Verfeinerter Bahnenalgorithmums

Carmen Stein, Yannic Maus

Januar 15, 2010

Inhaltsverzeichnis

Übersicht

Gruppe

Definition Gruppe

Sei G eine Menge und

$$\cdot : G \times G \to G, (g_1, g_2) \mapsto g_1 \cdot g_2 = g_1 g_2$$

eine Abbildung (genannt Verknüpfung). Dann heißt $(G, \cdot)$ eine **Gruppe**, wenn gilt:

1. $(g_1 g_2) g_3 = g_1 (g_2 g_3) \quad \forall g_1, g_2, g_3 \in G$
2. Es existiert $1 \in G$ mit $1g = g1 = g \quad \forall g \in G$
3. Zu jedem $g \in G$ existiert $g^{-1} \in G$ mit $gg^{-1} = g^{-1}g = 1$

Ordnung

Definition Gruppenordnung

Eine Gruppe G heißt **endlich**, falls die zugrundeliegende Menge G endlich ist.
$|G|$:= **Ordnung** von G := Anzahl der Elemente
Falls G nicht endlich ist, schreibt man $|G| := \infty$

Untergruppe

Definition Untergruppe

G Gruppe, $U \subseteq G$. U heißt **Untergruppe** von G, falls gilt:

1. $U \neq \emptyset$
2. $g, h \in U \Rightarrow gh^{-1} \in U$

Wir schreiben: $U \leq G$

Bemerkung

Eine Untergruppe ist wieder eine Gruppe bzgl. der Einschränkung der Verknüpfung.

Operation

Definition Operation

G Gruppe, M Menge. Eine Abbildung $G \times M \to M : (g, m) \mapsto gm$ mit

1. neutrales Element: $1m = m \quad \forall m \in M$
2. „Assoziativität“: $(g_1 g_2)m = g_1(g_2 m) \quad \forall m \in M, g_1, g_2 \in G$

heißt (Links-)**Operation** von G auf M.
Man nennt die Menge M ausgestattet mit der Operation der Gruppe G auch $G-$Menge.

Definition Bahn

G operiere auf M, $m \in M$.

$$Gm := \{gm \mid g \in G\}$$

heißt die **Bahn** von m unter G.
Bemerkung: Gm ist endlich, wenn G oder M endlich ist.

Erzeugnis

Definition Erzeugnis

Sei $M \subseteq G$, G Gruppe.

1. $$\langle M \rangle := \bigcap_{U \leq G, M \subseteq U} U$$

   heißt das **Erzeugnis** von M. $\langle M \rangle$ ist die kleinste Untergruppe, die M enthält.
   Falls $M = \{g_1, \cdots, g_n\}$, schreibt man auch: $\langle M \rangle := \langle g_1, \cdots, g_n \rangle$

Erzeugnis

Definition Erzeugnis

Sei $M \subseteq G$, G Gruppe.

1. $$\langle M \rangle := \bigcap_{U \leq G, M \subseteq U} U$$
   heißt das **Erzeugnis** von M. $\langle M \rangle$ ist die kleinste Untergruppe, die M enthält.
   Falls $M = \{g_1, \cdots, g_n\}$, schreibt man auch: $\langle M \rangle := \langle g_1, \cdots, g_n \rangle$
2. Falls $G = \langle M \rangle$, so heißt M ein **Erzeugendensystem** von G.

Erzeugnis

Definition Erzeugnis

Sei $M \subseteq G$, G Gruppe.

1. $$\langle M \rangle := \bigcap_{U \leq G, M \subseteq U} U$$
   heißt das **Erzeugnis** von M. $\langle M \rangle$ ist die kleinste Untergruppe, die M enthält.
   Falls $M = \{g_1, \cdots, g_n\}$, schreibt man auch: $\langle M \rangle := \langle g_1, \cdots, g_n \rangle$
2. Falls $G = \langle M \rangle$, so heißt M ein **Erzeugendensystem** von G.
3. Eine Gruppe, die von einem Element erzeugt wird, heißt **zyklisch**.

alternative Definition Erzeugnis

Alternative Definition: Sei $M \subseteq G$, G Gruppe.

1. $\langle M \rangle := \{m_1 \cdots m_n \mid m_i \in M \text{ oder } m_i^{-1} \in M, n \in \mathbb{N}_0\}$

alternative Definition Erzeugnis

Alternative Definition: Sei $M \subseteq G$, G Gruppe.

1. $\langle M \rangle := \{m_1 \cdots m_n \mid m_i \in M \text{ oder } m_i^{-1} \in M, n \in \mathbb{N}_0\}$
2. das leere Produkt ist als 1 definiert

alternative Definition Erzeugnis

Alternative Definition: Sei $M \subseteq G$, G Gruppe.

1. $\langle M \rangle := \{m_1 \cdots m_n \mid m_i \in M \text{ oder } m_i^{-1} \in M, n \in \mathbb{N}_0\}$
2. das leere Produkt ist als 1 definiert
3. algorithmisch besser nutzbar

Übersicht

Bahnenalgorithmus

Algorithm 1 einfacher Bahnenalgorithmus I

Eingabe: $E = \{g_1, \ldots, g_n\}$ Gruppe $G = \langle E \rangle$, Operation von G auf M, $m \in M$.

Ausgabe: Die Bahn Gm

```
Bahn := {m}
for each e ∈ Bahn do
   for each g ∈ E do
      newElement := g · e
      if newElement ∉ Bahn then
         Bahn := Bahn ∪ {newElement}
      end if
   end for
end for
```

einfacher Bahnenalgorithmus II

Bemerkungen

Falls $\mid G \mid = \infty$, so füge zu E noch die inversen Elemente der Erzeuger hinzu: $E = \{g_1, \ldots g_n, g_1^{-1} \ldots g_n^{-1}\}$

Übersicht

Restklasse

Bemerkung

- *$U \leq G$ operiert auf G durch inverse Rechtsmultiplikation:*
  $U \times G \to G : (u, g) \mapsto gu^{-1}$.
- *Die Bahn*
  $$gU := \{gu \mid u \in U\}$$
  von $g \in G$ unter der Operation heißt **Restklasse** *von g nach U.*
- *Menge der Restklassen von G nach U bezeichnet man als G/U*
- *ein Restklassenvertretersystem bezeichnet man als* **Transversale**.

Stabilisator

Definition Stabilisator

G operiere auf M, $m \in M$.

$$\mathrm{Stab}_G(m) := \{g \in G \mid gm = m\}$$

heißt **Stabilisator** von m in G.

Bemerkung

$Stab_G(m) \leq G$.

Stabilisator

Satz

$| Gm | \cdot | Stab_G(m) | = | G |$

Beweis.

Tafel □

Übersicht

Verfeinerter Bahnenalgorithmus I

Algorithmus

Gegeben: $E = \{g_1, \ldots, g_n\}$ *Gruppe* $G = \langle E \rangle$, *Operation auf* M, $m \in M$.
Gesucht:

1. *Die Bahn* Gm
2. *Erzeugendensystem* E_S *von* $Stab_G(m)$
3. *Transversale von* $G/Stab_G(m)$. *Genauer eine Abbildung mit* $\omega : Gm \to G$ *mit* $\omega(e)m = e$

Verfeinerter Bahnenalgorithmus II

Algorithm 2 verfeinerter Bahnenalgorithmus

```
Bahn := {m}
for each e ∈ Bahn do
    for each g ∈ E do
        newElement := g · e
        if newElement ∉ Bahn then
            Bahn := Bahn ∪ {newElement}
        end if
    end for
end for
```

Verfeinerter Bahnenalgorithmus II

Algorithm 3 verfeinerter Bahnenalgorithmus

$Bahn := \{m\}$
$\omega(m) = id$
$E_S = \emptyset$
for *each* $e \in Bahn$ **do**
  for *each* $g \in E$ **do**
    $newElement := g \cdot e$
    if $newElement \notin Bahn$ **then**
      $Bahn := Bahn \cup \{newElement\}$
    end if
  end for
end for

Verfeinerter Bahnenalgorithmus II

Algorithm 4 verfeinerter Bahnenalgorithmus

$Bahn := \{m\}$
$\omega(m) = id$
$E_S = \emptyset$
for *each* $e \in Bahn$ **do**
  for *each* $g \in E$ **do**
    $newElement := g \cdot e$
    if $newElement \notin Bahn$ **then**
      $Bahn := Bahn \cup \{newElement\}$
      $\omega(newElement) := g \cdot \omega(e)$
    end if
  end for
end for

Verfeinerter Bahnenalgorithmus II

Algorithm 5 verfeinerter Bahnenalgorithmus

$Bahn := \{m\}$
$\omega(m) = id$
$E_S = \emptyset$
for *each* $e \in Bahn$ **do**
  for *each* $g \in E$ **do**
    $newElement := g \cdot e$
    if $newElement \notin Bahn$ **then**
      $Bahn := Bahn \cup \{newElement\}$
      $\omega(newElement) := g \cdot \omega(e)$
    else
      $E_S := E_S \cup \{\omega(ge)^{-1} \cdot g \cdot (\omega(e)\}$
    end if
  end for
end for

Anwendung

Anwendung

- Ordnung einer Gruppe berechnen (Beispiel)

Anwendung

Anwendung

- Ordnung einer Gruppe berechnen (Beispiel)
- Feulner, Automorphismengruppen berechnen

Anwendung

Anwendung

- Ordnung einer Gruppe berechnen (Beispiel)
- Feulner, Automorphismengruppen berechnen
- Leons Algorithmus beruht auf der Grundidee, schränkt aber die Gruppe ein

Implementierung

Implementierung

- einfacher Bahnenalgorithmus in C++, verschiedene Suchstrukturen (Listen, Hashmaps)

Implementierung

Implementierung

- einfacher Bahnenalgorithmus in C++, verschiedene Suchstrukturen (Listen, Hashmaps)
- verfeinerter Bahnenalgorithmus in GAP (Groups, Algorithms, Programming) implementiert (auf Listen basierend)

Implementierung

Implementierung

- einfacher Bahnenalgorithmus in C++, verschiedene Suchstrukturen (Listen, Hashmaps)
- verfeinerter Bahnenalgorithmus in GAP (Groups, Algorithms, Programming) implementiert (auf Listen basierend)
- Beispiel mit Hamming Code [7,4]