

# The Andrews-Curtis Conjecture

Gilbert Baumslag, Bruno Buchberger, Timothy Daly

March 16, 2005

**Abstract**

# Contents

<b>1</b>	<b>The Problem</b>	<b>3</b>
<b>2</b>	<b>The Representation</b>	<b>3</b>
2.1	The inverse function <i>inv</i> . . . . .	4
2.2	The pinch function <i>pinch</i> . . . . .	4
2.3	The Andrews-Curtis Transformation functions . . . . .	5
2.3.1	T1: $x \rightarrow xy ; y \rightarrow y$ . . . . .	5
2.3.2	T2: $x \rightarrow X ; y \rightarrow y$ . . . . .	5
2.3.3	T3: $x \rightarrow axA ; y \rightarrow y$ . . . . .	5
2.3.4	T4: $x \rightarrow x ; y \rightarrow yx$ . . . . .	6
2.3.5	T5: $x \rightarrow x ; y \rightarrow Y$ . . . . .	6
2.3.6	T6: $x \rightarrow x ; y \rightarrow ayA$ . . . . .	6
2.3.7	T7: $x \rightarrow bxB ; y \rightarrow y$ . . . . .	6
2.3.8	T8: $x \rightarrow x ; y \rightarrow byB$ . . . . .	7
2.3.9	T9: $x \rightarrow Axa ; y \rightarrow y$ . . . . .	7
2.3.10	T10: $x \rightarrow x ; y \rightarrow Aya$ . . . . .	7
2.3.11	T11: $x \rightarrow Bxb ; y \rightarrow y$ . . . . .	7
2.3.12	T12: $x \rightarrow x ; y \rightarrow Byb$ . . . . .	8
<b>3</b>	<b>First Steps</b>	<b>8</b>
3.1	Apply each transformation . . . . .	8
3.2	Compose T1 with all . . . . .	8
3.3	Compose T2 with all . . . . .	9
3.4	Compose T3 with all . . . . .	9
3.5	Compose T4 with all . . . . .	10
3.6	Compose T5 with all . . . . .	10
3.7	Compose T6 with all . . . . .	11
3.8	Compose T7 with all . . . . .	11
3.9	Compose T8 with all . . . . .	11
3.10	Compose T9 with all . . . . .	12
3.11	Compose T10 with all . . . . .	12
3.12	Compose T11 with all . . . . .	13
3.13	Compose T12 with all . . . . .	13
<b>4</b>	<b>A Matrix Representation</b>	<b>14</b>
<b>5</b>	<b>Multiplicative Invariants of Special 2-Complexes</b>	<b>15</b>
<b>6</b>	<b>This Week's Finds in Mathematical Physics</b>	<b>15</b>
<b>7</b>	<b>A potential smooth counterexample in dimension 4 . . .</b>	<b>20</b>
<b>8</b>	<b>Canada Research Chair in Combinatorial Algebra</b>	<b>20</b>
<b>9</b>	<b>Balanced presentations of the trivial group on . . .</b>	<b>21</b>

10 Genetic algorithms and the Andrews-Curtis conjecture	21
11 A remark on the stable Andrews-Curtis conjecture	22
12 Algebraic Geometry over Groups	23
13 Representation theory, topological field theory, and...	27

## 1 The Problem

We are designing this program to work with the exact presentation:

$$G = \langle a, b ; abaBAB, AAbbb \rangle$$

where lower case letters are generators and uppercase letters are their inverses. This presentation is known to be trivial. That is, it is known to be exactly:

$$F = \langle a, b ; \rangle$$

We are given 12 transformations. Each transformation specifies a way to rewrite the two relators. If the first relator is called  $x$ , its inverse is  $X$ , the second relator is called  $y$ , and its inverse is  $Y$  then we have the Andrews-Curtis Transformations:

$$\mathbf{T1: } x \rightarrow xy ; y \rightarrow y$$

$$\mathbf{T2: } x \rightarrow X ; y \rightarrow y$$

$$\mathbf{T3: } x \rightarrow axA ; y \rightarrow y$$

$$\mathbf{T4: } x \rightarrow x ; y \rightarrow yx$$

$$\mathbf{T5: } x \rightarrow x ; y \rightarrow Y$$

$$\mathbf{T6: } x \rightarrow x ; y \rightarrow ayA$$

$$\mathbf{T7: } x \rightarrow bxB ; y \rightarrow y$$

$$\mathbf{T8: } x \rightarrow x ; y \rightarrow byB$$

$$\mathbf{T9: } x \rightarrow Axa ; y \rightarrow y$$

$$\mathbf{T10: } x \rightarrow x ; y \rightarrow Aya$$

$$\mathbf{T11: } x \rightarrow Bxb ; y \rightarrow y$$

$$\mathbf{T12: } x \rightarrow x ; y \rightarrow Byb$$

## 2 The Representation

In order to implement these transformations we keep both relators in one structure we call a **tuple**. In this implementation a **tuple** is represented as a lisp cons cell. The car of the cell is a string representing the first relator. The cdr of the cell is a string representing the second relator.

Each of the twelve transformation functions takes this pair of relators and returns a new pair containing the transformation. This generates a lot of cons cells.

A more efficient, but not yet implemented, implementation will be to destructively modify the tuple and reuse the cons cell. A further optimization is to keep a pool of strings of a given length and reuse the string storage. This is future work.

## 2.1 The inverse function *inv*

We need to be able to calculate the inverse of a given generator. So the inverse of 'a' is 'A', 'A' is 'a', 'b' is 'B' and 'B' is 'b'. This function, given a string composed of generators rewrites the string by reversing the string and doing a generator by generator replacement.

```
(defun inv (word)
  "given a word, construct its inverse"
  (let ((result (reverse word)))
    (dotimes (i (length result))
      (setf (elt result i)
            (cond
              ((eq (elt result i) #\a) #\A)
              ((eq (elt result i) #\b) #\B)
              ((eq (elt result i) #\A) #\a)
              ((eq (elt result i) #\B) #\b))))
    result))
```

## 2.2 The pinch function *pinch*

We need to elide generators which are concatenated to their inverse. So the combinations 'aA', 'Aa', 'bB' and 'Bb' disappear from the string. We stack the first generator in the string and then compare it with the second generator. If the next element is an inverse we pop the generator from the stack. Otherwise, we push the next generator from the string and continue the process.

For words of length 0 or 1 there can be no pinching so we simply return the word.

Note that this operation constructs a new word. An optimization is to make the word process destructive to save space. This is not yet implemented.

```
(defun pinch (word)
  "given a word pinch out all inverse elements"
  (let ((stack nil) (wlen (- (length word) 1)))
    (if (> wlen 0)
        (progn
          (push (elt word 0) stack)
          (dotimes (i wlen)
            (cond
              ((and (eq (first stack) #\a) (eq (elt word (+ i 1)) #\A)) (pop stack))
              ((and (eq (first stack) #\A) (eq (elt word (+ i 1)) #\a)) (pop stack))
              ((and (eq (first stack) #\b) (eq (elt word (+ i 1)) #\B)) (pop stack))
              ((and (eq (first stack) #\B) (eq (elt word (+ i 1)) #\b)) (pop stack))
              (t (push (elt word (+ i 1)) stack))))
          (concatenate 'string (reverse stack)))
        word)))
```

## 2.3 The Andrews-Curtis Transformation functions

These transformations have been rewritten. One optimization is that each transformation used to construct a new cons cell. We eliminate this waste by reusing the cons cell. The `[[rplaca]]` function replaces the first element of the `[[cons]]` cell (known as the `[[car]]`) "in place". Similarly, the `[[rplacd]]` function replaces the second element of the `[[cons]]` cell (known as the `[[cdr]]`) "in place". This optimization saves both space because we allocate less memory, and time because we don't have to reclaim memory we never use.

### 2.3.1 T1: $x \rightarrow xy ; y \rightarrow y$

```
(defun t1 (tuple)
  "x -> xy ; y -> y"
  (cons
    (pinch (concatenate 'string (car tuple) (cdr tuple)))
    (cdr tuple)))

(defun t1 (tuple)
  "x -> xy ; y -> y"
  (rplaca tuple (pinch (concatenate 'string (car tuple) (cdr tuple)))))
```

### 2.3.2 T2: $x \rightarrow X ; y \rightarrow y$

```
(defun t2 (tuple)
  "x -> X ; y -> y"
  (cons
    (inv (car tuple))
    (cdr tuple)))

(defun t2 (tuple)
  "x -> X ; y -> y"
  (rplaca tuple (inv (car tuple))))
```

### 2.3.3 T3: $x \rightarrow axA ; y \rightarrow y$

```
(defun t3 (tuple)
  "x -> axA ; y -> y"
  (cons
    (pinch (concatenate 'string "a" (car tuple) "A"))
    (cdr tuple)))

(defun t3 (tuple)
  "x -> axA ; y -> y"
  (rplaca tuple (pinch (concatenate 'string "a" (car tuple) "A"))))
```

**2.3.4** T4:  $x \rightarrow x$  ;  $y \rightarrow yx$

```
(defun t4 (tuple)
  "x -> x ; y -> yx"
  (cons
    (car tuple)
    (pinch (concatenate 'string (cdr tuple) (car tuple)))))
```

```
(defun t4 (tuple)
  "x -> x ; y -> yx"
  (rplacd tuple (pinch (concatenate 'string (cdr tuple) (car tuple)))))
```

**2.3.5** T5:  $x \rightarrow x$  ;  $y \rightarrow Y$

```
(defun t5 (tuple)
  "x -> x ; y -> Y"
  (cons
    (car tuple)
    (inv (cdr tuple))))
```

```
(defun t5 (tuple)
  "x -> x ; y -> Y"
  (rplacd tuple (inv (cdr tuple))))
```

**2.3.6** T6:  $x \rightarrow x$  ;  $y \rightarrow ayA$

```
(defun t6 (tuple)
  "x -> x ; y -> ayA"
  (cons
    (car tuple)
    (pinch (concatenate 'string "a" (cdr tuple) "A"))))
```

```
(defun t6 (tuple)
  "x -> x ; y -> ayA"
  (rplacd tuple (pinch (concatenate 'string "a" (cdr tuple) "A"))))
```

**2.3.7** T7:  $x \rightarrow bxB$  ;  $y \rightarrow y$

```
(defun t7 (tuple)
  "x -> bxB ; y -> y"
  (cons
    (pinch (concatenate 'string "b" (car tuple) "B"))
    (cdr tuple)))
```

```
(defun t7 (tuple)
  "x -> bxB ; y -> y"
  (rplaca tuple (pinch (concatenate 'string "b" (car tuple) "B"))))
```

**2.3.8** T8: x -> x ; y -> byB

```
(defun t8 (tuple)
  "x -> x ; y -> byB"
  (cons
    (car tuple)
    (pinch (concatenate 'string "b" (cdr tuple) "B"))))
```

```
(defun t8 (tuple)
  "x -> x ; y -> byB"
  (rplacd tuple (pinch (concatenate 'string "b" (cdr tuple) "B"))))
```

**2.3.9** T9: x -> Axa ; y -> y

```
(defun t9 (tuple)
  "x -> Axa ; y -> y"
  (cons
    (pinch (concatenate 'string "A" (car tuple) "a"))
    (cdr tuple)))
```

```
(defun t9 (tuple)
  "x -> Axa ; y -> y"
  (rplaca tuple (pinch (concatenate 'string "A" (car tuple) "a"))))
```

**2.3.10** T10: x -> x ; y -> Aya

```
(defun t10 (tuple)
  "x -> x ; y -> Aya"
  (cons
    (car tuple)
    (pinch (concatenate 'string "A" (cdr tuple) "a"))))
```

```
(defun t10 (tuple)
  "x -> x ; y -> Aya"
  (rplacd tuple (pinch (concatenate 'string "A" (cdr tuple) "a"))))
```

**2.3.11** T11: x -> Bxb ; y -> y

```
(defun t11 (tuple)
  "x -> Bxb ; y -> y"
  (cons
    (pinch (concatenate 'string "B" (car tuple) "b"))
    (cdr tuple)))
```

```
(defun t11 (tuple)
  "x -> Bxb ; y -> y"
  (rplaca tuple (pinch (concatenate 'string "B" (car tuple) "b"))))
```



**2.3.12** T12: `x -> x ; y -> Byb`

```
(defun t12 (tuple)
  "x -> x ; y -> Byb"
  (cons
    (car tuple)
    (pinch (concatenate 'string "B" (cdr tuple) "b"))))
```

```
(defun t12 (tuple)
  "x -> x ; y -> Byb"
  (rplacd tuple (pinch (concatenate 'string "B" (cdr tuple) "b"))))
```

## 3 First Steps

### 3.1 Apply each transformation

If we create a tuple from the two relators thus:

```
(setq r (cons "abaBAB" "AAbbb"))
```

and call each transformation on `r` thus:

```
(dolist (x '(t1 t2 t3 t4 t5 t6 t7 t8 t9 t10 t11 t12))
  (print (list x (funcall x r))))
```

we get the results:

```
(T1 ("abaBABAAbbb" . "AAbbb"))
(T2 ("babABA" . "AAbbb"))
(T3 ("aabaBABA" . "AAbbb"))
(T4 ("abaBAB" . "AAbbbabaBAB"))
(T5 ("abaBAB" . "BBBaa"))
(T6 ("abaBAB" . "AbbbA"))
(T7 ("babaBABB" . "AAbbb"))
(T8 ("abaBAB" . "bAAbb"))
(T9 ("baBABA" . "AAbbb"))
(T10 ("abaBAB" . "AAAbbba"))
(T11 ("BabaBA" . "AAbbb"))
(T12 ("abaBAB" . "BAAbbbb"))
```

### 3.2 Compose T1 with all

```
(dolist (x '(t1 t2 t3 t4 t5 t6 t7 t8 t9 t10 t11 t12))
  (print (list 't1 (list x (t1 (funcall x r))))))
```

We get the results:

```

(T1 (T1 ("abaBABAAbbbAAAbbb" . "AAAbbb")))
(T1 (T2 ("babABAAAbbb" . "AAAbbb")))
(T1 (T3 ("aabaBABAAbbb" . "AAAbbb")))
(T1 (T4 ("abaBABAAbbbabaBAB" . "AAAbbbabaBAB")))
(T1 (T5 ("abaBABBBBaa" . "BBBaa")))
(T1 (T6 ("abaBABAbbba" . "Abbba")))
(T1 (T7 ("babaBABBAAbbb" . "AAAbbb")))
(T1 (T8 ("abaBAAAbb" . "bAAbb")))
(T1 (T9 ("baBABAbbb" . "AAAbbb")))
(T1 (T10 ("abaBABAAbbba" . "AAAAbbba")))
(T1 (T11 ("BabaBAAAbbb" . "AAAbbb")))
(T1 (T12 ("abaBABBAAbbbb" . "BAAbbbb")))

```

### 3.3 Compose T2 with all

```

(dolist (x '(t1 t2 t3 t4 t5 t6 t7 t8 t9 t10 t11 t12))
  (print (list 't2 (list x (t2 (funcall x r))))))

```

We get the results:

```

(T2 (T1 ("BBBaababABA" . "AAAbbb")))
(T2 (T2 ("abaBAB" . "AAAbbb")))
(T2 (T3 ("ababABAA" . "AAAbbb")))
(T2 (T4 ("babABA" . "AAAbbbabaBAB")))
(T2 (T5 ("babABA" . "BBBaa")))
(T2 (T6 ("babABA" . "Abbba")))
(T2 (T7 ("bbabABAB" . "AAAbbb")))
(T2 (T8 ("babABA" . "bAAbb")))
(T2 (T9 ("AbabAB" . "AAAbbb")))
(T2 (T10 ("babABA" . "AAAAbbba")))
(T2 (T11 ("abABAb" . "AAAbbb")))
(T2 (T12 ("babABA" . "BAAbbbb")))

```

### 3.4 Compose T3 with all

```

(dolist (x '(t1 t2 t3 t4 t5 t6 t7 t8 t9 t10 t11 t12))
  (print (list 't3 (list x (t3 (funcall x r))))))

```

We get the results:

```

(T3 (T1 ("aabaBABAAbbba" . "AAAbbb")))
(T3 (T2 ("ababABAA" . "AAAbbb")))
(T3 (T3 ("aaabaBABA" . "AAAbbb")))
(T3 (T4 ("aabaBABA" . "AAAbbbabaBAB")))
(T3 (T5 ("aabaBABA" . "BBBaa")))
(T3 (T6 ("aabaBABA" . "Abbba")))
(T3 (T7 ("ababaBABBA" . "AAAbbb")))

```

```

(T3 (T8 ("aabaBABA" . "bAAAbb")))
(T3 (T9 ("abaBAB" . "AAbbb")))
(T3 (T10 ("aabaBABA" . "AAAbbba")))
(T3 (T11 ("aBabaBAA" . "AAbbb")))
(T3 (T12 ("aabaBABA" . "BAAbbbb")))

```

### 3.5 Compose T4 with all

```

(dolist (x '(t1 t2 t3 t4 t5 t6 t7 t8 t9 t10 t11 t12))
  (print (list 't4 (list x (t4 (funcall x r))))))

```

We get the results:

```

(T4 (T1 ("abaBABAAbbb" . "AAbbbabaBABAAbbb")))
(T4 (T2 ("babABA" . "AAbbbbabABA")))
(T4 (T3 ("aabaBABA" . "AAbbbaabaBABA")))
(T4 (T4 ("abaBAB" . "AAbbbabaBABabaBAB")))
(T4 (T5 ("abaBAB" . "BBBaaabaBAB")))
(T4 (T6 ("abaBAB" . "AbbbbabaBAB")))
(T4 (T7 ("babaBABB" . "AAbbbbabaBABB")))
(T4 (T8 ("abaBAB" . "bAAAbbabaBAB")))
(T4 (T9 ("baBABA" . "AAbbbbabaBABA")))
(T4 (T10 ("abaBAB" . "AAAbbbaabaBAB")))
(T4 (T11 ("BabaBA" . "AAbbabaBA")))
(T4 (T12 ("abaBAB" . "BAAbbbbabaBAB")))

```

### 3.6 Compose T5 with all

```

(dolist (x '(t1 t2 t3 t4 t5 t6 t7 t8 t9 t10 t11 t12))
  (print (list 't5 (list x (t5 (funcall x r))))))

```

We get the results:

```

(T5 (T1 ("abaBABAAbbb" . "BBBaa")))
(T5 (T2 ("babABA" . "BBBaa")))
(T5 (T3 ("aabaBABA" . "BBBaa")))
(T5 (T4 ("abaBAB" . "babABABBBaa")))
(T5 (T5 ("abaBAB" . "AAbbb")))
(T5 (T6 ("abaBAB" . "aBBBa")))
(T5 (T7 ("babaBABB" . "BBBaa")))
(T5 (T8 ("abaBAB" . "BBaaB")))
(T5 (T9 ("baBABA" . "BBBaa")))
(T5 (T10 ("abaBAB" . "ABBBaaa")))
(T5 (T11 ("BabaBA" . "BBBaa")))
(T5 (T12 ("abaBAB" . "BBBaaab")))

```

### 3.7 Compose T6 with all

```
(dolist (x '(t1 t2 t3 t4 t5 t6 t7 t8 t9 t10 t11 t12))
  (print (list 't6 (list x (t6 (funcall x r))))))
```

We get the results:

```
(T6 (T1 ("abaBABAAbbb" . "AbbbA")))
(T6 (T2 ("babABA" . "AbbbA")))
(T6 (T3 ("aabaBABA" . "AbbbA")))
(T6 (T4 ("abaBAB" . "AbbbabaBABA")))
(T6 (T5 ("abaBAB" . "aBBBa")))
(T6 (T6 ("abaBAB" . "bbbAA")))
(T6 (T7 ("babaBABB" . "AbbbA")))
(T6 (T8 ("abaBAB" . "abAAbbA")))
(T6 (T9 ("baBABa" . "AbbbA")))
(T6 (T10 ("abaBAB" . "AAbbb")))
(T6 (T11 ("BabaBA" . "AbbbA")))
(T6 (T12 ("abaBAB" . "aBAAbbbbA")))
```

### 3.8 Compose T7 with all

```
(dolist (x '(t1 t2 t3 t4 t5 t6 t7 t8 t9 t10 t11 t12))
  (print (list 't7 (list x (t7 (funcall x r))))))
```

We get the results:

```
(T7 (T1 ("babaBABAAbb" . "AAbbb")))
(T7 (T2 ("bbabABAB" . "AAbbb")))
(T7 (T3 ("baabaBABAB" . "AAbbb")))
(T7 (T4 ("babaBABB" . "AAbbbabaBAB")))
(T7 (T5 ("babaBABB" . "BBBaa")))
(T7 (T6 ("babaBABB" . "AbbbA")))
(T7 (T7 ("bbabaBABBB" . "AAbbb")))
(T7 (T8 ("babaBABB" . "bAAbb")))
(T7 (T9 ("bbaBABaB" . "AAbbb")))
(T7 (T10 ("babaBABB" . "AAAbbba")))
(T7 (T11 ("abaBAB" . "AAbbb")))
(T7 (T12 ("babaBABB" . "BAAbbbb")))
```

### 3.9 Compose T8 with all

```
(dolist (x '(t1 t2 t3 t4 t5 t6 t7 t8 t9 t10 t11 t12))
  (print (list 't8 (list x (t8 (funcall x r))))))
```

We get the results:

```

(T8 (T1 ("abaBABAAbbb" . "bAAbb")))
(T8 (T2 ("babABA" . "bAAAbb")))
(T8 (T3 ("aabaBABA" . "bAAAbb")))
(T8 (T4 ("abaBAB" . "bAAAbbbabaBABB")))
(T8 (T5 ("abaBAB" . "BBaaB")))
(T8 (T6 ("abaBAB" . "bAbbbAB")))
(T8 (T7 ("babaBABB" . "bAAbb")))
(T8 (T8 ("abaBAB" . "bbAAAb")))
(T8 (T9 ("baBABA" . "bAAAbb")))
(T8 (T10 ("abaBAB" . "bAAAbbbbaB")))
(T8 (T11 ("BabaBA" . "bAAbb")))
(T8 (T12 ("abaBAB" . "AAAbbb")))

```

### 3.10 Compose T9 with all

```

(dolist (x '(t1 t2 t3 t4 t5 t6 t7 t8 t9 t10 t11 t12))
  (print (list 't9 (list x (t9 (funcall x r))))))

```

We get the results:

```

(T9 (T1 ("baBABAAbbba" . "AAAbbb")))
(T9 (T2 ("AbabAB" . "AAAbbb")))
(T9 (T3 ("abaBAB" . "AAAbbb")))
(T9 (T4 ("baBABA" . "AAAbbbabaBAB")))
(T9 (T5 ("baBABA" . "BBBaa")))
(T9 (T6 ("baBABA" . "AbbbA")))
(T9 (T7 ("AbabaBABBa" . "AAAbbb")))
(T9 (T8 ("baBABA" . "bAAAbb")))
(T9 (T9 ("AbaBABaa" . "AAAbbb")))
(T9 (T10 ("baBABA" . "AAAbbbba")))
(T9 (T11 ("ABabaB" . "AAAbbb")))
(T9 (T12 ("baBABA" . "BAAbbbb")))

```

### 3.11 Compose T10 with all

```

(dolist (x '(t1 t2 t3 t4 t5 t6 t7 t8 t9 t10 t11 t12))
  (print (list 't10 (list x (t10 (funcall x r))))))

```

We get the results:

```

(T10 (T1 ("abaBABAAbbb" . "AAAbbbba")))
(T10 (T2 ("babABA" . "AAAbbbba")))
(T10 (T3 ("aabaBABA" . "AAAbbbba")))
(T10 (T4 ("abaBAB" . "AAAbbbabaBABA")))
(T10 (T5 ("abaBAB" . "ABBBaaa")))

```

```

(T10 (T6 ("abaBAB" . "AAbbb")))
(T10 (T7 ("babaBABB" . "AAAAbba")))
(T10 (T8 ("abaBAB" . "AbAAbba")))
(T10 (T9 ("baBABa" . "AAAAbba")))
(T10 (T10 ("abaBAB" . "AAAAAbbaa")))
(T10 (T11 ("BabaBA" . "AAAAbba")))
(T10 (T12 ("abaBAB" . "ABAAbba")))

```

### 3.12 Compose T11 with all

```

(dolist (x '(t1 t2 t3 t4 t5 t6 t7 t8 t9 t10 t11 t12))
  (print (list 't11 (list x (t11 (funcall x r))))))

```

We get the results:

```

(T11 (T1 ("BabaBABAAbbbb" . "AAbbb")))
(T11 (T2 ("abABAb" . "AAbbb")))
(T11 (T3 ("BaabaBABA" . "AAbbb")))
(T11 (T4 ("BabaBA" . "AAAbbaBABA")))
(T11 (T5 ("BabaBA" . "BBBaa")))
(T11 (T6 ("BabaBA" . "AbbaA")))
(T11 (T7 ("abaBAB" . "AAbbb")))
(T11 (T8 ("BabaBA" . "bAAAb")))
(T11 (T9 ("aBABab" . "AAbbb")))
(T11 (T10 ("BabaBA" . "AAAAbba")))
(T11 (T11 ("BBabaBA" . "AAbbb")))
(T11 (T12 ("BabaBA" . "BAAbbbb")))

```

### 3.13 Compose T12 with all

```

(dolist (x '(t1 t2 t3 t4 t5 t6 t7 t8 t9 t10 t11 t12))
  (print (list 't12 (list x (t12 (funcall x r))))))

```

We get the results:

```

(T12 (T1 ("abaBABAAbbb" . "BAAbbbb")))
(T12 (T2 ("babABA" . "BAAbbbb")))
(T12 (T3 ("aabaBABA" . "BAAbbbb")))
(T12 (T4 ("abaBAB" . "BAAbbaBABA")))
(T12 (T5 ("abaBAB" . "BBBBaab")))
(T12 (T6 ("abaBAB" . "BAAbbaAb")))
(T12 (T7 ("babaBABB" . "BAAbbbb")))
(T12 (T8 ("abaBAB" . "AAbbb")))
(T12 (T9 ("baBABa" . "BAAbbbb")))
(T12 (T10 ("abaBAB" . "BAAAbbab")))

```

(T12 (T11 ("BabaBA" . "BAAbbbb")))
  
(T12 (T12 ("abaBAB" . "BBAAbbbb"))) )

## 4 A Matrix Representation

Given the group

$$G = \langle a, b; \text{abaBAB}, \text{AAbbb} \rangle$$

we can consider the case where the generators are given by 2 x 2 matrices with symbolic entries. So we allow

$$a = \begin{bmatrix} c & d \\ e & f \end{bmatrix} \text{ and } b = \begin{bmatrix} g & h \\ i & j \end{bmatrix}$$

and their inverses are:

$$A = \begin{bmatrix} \frac{f}{c f - d e} & -\frac{d}{c f - d e} \\ -\frac{e}{c f - d e} & \frac{c}{c f - d e} \end{bmatrix} \text{ and } B = \begin{bmatrix} \frac{j}{g j - h i} & -\frac{h}{g j - h i} \\ -\frac{i}{g j - h i} & \frac{g}{g j - h i} \end{bmatrix}$$

then by simple multiplication we know that the relators are:

$$\text{abaBAB} = \begin{bmatrix} c & d \\ e & f \end{bmatrix} \begin{bmatrix} g & h \\ i & j \end{bmatrix} \begin{bmatrix} c & d \\ e & f \end{bmatrix} \begin{bmatrix} \frac{j}{g j - h i} & -\frac{h}{g j - h i} \\ -\frac{i}{g j - h i} & \frac{g}{g j - h i} \end{bmatrix} \begin{bmatrix} \frac{f}{c f - d e} & -\frac{d}{c f - d e} \\ -\frac{e}{c f - d e} & \frac{c}{c f - d e} \end{bmatrix}$$

and

$$\text{AAbbb} = \begin{bmatrix} \frac{f}{c f - d e} & -\frac{d}{c f - d e} \\ -\frac{e}{c f - d e} & \frac{c}{c f - d e} \end{bmatrix} \begin{bmatrix} \frac{f}{c f - d e} & -\frac{d}{c f - d e} \\ -\frac{e}{c f - d e} & \frac{c}{c f - d e} \end{bmatrix} \begin{bmatrix} g & h \\ i & j \end{bmatrix} \begin{bmatrix} g & h \\ i & j \end{bmatrix} \begin{bmatrix} g & h \\ i & j \end{bmatrix}$$

or, in fully expanded form

$$\text{abaBAB} = \frac{d e f j^3 + ((-d f^2 + c d f + d^2 e) i + (c e f + d e^2) h + (-d e + c^2) f g) j^2 + ((-2 d^2 f + c d^2) i^2 + ((-c f^2 + 3 c d f - d^2) h + (-d e + c^2) g) i + (d e + c^2) g^2) i + (e f^2 j^3 + ((-f^3 + c f^2 + d e f) i + 2 e^2 f h + (-e f^2 + c e f) g) j^2 + ((-2 d f^2 + c d f) i^2 + ((-e f^2 + 2 c e f + d e^2) h + (-d e + c^2) g) i + (d e + c^2) g^2) i + (-e f - c e^2) g + (d e + c^2) g^2) i + (-e f - c e^2) g + (d e + c^2) g^2}{c^2 f^2 - 2 c d e f + d^2 e^2}$$

and

$$\text{AAbbb} = \frac{(-d f - c d) i j^2 + ((f^2 + d e) h + (-d f - c d) g) i j + (-d f - c d) h i^2 + ((2 f^2 + 2 d e) g h + (-d f - c d) g^2) i + (f^2 + d e) g^3}{c^2 f^2 - 2 c d e f + d^2 e^2} + \frac{(d e + c^2) i j^2 + ((-e f - c e) h + (d e + c^2) g) i j + (d e + c^2) h i^2 + ((-2 e f - 2 c e) g h + (d e + c^2) g^2) i + (-e f - c e) g + (d e + c^2) g^2}{c^2 f^2 - 2 c d e f + d^2 e^2}$$

## 5 Multiplicative Invariants of Special 2-Complexes

*Multiplicative Invariants of Special 2-Complexes [5]*

**John Rosson**(Portland State)

### ABSTRACT

This dissertation constructs total invariants to distinguish between simple homotopy classes of simply connected 2-complexes. Frank Quinn developed multiplicative invariants of 2 dimensional CW-complexes (2-complexes) by using concepts from Topological Quantum Field Theory. One motivation for this dissertation was to get invariants that could survive stabilization and therefore potentially distinguish Andrews-Curtis classes of 2-complexes. Such invariants would help resolve the well known Andrews Curtis Conjecture that no nontrivial Andrews Curtis Classes exist.

In this dissertation we follow Quinn's philosophy, but from the top down. We first construct a canonical decomposition of a special 2-complex. Second, we introduce a category of algebraic objects, called a double semigroup, which allows free constructions and quotients. Next, we build a particular double semigroup whose elements are a complete set of invariants for homeomorphism classes of special 2-complexes. Finally, we construct a double semigroup whose elements are a complete set of Andrews Curtis invariants.

Tuesday, June 11, 2002

### DISSERTATION COMMITTEE

M. Paul Latiolais, Chairman

Andrew M. Fraser

Joyce O'Halloran

Serge Preston

Erik Bodegom, Graduate Studies Rep.

## 6 This Week's Finds in Mathematical Physics

*This Week's Finds in Mathematical Physics (Week 23)[4]*

**John Baez**(University of California Riverside)

I will soon revert to my older style, in which I list piles of new papers as they accumulate on my desk. This time, though, I want to describe Frank Quinn's work on the Andrews-Curtis conjecture using topological quantum field theories (TQFTs), as promised. Then, if you'll pardon me, I'll list the contents of a book I've just finished editing. It is such a relief to be done that I cannot resist.

So -



Topological quantum invariants and the Andrews-Curtis conjecture (Progress report), by Frank Quinn, preprint, Sept. 1993.

Lectures on axiomatic topological quantum field theory, by Frank Quinn, to appear in the proceedings of the Park City Geometry Institute.

On the Andrews-Curtis conjecture and related problems, by Wolfgang Metzler, in Combinatorial Methods in Topology and Algebraic Geometry, Contemporary Mathematics 44, AMS, 1985.

Last week I described - in a pretty sketchy way - how the 4-color theorem and the Beraha conjecture are related to TQFTs. These can be regarded as two very hard problems in 2-dimensional topology - one solved by a mixture of cleverness and extreme brute force, the other still open. There is another hard problem in 2-dimensional topology called the Andrews-Curtis conjecture, which Quinn is working on using TQFT methods, which I'll talk about this time. I don't know too much about this stuff, so I hope any experts out there will correct my inevitable mistakes.

Actually, this conjecture is easiest to describe in a purely algebraic way, so I'll start there. Hopefully most of you know the concept of a "presentation" of a group in terms of generators and relations. For example, the group  $Z_n$  (integers mod  $n$ ) has the presentation  $\langle x | x^n \rangle$ . This means, roughly, that we form all products of the "generator"  $x$  and its inverse, and then mod out by the "relation"  $x^n = 1$ . A bit more interesting is the dihedral group  $D_n$  of symmetries of a regular  $n$ -gon, counting rotations and reflections, with presentation  $\langle x, y | x^n, y^2, (xy)^2 \rangle$ . Here  $x$  corresponds to a clockwise rotation by  $1/n$ -th of a turn, and  $y$  corresponds to a reflection.

A group always has lots of different presentations, so a natural problem is to decide whether two different presentations give the same group (or, strictly speaking, isomorphic groups). It'd be nice to have an algorithm for deciding this question. But it's a famous result of mathematical logic that there is no such algorithm!

If two presentations give the same group, one can get from one to the other by a sequence of the following easy steps, called Tietze moves:

Throw in an extra new generator  $x$  together with the extra new relation  $xg^{-1}$  where  $g$  is a product of the previous generators and their inverses.

The inverse of 1) - remove a generator  $x$  together with the relation  $xg^{-1}$ , if possible (the relation  $xg^{-1}$  needs to be there!).

Throw in a new relation that's a consequence of existing relations.

The inverse of 2) - remove a relation that's a consequence of other relations.

So if one has two presentations and wants to see if they give same group, you could always set up a program that blindly tries using these Tietze moves in all possible ways to transform one presentation into the other. If they are the same it'll eventually catch on! But if they're not it'll chug on forever. There's no algorithmic way to tell \*when\* it should give up and admit the two presentations

give different groups! - which is why we say there is no "decision procedure" for this problem.

In one form, the Andrews-Curtis conjecture goes as follows. Remember that the trivial group is the group with just the identity element; it has a presentation  $\langle x|x \rangle$ . Suppose we have some other "balanced" presentation of the trivial group, that is a presentation with just as many generators as relations:  $\langle x_1, \dots, x_n | r_1, \dots, r_n \rangle$ . Then the conjecture is that it can be reduced to the presentation  $\langle x|x \rangle$  by a sequence of the following moves that keep the presentation balanced:

Throw in an extra new generator  $x$  together with the extra new relation  $x$ .

The inverse of 1.

Permute the relations

Change  $r_1$  to  $r_1^{-1}$

Change  $r_1$  to  $r_1 r_2$

Change  $r_1$  to  $g r_1 g^{-1}$  for any  $g$ .

The experts seem to think this conjecture is probably false - but nobody has disproved it. Metzler lists a few presentations of the trivial group that might be counterexamples: nobody has ever found a way to use moves 1-6 to boil them down to the presentation  $\langle x|x \rangle$ . For example,

$\langle a, b | b^5 a^{-4}, aba(bab)^{-1} \rangle$ .

Try it!

The Andrews-Curtis conjecture is interesting mainly for its implications in topology. When they first stated their conjecture they noted a number of topological consequences, and the referee of the paper noted one more. For example, it would shed some light on the Poincare conjecture (although not settle it) as follows. Recall that the Poincare conjecture says every 3-dimensional manifold homotopic to a 3-sphere is homeomorphic to a 3-sphere. The Andrews-Curtis conjecture implies that if the Poincare conjecture is false, any counterexample can in fact be embedded (topologically) in  $R^4$ !

It was the referee (does anyone know who that was) who noted that the Andrews-Curtis conjecture can be formulated in terms of "CW complexes." This is how Quinn thinks about it, so I suppose I should say what those are.

A 0-complex is simply a set of points given the discrete topology. We call the points "0-cells." To get a 1-complex, we take a set of "1-cells," that is, closed unit intervals, and glue their ends on to the 0-cells in any way we want. In other words, we get a graph, possibly with some edges having both ends at the same vertex. To get a 2-complex, we take a set of "2-cells," that is, 2-dimensional closed disks, and glue their boundaries onto our 1-complex by any continuous map. And so on, with the "n-cells" being just copies of the closed unit ball in  $R^n$ .

CW complexes were invented by J. H. C. Whitehead in 1949 and are a key tool in algebraic topology. (The word "CW," by the way, seems to come from "closure-finite" and "weak" – as in "weak topology.") They are a nice class of topological spaces since on the one hand, being built up by gluing simple pieces together, one can really understand them, and on the other hand, they are actually quite general. In fact, if one is interested in the usual invariants studied in algebraic topology (homology and cohomology groups, homotopy groups and the like), CW complexes are pretty much good enough. More precisely, Whitehead proved a "CW approximation theorem" saying that any halfway decent topological space (i.e., any "compactly generated" space) is "weakly homotopy equivalent" to a CW complex. I won't burden you with the definitions here; I learned this stuff once upon a time from

4) *Elements of Homotopy Theory*, by George W. Whitehead, Springer-Verlag, Berlin, 1978. ISBN 0-387-90336-4

Anyway, the Andrews-Curtis conjecture can be thought of as being about 2-complexes. In fact, a group presentation can be regarded as instructions for building up a 2-complex – start with a point, glue on 1-cells, one for each generator (obtaining a "bouquet of circles") and then glue on 2-cells, one for each relation, attaching their boundaries to the 1-cells in the manner prescribed by the relation. This 2-complex will have fundamental group equal to the group given by the presentation. The moves 1-6 above can be thought of as operations on these 2-complexes. So one can translate the Andrews-Curtis conjecture into a statement about 2-complexes. And at this point I guess I'm going to start getting more technical...

One topological statement of the Andrews-Curtis conjecture is that "if two 2-complexes are simply equivalent then one can be 2-deformed to the other." I don't understand this as well as I want, so I won't explain it; instead, I'll briefly explain the (weaker?) version corresponding more closely to the algebraic statement above, namely "if  $X$  is a contractible 2-complex, it can be 2-deformed to a point." Being "contractible" means that as far as homotopy theory goes  $X$  is just like a point. (E.g., the unit disk is contractible, while the circle is not.) And a "2-deformation" roughly means a sequence of moves consisting of adding or deleting 1-cells or 2-cells in a way that doesn't affect things as far as homotopy goes, or doing homotopies of attaching maps of 2-cells. The interesting thing about these formulations of the Andrews-Curtis conjecture is that their analogs for  $n > 2$  are true and in fact were shown by J H C Whitehead in 1939!

Quinn's goal is to cook up invariants of 2-complexes that might detect counterexamples to the Andrews-Curtis conjecture, i.e., invariants under 2-deformation. He wants to do it using 1+1-dimensional TQFTs of a sort that assign vector spaces to 1-complexes and linear maps to 2-complexes. Traditionally, TQFTs assign vector spaces to  $n$ -manifolds and linear maps to  $(n + 1)$ -manifolds. Quinn calls his TQFTs "modular" because they have a lot of formal similarities to the kind of TQFTs that come up in string theory (where the modular group reigns supreme). He gives a thorough axiomatic description of

modular TQFTs in his lecture notes, and this is actually the most fascinating aspect for me, more so than the Andrews-Curtis conjecture per se, since it bears on physics.

The problem with coming up with an TQFT invariant that can catch counterexamples to the Andrews-Curtis conjecture is an interesting "stabilization" property that 2-complexes have. Namely, if two 2-complexes are simply equivalent, one can wedge them both with some large number  $k$  of 2-spheres and get complexes which are 2-deformable to each other. It turns out that this means we want to find a TQFT such that  $Z(S^2)^k = 0$ . And so Quinn considers TQFTs based, not on the complex numbers, but on integers mod  $p$ .

A TQFT of his sort amounts to finding a symmetric tensor category of vector spaces and an object  $A$  in this category with some special properties corresponding to the fact that it is the vector space corresponding to the unit interval  $[0, 1]$ , which is the basic 1-complex from which one can build up more fancy ones. The kind of category he uses has been described by:

- 5) S. Gelfand and D. Kazhdan, *Examples of tensor categories*,  
*Invent. Math.* 109 (1992) 595-617.

It is formed by starting with the category of representations of an algebraic group in characteristic  $p$ , and then making a semisimple category out of this in a manner strongly reminiscent of what they do in the theory of quantum groups at roots of unity. (See "week5" for a bit more about this.) The object  $A$  is taken to be the sum of one copy of each irreducible representation. (Again, this is strikingly reminiscent, and no doubt based on, what occurs in the physics of the Wess-Zumino-Witten model, where quantum groups at roots of unity play the role a finite group is playing here.)

So, to round off a long story, Quinn and Ivelini Bobtcheva are currently engaged in some rather massive computer calculations in order to actually explicitly obtain the data necessary to calculate in the TQFTs of this form. They have been looking at the groups  $SL(2)$ ,  $SL(3)$ ,  $Sp(4)$  and  $G_2$  over  $Z_p$ , where  $p$  is small (up to 19 for the  $SL(2)$  case). They are finding some interesting stuff just by calculating the TQFT invariants of the 2-complexes corresponding to the presentations  $\langle x|x^n \rangle$ . (Note that  $n = 0$  gives a space that's a wedge of a circle and  $S^2$ , while  $n = 1$  gives a disk.) Namely, they are finding periodicity in  $n$ .

But they haven't found any counterexamples to the Andrews-Curtis conjecture yet!

- 6) *Knots and Quantum Gravity*, ed. John Baez,  
Oxford University Press (to appear).

This is the proceedings of a workshop held at U.C. Riverside; a large percentage of the papers contain new results. Let me simply list them:

The Loop Formulation of Gauge Theory and Gravity, by Renate Loll

Representation Theory of Analytic Holonomy C\* Algebras, by Abhay Ashtekar and Jerzy Lewandowski

The Gauss Linking Number in Quantum Gravity,  
by Rodolfo Gambini and Jorge Pullin  
(currently available as gr-qc/9310025)

Vassiliev Invariants and the Loop States in Quantum Gravity,  
by Louis H. Kauffman (soon to be on gr-qc)

Geometric Structures and Loop Variables in  
(2+1)-Dimensional Gravity,  
by Steven Carlip (currently available as gr-qc/9309020)

From Chern-Simons to WZW via Path Integrals, by Dana S. Fine

Topological Field Theory as the Key to Quantum Gravity,  
by Louis Crane (currently available as hep-th/9308126)

Strings, Loops, Knots and Gauge Fields, by John Baez (currently available as hep-th/9309067)

BF Theories and 2-knots,  
by Paolo Cotta-Ramusino and Maurizio Martellini

Knotted Surfaces, Braid Movies, and Beyond,  
by J. Scott Carter and Masahico Saito

## 7 A potential smooth counterexample in dimension 4...

*A potential smooth counterexample in dimension 4 to the Poincare conjecture, the Schonflies conjecture, and the Andrews-Curtis conjecture*[3]

**Selman Akbulut and R. Kirby**

## 8 Canada Research Chair in Combinatorial Algebra

*Canada Research Chair in Combinatorial Algebra*[2]

**Alexei Miasnikov**(McGill)

A second focus of Dr. Miasnikov's program is to find a counterexample to the Andrews-Curtis Conjecture, which holds that every balanced presentation of a trivial group can be transformed into the standard presentation by a finite sequence of elementary transformations. Dr. Miasnikov posits that the Andrews-Curtis Conjecture is false, and his proposed goal is to find a counterexample in a finite quotient using computational methods.

## 9 Balanced presentations of the trivial group on . . .

*Balanced presentations of the trivial group on two generators and the Andrews-Curtis conjecture*[7]

Alexei D. Miasnikov and Alexei G. Myasnikov (City College of New York)

Title: Balanced presentations of the trivial group on two generators and the Andrews-Curtis conjecture

Authors: Alexei D. Miasnikov, Alexei G. Myasnikov

Categories: GR Group Theory

Math Subject Class: 20E05, 20F05,  
68T05 (Primary),  
57M05, 57M20. (Secondary)

Journal reference: In W. Kantor and A. Seress, editors,  
Groups and Computation III, volume 23,  
(2001) 257-263, Berlin

Comments: 7 pages, no figures

**Abstract:** The Andrews-Curtis conjecture states that every balanced presentation of the trivial group can be reduced to the standard one by a sequence of the elementary Nielsen transformations and conjugations. In this paper we describe all balanced presentations of the trivial group on two generators and with the total length of relators  $\leq 12$ . We show that all these presentations satisfy the Andrews-Curtis conjecture.

From: Alexei Miasnikov <alex@groups.sci.ccny.cuny.edu>

Date: Mon, 21 Apr 2003 21:01:49 GMT (11kb)

@article{math.GR/0304305,

title = {{Balanced presentations of the trivial group on two generators and the Andrews-Curtis conjecture}},

author = {Alexei D. Miasnikov and Alexei G. Myasnikov},

journal = {In W. Kantor and A. Seress, editors,  
Groups and Computation III, volume},

volume = 23,

year = 2001,

pages = {257--263},

eprint = {arXiv:math.GR/0304305}}

## 10 Genetic algorithms and the Andrews-Curtis conjecture

*Genetic algorithms and the Andrews-Curtis conjecture*[6]

Alexei D. Miasnikov (City College of New York)

Title: Genetic algorithms and the Andrews-Curtis conjecture

Author: Alexei D. Miasnikov

Categories: GR Group Theory

Math Subject Class: 20E05, 20F05, 68T05 (Primary) 57M05,57M20

Journal reference:

International Journal of Algebra and Computation,

Vol. 9 No. 6, (1999) 671-686 Comments: 19 pages

**Abstract:** The Andrews-Curtis conjecture claims that every balanced presentation of the trivial group can be transformed into the trivial presentation by a finite sequence of "elementary transformations" which are Nielsen transformations together with an arbitrary conjugation of a relator. It is believed that the Andrews-Curtis conjecture is false; however, not so many possible counterexamples are known. It is not a trivial matter to verify whether the conjecture holds for a given balanced presentation or not. The purpose of this paper is to describe some non-deterministic methods, called Genetic Algorithms, designed to test the validity of the Andrews-Curtis conjecture. Using such algorithm we have been able to prove that all known (to us) balanced presentations of the trivial group where the total length of the relators is at most 12 satisfy the conjecture. In particular, the Andrews-Curtis conjecture holds for the presentation  $\langle x, y | xyx = yxy, x^2 = y^3 \rangle$  which was one of the well known potential counterexamples.

From: Alexei Miasnikov <alex@groups.sci.ccny.cuny.edu>

Date: Mon, 21 Apr 2003 21:07:18 GMT (13kb)

@misc{math.GR/0304306,

title =

{Genetic algorithms and the Andrews-Curtis conjecture},

author = {Alexei D. Miasnikov},

howpublished =

{International Journal of Algebra and Computation,

Vol. 9 No. 6, (1999) 671-686},

eprint = {arXiv:math.GR/0304306}}

## 11 A remark on the stable Andrews-Curtis conjecture

*A remark on the stable Andrews-Curtis conjecture*[1]

**Andrew Casson** (Yale)

The stable Andrews-Curtis conjecture in combinatorial group theory is equivalent to the conjecture that every finite contractible 2-complex can be reduced to a single point by elementary expansions and collapses through complexes of

dimension at most 3. In group-theoretic terms, this means that every presentation of the trivial group with equal numbers of generators and relators can be simplified to standard form by elementary moves corresponding to "addition of cells" or "handle-slides", together with "stabilization" moves that increase or decrease the number of generators (and relators). The role of the stabilization moves is unclear; conceivably any presentation that can be simplified using the full set of moves can also be simplified without stabilization. I will discuss several equivalent forms of the conjecture, and present (rather weak) evidence that presentations can be "improved" by stabilization.

## 12 Algebraic Geometry over Groups

*Algebraic Geometry over Groups*[8]

**Gilbert Baumslag, A. Myasnikov and V. Remeslennikov**

2000/01 The Andrews-Curtis conjecture, a problem on locally finite groups, some aspects of Grigorchuk groups, and the paper 'Algebraic Geometry over Groups' by G. Baumslag, A. Myasnikov, and V. Remeslennikov which appeared in *J. Algebra* 219 (1999).

2000/01 The Andrews-Curtis conjecture, a problem on locally finite groups, some aspects of Grigorchuk groups, and the paper 'Algebraic Geometry over Groups' by G. Baumslag, A. Myasnikov, and V. Remeslennikov which appeared in *J. Algebra* 219 (1999).

Algebraic Geometry over Groups I. Algebraic Sets and Ideal Theory

Gilbert Baumslag, Alexei Myasnikov, and Vladimir Remeslennikov

Received 8 January 1999. Available online 10 April 2002.

Abstract

The object of this paper, which is the first in a series of three, is to lay the foundations of the theory of ideals and algebraic sets over groups.

References

AL. L. Auslander,  
On a problem of Philip Hall.  
*Ann. of Math.* (2) 86  
(1967), pp. 112116. MathSciNet

BH. H. Bass,  
Groups acting on non-archimedean trees.  
*Arboreal Group Theory* (1991), pp. 69130.

BB. B. Baumslag,  
Residually free groups.  
*Proc. London Math. Soc.* 17 (1967),



pp. 402418. MathSciNet

BG. G. Baumslag,  
On generalized free products. Math. Z. 7 (1962),  
pp. 423438. MathSciNet

BMR1. G. Baumslag, A. Myasnikov and V. Remeslennikov,  
Residually hyperbolic groups and approximation theorems  
for extensions of centralizers.  
Proc. Inst. Appl. Math Russian Acad. Sci. 24 (1996),  
pp. 337.

BMR2. G. Baumslag, A. Myasnikov, and, V. Remeslennikov,  
Discriminating completions of hyperbolic groups,  
submitted for publication.

BMR0. G. Baumslag, A. Myasnikov and V. Roman\`kov,  
Two theorems about equationally Noetherian groups.  
J. Algebra 194 (1997), pp. 654664.  
Abstract | Abstract + References | PDF (176 K) | MathSciNet

BR. R. Bryant, The verbal topology of a group.  
J. Algebra 48 (1977), pp. 340346. MathSciNet

CK. C. C. Chang and H. J. Keisler.  
Model Theory,  
North-Holland, New York (1973).

CE. L. P. Comerford and C. C. Edmunds,  
Quadratic equations over free groups and free products.  
J. Algebra 68 (1981), pp. 276297. MathSciNet

FGMRS. B. Fine, A. M. Gaglione, A. Miasnikov,  
G. Rosenberger and D. Spellman,  
A classification of fully residual free groups of  
rank three or less.  
J. Algebra 200 (1998), pp. 571605. Abstract | Abstract  
+ References | PDF (355 K) | MathSciNet

GK. R. I. Grigorchuk and P. F. Kurchanov,  
Some questions of group theory related to geometry.  
Itogi Nauki i Techniki, Sovremennye  
Problemy Matematiki, Fundamental'nye Napravlenia, VINITI,  
58Encyclopedia of Mathematical Sciences (1990).

GV. V. Guba,

Equivalence of infinite systems of equations in free groups and semigroups to finite subsystems. Mat. Zametki 40 (1986), pp. 321324.

HR. R. Hartshorne.  
Algebraic Geometry,  
Springer-Verlag, New York (1977).

HZ. E. Hrushovski and B. Zilber,  
Zariski geometries.  
J. Amer. Math. Soc. 9 (1996), pp. 156. MathSciNet

KMS. A. Karrass, W. Magnus and D. Solitar,  
Elements of finite order in groups with  
a single defining relation.  
Comm. Pure Appl. Math. 13 (1960), pp. 458466.

KM. O. Kharlampovich and A. Myasnikov,  
Irreducible affine varieties over a free group.  
I: irreducibility of quadratic equations and  
Nullstellensatz. J. Algebra 200 (1998), pp. 472516.  
Abstract | Abstract + References | PDF (445 K) | MathSciNet

KM. O. Kharlampovich and A. Myasnikov,  
Irreducible affine varieties over a free group.  
II: systems in triangular quasi-quadratic form and  
description of residually free groups.  
J. Algebra 200 (1988), pp. 517570.

LR1. R. C. Lyndon,  
The equation  $a^2b^2 = c^2$  in free groups.  
Michigan Math. J. 6 (1959), pp. 155164. MathSciNet

LR2. R. C. Lyndon,  
Groups with parametric exponents.  
Trans. Amer. Math. Soc. 96 (1960), pp. 518533. MathSciNet

LR3. R. C. Lyndon,  
Equation in free groups.  
Trans. Amer. Math. Soc. 96 (1960), pp. 445457. MathSciNet

MKS. W. Magnus, A. Karrass and D. Solitar.  
Combinatorial Group Theory:  
Presentations of Groups in Terms of Generators and Relations,  
Wiley-Interscience, New York/London/Sydney (1966).

- MG. G. S. Makanin,  
 Decidability of the universal and positive theories  
 of a free group.  
 Izv. Akad. Nauk SSSR Ser. Math. 48 (1985), pp. 735749.
- MR1. A. G. Myasnikov and V. N. Remeslennikov,  
 Exponential groups I:  
 Foundations of the theory and tensor completion.  
 Siberian Math. J. 35 (1994), pp. 11061118.
- MR2. A. G. Myasnikov and V. N. Remeslennikov,  
 Exponential groups II:  
 Extensions of centralizers and tensor completion of  
 CSA-groups.  
 Internat. J. Algebra Comput. 6 (1996),  
 pp. 687711. MathSciNet
- PB. B. Plotkin,  
 Varieties of algebras and algebraic varieties.  
 Categories of algebraic varieties,  
 Hebrew University, Jerusalem, 1996, preprint.
- RA1. A. Razborov,  
 On systems of equations in a free group.  
 Math. USSR-Izv. 25 (1985), pp. 115162.
- RA2. A. Razborov,  
 On systems of equations in a free group,  
 Ph.D. thesis, Steklov Math. Institute, Moscow, 1987.
- RV1. V. N. Remeslennikov,  
 E-free groups.  
 Siberian Math. J. 30 (1989), pp. 153157.
- RV2. V. N. Remeslennikov,  
 Matrix representation of finitely generated  
 metabelian groups.  
 Algebra i Logika 8 (1969), pp. 7276.
- SW. W. R. Scott,  
 Algebraically closed groups.  
 Proc. Amer. Math. Soc. 2 (1951), pp. 118212.
- SM. M. P. Schutzenberger,  
 Sur l'equation  $a^{2h} = b^2 + mc^2 + p$  dans un group libre.

C.R. Acad. Sci. Paris 248 (1959), pp. 24352436. MathSciNet

SJ. J. Stallings,  
Finiteness properties of matrix representations.  
Ann. of Math. 124 (1986), pp. 337346. MathSciNet

WB. B. A. F. Wehrfritz.  
Infinite Linear Groups,  
Springer-Verlag, New York/Heidelberg/Berlin (1973).

## 13 Representation theory, topological field theory, and...

*Representation theory, topological field theory, and the Andrews-Curtis conjecture*[9]  
**Frank Quinn**

Title: Representation theory, topological field theory,  
and the Andrews-Curtis conjecture

Author: Frank Quinn

Categories: (QA Quantum Algebra; GR Group Theory)

Comments: 7 pages.

ADMIN NOTE: source file was garbled, partially salvaged 19Feb2001

**Abstract:** We pose a representation-theoretic question motivated by an attempt to resolve the Andrews-Curtis conjecture. Roughly, is there a triangular Hopf algebra with a collection of self-dual irreducible representations  $V_i$  so that the product of any two decomposes as a sum of copies of the  $V_i$ , and  $\sum(\text{rank } V_i)^2 = 0$ ? This data can be used to construct a 'topological quantum field theory' on 2-complexes which stands a good chance of detecting counterexamples to the conjecture.

From: Frank Quinn <quinn@math.vt.edu>

Date: Thu, 13 Feb 1992 14:42 EDT (7kb)

Revised: Fri, 14 Feb 1992 14:32 EDT

@misc{hep-th/9202044,

title = {{Representation theory, topological field theory,  
and the Andrews-Curtis conjecture}},

author = {Frank Quinn},

eprint = {arXiv:hep-th/9202044}}

## References

- [1] Casson, Andrew,  
“A remark on the stable Andrews-Curtis conjecture”,  
“[http://www.math.yale.edu/users/yair/topsem/casson\\_abs.html](http://www.math.yale.edu/users/yair/topsem/casson_abs.html)”
- [2] Miasnikov, Alexei,  
“Canada Research Chair in Combinatorial Algebra”,  
“<http://www.chairs.gc.ca>”
- [3] Akbulut, Selman; Kirby, R.,  
“A potential smooth counterexample in dimension 4 to the Poincare conjecture, the Schonflies conjecture, and the Andrew-Curtis conjecture”,  
“<http://www.mth.msu.edu/~akbulut/vita/vita.html>”,  
Topology 24, no. 4 (1985), 375-390.
- [4] Baez, John,  
“This Week’s Finds in Mathematical Physics (Week 23)”,  
“<http://math.ucr.edu/home/baez/week23.html>”
- [5] Rosson, John,  
“Multiplicative Invariants of Special 2-Complexes”,  
“<http://www.sysc.pdx.edu/newabstracts/abstract-rosson.htm>”
- [6] Miasnikov, Alexei,  
“Genetic algorithms and the Andrews-Curtis conjecture”,  
“<http://front.math.ucdavis.edu/math.GR/0304306>”
- [7] Miasnikov, Alexei D.; Myasnikov, Alexei G.  
“Balanced presentations of the trivial group on two generators and the Andrews-Curtis conjecture”,  
“<http://front.math.ucdavis.edu/math.GR/0304305>”
- [8] Baumslag, Gilbert, Myasnikov, A. and Remeslennikov,  
“Algebraic Geometry over Groups”,  
“<http://front.math.ucdavis.edu/math.GR/0304305>” Journal of Algebra Volume 219, Issue 1 , 1 September 1999, Pages 16-79
- [9] Quinn, Frank  
“Representation theory, topological field theory, and the Andrews-Curtis conjecture”,  
“<http://front.math.ucdavis.edu/hep-th/9202044>”