# Algorithmic Nearring Theory

## Current Developments

Franz Binder

Department of Algebra
Johannes Kepler University Linz, Austria

Conference on

Nearrings, Nearfields,
and related topics

Vorau 2009

# Outline

1. General Ideas
   - Why Algorithmic?
   - Chain Conditiions
   - Computing Orbits

2. Finite Nearrings
   - Finite Transformations Nearrings
   - Compute Additive Generators
   - Use Nearring Generators Directly

# Why Algorithmic?

- Because its modern.
- Some computations may have to get ideas for conjectures.
- Modern technology provides us with the necessary computing power. We shoud use it.
- Essentially all applications of mathematics use computation.
- Algorithmic theories are a useful refinement of classical ones.

# Chain Conditions

## Definition

An ordered set has ACC iff

- for each ascending chain $a_1 \leq a_2 \leq a_3 \leq \dots$, there is $n \in \mathbb{N}$ such that $a_n = a_{n+1} = a_{n+2} = \dots$.
- for each ascending chain $a_1 \leq a_2 \leq a_3 \leq \dots$, there is $n \in \mathbb{N}$ such that $a_n = a_{n+1}$.
- if there is a proper ascending chains $a_1 < a_2 < a_3 < \dots$, then we get a contradiction.

## Example

Consider the 2-element lattice, $0 < 1$, and the chain starting with $0 \leq 0 \leq 0 \leq 0 \leq 0 \leq 0 \leq \dots$

- It is undecidable whether it will move up to 1 eventually.
- We can easily construct two equal elements in this chain.
- Negative statements contain no construction.

# Use ACC for Algorithms

## Proposition

*Let*

- $\mathscr{C}$ *be a class of subgroups with ACC;*
- $T : \mathscr{C} \to \mathscr{C}$*, monotone;*
- $\mathscr{D} = \{ H \in \mathscr{C} \mid T(H) = H \}$*;*
- $H_0 \in \mathscr{C}$.

*Then we can compute the smallest element of* $\mathscr{D}$ *containing* $H_0$.

## Proof.

We define:
$H_{n+1} = T(H_n)$.
Then $H_n \subseteq H_{n+1}$,
By ACC, there is some $n$ such that $H_n = H_{n+1}$, which is the solution. □

## Remark

In situations like this, we need the first place of equality in the ascending chain.

# Computing Orbits in Nearring Modules

## Example

Let $R$ be a nearring acting on a group $G$ such that

- $R$ is generated by a finite set $E$;
- $\langle H \cup H^e \rangle$ is f.g.,for each f.g. subgroup $H$ of $G$, and for each $e \in E$;
- $G$ has ACC for f.g. subgroups.

Then we can compute the orbit $g^N$, for each $g \in G$.

## Proof.

We start with
$H_0 = \langle g^E \rangle = \langle g^{e_1}, \dots, g^{e_n} \rangle$.
Using $T(H) = \langle H \cup H^E \rangle = \langle H \cup H^{e_1} \rangle \cup \dots \cup \langle H \cup H^{e_n} \rangle$
we note that $g^N$ is the smallest $T$-invariant subgroup containing $H_0$.
Thus it can be computed using ACC for f.g. subgroups. $\square$

# Finite Transformation Nearrings.

## Problem

Let

- $G$ be a small group;
- $E$ be a small set of transformations on $G$;
- $R$ be the nearring generated by $E$.

Compute information about $R$ without looping over a big set.

## Membership Problem

Given $f \in M(G)$, decide whether $f \in R$ or $f \notin R$.

## Convention

small: Your age

big: Any (your age)-digit number

## Remark

$M(G)$ is big.

# Compute Additive Generators.

## Remark

- Algorithmic group theory ist well established and implemented (e.g. in GAP).
- If we have generators of the additive group of the nearring $R$, we can solve the membership problem, determine its size, etc.

## Proposition

*If*

- *$e$ is an endomorphism of $G$;*
- *$H \leq G$;*
- *$H$ is generated by $F$;*

*Then $\langle H \cup H^e \rangle = \langle F \cup F^e \rangle$.*

## Corollary

*It is easy to compute additive generators of d.g. transformation nearrings.*

# Nearly D.g. Nearrings.

## Proposition

*If each generator e of the nearring R has the property* $e(g_1 + g_2 + g_3) \in \langle e(g_1+g_2), e(g_1+g_3), e(g_2+g_3), e(g_1), e(g_2), e(g_3), e_0\rangle$, *then it is also rather easy to compute additive generators.*

## Remark

All quadratic polynomials over rings have this property.

## Remark

- This approach is not always applicable.
- Even if it is, it will never help us to deal with nearrings that are too for being mastered by general group methods.
- A method to find better generators (of "lower degree") would be helpful.
- We do not have even a method to compute distributive generators of a d.g. nearring given by non-distributive generators.

# Use Nearring Generators Directly.

## Remark

Let $G$ be a finite group, and $N \leq M(G)$ generated by $E$.

- $G$ is an $R$-module, in the natural way;
- The same for $G \times G$, $G^3, \dots$.
- We can compute efficiently $g^R$ (orbits), $(g, h)^R$ (interpolation), ....
- These $R$-modules are in fact just small groups with operators.

## Theorem

*The knowledge of these $R$-modules can be transformed back to solve some problems about $R$:*

- *2-interpolation property?*
- *is $R$ tame on $G$?*
- *Is $R$ distributive?*
- *Is $R$ a ring?*
- *Is $R = M(G)$ ?*
- *Is $R$ 2-primitive on $G$?*
- *Do these apply to the zero-symmetric part of $R$?*

# Generators for the Pierce Decomposition.

## Theorem

*Let $R$ be a nearring generated by $E$, and $i$ an idempotent. Then*
$R^+ = Z \ltimes_R K$
*where* $K = \{ ir \mid r \in R \}$,
$Z = \{ r \mid ir = 0 \}$, *and*
$K \leq_R R^+$, $Z \trianglelefteq_R R^+$.
*In addition,*

- *$K$ is generated by $\{ ie \mid e \in E \}$ as an $R$-subgroup;*
- *$Z$ is generated by $\{ e - ie \mid e \in E \}$ as a right ideal.*

## Remark

We can compute orbits like $g^K$ or $g^Z$ using just these generators.

## Remark

We can repeat this step.

# Summary

- Studying algorithms provides a useful refinement of classical theories.
- A lot of non so obvious algorithms for nearrings have been developed and implemented in SONATA.
- Finding algorithms does not necessarily mean to deal with machine models and possible deficiencies of present computers.

- Outlook
  - Algorithms for infinite nearrings have not yet been studied systematically.
  - Some important problems (like membership) still open.
  - Help welcome!

# Already Published I


- F. Binder, P. Mayr: Algorithms for finite near-rings and their $N$-gorups.
  *Journal of Symbolic Computation*.
  2000.
- Sonata-Team:
  Algorithms for Near-rings of Non-linear Transformation.
  *ISSAC 2000*, ACM, 2000.
- Sonata-Team:
  SONATA, System Of Nearrings And Their Applications.
  www.algebra.uni-linz.ac.at/sonata