The value of information in algebraic and geometric decision problems

Maurice Chiodo
The University of Melbourne

June 8, 2010

Motivation:

Motivation:

The triviality problem, of deciding whether a finite presentation of a group defines the trivial group, is algorithmically undecidable.

Motivation:

The triviality problem, of deciding whether a finite presentation of a group defines the trivial group, is algorithmically undecidable. So is the problem of determining if a finite presentation defines a non-trivial free product, or determining if one finite presentation embeds into another finite presentation (as groups).

Motivation:

The triviality problem, of deciding whether a finite presentation of a group defines the trivial group, is algorithmically undecidable. So is the problem of determining if a finite presentation defines a non-trivial free product, or determining if one finite presentation embeds into another finite presentation (as groups).

Questions

Motivation:

The triviality problem, of deciding whether a finite presentation of a group defines the trivial group, is algorithmically undecidable. So is the problem of determining if a finite presentation defines a non-trivial free product, or determining if one finite presentation embeds into another finite presentation (as groups).

Questions

-Is there an algorithm to produce a non-trivial element from a finite presentation of a non-trivial group?

Motivation:

The triviality problem, of deciding whether a finite presentation of a group defines the trivial group, is algorithmically undecidable. So is the problem of determining if a finite presentation defines a non-trivial free product, or determining if one finite presentation embeds into another finite presentation (as groups).

Questions

-Is there an algorithm to produce a non-trivial element from a finite presentation of a non-trivial group?
-Is there an algorithm to decompose a finite presentation of a non-trivial free product into two non-trivial finitely presented factors?

Motivation:

The triviality problem, of deciding whether a finite presentation of a group defines the trivial group, is algorithmically undecidable. So is the problem of determining if a finite presentation defines a non-trivial free product, or determining if one finite presentation embeds into another finite presentation (as groups).

Questions

-Is there an algorithm to produce a non-trivial element from a finite presentation of a non-trivial group?
-Is there an algorithm to decompose a finite presentation of a non-trivial free product into two non-trivial finitely presented factors?
-Is there an algorithm to construct an embedding from one finitely presented group into another in which it embeds?

Motivation:

The triviality problem, of deciding whether a finite presentation of a group defines the trivial group, is algorithmically undecidable. So is the problem of determining if a finite presentation defines a non-trivial free product, or determining if one finite presentation embeds into another finite presentation (as groups).

Questions

-Is there an algorithm to produce a non-trivial element from a finite presentation of a non-trivial group?
-Is there an algorithm to decompose a finite presentation of a non-trivial free product into two non-trivial finitely presented factors?
-Is there an algorithm to construct an embedding from one finitely presented group into another in which it embeds?
-Do such results have applications elsewhere?

The most common way to tackle decision problems in group theory has been to encode the following recursion-theoretic fact into group presentations:

The most common way to tackle decision problems in group theory has been to encode the following recursion-theoretic fact into group presentations:

There is a recursively enumerable set $\mathbb{K}$ which is not recursive.

The most common way to tackle decision problems in group theory has been to encode the following recursion-theoretic fact into group presentations:

There is a recursively enumerable set $\mathbb{K}$ which is not recursive.

However, this does not work in any obvious way for our problems. So we take a modified approach as follows:

The most common way to tackle decision problems in group theory has been to encode the following recursion-theoretic fact into group presentations:

There is a recursively enumerable set $\mathbb{K}$ which is not recursive.

However, this does not work in any obvious way for our problems. So we take a modified approach as follows:

1. Develop stronger results in recursion theory.

The most common way to tackle decision problems in group theory has been to encode the following recursion-theoretic fact into group presentations:

There is a recursively enumerable set $\mathbb{K}$ which is not recursive.

However, this does not work in any obvious way for our problems. So we take a modified approach as follows:

1. Develop stronger results in recursion theory.
2. Encode these into group presentations, incorporating some of the existing encoding techniques (Boone, Adian-Rabin).

The most common way to tackle decision problems in group theory has been to encode the following recursion-theoretic fact into group presentations:

There is a recursively enumerable set $\mathbb{K}$ which is not recursive.

However, this does not work in any obvious way for our problems. So we take a modified approach as follows:

1. Develop stronger results in recursion theory.
2. Encode these into group presentations, incorporating some of the existing encoding techniques (Boone, Adian-Rabin).
3. Encode these presentations into closed 4-manifolds, using an existing construction by Markov.

The most common way to tackle decision problems in group theory has been to encode the following recursion-theoretic fact into group presentations:

There is a recursively enumerable set $\mathbb{K}$ which is not recursive.

However, this does not work in any obvious way for our problems. So we take a modified approach as follows:

1. Develop stronger results in recursion theory.
2. Encode these into group presentations, incorporating some of the existing encoding techniques (Boone, Adian-Rabin).
3. Encode these presentations into closed 4-manifolds, using an existing construction by Markov.

The main engine for our results:

The most common way to tackle decision problems in group theory has been to encode the following recursion-theoretic fact into group presentations:

There is a recursively enumerable set $\mathbb{K}$ which is not recursive.

However, this does not work in any obvious way for our problems. So we take a modified approach as follows:

1. Develop stronger results in recursion theory.
2. Encode these into group presentations, incorporating some of the existing encoding techniques (Boone, Adian-Rabin).
3. Encode these presentations into closed 4-manifolds, using an existing construction by Markov.

The main engine for our results:
You can't say anything about the complement of an r.e. set!

1. Recursion theory preliminaries:

1. Recursion theory preliminaries:

We define $\varphi_m : \mathbb{N} \to \mathbb{N}$ to be the m^{th} **partial recursive function**.

1. Recursion theory preliminaries:

We define $\varphi_m : \mathbb{N} \to \mathbb{N}$ to be the m^{th} **partial recursive function**. The m^{th} **partial recursive set** W_m is then the domain of φ_m.

1. Recursion theory preliminaries:

We define $\varphi_m : \mathbb{N} \to \mathbb{N}$ to be the m^{th} **partial recursive function**. The m^{th} **partial recursive set** W_m is then the domain of φ_m. We also use **Cantor's pairing function** $\langle .,. \rangle : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$, $\langle x, y \rangle := \frac{1}{2}(x+y)(x+y+1)+y$, which is a (computable) bijection from $\mathbb{N} \times \mathbb{N}$ to $\mathbb{N}$.

1. Recursion theory preliminaries:

We define $\varphi_m : \mathbb{N} \to \mathbb{N}$ to be the m^{th} **partial recursive function**.
The m^{th} **partial recursive set** W_m is then the domain of φ_m.
We also use **Cantor's pairing function** $\langle .,. \rangle : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$,
$\langle x, y \rangle := \frac{1}{2}(x+y)(x+y+1) + y$, which is a (computable) bijection from $\mathbb{N} \times \mathbb{N}$ to $\mathbb{N}$.

Theorem (smn theorem)

1. Recursion theory preliminaries:

We define $\varphi_m : \mathbb{N} \to \mathbb{N}$ to be the m^{th} **partial recursive function**. The m^{th} **partial recursive set** W_m is then the domain of φ_m. We also use **Cantor's pairing function** $\langle ., . \rangle : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$, $\langle x, y \rangle := \frac{1}{2}(x + y)(x + y + 1) + y$, which is a (computable) bijection from $\mathbb{N} \times \mathbb{N}$ to $\mathbb{N}$.

Theorem (smn theorem)

For all $m, n \in \mathbb{N}$, a partial function $f : \mathbb{N}^{m+n} \to \mathbb{N}$ is partial-recursive if and only if there is a recursive function $s : \mathbb{N}^m \to \mathbb{N}$ such that, for all $e_1, \ldots, e_m, x_1, \ldots, x_n \in \mathbb{N}$ we have that $f(e_1, \ldots, e_m, x_1, \ldots, x_n) = \varphi_{s(e_1,\ldots,e_m)}(\langle x_1, \ldots, x_n \rangle)$.

1. Recursion theory preliminaries:

We define $\varphi_m : \mathbb{N} \to \mathbb{N}$ to be the m^{th} **partial recursive function**. The m^{th} **partial recursive set** W_m is then the domain of φ_m. We also use **Cantor's pairing function** $\langle ., . \rangle : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$, $\langle x, y \rangle := \frac{1}{2}(x + y)(x + y + 1) + y$, which is a (computable) bijection from $\mathbb{N} \times \mathbb{N}$ to $\mathbb{N}$.

Theorem (smn theorem)

For all $m, n \in \mathbb{N}$, a partial function $f : \mathbb{N}^{m+n} \to \mathbb{N}$ is partial-recursive if and only if there is a recursive function $s : \mathbb{N}^m \to \mathbb{N}$ such that, for all $e_1, \ldots, e_m, x_1, \ldots, x_n \in \mathbb{N}$ we have that $f(e_1, \ldots, e_m, x_1, \ldots, x_n) = \varphi_{s(e_1,\ldots,e_m)}(\langle x_1, \ldots, x_n \rangle)$.

That is to say, a partial function $f : \mathbb{N}^k \to \mathbb{N}$ is partial recursive if and only if, whenever we hold some of its variables fixed, the remaining function is partial recursive.

Theorem (Kleene recursion theorem)

Theorem (Kleene recursion theorem)

Let $f : \mathbb{N} \to \mathbb{N}$ be a recursive function. Then there exists $n \in \mathbb{N}$ with $\varphi_n = \varphi_{f(n)}$ (as functions).

Theorem (Kleene recursion theorem)

Let $f : \mathbb{N} \to \mathbb{N}$ be a recursive function. Then there exists $n \in \mathbb{N}$ with $\varphi_n = \varphi_{f(n)}$ (as functions).

So every recursive function f has a 'fixed point' on indices of partial recursive functions.

Theorem (Kleene recursion theorem)

Let $f : \mathbb{N} \to \mathbb{N}$ be a recursive function. Then there exists $n \in \mathbb{N}$ with $\varphi_n = \varphi_{f(n)}$ (as functions).

So every recursive function f has a 'fixed point' on indices of partial recursive functions.

We now have all the tools necessary to prove our recursion theory results:

Lemma (First recursion theory result)

Lemma (First recursion theory result)

Fix any $k > 0$*. Then there is no partial recursive function* $g : \mathbb{N}^{k+2} \to \mathbb{N}$ *such that, given* $n, x_0, \ldots, x_k \in \mathbb{N}$ *satisfying* $|\{x_0, \ldots, x_k\} \cap W_n| \leq 1$*, we have that* $g(n, x_0, \ldots, x_k)$ *halts with output* $x_i \in \{x_0, \ldots, x_k\}$ *such that* $x_i \notin W_n$*.*

Lemma (First recursion theory result)

Fix any $k > 0$. Then there is no partial recursive function $g : \mathbb{N}^{k+2} \to \mathbb{N}$ such that, given $n, x_0, \ldots, x_k \in \mathbb{N}$ satisfying $|\{x_0, \ldots, x_k\} \cap W_n| \leq 1$, we have that $g(n, x_0, \ldots, x_k)$ halts with output $x_i \in \{x_0, \ldots, x_k\}$ such that $x_i \notin W_n$.

That is, given a recursively enumerable set W_n and k elements, at most one of which lies in W_n, we can't recursively pick one lying OUTSIDE W_n.

Lemma (First recursion theory result)

Fix any $k > 0$. Then there is no partial recursive function $g : \mathbb{N}^{k+2} \to \mathbb{N}$ such that, given $n, x_0, \ldots, x_k \in \mathbb{N}$ satisfying $|\{x_0, \ldots, x_k\} \cap W_n| \leq 1$, we have that $g(n, x_0, \ldots, x_k)$ halts with output $x_i \in \{x_0, \ldots, x_k\}$ such that $x_i \notin W_n$.

That is, given a recursively enumerable set W_n and k elements, at most one of which lies in W_n, we can't recursively pick one lying OUTSIDE W_n.

(We say nothing about the behaviour of g when the input is not 'valid'.)

Proof.

Proof.

Assume such a g exists. Define $f : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ by

$$f(n,m) := \begin{cases} 0 \text{ if } g(n,0,\ldots,k) = j \in \{0,\ldots,k\} \text{ and } m = j \\ \uparrow \text{ in all other cases} \end{cases}$$

Proof.

Assume such a g exists. Define $f : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ by

$$f(n,m) := \begin{cases} 0 \text{ if } g(n,0,\dots,k) = j \in \{0,\dots,k\} \text{ and } m = j \\ \uparrow \ \text{ in all other cases} \end{cases}$$

Then f is partial-recursive, since g is. By the smn theorem, there exists a recursive function $s : \mathbb{N} \to \mathbb{N}$ such that $f(n,m) = \varphi_{s(n)}(m)$ for all m, n. Since s is recursive, the Kleene recursion theorem shows that there must be some n' such that $\varphi_{s(n')} = \varphi_{n'}$. Thus $f(n',m) = \varphi_{n'}(m)$ for all $m \in \mathbb{N}$. Moreover, by definition, $\varphi_{n'}(m)$ can halt on at most one of the cases $m = 0, \dots, m = k$ (if at all), and no other values. Thus $|\{0,\dots,k\} \cap W_{n'}| \leq 1$. So $g(n', 0, \dots, k)$ will halt and output $j \in \{0,\dots,k\} \setminus W_{n'}$ (by construction of g). But since $g(n',0,\dots,k)$ halts with output j in $\{0,\dots,k\}$, then $f(n',j)$ halts (by construction of f). Hence $\varphi_{n'}(j)$ halts (by definition of $\varphi_{n'}$), and so $j \in W_{n'}$ since $W_{n'}$ is precisely the halting set of $\varphi_{n'}$. Thus we have a contradiction, as we showed $j \notin W_{n'}$, so no such g can exist.

Proof.

Assume such a g exists. Define $f : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ by

$$f(n,m) := \begin{cases} 0 \text{ if } g(n,0,\ldots,k) = j \in \{0,\ldots,k\} \text{ and } m = j \\ \uparrow \text{ in all other cases} \end{cases}$$

Then f is partial-recursive, since g is. By the smn theorem, there exists a recursive function $s : \mathbb{N} \to \mathbb{N}$ such that $f(n,m) = \varphi_{s(n)}(m)$ for all m, n. Since s is recursive, the Kleene recursion theorem shows that there must be some n' such that $\varphi_{s(n')} = \varphi_{n'}$. Thus $f(n',m) = \varphi_{n'}(m)$ for all $m \in \mathbb{N}$. Moreover, by definition, $\varphi_{n'}(m)$ can halt on at most one of the cases $m = 0, \ldots, m = k$ (if at all), and no other values. Thus $|\{0,\ldots,k\} \cap W_{n'}| \leq 1$. So $g(n',0,\ldots,k)$ will halt and output $j \in \{0,\ldots,k\} \setminus W_{n'}$ (by construction of g). But since $g(n',0,\ldots,k)$ halts with output j in $\{0,\ldots,k\}$, then $f(n',j)$ halts (by construction of f). Hence $\varphi_{n'}(j)$ halts (by definition of $\varphi_{n'}$), and so $j \in W_{n'}$ since $W_{n'}$ is precisely the halting set of $\varphi_{n'}$. Thus we have a contradiction, as we showed $j \notin W_{n'}$, so no such g can exist.

Proof.

Assume such a g exists. Define $f : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ by

$$f(n, m) := \begin{cases} 0 \text{ if } g(n, 0, \ldots, k) = j \in \{0, \ldots, k\} \text{ and } m = j \\ \uparrow \quad \text{in all other cases} \end{cases}$$

Then f is partial-recursive, since g is. By the smn theorem, there exists a recursive function $s : \mathbb{N} \to \mathbb{N}$ such that $f(n, m) = \varphi_{s(n)}(m)$ for all m, n. Since s is recursive, the Kleene recursion theorem shows that there must be some n' such that $\varphi_{s(n')} = \varphi_{n'}$. Thus $f(n', m) = \varphi_{n'}(m)$ for all $m \in \mathbb{N}$. Moreover, by definition, $\varphi_{n'}(m)$ can halt on at most one of the cases $m = 0, \ldots, m = k$ (if at all), and no other values. Thus $|\{0, \ldots, k\} \cap W_{n'}| \leq 1$. So $g(n', 0, \ldots, k)$ will halt and output $j \in \{0, \ldots, k\} \setminus W_{n'}$ (by construction of g). But since $g(n', 0, \ldots, k)$ halts with output j in $\{0, \ldots, k\}$, then $f(n', j)$ halts (by construction of f). Hence $\varphi_{n'}(j)$ halts (by definition of $\varphi_{n'}$), and so $j \in W_{n'}$ since $W_{n'}$ is precisely the halting set of $\varphi_{n'}$. Thus we have a contradiction, as we showed $j \notin W_{n'}$, so no such g can exist.

Proof.

Assume such a g exists. Define $f : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ by

$$f(n,m) := \begin{cases} 0 \text{ if } g(n,0,\ldots,k) = j \in \{0,\ldots,k\} \text{ and } m = j \\ \uparrow \text{ in all other cases} \end{cases}$$

Then f is partial-recursive, since g is. By the smn theorem, there exists a recursive function $s : \mathbb{N} \to \mathbb{N}$ such that $f(n,m) = \varphi_{s(n)}(m)$ for all m, n. Since s is recursive, the Kleene recursion theorem shows that there must be some n' such that $\varphi_{s(n')} = \varphi_{n'}$. Thus $f(n', m) = \varphi_{n'}(m)$ for all $m \in \mathbb{N}$. Moreover, by definition, $\varphi_{n'}(m)$ can halt on at most one of the cases $m = 0, \ldots, m = k$ (if at all), and no other values. Thus $|\{0,\ldots,k\} \cap W_{n'}| \leq 1$. So $g(n', 0, \ldots, k)$ will halt and output $j \in \{0,\ldots,k\} \setminus W_{n'}$ (by construction of g). But since $g(n', 0, \ldots, k)$ halts with output j in $\{0,\ldots,k\}$, then $f(n', j)$ halts (by construction of f). Hence $\varphi_{n'}(j)$ halts (by definition of $\varphi_{n'}$), and so $j \in W_{n'}$ since $W_{n'}$ is precisely the halting set of $\varphi_{n'}$. Thus we have a contradiction, as we showed $j \notin W_{n'}$, so no such g can exist.

Proof.

Assume such a g exists. Define $f : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ by

$$f(n, m) := \begin{cases} 0 \text{ if } g(n, 0, \ldots, k) = j \in \{0, \ldots, k\} \text{ and } m = j \\ \uparrow \text{ in all other cases} \end{cases}$$

Then f is partial-recursive, since g is. By the smn theorem, there exists a recursive function $s : \mathbb{N} \to \mathbb{N}$ such that $f(n, m) = \varphi_{s(n)}(m)$ for all m, n. Since s is recursive, the Kleene recursion theorem shows that there must be some n' such that $\varphi_{s(n')} = \varphi_{n'}$. Thus $f(n', m) = \varphi_{n'}(m)$ for all $m \in \mathbb{N}$. Moreover, by definition, $\varphi_{n'}(m)$ can halt on at most one of the cases $m = 0, \ldots, m = k$ (if at all), and no other values. Thus $|\{0, \ldots, k\} \cap W_{n'}| \leq 1$. So $g(n', 0, \ldots, k)$ will halt and output $j \in \{0, \ldots, k\} \setminus W_{n'}$ (by construction of g). But since $g(n', 0, \ldots, k)$ halts with output j in $\{0, \ldots, k\}$, then $f(n', j)$ halts (by construction of f). Hence $\varphi_{n'}(j)$ halts (by definition of $\varphi_{n'}$), and so $j \in W_{n'}$ since $W_{n'}$ is precisely the halting set of $\varphi_{n'}$. Thus we have a contradiction, as we showed $j \notin W_{n'}$, so no such g can exist.

Proof.

Assume such a g exists. Define $f : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ by

$$f(n,m) := \begin{cases} 0 \text{ if } g(n,0,\dots,k) = j \in \{0,\dots,k\} \text{ and } m = j \\ \uparrow \ \text{ in all other cases} \end{cases}$$

Then f is partial-recursive, since g is. By the smn theorem, there exists a recursive function $s : \mathbb{N} \to \mathbb{N}$ such that $f(n,m) = \varphi_{s(n)}(m)$ for all m, n. Since s is recursive, the Kleene recursion theorem shows that there must be some n' such that $\varphi_{s(n')} = \varphi_{n'}$. Thus $f(n',m) = \varphi_{n'}(m)$ for all $m \in \mathbb{N}$. Moreover, by definition, $\varphi_{n'}(m)$ can halt on at most one of the cases $m = 0, \dots, m = k$ (if at all), and no other values. Thus $|\{0,\dots,k\} \cap W_{n'}| \leq 1$. So $g(n',0,\dots,k)$ will halt and output $j \in \{0,\dots,k\} \setminus W_{n'}$ (by construction of g). But since $g(n',0,\dots,k)$ halts with output j in $\{0,\dots,k\}$, then $f(n',j)$ halts (by construction of f). Hence $\varphi_{n'}(j)$ halts (by definition of $\varphi_{n'}$), and so $j \in W_{n'}$ since $W_{n'}$ is precisely the halting set of $\varphi_{n'}$. Thus we have a contradiction, as we showed $j \notin W_{n'}$, so no such g can exist.

Proof.

Assume such a g exists. Define $f : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ by

$$f(n,m) := \begin{cases} 0 \text{ if } g(n,0,\ldots,k) = j \in \{0,\ldots,k\} \text{ and } m = j \\ \uparrow \ \text{ in all other cases} \end{cases}$$

Then f is partial-recursive, since g is. By the smn theorem, there exists a recursive function $s : \mathbb{N} \to \mathbb{N}$ such that $f(n,m) = \varphi_{s(n)}(m)$ for all m, n. Since s is recursive, the Kleene recursion theorem shows that there must be some n' such that $\varphi_{s(n')} = \varphi_{n'}$. Thus $f(n',m) = \varphi_{n'}(m)$ for all $m \in \mathbb{N}$. Moreover, by definition, $\varphi_{n'}(m)$ can halt on at most one of the cases $m = 0, \ldots, m = k$ (if at all), and no other values. Thus $|\{0,\ldots,k\} \cap W_{n'}| \leq 1$. So $g(n',0,\ldots,k)$ will halt and output $j \in \{0,\ldots,k\} \setminus W_{n'}$ (by construction of g). But since $g(n',0,\ldots,k)$ halts with output j in $\{0,\ldots,k\}$, then $f(n',j)$ halts (by construction of f). Hence $\varphi_{n'}(j)$ halts (by definition of $\varphi_{n'}$), and so $j \in W_{n'}$ since $W_{n'}$ is precisely the halting set of $\varphi_{n'}$. Thus we have a contradiction, as we showed $j \notin W_{n'}$, so no such g can exist.

Proof.

Assume such a g exists. Define $f : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ by

$$f(n, m) := \begin{cases} 0 \text{ if } g(n, 0, \ldots, k) = j \in \{0, \ldots, k\} \text{ and } m = j \\ \uparrow \ \text{ in all other cases} \end{cases}$$

Then f is partial-recursive, since g is. By the smn theorem, there exists a recursive function $s : \mathbb{N} \to \mathbb{N}$ such that $f(n, m) = \varphi_{s(n)}(m)$ for all m, n. Since s is recursive, the Kleene recursion theorem shows that there must be some n' such that $\varphi_{s(n')} = \varphi_{n'}$. Thus $f(n', m) = \varphi_{n'}(m)$ for all $m \in \mathbb{N}$. Moreover, by definition, $\varphi_{n'}(m)$ can halt on at most one of the cases $m = 0, \ldots, m = k$ (if at all), and no other values. Thus $|\{0, \ldots, k\} \cap W_{n'}| \leq 1$. So $g(n', 0, \ldots, k)$ will halt and output $j \in \{0, \ldots, k\} \setminus W_{n'}$ (by construction of g). But since $g(n', 0, \ldots, k)$ halts with output j in $\{0, \ldots, k\}$, then $f(n', j)$ halts (by construction of f). Hence $\varphi_{n'}(j)$ halts (by definition of $\varphi_{n'}$), and so $j \in W_{n'}$ since $W_{n'}$ is precisely the halting set of $\varphi_{n'}$. Thus we have a contradiction, as we showed $j \notin W_{n'}$, so no such g can exist.

In a very similar manner, we can prove the following:

In a very similar manner, we can prove the following:

Lemma (Second recursion theory result)

In a very similar manner, we can prove the following:

Lemma (Second recursion theory result)

Fix any $k > 0$*. Then there is no partial recursive function* $g : \mathbb{N}^{k+2} \to \mathbb{N}$ *such that, given* $n, x_0, \ldots, x_k \in \mathbb{N}$ *satisfying* $\{x_0, \ldots, x_k\} \not\subseteq W_n$*, we have that* $g(n, x_0, \ldots, x_k)$ *halts with output* $x_i \in \{x_0, \ldots, x_k\}$ *such that* $\{x_0, \ldots, \hat{x}_i, \ldots x_k\} \not\subseteq W_n$*.*

In a very similar manner, we can prove the following:

Lemma (Second recursion theory result)

Fix any $k > 0$. Then there is no partial recursive function $g : \mathbb{N}^{k+2} \to \mathbb{N}$ such that, given $n, x_0, \dots, x_k \in \mathbb{N}$ satisfying $\{x_0, \dots, x_k\} \not\subseteq W_n$, we have that $g(n, x_0, \dots, x_k)$ halts with output $x_i \in \{x_0, \dots, x_k\}$ such that $\{x_0, \dots, \hat{x}_i, \dots x_k\} \not\subseteq W_n$.

That is, given a recursively enumerable set W_n and a finite set $F \not\subseteq W_n$, we can't, in general, recursively find a proper subset $A \subset F$ such that $A \not\subseteq W_n$.

2. Group theory results:

2. Group theory results:

From hereon, if P is a presentation of a group, the we denote by $\overline{P}$ the group presented by P.

2. Group theory results:

From hereon, if P is a presentation of a group, the we denote by $\overline{P}$ the group presented by P.

Theorem (Boone-Adian-Rabin)

2. Group theory results:

From hereon, if P is a presentation of a group, the we denote by $\overline{P}$ the group presented by P.

Theorem (Boone-Adian-Rabin)

We have an explicit algorithm that, on input of $m, n \in \mathbb{N}$, constructs a finite presentation $\Pi_{m,n}$ such that:

2. Group theory results:

From hereon, if P is a presentation of a group, the we denote by $\overline{P}$ the group presented by P.

Theorem (Boone-Adian-Rabin)

We have an explicit algorithm that, on input of $m, n \in \mathbb{N}$, *constructs a finite presentation* $\Pi_{m,n}$ *such that:*
1. $\overline{\Pi}_{m,n} \cong \{e\}$ *if and only if* $n \in W_m$.

2. Group theory results:

From hereon, if P is a presentation of a group, the we denote by $\overline{P}$ the group presented by P.

Theorem (Boone-Adian-Rabin)

We have an explicit algorithm that, on input of $m, n \in \mathbb{N}$, constructs a finite presentation $\Pi_{m,n}$ such that:
1. $\overline{\Pi}_{m,n} \cong \{e\}$ if and only if $n \in W_m$.
2. If $\overline{\Pi}_{m,n}$ is non-trivial then it is perfect, 2-generated, torsion free, and freely indecomposable (ie: can't be expressed as a non-trivial free product).

2. Group theory results:

From hereon, if P is a presentation of a group, the we denote by $\overline{P}$ the group presented by P.

Theorem (Boone-Adian-Rabin)

We have an explicit algorithm that, on input of $m, n \in \mathbb{N}$, constructs a finite presentation $\Pi_{m,n}$ such that:
1. $\overline{\Pi}_{m,n} \cong \{e\}$ if and only if $n \in W_m$.
2. If $\overline{\Pi}_{m,n}$ is non-trivial then it is perfect, 2-generated, torsion free, and freely indecomposable (ie: can't be expressed as a non-trivial free product).

This is VERY useful to us, as it allows us to turn information about numbers into information about groups (and vice versa).

2. Group theory results:

From hereon, if P is a presentation of a group, the we denote by $\overline{P}$ the group presented by P.

Theorem (Boone-Adian-Rabin)

We have an explicit algorithm that, on input of $m, n \in \mathbb{N}$, constructs a finite presentation $\Pi_{m,n}$ such that:
1. $\overline{\Pi}_{m,n} \cong \{e\}$ if and only if $n \in W_m$.
2. If $\overline{\Pi}_{m,n}$ is non-trivial then it is perfect, 2-generated, torsion free, and freely indecomposable (ie: can't be expressed as a non-trivial free product).

This is VERY useful to us, as it allows us to turn information about numbers into information about groups (and vice versa).

We need one more preliminary result on free products of groups:

Theorem (Grushko-Neumann decomposition)

Theorem (Grushko-Neumann decomposition)

Let P be a finite presentation of a group that splits as a free product $A_1 * \cdots * A_n$, with all the A_i indecomposable. Let $B_1 * \cdots * B_k$ be another such splitting into indecomposable groups. Then $n = k$, and there exists a permutation $\sigma \in S_n$ such that $A_i \cong B_{\sigma(i)}$ for all $1 \leq i \leq n$.

Theorem (Grushko-Neumann decomposition)

Let P be a finite presentation of a group that splits as a free product $A_1 * \cdots * A_n$, with all the A_i indecomposable. Let $B_1 * \cdots * B_k$ be another such splitting into indecomposable groups. Then $n = k$, and there exists a permutation $\sigma \in S_n$ such that $A_i \cong B_{\sigma(i)}$ for all $1 \leq i \leq n$.

Using this, and our $\Pi_{m,n}$ presentations, we can now show:

Theorem (Grushko-Neumann decomposition)

Let P be a finite presentation of a group that splits as a free product $A_1 * \cdots * A_n$, with all the A_i indecomposable. Let $B_1 * \cdots * B_k$ be another such splitting into indecomposable groups. Then $n = k$, and there exists a permutation $\sigma \in S_n$ such that $A_i \cong B_{\sigma(i)}$ for all $1 \leq i \leq n$.

Using this, and our $\Pi_{m,n}$ presentations, we can now show:

Lemma (Encoding recursion theory into groups)

Theorem (Grushko-Neumann decomposition)

Let P be a finite presentation of a group that splits as a free product $A_1 * \cdots * A_n$, with all the A_i indecomposable. Let $B_1 * \cdots * B_k$ be another such splitting into indecomposable groups. Then $n = k$, and there exists a permutation $\sigma \in S_n$ such that $A_i \cong B_{\sigma(i)}$ for all $1 \leq i \leq n$.

Using this, and our $\Pi_{m,n}$ presentations, we can now show:

Lemma (Encoding recursion theory into groups)

There is no algorithm that, on input of a finite presentation of the form $Q = \Pi_{n,a} * \Pi_{n,b} * \Pi_{n,c}$, where $|\{a, b, c\} \cap W_n| \leq 1$, outputs two finite presentations A, B of non-trivial groups such that $\overline{Q} \cong \overline{A} * \overline{B}$.

Proof.

Proof.

We proceed by contradiction. As $|\{a, b, c\} \cap W_n| \leq 1$, we must have at least two of $\overline{\Pi}_{n,a}, \overline{\Pi}_{n,b}, \overline{\Pi}_{n,c}$ are non-trivial. So split $\overline{Q}$ as $\overline{A} * \overline{B}$, with $\overline{A}, \overline{B}$ both non-trivial. We consider 2 cases:

Proof.

We proceed by contradiction. As $|\{a, b, c\} \cap W_n| \leq 1$, we must have at least two of $\overline{\Pi}_{n,a}, \overline{\Pi}_{n,b}, \overline{\Pi}_{n,c}$ are non-trivial. So split $\overline{Q}$ as $\overline{A} * \overline{B}$, with $\overline{A}, \overline{B}$ both non-trivial. We consider 2 cases:
Case 1. Precisely one of a, b, c lies in W_n. If $a \in W_n$, then $\overline{Q} \cong \overline{\Pi}_{n,b} * \overline{\Pi}_{n,c}$. This is an indecomposable splitting, so $\overline{A}$ must be isomorphic to at least one of $\overline{\Pi}_{n,b}, \overline{\Pi}_{n,c}$. The same idea works if instead $b \in W_n$ or $c \in W_n$. So, regardless of which of a, b, c lie in W_n, $\overline{A}$ is isomorphic to at least one of $\overline{\Pi}_{n,a}, \overline{\Pi}_{n,b}, \overline{\Pi}_{n,c}$.

Proof.

We proceed by contradiction. As $|\{a,b,c\} \cap W_n| \leq 1$, we must have at least two of $\overline{\Pi}_{n,a}, \overline{\Pi}_{n,b}, \overline{\Pi}_{n,c}$ are non-trivial. So split $\overline{Q}$ as $\overline{A} * \overline{B}$, with $\overline{A}, \overline{B}$ both non-trivial. We consider 2 cases:
Case 1. Precisely one of a, b, c lies in W_n. If $a \in W_n$, then $\overline{Q} \cong \overline{\Pi}_{n,b} * \overline{\Pi}_{n,c}$. This is an indecomposable splitting, so $\overline{A}$ must be isomorphic to at least one of $\overline{\Pi}_{n,b}, \overline{\Pi}_{n,c}$. The same idea works if instead $b \in W_n$ or $c \in W_n$. So, regardless of which of a, b, c lie in W_n, $\overline{A}$ is isomorphic to at least one of $\overline{\Pi}_{n,a}, \overline{\Pi}_{n,b}, \overline{\Pi}_{n,c}$.
Case 2. None of a, b, c lie in W_n. Then $\overline{Q} \cong \overline{\Pi}_{n,a} * \overline{\Pi}_{n,b} * \overline{\Pi}_{n,c}$ is a splitting into indecomposable groups. Thus precisely one of $\overline{A}, \overline{B}$ splits as a free product; the other does not. Hence, at least one of $\overline{A}, \overline{B}$ must be isomorphic to at least one of $\overline{\Pi}_{n,a}, \overline{\Pi}_{n,b}, \overline{\Pi}_{n,c}$.

Proof.

We proceed by contradiction. As $|\{a, b, c\} \cap W_n| \leq 1$, we must have at least two of $\overline{\Pi}_{n,a}, \overline{\Pi}_{n,b}, \overline{\Pi}_{n,c}$ are non-trivial. So split $\overline{Q}$ as $\overline{A} * \overline{B}$, with $\overline{A}, \overline{B}$ both non-trivial. We consider 2 cases:

Case 1. Precisely one of a, b, c lies in W_n. If $a \in W_n$, then $\overline{Q} \cong \overline{\Pi}_{n,b} * \overline{\Pi}_{n,c}$. This is an indecomposable splitting, so $\overline{A}$ must be isomorphic to at least one of $\overline{\Pi}_{n,b}, \overline{\Pi}_{n,c}$. The same idea works if instead $b \in W_n$ or $c \in W_n$. So, regardless of which of a, b, c lie in W_n, $\overline{A}$ is isomorphic to at least one of $\overline{\Pi}_{n,a}, \overline{\Pi}_{n,b}, \overline{\Pi}_{n,c}$.

Case 2. None of a, b, c lie in W_n. Then $\overline{Q} \cong \overline{\Pi}_{n,a} * \overline{\Pi}_{n,b} * \overline{\Pi}_{n,c}$ is a splitting into indecomposable groups. Thus precisely one of $\overline{A}, \overline{B}$ splits as a free product; the other does not. Hence, at least one of $\overline{A}, \overline{B}$ must be isomorphic to at least one of $\overline{\Pi}_{n,a}, \overline{\Pi}_{n,b}, \overline{\Pi}_{n,c}$.

In either case, at least one of $\overline{\Pi}_{n,a}, \overline{\Pi}_{n,b}, \overline{\Pi}_{n,c}$ is isomorphic to at least one of $\overline{A}, \overline{B}$; these latter two being non-trivial groups. We can recursively begin searching for such an isomorphism. This process will eventually halt, and thus give one of $\overline{\Pi}_{n,a}, \overline{\Pi}_{n,b}, \overline{\Pi}_{n,c}$ as non-trivial, and hence one of a, b, c not in W_n. This contradicts our first recursion theory result.

As an immediate consequence we have:

As an immediate consequence we have:

Theorem (C. 2010)

As an immediate consequence we have:

Theorem (C. 2010)

There is no algorithm that, on input of a finite presentation P of a group that is a free product of two non-trivial finitely presented groups, outputs two finite presentations P_1, P_2 which represent non-trivial groups and whose free product is isomorphic to $\overline{P}$.

As an immediate consequence we have:

Theorem (C. 2010)

There is no algorithm that, on input of a finite presentation P of a group that is a free product of two non-trivial finitely presented groups, outputs two finite presentations P_1, P_2 which represent non-trivial groups and whose free product is isomorphic to $\overline{P}$.

That is, there is no algorithm that, on input of a finite presentation of a non-trivial free product, algorithmically splits it (For if we could, then we could split $\Pi_{n,a} * \Pi_{n,b} * \Pi_{n,c}$).

In a very similar style, using our second recursion theory result and a slightly different way of combining the $\Pi_{m,n}$ groups, we can show the following:

In a very similar style, using our second recursion theory result and a slightly different way of combining the $\Pi_{m,n}$ groups, we can show the following:

Theorem (C. 2010)

In a very similar style, using our second recursion theory result and a slightly different way of combining the $\Pi_{m,n}$ groups, we can show the following:

Theorem (C. 2010)

Fix any $k > 0$. Then there is no algorithm that, on input of a finite presentation $P = \langle X | R \rangle$ of a non-trivial group $\overline{P}$, outputs a word w on X of length at most k such that w is non-trivial in $\overline{P}$.

In a very similar style, using our second recursion theory result and a slightly different way of combining the $\Pi_{m,n}$ groups, we can show the following:

Theorem (C. 2010)

Fix any $k > 0$. Then there is no algorithm that, on input of a finite presentation $P = \langle X | R \rangle$ of a non-trivial group $\overline{P}$, outputs a word w on X of length at most k such that w is non-trivial in $\overline{P}$.

So if there was an algorithm to output a non-trivial element from a non-trivial group, then there would be no bound on the length of the words which it could output. Hence, knowing a group is non-trivial is NOT enough to be able to algorithmically output a non-trivial generator (which is the first place one would naively look for a non-trivial element).

Again, using our existing recursion theory results, our $\Pi_{m,n}$ groups, and a straightforward application of a result by Adian-Rabin, we can show the following:

Again, using our existing recursion theory results, our $\Pi_{m,n}$ groups, and a straightforward application of a result by Adian-Rabin, we can show the following:

Lemma

There is no algorithm that, on input of a finite presentation $P = \langle X|R \rangle$ of a group with torsion, and some finite n which is the order of some element of $\overline{P}$, outputs a word w on X which represents any torsion element of $\overline{P}$ (not necessarily of order n).

Again, using our existing recursion theory results, our $\Pi_{m,n}$ groups, and a straightforward application of a result by Adian-Rabin, we can show the following:

Lemma

There is no algorithm that, on input of a finite presentation $P = \langle X|R \rangle$ of a group with torsion, and some finite n which is the order of some element of $\overline{P}$, outputs a word w on X which represents any torsion element of $\overline{P}$ (not necessarily of order n).

So knowing a group has torsion, AND the order of some torsion element, is still not enough to algorithmically construct a torsion element. We can easily use this to show the following result:

Again, using our existing recursion theory results, our $\Pi_{m,n}$ groups, and a straightforward application of a result by Adian-Rabin, we can show the following:

Lemma

There is no algorithm that, on input of a finite presentation $P = \langle X|R \rangle$ of a group with torsion, and some finite n which is the order of some element of $\overline{P}$, outputs a word w on X which represents any torsion element of $\overline{P}$ (not necessarily of order n).

So knowing a group has torsion, AND the order of some torsion element, is still not enough to algorithmically construct a torsion element. We can easily use this to show the following result:

Theorem (C. 2010)

There is no algorithm that, on input of two finite presentations $P = \langle X|R \rangle$ and $Q = \langle Y|S \rangle$ such that $\overline{P}$ embeds in $\overline{Q}$, outputs an explicit map $\theta : X \to W(Y)$ such that θ extends to an embedding $\overline{\theta} : \overline{P} \hookrightarrow \overline{Q}$.

Proof.

Proof.

Just take Q to be a presentation of a group with torsion of order n. Then $\overline{\langle t | t^n \rangle}$ embeds in $\overline{Q}$. But being able to construct such an embedding would enable us to identify a torsion element (the image of t), which contradicts the previous lemma. □

Proof.
Just take Q to be a presentation of a group with torsion of order n. Then $\overline{\langle t | t^n \rangle}$ embeds in $\overline{Q}$. But being able to construct such an embedding would enable us to identify a torsion element (the image of t), which contradicts the previous lemma. □

So knowing that $\overline{P}$ embeds in $\overline{Q}$ is not NOT sufficient to construct such an embedding. Compare this with the fact that:

Proof.
Just take Q to be a presentation of a group with torsion of order n. Then $\overline{\langle t | t^n \rangle}$ embeds in $\overline{Q}$. But being able to construct such an embedding would enable us to identify a torsion element (the image of t), which contradicts the previous lemma. □

So knowing that $\overline{P}$ embeds in $\overline{Q}$ is not NOT sufficient to construct such an embedding. Compare this with the fact that:

1. Knowing $\overline{P}$ surjects onto $\overline{Q}$ is enough to construct a surjection.

Proof.

Just take Q to be a presentation of a group with torsion of order n. Then $\overline{\langle t | t^n \rangle}$ embeds in $\overline{Q}$. But being able to construct such an embedding would enable us to identify a torsion element (the image of t), which contradicts the previous lemma. □

So knowing that $\overline{P}$ embeds in $\overline{Q}$ is not NOT sufficient to construct such an embedding. Compare this with the fact that:

1. Knowing $\overline{P}$ surjects onto $\overline{Q}$ is enough to construct a surjection.
2. Knowing $\overline{P} \cong \overline{Q}$ is enough to construct an isomorphism.

3. Applications to 4-Manifolds:

3. Applications to 4-Manifolds:

Theorem (Markov)

3. Applications to 4-Manifolds:

Theorem (Markov)

There is a recursive procedure that, on input of a finite presentation $P = \langle X | R \rangle$ *of a group, constructs a finite triangulation* $M(P)$ *of a closed 4-manifold with the following properties:*

3. Applications to 4-Manifolds:

Theorem (Markov)

There is a recursive procedure that, on input of a finite presentation $P = \langle X | R \rangle$ of a group, constructs a finite triangulation $M(P)$ of a closed 4-manifold with the following properties:

1. $\pi_1(M(P)) \cong \overline{P}$.

3. Applications to 4-Manifolds:

Theorem (Markov)

There is a recursive procedure that, on input of a finite presentation $P = \langle X | R \rangle$ of a group, constructs a finite triangulation $M(P)$ of a closed 4-manifold with the following properties:

1. $\pi_1(M(P)) \cong \overline{P}$.

2. If P and Q are finite presentations, then $M(P * Q)$ is homeomorphic to the connect sum $M(P) \# M(Q)$.

3. Applications to 4-Manifolds:

Theorem (Markov)

There is a recursive procedure that, on input of a finite presentation $P = \langle X | R \rangle$ of a group, constructs a finite triangulation $M(P)$ of a closed 4-manifold with the following properties:

1. $\pi_1(M(P)) \cong \overline{P}$.

2. If P and Q are finite presentations, then $M(P * Q)$ is homeomorphic to the connect sum $M(P)\#M(Q)$.

Combining this with our first group theory lemma regarding splitting $\Pi_{n,a} * \Pi_{n,b} * \Pi_{n,c}$, and the fact that we can 'read off' a presentation for the fundamental group of a finite triangulation, we get:

Corollary (C. 2010)

Corollary (C. 2010)

There is no algorithm that, on input of a finite triangulation of a closed 4-manifold M which splits as a connect sum of two non-simply connected manifolds, outputs two finite triangulations of non-simply connected closed 4-manifolds M_1, M_2 whose connect sum is homeomorphic to M.

Corollary (C. 2010)

There is no algorithm that, on input of a finite triangulation of a closed 4-manifold M which splits as a connect sum of two non-simply connected manifolds, outputs two finite triangulations of non-simply connected closed 4-manifolds M_1, M_2 whose connect sum is homeomorphic to M.

So just knowing that a 4-manifold splits as a connect sum of non-simply connected pieces is NOT enough to be able to split it as such. (If we could, then the Markov construction would allow us to split $\Pi_{n,a} * \Pi_{n,b} * \Pi_{n,c}$).

We can also apply the Markov construction to carry over some of our other algorithmic results from algebra to geometry.

We can also apply the Markov construction to carry over some of our other algorithmic results from algebra to geometry.

Corollary (C. 2010)

We can also apply the Markov construction to carry over some of our other algorithmic results from algebra to geometry.

Corollary (C. 2010)

There is no algorithm that, on input of a finite triangulation of a closed 4-manifold M such that $\pi_1(M)$ has torsion, outputs an essential loop γ in M which represents a torsion element in $\pi_1(M)$.

We can also apply the Markov construction to carry over some of our other algorithmic results from algebra to geometry.

Corollary (C. 2010)

There is no algorithm that, on input of a finite triangulation of a closed 4-manifold M such that $\pi_1(M)$ has torsion, outputs an essential loop γ in M which represents a torsion element in $\pi_1(M)$.

If we could do this, then the Markov construction would allow us to construct a torsion element in any finite presentation of a torsion group, which we showed impossible in our earlier lemma.

Full details of the material presented here can be found in the preprint:

M. Chiodo, *Finding non-trivial elements and splittings in groups*, arXiv:1002.2786v3 (2010).