*X,-1*Y]} =_R \texttt{3} \\
\mapsto_{eliminiere} & \texttt{[1*X,1*Y]} =_R \texttt{7, [2*Y]} =_R \texttt{4} \\
\mapsto_{eliminiere} & \texttt{[1*X]} =_R \texttt{5, [2*Y]} =_R \texttt{4} \quad \diamond
\end{array}
$$

Lineare Gleichungssysteme R

Um eindeutig bestimmte (determinierte) Variablen an ihren Wert zu binden, fügen wir zwei Regeln hinzu:

```
binden @ [A*X] =_R B <=> X is B/A.

gebunden @ P =_R B <=>
    entferne(A*X,P,P1), nonvar(X) |
    B1 is B - X*A,
    P1 =_R B1.
```

Anmerkung

Die erste Regel `binden` löst eine Gleichung, in der nur noch eine Variable vorkommt, nach dieser Variablen auf und bindet die Variable an den berechneten Wert. Die zweite Regel `gebunden` vereinfacht eine Gleichung, deren Polynom eine gebundene Variable enthält, indem die Konstante der Gleichung entsprechend aktualisiert wird.

Zum Beispiel wird eine Ungleichung $p > b$ zu $p - s = b \wedge s > 0$ transformiert, wobei s die neu eingeführte Variable ist. Man nennt sie Schlupfvariablen (engl. slack variables). Sie sind so gewählt, daß sie immer nichtnegativ sind. Die verbleibenden einfachen Ungleichungen werden so lange ignoriert, bis ein Wert für die Schlupfvariable bekannt ist.

Man verliert allerdings dabei die Vollständigkeit. Entweder eine strengere Normalform für Gleichungen nur mit Schlupfvariablen (der Ansatz in CHIP), die die Konsistenz garantiert, oder man wendet erweiterte Eliminationsverfahren an, um implizite Gleichheiten zwischen Schlupfvariablen zu entdecken (der Ansatz in CLP(R) und Prolog III).

In CLP-Sprachen kommen Erweiterungen der Gauß'schen Elimination (quadratisch) und des Simplexverfahrens (exponentiell, oft polynomial) zum Einsatz.

Anwendungsbeispiel: Finanzwesen

```
%       D: Darlehenshoehe
%       T: Dauer der Rueckzahlung in Monaten
%       Z: Zinssatz pro Monat
%       R: Monatliche Rueckzahlungsrate
%       S: Restschuld nach T Monaten

mortgage(D, T, Z, R, S) :-
        T =_R 0,
        D =_R S.

mortgage(D, T, Z, R, S)  :-
        T > 0,
        T1 =_R T - 1,
        D1 =_R D + D*Z - R,
        mortgage(D1, T1, Z, R, S).
```

Testbeispiele für `mortgage`

```
:- mortgage(100000,360,0.01,1025,S).

S = 12625.896680787611

:- mortgage(D,360,0.01,1025,0).

D = 99648.7893560409

:- S=<0, mortgage(100000,T,0.01,1025,S).

T= 374, S = -807.9642032884383

:- mortgage(D,360,0.01,R,0).

R = 0.0102861259692550 55*D
```

Anmerkung

Das Ziel `mortgage(100000,360,0.01,1025,S)` liefert die Antwort `S`$\doteq$`12625.90`. Man kann sich nun fragen, welchen Betrag man in diesem Zeitraum zu diesen Konditionen vollständig zurückzahlen kann: Das Ziel `mortgage(D,360,0.01,1025,0)` liefert `D`$\doteq$`99648.79`, einen nur geringfügig niedrigeren Betrag (Zinseszinseffekt).

Umgekehrt können wir uns fragen, wieviel Monate lang wir die `100000` zurückzahlen müssen. Das Ziel `S=<0, mortgage(100000,T,0.01,1025,S)` liefert `T`$\doteq$`374` und `S`$\doteq$`-807.96`.

Wie verhalten sich allgemein Darlehenshöhe und Rückzahlungsbetrag zueinander, wenn man zu obigen Konditionen nach 30 Jahren schuldenfrei sein will? Das Ziel `mortgage(D,360,0.01,R,0)` liefert die intensionale Antwort `R`$=_R$`0.0102861198*D`.

Beliebig weit läßt sich aber die Verallgemeinerung des Zieles nicht treiben: Sollte ein Ziel den Zinssatz `Z` offen lassen, so wird nach mehr als einem Rekursionschritt die Gleichung `D1` $=_R$ `D + D*Z - R` nicht mehr linear, da das neue `D` dann ebenfalls nicht mehr determiniert ist.

Projektion für Gleichungen

Eliminiere alle Vorkommen der Variable `X`.

p2 @ project(X), PX1 $=_R$ B1 $\backslash$ PX2 $=_R$ B2 <=>
  entferne(A1*X,PX1,P1), entferne(A2*X,PX2,P2)
  |
  multipliziere_poly_konst(P1-B1,A2/A1,P3-B3),
  subtrahiere_poly_poly(P2-B2,P3-B3,P4-B4),
  P4 $=_R$ B4.

p1 @ project(X) $\backslash$ PX1 $=_R$ B1 <=>
  entferne(A1*X,PX1,P1) | true.

p0 @ project(X) <=> true.

Projektion fr Ungleichungen (I)

Geg. Constraintproblem ber Variablen $\mathcal{X}$, gesucht Constraints ber $\mathcal{X}' \subseteq \mathcal{X}$.

Beispiel:

$$\begin{array}{rcl} X+Y & \leq & 1 \\ X-Y & \leq & 1 \\ -X+Y & \leq & 1 \\ -X-Y & \leq & 1 \end{array}$$

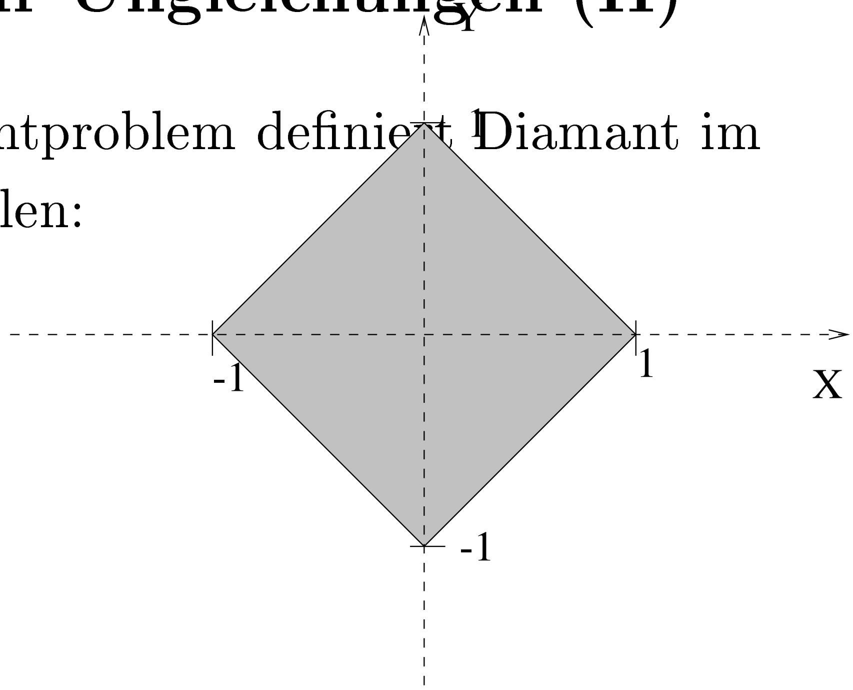
Resultat der Projektion:

$$-1 \leq X$$

$$X \leq 1$$

Projektion fr Ungleichungen (II)

Erstes Constraintproblem definiert Diamant im Zweidimensionalen:



Zweites Constraintproblem: Projektion of X Linie

Fourier Elimination

Elimination einer Variablen Y. Geg: Menge C von linear Ungleichungen.

Aufteilung von C in:

$$\begin{aligned} C^0 &= \text{Ungleichungen ohne } Y \\ C^+ &= \text{Ungleichungen } Y \leq t \\ C^- &= \text{Ungleichungen } t \leq Y \end{aligned}$$

t enthält nicht Y. Die Fourier-Elimination ist:

- Definiere C^{Y-elem} durch Bilden aller Ungleichungen

  $t_1 \leq t_2 \;\text{ mit } t_1 \leq Y \in C^- \text{ und } Y \leq t_2 \in C^+$

- Ersetze $C = C^0 \cup C^- \cup C^+$ durch $C' = C^0 \cup C^{Y-elem}$

C' ist Projektion von C auf $var(C)\backslash\{Y\}$.

Iteration!

Beispiel

Diamond:

$$\begin{array}{llll} X+Y & \leq 1 & X-Y & \leq 1 \\ -X+Y & \leq 1 & -X-Y & \leq 1 \end{array}$$

Aufspaltung:

$$\begin{aligned} C^0 &= \{\} \\ C^+ &= \{X-1 \leq Y, -1-X \leq Y\} \\ C^- &= \{Y \leq 1-X, Y \leq 1+X\} \end{aligned}$$

Fourierelimination:

$$\begin{array}{lllll} X-1 \leq Y & \text{mit} & Y \leq 1-X & \text{ergibt} & X \leq 1 \\ X-1 \leq Y & \text{mit} & Y \leq 1+X & \text{ergibt} & -1 \leq 1 \\ -1-X \leq \mathrm{Y} & \text{mit} & Y \leq 1-X & \text{ergibt} & -1 \leq 1 \\ -1-X \leq Y & \text{mit} & Y \leq 1+X & \text{ergibt} & -1 \leq X \end{array}$$

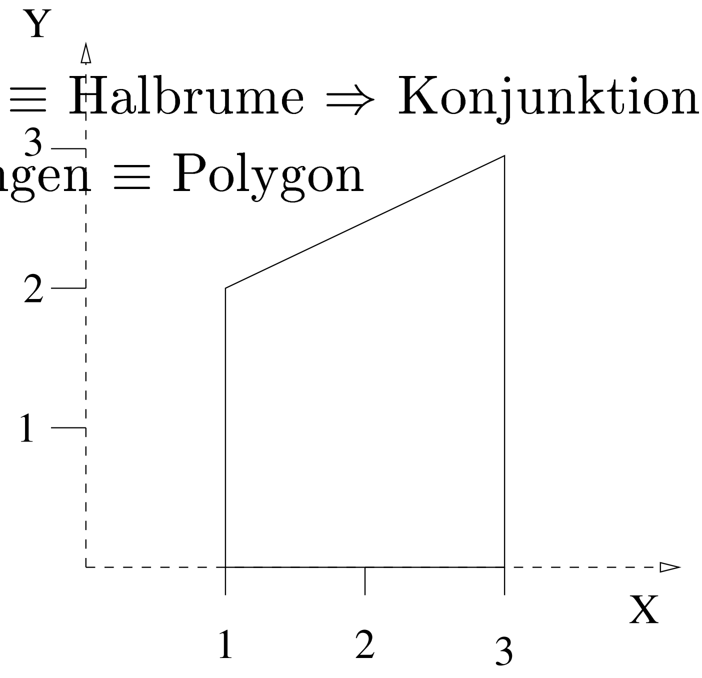
Simplex Algorithmus

Problem: Gegeben lineare Funktion, suche Minimum, wobei eine Menge von Ungleichungen eingehalten werden mu

Beispiel: Minimiere $X - Y$ unter

$$1 \leq X \wedge X \leq 3 \wedge 0 \leq Y \wedge 2Y - X \leq 3$$

Ungleichungen $\equiv$ Halbrume $\Rightarrow$ Konjunktion von Ungleichungen $\equiv$ Polygon



Simplex Form

Man kann das Problem (C, f) umformulieren: Simplexform $(C_E \wedge C_I, f)$, wobei f zu minimierende lineare Funktion, C_E lineare Gleichungen, und C_I Ungleichungen der Form $X \geq 0$ fr alle X in C_E und f sind.
Beispiel: Minimiere $X - Y$ unter

$$1 \leq X \wedge X \leq 3 \wedge 0 \leq Y \wedge 2Y - X \leq 3$$

kann in Simplexform gebracht: minimiere $X - Y$ unter

$$\begin{aligned}
&X - S2 = 1 \wedge \\
&X + S_3 = 3 \wedge \\
&-X + 2Y + S_1 = 3 \wedge \\
&X \geq 0 \wedge Y \geq 0 \wedge S_1 \geq 0 \wedge S_2 \geq 0 \wedge S_3 \geq 0
\end{aligned}$$

S_i heien Schlupfvariablen.

Basic Feasible Form

Falls C_E der Form $X_0 = b + a_1 X_1 + ... + a_n X_n$ mit $b \geq 0$, und X_0 kommt sonst in C_E und f nicht mehr vor.

Die Variablen, die nur einmal vorkommen, heien basic, die anderen Parameter.

Feasible solution: Parameter = 0.

Beispiel: Minimiere $f = 10 - Y - Z$ unter

$$\begin{array}{rcl} X & = & 3 - Y \wedge \\ T & = & 4 + 2Y - 2Z \wedge \\ X & \geq & 0 \wedge Y \geq 0 \wedge T \geq 0 \wedge Z \geq 0 \end{array}$$

Feasible solution: $Y = 0$, $Z = 0 \rightarrow f = 10$.

Anmerkung

Jede Simplexform kann durch Variableneliminati-
on in eine Basic Feasible Form gebracht werden.

Simplex (Forts.)

Idee des Simplex: jede basic feasible form entspricht einer Ecke im Lsungspolygon.

Gegeben eine Ecke (= basic feasible form) im Polygon ⇒ finde bessere Ecke (=bessere basic feasible form) im Polygon.

Dabei: Ein Parameter wird erhht, eine basic Variable auf 0 gesetzt ⇒ vertauschen eines Parameters mit einer basic Variable.

geht immer! (da Gleichungssystem in gelster Form)

Beispiel Minimiere $10 - Y - Z$ unter

$$\begin{aligned} X &= 3 - Y \wedge \\ T &= 4 + 2Y - 2Z \wedge \\ X &\geq 0 \wedge Y \geq 0 \wedge T \geq 0 \wedge Z \geq 0 \end{aligned}$$

feasible solution: $Y = 0$, $Z = 0 \rightarrow f = 10$.

Man kann die Loesung kleiner machen, wenn man Y vergrssert. Gleichungen, mit Y negativ $\rightarrow$ wieweit kann ich Y vergrern.

- $X = 3 - Y \Rightarrow Y = 3 - X$ und Maximalwert 3 fuer Y
- ersetze Y berall durch $3 - X$.
- Resultat: minimiere $7 + X - Z$ unter

$$\begin{aligned} Y &= 3 - X \wedge \\ T &= 10 - 2X - 2Z \wedge \\ X &\geq 0 \wedge Y \geq 0 \wedge T \geq 0 \wedge Z \geq 0 \end{aligned}$$

feasible solution: $X = 0$ und $Z = 0 \rightarrow f = 7$

Bespiel (Forts)

Z grer macht f kleiner. Gleichung
$T = 10 - 2X - 2Z$ begrenzt Z

daher: ersetze Z durch T mit $Z = 5 - X - 0.5T$

neue basic feasible form: minimiere
$2 + X + 0.5T$ unter

$$\begin{aligned} Y &= 3 - X \wedge \\ Z &= 5 - X - 0.5T \wedge \\ X &\geq 0 \wedge Y \geq 0 \wedge T \geq 0 \wedge Z \geq 0 \end{aligned}$$

feasible solution: $X = 0$ und $T = 0 \rightarrow f = 2$