Introduction

GAP examples

The orbit algorithm

Computing the stabiliser

Double cosets

Orbits by suborbits

Storing suborbits

Orbit-by-Suborbits

Finding homomorphisms

Problems

Orbits and Double Cosets

Max Neunhöffer

University of St Andrews

GAC 2010, Allahabad

Group actions and orbits

Let G be a group acting from the right on a set X:

$$A : X \times G \to X \qquad \text{(“action function”)}$$

with $A(x, 1) = x$ and $A(x, gh) = A(A(x, g), h)$ for all $x \in X$ and all $g, h \in G$.

Introduction
GAP examples
The orbit algorithm
Computing the stabiliser
Double cosets
Orbits by suborbits
Storing suborbits
Orbit-by-Suborbits
Finding homomorphisms
Problems

Group actions and orbits

Let G be a group acting from the right on a set X:

$$A : X \times G \to X \qquad \text{(“action function”)}$$

with $A(x, 1) = x$ and $A(x, gh) = A(A(x, g), h)$ for all $x \in X$ and all $g, h \in G$.

Notation

Write xg for $A(x, g)$ and xgh for $A(x, gh) = A(A(x, g), h)$.

Group actions and orbits

Let G be a group acting from the right on a set X:

$$A : X \times G \to X \qquad (\text{“action function”})$$

with $A(x, 1) = x$ and $A(x, gh) = A(A(x, g), h)$ for all $x \in X$ and all $g, h \in G$.

Notation

Write xg for $A(x, g)$ and xgh for $A(x, gh) = A(A(x, g), h)$.
Write xG for $\{xg \mid g \in G\}$ and call A transitive if $xG = X$.

Group actions and orbits

Let G be a group acting from the right on a set X:

$$A : X \times G \to X \qquad (\text{“action function”})$$

with $A(x, 1) = x$ and $A(x, gh) = A(A(x, g), h)$ for all $x \in X$ and all $g, h \in G$.

Notation

Write xg for $A(x, g)$ and xgh for $A(x, gh) = A(A(x, g), h)$.
Write xG for $\{xg \mid g \in G\}$ and call A transitive if $xG = X$.
Call xG the orbit of x under G.

Introduction
GAP examples
The orbit algorithm
Computing the stabiliser
Double cosets
Orbits by suborbits
Storing suborbits
Orbit-by-Suborbits
Finding homomorphisms
Problems

Group actions and orbits

Let G be a group acting from the right on a set X:

$$A : X \times G \to X \qquad (\text{“action function”})$$

with $A(x, 1) = x$ and $A(x, gh) = A(A(x, g), h)$ for all $x \in X$ and all $g, h \in G$.

Notation

Write xg for $A(x, g)$ and xgh for $A(x, gh) = A(A(x, g), h)$.
Write xG for $\{xg \mid g \in G\}$ and call A transitive if $xG = X$.
Call xG the orbit of x under G.
Call $\mathrm{Stab}_G(x) := \{g \in G \mid xg = x\} \le G$ the stabiliser.

Group actions and orbits

Let G be a group acting from the right on a set X:

$$A : X \times G \to X \qquad (\text{“action function”})$$

with $A(x, 1) = x$ and $A(x, gh) = A(A(x, g), h)$ for all $x \in X$ and all $g, h \in G$.

Notation

Write xg for $A(x, g)$ and xgh for $A(x, gh) = A(A(x, g), h)$.
Write xG for $\{xg \mid g \in G\}$ and call A transitive if $xG = X$.
Call xG the orbit of x under G.
Call $\mathrm{Stab}_G(x) := \{g \in G \mid xg = x\} \leq G$ the stabiliser.
Write gH for $\{gh \mid h \in H\}$ and Hg for $\{hg \mid h \in H\}$.

Group actions and orbits

Let G be a group acting from the right on a set X:

$$A : X \times G \to X \qquad (\text{“action function”})$$

with $A(x, 1) = x$ and $A(x, gh) = A(A(x, g), h)$ for all $x \in X$ and all $g, h \in G$.

Notation

Write xg for $A(x, g)$ and xgh for $A(x, gh) = A(A(x, g), h)$.
Write xG for $\{xg \mid g \in G\}$ and call A transitive if $xG = X$.
Call xG the orbit of x under G.
Call $\mathrm{Stab}_G(x) := \{g \in G \mid xg = x\} \leq G$ the stabiliser.
Write gH for $\{gh \mid h \in H\}$ and Hg for $\{hg \mid h \in H\}$.

Example

Let $H < G$, then

- H acts on G by $A : G \times H \to G, (g, h) \mapsto gh$.

Introduction

GAP examples

The orbit algorithm

Computing the stabiliser

Double cosets

Orbits by suborbits

Storing suborbits

Orbit-by-Suborbits

Finding homomorphisms

Problems

Group actions and orbits

Let G be a group acting from the right on a set X:

$$A : X \times G \to X \qquad \text{("action function")}$$

with $A(x, 1) = x$ and $A(x, gh) = A(A(x, g), h)$ for all $x \in X$ and all $g, h \in G$.

Notation

Write xg for $A(x, g)$ and xgh for $A(x, gh) = A(A(x, g), h)$.
Write xG for $\{xg \mid g \in G\}$ and call A transitive if $xG = X$.
Call xG the orbit of x under G.
Call $\mathrm{Stab}_G(x) := \{g \in G \mid xg = x\} \leq G$ the stabiliser.
Write gH for $\{gh \mid h \in H\}$ and Hg for $\{hg \mid h \in H\}$.

Example

Let $H < G$, then

- H acts on G by $A : G \times H \to G, (g, h) \mapsto gh$.
- G acts on the right cosets $X := \{Hg \mid g \in G\}$ by $A : X \times G \to X, (Hg, g') \mapsto Hgg'$.

Introduction
GAP examples
The orbit algorithm
Computing the stabiliser
Double cosets
Orbits by suborbits
Storing suborbits
Orbit-by-Suborbits
Finding homomorphisms
Problems

Orbit-Stabiliser Theorem/Double cosets

Theorem (Orbit-Stabiliser)

Let G act transitively on X and let $S := \mathrm{Stab}_G(x)$ for some $x \in X$. Then $|G| = |X| \cdot |S|$ and

$$\begin{array}{ccc} \{Sg \mid g \in G\} & \longrightarrow & X \\ Sg & \longmapsto & xg \end{array}$$

is well-defined and is a bijection.

Introduction

GAP examples

The orbit algorithm

Computing the stabiliser

Double cosets

Orbits by suborbits

Storing suborbits

Orbit-by-Suborbits

Finding homomorphisms

Problems

Orbit-Stabiliser Theorem/Double cosets

Theorem (Orbit-Stabiliser)

Let G act transitively on X and let $S := \mathrm{Stab}_G(x)$ for some $x \in X$. Then $|G| = |X| \cdot |S|$ and

$$\begin{array}{ccc} \{Sg \mid g \in G\} & \longrightarrow & X \\ Sg & \longmapsto & xg \end{array}$$

is well-defined and is a bijection.

Definition (Double cosets)

Let $H, K \leq G$ be two subgroups. Then

$$HgK := \{hgk \mid h \in H, k \in K\}$$

is called the H-K-double coset of g.

GAP examples

see other window

Introduction
GAP examples
The orbit algorithm
Computing the stabiliser
Double cosets
Orbits by suborbits
Storing suborbits
Orbit-by-Suborbits
Finding homomorphisms
Problems

Algorithm: EnumerateOrbit

Input: $G = \langle g_1, \ldots, g_k \rangle$ acting on X and $x \in X$.

1. Assign list $L := [x]$ and $i := 1$
2. While $i \leq$ Length(L) do
3. For j in $[1, 2, \ldots, k]$ do
4. Assign $y := L[i]g_j$
5. If $y \notin L$ then
6. Append y to the end of L
7. Assign $i := i + 1$

Introduction

GAP examples

The orbit algorithm

Computing the stabiliser

Double cosets

Orbits by suborbits

Storing suborbits

Orbit-by-Suborbits

Finding homomorphisms

Problems

Algorithm: ENUMERATEORBIT

Input: $G = \langle g_1, \ldots, g_k \rangle$ acting on X and $x \in X$.

1. Assign list $L := [x]$ and $i := 1$
2. While $i \leq$ Length(L) do
3. For j in $[1, 2, \ldots, k]$ do
4. Assign $y := L[i]g_j$
5. If $y \notin L$ then
6. Append y to the end of L
7. Assign $i := i + 1$

Fact (Correctness and termination)

If this terminates, then L contains the complete orbit xG.

Algorithm: EnumerateOrbit

Input: $G = \langle g_1, \ldots, g_k \rangle$ acting on X and $x \in X$.

1. Assign list $L := [x]$ and $i := 1$
2. While $i \leq$ Length(L) do
3. For j in $[1, 2, \ldots, k]$ do
4. Assign $y := L[i]g_j$
5. If $y \notin L$ then
6. Append y to the end of L
7. Assign $i := i + 1$

Fact (Correctness and termination)

If this terminates, then L contains the complete orbit xG. If xG is finite, then the algorithm terminates.

Introduction
GAP examples
The orbit algorithm
Computing the stabiliser
Double cosets
Orbits by suborbits
Storing suborbits
Orbit-by-Suborbits
Finding homomorphisms
Problems

Algorithm: EnumerateOrbit

Input: $G = \langle g_1, \ldots, g_k \rangle$ acting on X and $x \in X$.

1. Assign list $L := [x]$ and $i := 1$
2. While $i \leq$ Length(L) do
3. For j in $[1, 2, \ldots, k]$ do
4. Assign $y := L[i]g_j$
5. If $y \notin L$ then
6. Append y to the end of L
7. Assign $i := i + 1$

Fact (Correctness and termination)

If this terminates, then L contains the complete orbit xG.
If xG is finite, then the algorithm terminates.

Comment (Performance)

Crucial: Check efficiently if $y \notin L$.

Algorithm: ENUMERATEORBIT

Input: $G = \langle g_1, \ldots, g_k \rangle$ acting on X and $x \in X$.

1. Assign list $L := [x]$ and $i := 1$
2. While $i \leq$ Length(L) do
3. For j in $[1, 2, \ldots, k]$ do
4. Assign $y := L[i]g_j$
5. If $y \notin L$ then
6. Append y to the end of L
7. Assign $i := i + 1$

Fact (Correctness and termination)

If this terminates, then L contains the complete orbit xG.
If xG is finite, then the algorithm terminates.

Comment (Performance)

Crucial: Check efficiently if $y \notin L$.
$\Longrightarrow$ use hashing technique

Breadth first search — Schreier tree

$$G = \langle a, b, c \rangle$$

$$x \xrightarrow{a} xa$$

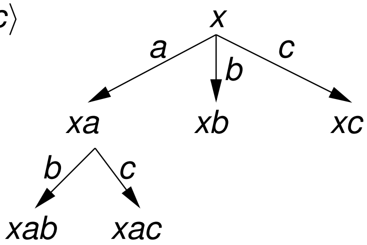
Breadth first search — Schreier tree

$G = \langle a, b, c \rangle$

$$\begin{tikzcd} & x \arrow[dl, "a"'] \arrow[d, "b"] \arrow[dr, "c"] & \\ xa & xb & xc \end{tikzcd}$$

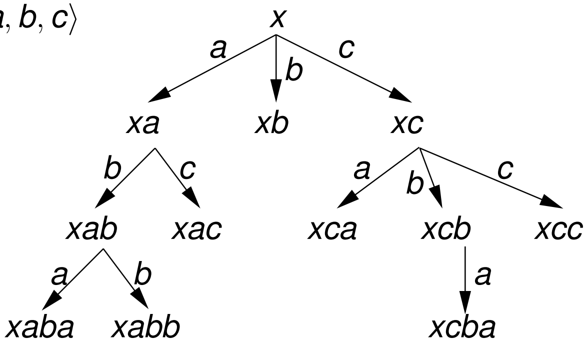
Breadth first search — Schreier tree

$G = \langle a, b, c \rangle$



Breadth first search — Schreier tree

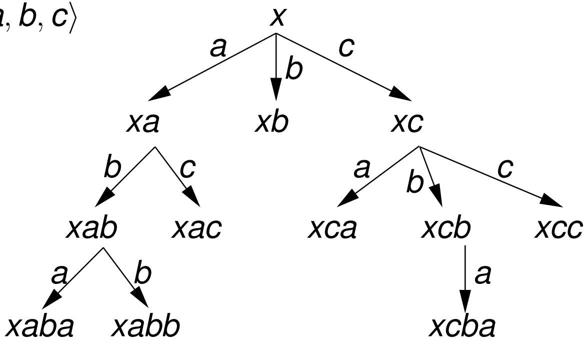
$G = \langle a, b, c \rangle$



Introduction
GAP examples
The orbit algorithm
Computing the stabiliser
Double cosets
Orbits by suborbits
Storing suborbits
Orbit-by-Suborbits
Finding homomorphisms
Problems

Breadth first search — Schreier tree

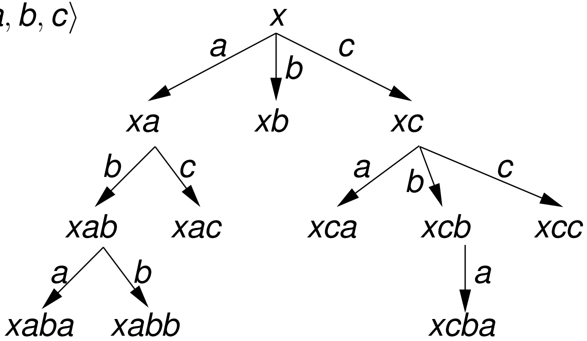
$G = \langle a, b, c \rangle$



Tree is discovered row by row.

Breadth first search — Schreier tree

$G = \langle a, b, c \rangle$

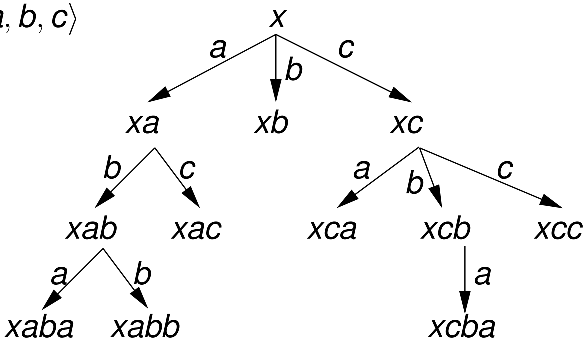


Tree is discovered row by row.

For each point we get a word in the generators.

Breadth first search — Schreier tree

$G = \langle a, b, c \rangle$



Tree is discovered row by row.

For each point we get a word in the generators.

These words are shortest possible!

Computing the stabiliser

Assume we run the standard orbit algorithm for $X = xG$.

Computing the stabiliser

Assume we run the standard orbit algorithm for $X = xG$.

Fact (Schreier generators)

Whenever we apply a generator g to a point xw (w a word in the generators) and find that $y := xwg$ is already known, it must be of the form xw' for a known word w'.

Computing the stabiliser

Assume we run the standard orbit algorithm for $X = xG$.

Fact (Schreier generators)

Whenever we apply a generator g to a point xw (w a word in the generators) and find that $y := xwg$ is already known, it must be of the form xw' for a known word w'. Then wgw'^{-1} fixes x and thus is contained in $\mathrm{Stab}_G(x)$.

Computing the stabiliser

Assume we run the standard orbit algorithm for $X = xG$.

Fact (Schreier generators)

Whenever we apply a generator g to a point xw (w a word in the generators) and find that $y := xwg$ is already known, it must be of the form xw' for a known word w'. Then wgw'^{-1} fixes x and thus is contained in $\mathrm{Stab}_G(x)$.

Theorem (Schreier's Lemma)

All these wgw'^{-1} together generate $\mathrm{Stab}_G(x)$.

Computing the stabiliser

Assume we run the standard orbit algorithm for $X = xG$.

Fact (Schreier generators)

Whenever we apply a generator g to a point xw (w a word in the generators) and find that $y := xwg$ is already known, it must be of the form xw' for a known word w'. Then wgw'^{-1} fixes x and thus is contained in $\mathrm{Stab}_G(x)$.

Theorem (Schreier's Lemma)

All these wgw'^{-1} together generate $\mathrm{Stab}_G(x)$.

Problem

There can be many such Schreier generators.

The Orbit-Stabiliser-Algorithm

Algorithm: EnumerateOrbitWithStabiliser

Input: $G = \langle g_1, \ldots, g_k \rangle$ acting on X and $x \in X$.

1. Assign list $L := [x]$ and $i := 1$ and $S := \{1\}$
2. While $i \leq$ Length(L) do
3. For j in $[1, 2, \ldots, k]$ do
4. Assign $y := L[i]g_j$
5. If $y \notin L$ then
6. Append y to the end of L
7. else
8. Assign $S := \langle S, wg_jw'^{-1} \rangle$
9. Assign $i := i + 1$

Double cosets

Assume we want to find S-H-double coset representatives for $S, H < G$.

Double cosets

Assume we want to find S-H-double coset representatives for $S, H < G$. Find a transitive action of G on X with $S = \mathrm{Stab}_G(x)$ for some $x \in X$.

Introduction
GAP examples
The orbit algorithm
Computing the stabiliser
Double cosets
Orbits by suborbits
Storing suborbits
Orbit-by-Suborbits
Finding homomorphisms
Problems

Double cosets

Assume we want to find S-H-double coset representatives for $S, H < G$. Find a transitive action of G on X with $S = \mathrm{Stab}_G(x)$ for some $x \in X$.

Theorem

In the above situation the map

$$F: \begin{array}{ccc} \{SgH \mid g \in G\} & \longrightarrow & \{yH \mid y \in X\} \\ SgH & \longmapsto & xgH \end{array}$$

between the set of S-H-double cosets and the set of H-suborbits is well-defined and is a bijection.

Double cosets

Assume we want to find S-H-double coset representatives for $S, H < G$. Find a transitive action of G on X with $S = \mathrm{Stab}_G(x)$ for some $x \in X$.

Theorem

In the above situation the map

$$\begin{array}{cccc} F: & \{SgH \mid g \in G\} & \longrightarrow & \{yH \mid y \in X\} \\ & SgH & \longmapsto & xgH \end{array}$$

between the set of S-H-double cosets and the set of H-suborbits is well-defined and is a bijection.

$\Longrightarrow$ We enumerate xG and then find all H-orbits in there.

Double cosets

Assume we want to find S-H-double coset representatives for $S, H < G$. Find a transitive action of G on X with $S = \mathrm{Stab}_G(x)$ for some $x \in X$.

Theorem

In the above situation the map

$$F: \begin{array}{ccc} \{SgH \mid g \in G\} & \longrightarrow & \{yH \mid y \in X\} \\ SgH & \longmapsto & xgH \end{array}$$

between the set of S-H-double cosets and the set of H-suborbits is well-defined and is a bijection.

$\Longrightarrow$ We enumerate xG and then find all H-orbits in there.

Without a better idea, we would simply enumerate H-orbits of points in xG which we have not yet covered.

Double cosets

Theorem

In the above situation the map

$$\begin{array}{rccc} F: & \{SgH \mid g \in G\} & \longrightarrow & \{yH \mid y \in X\} \\ & SgH & \longmapsto & xgH \end{array}$$

between the set of S-H-double cosets and the set of H-suborbits is well-defined and is a bijection.

Double cosets

Theorem

In the above situation the map

$$\begin{array}{rccc} F: & \{SgH \mid g \in G\} & \longrightarrow & \{yH \mid y \in X\} \\ & SgH & \longmapsto & xgH \end{array}$$

between the set of S-H-double cosets and the set of H-suborbits is well-defined and is a bijection.

Proof:

- $SgH = Sg'H$ iff $g' = sgh$ for some $s \in S$, $h \in H$.

Double cosets

Theorem

In the above situation the map

$$F: \begin{array}{ccc} \{SgH \mid g \in G\} & \longrightarrow & \{yH \mid y \in X\} \\ SgH & \longmapsto & xgH \end{array}$$

between the set of S-H-double cosets and the set of H-suborbits is well-defined and is a bijection.

Proof:

- $SgH = Sg'H$ iff $g' = sgh$ for some $s \in S$, $h \in H$.
- In that case $xgH = xsghH$. Thus F is well-defined.

Double cosets

Theorem

In the above situation the map

$$\begin{array}{rccc} F: & \{SgH \mid g \in G\} & \longrightarrow & \{yH \mid y \in X\} \\ & SgH & \longmapsto & xgH \end{array}$$

between the set of S-H-double cosets and the set of H-suborbits is well-defined and is a bijection.

Proof:

- $SgH = Sg'H$ iff $g' = sgh$ for some $s \in S$, $h \in H$.
- In that case $xgH = xsghH$. Thus F is well-defined.
- If $xgH = xg'H$, then there is an $h \in H$ such that $xgh = xg'$ and thus ghg'^{-1} fixes x and lies in S. Thus $g' = sgh$ for some $s \in S$ and some $h \in H$ and F is injective.

Introduction

GAP examples

The orbit algorithm

Computing the stabiliser

Double cosets

Orbits by suborbits

Storing suborbits

Orbit-by-Suborbits

Finding homomorphisms

Problems

Double cosets

Theorem

In the above situation the map

$$F: \begin{array}{ccc} \{SgH \mid g \in G\} & \longrightarrow & \{yH \mid y \in X\} \\ SgH & \longmapsto & xgH \end{array}$$

between the set of S-H-double cosets and the set of H-suborbits is well-defined and is a bijection.

Proof:

- $SgH = Sg'H$ iff $g' = sgh$ for some $s \in S$, $h \in H$.
- In that case $xgH = xsghH$. Thus F is well-defined.
- If $xgH = xg'H$, then there is an $h \in H$ such that $xgh = xg'$ and thus ghg'^{-1} fixes x and lies in S. Thus $g' = sgh$ for some $s \in S$ and some $h \in H$ and F is injective.
- F is surjective since the action is transitive.

Introduction
GAP examples
The orbit algorithm
Computing the stabiliser
Double cosets
Orbits by suborbits
Storing suborbits
Orbit-by-Suborbits
Finding homomorphisms
Problems

Storing U-suborbits

$U < G$ a helper subgroup $\longrightarrow$ archive U-suborbits!

Storing U-suborbits

$U < G$ a helper subgroup $\longrightarrow$ archive U-suborbits!

We want:

- given $x \in X$, store xU and compute $|xU|$

Storing U-suborbits

$U < G$ a helper subgroup $\longrightarrow$ archive U-suborbits!

We want:

- given $x \in X$, store xU and compute $|xU|$
- given $z \in X$, decide whether z lies in a stored xU

Introduction
GAP examples
The orbit algorithm
Computing the stabiliser
Double cosets
Orbits by suborbits
Storing suborbits
Orbit-by-Suborbits
Finding homomorphisms
Problems

Storing U-suborbits

$U < G$ a helper subgroup $\longrightarrow$ archive U-suborbits!

We want:

- given $x \in X$, store xU and compute $|xU|$
- given $z \in X$, decide whether z lies in a stored xU

To this end, let $\bar{}: X \to Y$ be a homomorphism of U-sets:

Storing U-suborbits

$U < G$ a helper subgroup $\longrightarrow$ archive U-suborbits!

We want:

- given $x \in X$, store xU and compute $|xU|$
- given $z \in X$, decide whether z lies in a stored xU

To this end, let $\bar{\ } : X \to Y$ be a homomorphism of U-sets:

- enumerate Y completely

Storing U-suborbits

$U < G$ a helper subgroup $\longrightarrow$ archive U-suborbits!

We want:

- given $x \in X$, store xU and compute $|xU|$
- given $z \in X$, decide whether z lies in a stored xU

To this end, let $\bar{\ } : X \to Y$ be a homomorphism of U-sets:

- enumerate Y completely
- choose one element in each U-orbit of Y arbitrarily

Introduction
GAP examples
The orbit algorithm
Computing the stabiliser
Double cosets
Orbits by suborbits
Storing suborbits
Orbit-by-Suborbits
Finding homomorphisms
Problems

Storing U-suborbits

$U < G$ a helper subgroup $\longrightarrow$ archive U-suborbits!

We want:

- given $x \in X$, store xU and compute $|xU|$
- given $z \in X$, decide whether z lies in a stored xU

To this end, let $\bar{\ } : X \to Y$ be a homomorphism of U-sets:

- enumerate Y completely
- choose one element in each U-orbit of Y arbitrarily
- call these U-minimal

Storing U-suborbits

$U < G$ a helper subgroup $\longrightarrow$ archive U-suborbits!

We want:

- given $x \in X$, store xU and compute $|xU|$
- given $z \in X$, decide whether z lies in a stored xU

To this end, let $\bar{\ } : X \to Y$ be a homomorphism of U-sets:

- enumerate Y completely
- choose one element in each U-orbit of Y arbitrarily
- call these U-minimal
- for $y \in Y$, store a $u_y \in U$ such that yu_y is U-minimal

Introduction

GAP examples

The orbit algorithm

Computing the stabiliser

Double cosets

Orbits by suborbits

Storing suborbits

Orbit-by-Suborbits

Finding homomorphisms

Problems

Storing U-suborbits

$U < G$ a helper subgroup $\longrightarrow$ archive U-suborbits!

We want:

- given $x \in X$, store xU and compute $|xU|$
- given $z \in X$, decide whether z lies in a stored xU

To this end, let $\bar{\ } : X \to Y$ be a homomorphism of U-sets:

- enumerate Y completely
- choose one element in each U-orbit of Y arbitrarily
- call these U-minimal
- for $y \in Y$, store a $u_y \in U$ such that yu_y is U-minimal
- for U-minimal $y \in Y$, store generators of $\mathrm{Stab}_U(y)$

Storing U-suborbits

$U < G$ a helper subgroup $\longrightarrow$ archive U-suborbits!

We want:

- given $x \in X$, store xU and compute $|xU|$
- given $z \in X$, decide whether z lies in a stored xU

To this end, let $\bar{ }: X \to Y$ be a homomorphism of U-sets:

- enumerate Y completely
- choose one element in each U-orbit of Y arbitrarily
- call these U-minimal
- for $y \in Y$, store a $u_y \in U$ such that yu_y is U-minimal
- for U-minimal $y \in Y$, store generators of $\mathrm{Stab}_U(y)$
- call $x \in X$ U-minimal, if $\bar{x} \in Y$ is U-minimal

Introduction
GAP examples
The orbit algorithm
Computing the stabiliser
Double cosets
Orbits by suborbits
Storing suborbits
Orbit-by-Suborbits
Finding homomorphisms
Problems

Storing U-suborbits

$U < G$ a helper subgroup $\longrightarrow$ archive U-suborbits!

We want:

- given $x \in X$, store xU and compute $|xU|$
- given $z \in X$, decide whether z lies in a stored xU

To this end, let $\bar{~} : X \to Y$ be a homomorphism of U-sets:

- enumerate Y completely
- choose one element in each U-orbit of Y arbitrarily
- call these U-minimal
- for $y \in Y$, store a $u_y \in U$ such that yu_y is U-minimal
- for U-minimal $y \in Y$, store generators of $\mathrm{Stab}_U(y)$
- call $x \in X$ U-minimal, if $\bar{x} \in Y$ is U-minimal

Algorithm

Store xU by storing all U-minimal elements in xU.

Introduction
GAP examples
The orbit algorithm
Computing the stabiliser
Double cosets
Orbits by suborbits
Storing suborbits
Orbit-by-Suborbits
Finding homomorphisms
Problems

Storing U-suborbits II

If $x \in X$ is U-minimal (i.e. $\bar{x} \in Y$ is U-minimal), then

Storing U-suborbits II

If $x \in X$ is U-minimal (i.e. $\bar{x} \in Y$ is U-minimal), then

$x\mathrm{Stab}_U(\bar{x})$ is the set of U-minimal elements in xU.

Storing U-suborbits II

If $x \in X$ is U-minimal (i.e. $\bar{x} \in Y$ is U-minimal), then

$x\mathrm{Stab}_U(\bar{x})$ is the set of U-minimal elements in xU.

Algorithm (Storing xU)

Input: $x \in X$
look up $u_{\bar{x}}$ and compute $z := xu_{\bar{x}}$
enumerate and store $z\mathrm{Stab}_U(\bar{z})$
find $\mathrm{Stab}_U(z) \leq \mathrm{Stab}_U(\bar{z})$ and thus $|zU| = |xU|$

Storing U-suborbits II

If $x \in X$ is U-minimal (i.e. $\bar{x} \in Y$ is U-minimal), then

$x\mathrm{Stab}_U(\bar{x})$ is the set of U-minimal elements in xU.

Algorithm (Storing xU)

Input: $x \in X$
look up $u_{\bar{x}}$ and compute $z := xu_{\bar{x}}$
enumerate and store $z\mathrm{Stab}_U(\bar{z})$
find $\mathrm{Stab}_U(z) \leq \mathrm{Stab}_U(\bar{z})$ and thus $|zU| = |xU|$

Algorithm (Looking up $z \in X$)

Input: $z \in X$, some stored xU
look up $u_{\bar{z}}$ and compute $w := zu_{\bar{z}}$
look up w in list of stored points
$z \in xU$ iff w already stored

Orbit by suborbits

Algorithm (Orbit by suborbits)

Input: $G = \langle g_1, \ldots, g_r \rangle$ acting on X, $x \in X$
store xU and set $l := [x]$
repeat forever:
    for z in l:
        for g in $[g_1, \ldots, g_r]$:
            if zgU already stored:
                compute stabiliser element
            else:
                store zgU
                append zg to l
    exit if orbit and stabiliser ready

Output: l, U-suborbits, generators for $\mathrm{Stab}_G(x)$

Orbit by suborbits

Algorithm (Orbit by suborbits)

Input: $G = \langle g_1, \ldots, g_r \rangle$ acting on X, $x \in X$
store xU and set $l := [x]$
repeat forever:
    for z in l:
        for g in $[g_1, \ldots, g_r]$:
            if zgU already stored:
                compute stabiliser element
            else:
                store zgU
                append zg to l
    exit if orbit and stabiliser ready
    for z in l:
        for u in generators of U:
            append zu to l
Output: l, U-suborbits, generators for $\mathrm{Stab}_G(x)$

Finding homomorphisms

Let G act linearly on a F-vectorspace M:

$$\rho : G \to \mathrm{End}_F(M)$$

Introduction
GAP examples
The orbit algorithm
Computing the stabiliser
Double cosets
Orbits by suborbits
Storing suborbits
Orbit-by-Suborbits
Finding homomorphisms
Problems

Finding homomorphisms

Let G act linearly on a F-vectorspace M:

$$\rho : G \to \mathrm{End}_F(M)$$

$N < M$ a G-invariant subspace,
$\pi : M \to M/N$ the canonical map.

Introduction
GAP examples
The orbit algorithm
Computing the stabiliser
Double cosets
Orbits by suborbits
Storing suborbits
Orbit-by-Suborbits
Finding homomorphisms
Problems

Finding homomorphisms

Let G act linearly on a F-vectorspace M:

$$\rho : G \to \mathrm{End}_F(M)$$

$N < M$ a G-invariant subspace,
$\pi : M \to M/N$ the canonical map.

Then the following diagram commutes for all $g \in G$:

$$\begin{tikzcd} M \arrow[r, "\cdot g"] \arrow[d, "\pi"'] & M \arrow[d, "\pi"] \\ M/N \arrow[r, "\cdot g"] & M/N \end{tikzcd}$$

with the induced action on M/N.

Introduction
GAP examples
The orbit algorithm
Computing the stabiliser
Double cosets
Orbits by suborbits
Storing suborbits
Orbit-by-Suborbits
Finding homomorphisms
Problems

Finding homomorphisms

Let G act linearly on a F-vectorspace M:

$$\rho : G \to \mathrm{End}_F(M)$$

$N < M$ a G-invariant subspace,
$\pi : M \to M/N$ the canonical map.

Then the following diagram commutes for all $g \in G$:

$$\begin{tikzcd} M \arrow[r, "\cdot g"] \arrow[d, "\pi"'] & M \arrow[d, "\pi"] \\ M/N \arrow[r, "\cdot g"] & M/N \end{tikzcd}$$

with the induced action on M/N.

The same holds for the projective action, if we replace

- M by $\mathbb{P}(M)$ and
- $\mathbb{P}(M/N)$ by $\mathbb{P}(M/N) \cup \{0\}$.

Introduction

GAP examples

The orbit algorithm

Computing the stabiliser

Double cosets

Orbits by suborbits

Storing suborbits

Orbit-by-Suborbits

Finding homomorphisms

Problems

Advantages and Problems

Advantages:

- Saves both time and space.

Introduction

GAP examples

The orbit algorithm

Computing the stabiliser

Double cosets

Orbits by suborbits

Storing suborbits

Orbit-by-Suborbits

Finding homomorphisms

Problems

Advantages and Problems

Advantages:

- Saves both time and space.
- Can be iterated to use a chain of helper subgroups.

Introduction
GAP examples
The orbit algorithm
Computing the stabiliser
Double cosets
Orbits by suborbits
Storing suborbits
Orbit-by-Suborbits
Finding homomorphisms
Problems

Advantages and Problems

Advantages:

- Saves both time and space.
- Can be iterated to use a chain of helper subgroups.
- Still provides a sort of Schreier tree.

Introduction

GAP examples

The orbit algorithm

Computing the stabiliser

Double cosets

Orbits by suborbits

Storing suborbits

Orbit-by-Suborbits

Finding homomorphisms

Problems

Advantages and Problems

Advantages:

- Saves both time and space.
- Can be iterated to use a chain of helper subgroups.
- Still provides a sort of Schreier tree.
- Gives access to giant orbits.

Introduction
GAP examples
The orbit algorithm
Computing the stabiliser
Double cosets
Orbits by suborbits
Storing suborbits
Orbit-by-Suborbits
Finding homomorphisms
Problems

Advantages and Problems

Advantages:

- Saves both time and space.
- Can be iterated to use a chain of helper subgroups.
- Still provides a sort of Schreier tree.
- Gives access to giant orbits.

Problems:

- Needs helper subgroup U.

Introduction

GAP examples

The orbit algorithm

Computing the stabiliser

Double cosets

Orbits by suborbits

Storing suborbits

Orbit-by-Suborbits

Finding homomorphisms

Problems

Advantages and Problems

Advantages:

- Saves both time and space.
- Can be iterated to use a chain of helper subgroups.
- Still provides a sort of Schreier tree.
- Gives access to giant orbits.

Problems:

- Needs helper subgroup U.
- Needs homomorphism.

Advantages and Problems

Advantages:

- Saves both time and space.
- Can be iterated to use a chain of helper subgroups.
- Still provides a sort of Schreier tree.
- Gives access to giant orbits.

Problems:

- Needs helper subgroup U.
- Needs homomorphism.
- Needs mostly manual search and preparation.

Introduction

GAP examples

The orbit algorithm

Computing the stabiliser

Double cosets

Orbits by suborbits
- Storing suborbits
- Orbit-by-Suborbits
- Finding homomorphisms
- Problems

Advantages and Problems

Advantages:

- Saves both time and space.
- Can be iterated to use a chain of helper subgroups.
- Still provides a sort of Schreier tree.
- Gives access to giant orbits.

Problems:

- Needs helper subgroup U.
- Needs homomorphism.
- Needs mostly manual search and preparation.
- Sometimes helper subgroups do not exist.

Introduction
GAP examples
The orbit algorithm
Computing the stabiliser
Double cosets
Orbits by suborbits
Storing suborbits
Orbit-by-Suborbits
Finding homomorphisms
Problems

The End

Bibliography

- Frank Lübeck and Max Neunhöffer.
  Enumerating large orbits and direct condensation.
  Experiment. Math., 10(2):197–205, 2001.

- Jürgen Müller, Max Neunhöffer, and Robert A. Wilson.
  Enumerating big orbits and an application: *B* acting on the cosets of Fi_{23}.
  J. Algebra, 314(1):75–96, 2007.