MIX is a 2-MCFL

Sylvain Salvati

INRIA Bordeaux Sud-Ouest, LaBRI, Université de Bordeaux

Workshop on Multiple Context-Free Grammars and Related Formalisms

Outline

Outline

The *MIX* language

$$MIX = \{w \in \{a; b; c\}^* || w|_a = |w|_b = |w|_c\}$$

The *MIX* language

$MIX = \{w \in \{a; b; c\}^* || w|_a = |w|_b = |w|_c\}$
The *MIX* language and computational linguists:

- Joshi 1985: [*MIX*] *represents the extreme case of the degree of free word order permitted in a language. This extreme case is linguistically not relevant. [. . .] TAGs also cannot generate this language although for TAGs the proof is not in hand yet.*

The *MIX* language

$MIX = \{w \in \{a; b; c\}^* || w|_a = |w|_b = |w|_c\}$

The *MIX* language and computational linguists:

- Joshi 1985: [*MIX*] *represents the extreme case of the degree of free word order permitted in a language. This extreme case is linguistically not relevant. [...] TAGs also cannot generate this language although for TAGs the proof is not in hand yet.*
- Joshi *et al.* 1990: *Mildly Context Sensitive Grammars capture only certain kinds of dependencies, e.g, nested dependencies and certain limited kinds of crossing dependencies (e.g., in the subordinate clause constructions in Dutch or some variations of them but perhaps not in the so-called MIX (or Bach) language) [...] MCTAGS also belong to Mildly Context Sensitive Grammars...*

The O_2 language

$$O_2 = \{w \in \{a; \overline{a}; b; \overline{b}\}^* \mid |w|_a = |w|_{\overline{a}} \wedge |w|_b = |w|_{\overline{b}}\}$$

The O_2 language

$O_2 = \{w \in \{a; \overline{a}; b; \overline{b}\}^* \mid |w|_a = |w|_{\overline{a}} \wedge |w|_b = |w|_{\overline{b}}\}$

The O_2 language is of interest in computational group theory:

- the monoid homomorphism $z : \{a; \overline{a}; b; \overline{b}\}^* \to \mathbb{Z}^2$ such that $z(a) = (1, 0)$, $z(\overline{a}) = (-1, 0)$, $z(b) = (0, 1)$, $z(\overline{b}) = (0, -1)$ has O_2 as kernel.

The O_2 language

$O_2 = \{w \in \{a; \overline{a}; b; \overline{b}\}^* \mid |w|_a = |w|_{\overline{a}} \wedge |w|_b = |w|_{\overline{b}}\}$
The O_2 language is of interest in computational group theory:

- the monoid homomorphism $z : \{a; \overline{a}; b; \overline{b}\}^* \to \mathbb{Z}^2$ such that $z(a) = (1, 0)$, $z(\overline{a}) = (-1, 0)$, $z(b) = (0, 1)$, $z(\overline{b}) = (0, -1)$ has O_2 as kernel.
- O_2 is a group language that is not context free. An open question is whether it is an indexed language.

MIX and O_2 are rationally equivalent

The following transductions are due to Kanazawa:

- There is a rational transduction from O_2 to MIX: let $R = \{a|b|\overline{a}\overline{b}\}^*$, then $MIX = h(O_2 \cap R)$ if $h(a) = a$, $h(b) = b$, $h(\overline{a}) = c$ and $h(\overline{b}) = \epsilon$.

MIX and O_2 are rationally equivalent

The following transductions are due to Kanazawa:

- There is a rational transduction from O_2 to MIX: let $R = \{a|b|\overline{a}\overline{b}\}^*$, then $MIX = h(O_2 \cap R)$ if $h(a) = a$, $h(b) = b$, $h(\overline{a}) = c$ and $h(\overline{b}) = \epsilon$.
- There is a rational transduction from MIX to O_2: let $R = \{abab|cc|cbcb|aa\}^*$, then O_2, then $O_2 = g^{-1}(MIX \cap R)$ with $g(a) = abab$, $g(\overline{a}) = cc$, $g(b) = cbcb$ $g(\overline{b}) = aa$.

MIX and O_2 are rationally equivalent

The following transductions are due to Kanazawa:

- There is a rational transduction from O_2 to MIX: let $R = \{a|b|\overline{a}\overline{b}\}^*$, then $MIX = h(O_2 \cap R)$ if $h(a) = a$, $h(b) = b$, $h(\overline{a}) = c$ and $h(\overline{b}) = \epsilon$.
- There is a rational transduction from MIX to O_2: let $R = \{abab|cc|cbcb|aa\}^*$, then O_2, then $O_2 = g^{-1}(MIX \cap R)$ with $g(a) = abab$, $g(\overline{a}) = cc$, $g(b) = cbcb$ $g(\overline{b}) = aa$.

NB: $w \in MIX \cap R$ iff $|w|_{abab} + |w|_{aa} = |w|_{cbcb} + |w|_{abab} = |w|_{cc} + |w|_{cbcb}$ iff $|w|_{abab} = |w|_{cc}$ and $|w|_{cbcb} = |w|_{aa}$.

MIX and O_2 are rationally equivalent

The following transductions are due to Kanazawa:

- There is a rational transduction from O_2 to MIX: let $R = \{a|b|\overline{a}\overline{b}\}^*$, then $MIX = h(O_2 \cap R)$ if $h(a) = a$, $h(b) = b$, $h(\overline{a}) = c$ and $h(\overline{b}) = \epsilon$.
- There is a rational transduction from MIX to O_2: let $R = \{abab|cc|cbcb|aa\}^*$, then O_2, then $O_2 = g^{-1}(MIX \cap R)$ with $g(a) = abab$, $g(\overline{a}) = cc$, $g(b) = cbcb$ $g(\overline{b}) = aa$.

Thus MIX belongs to a rational cone iff O_2 does.

Outline

The *MIX* and O_2 languages

Multiple Context Free Grammars (MCFGs)

A grammar for O_2

A Theorem on Jordan curves

Conclusion and conjectures

Multiple Context Free Grammars

MCFGs are context free grammars of tuples of strings. A MCFG, G is a tuple (N, T, S, R) where:

- N is a ranked alphabet of non-terminals,

Multiple Context Free Grammars

MCFGs are context free grammars of tuples of strings. A MCFG, G is a tuple (N, T, S, R) where:

- N is a ranked alphabet of non-terminals,
- T is the alphabet of terminals,

Multiple Context Free Grammars

MCFGs are context free grammars of tuples of strings. A MCFG, G is a tuple (N, T, S, R) where:

- N is a ranked alphabet of non-terminals,
- T is the alphabet of terminals,
- S is an element of N of arity 1,

Multiple Context Free Grammars

MCFGs are context free grammars of tuples of strings. A MCFG, G is a tuple (N, T, S, R) where:

- N is a ranked alphabet of non-terminals,
- T is the alphabet of terminals,
- S is an element of N of arity 1,
- R is the set of rules of the form:

$$A(s_1, \ldots, s_n) \text{:-} B_1(x_1^1, \ldots, x_{k_1}^1), \ldots, B_m(x_1^m, \ldots, x_{k_m}^m)$$

where the s_i are strings of $T \cup \{x_j^i | i \in [1, m], j \in [1, k_i]\}$ so that x_j^i has at most one occurrence in $s_1 \ldots s_n$.

Multiple Context Free Grammars

MCFGs are context free grammars of tuples of strings. A MCFG, G is a tuple (N, T, S, R) where:

- N is a ranked alphabet of non-terminals,
- T is the alphabet of terminals,
- S is an element of N of arity 1,
- R is the set of rules of the form:

$$A(s_1, \dots, s_n)\text{:-}B_1(x_1^1, \dots, x_{k_1}^1), \dots, B_m(x_1^m, \dots, x_{k_m}^m)$$

where the s_i are strings of $T \cup \{x_j^i | i \in [1, m], j \in [1, k_i]\}$ so that x_j^i has at most one occurrence in $s_1 \dots s_n$.
If $B_1(s_1^1, \dots, s_{k_1}^1)$, ... and $B_m(s_1^m, \dots, s_{k_m}^m)$ are derivable then $A(\sigma(s_1), \dots, \sigma(s_n))$ where $\sigma(x_j^i) = s_j^i$ is derivable.

Multiple Context Free Grammars

MCFGs are context free grammars of tuples of strings. A MCFG, G is a tuple (N, T, S, R) where:

- N is a ranked alphabet of non-terminals,
- T is the alphabet of terminals,
- S is an element of N of arity 1,
- R is the set of rules of the form:

$$A(s_1, \ldots, s_n) \text{:-} B_1(x_1^1, \ldots, x_{k_1}^1), \ldots, B_m(x_1^m, \ldots, x_{k_m}^m)$$

where the s_i are strings of $T \cup \{x_j^i | i \in [1, m], j \in [1, k_i]\}$ so that x_j^i has at most one occurrence in $s_1 \ldots s_n$.
If $B_1(s_1^1, \ldots, s_{k_1}^1)$, ... and $B_m(s_1^m, \ldots, s_{k_m}^m)$ are derivable then $A(\sigma(s_1), \ldots, \sigma(s_n))$ where $\sigma(x_j^i) = s_j^i$ is derivable.

The language defined by G is $\{s | S(s) \text{ is derivable}\}$.

Multiple Context Free Grammars

MCFGs are context free grammars of tuples of strings. A MCFG, G is a tuple (N, T, S, R) where:

- N is a ranked alphabet of non-terminals,
- T is the alphabet of terminals,
- S is an element of N of arity 1,
- R is the set of rules of the form:

$$A(s_1, \ldots, s_n) \text{:-} B_1(x_1^1, \ldots, x_{k_1}^1), \ldots, B_m(x_1^m, \ldots, x_{k_m}^m)$$

where the s_i are strings of $T \cup \{x_j^i | i \in [1, m], j \in [1, k_i]\}$ so that x_j^i has at most one occurrence in $s_1 \ldots s_n$.
If $B_1(s_1^1, \ldots, s_{k_1}^1)$, ... and $B_m(s_1^m, \ldots, s_{k_m}^m)$ are derivable then $A(\sigma(s_1), \ldots, \sigma(s_n))$ where $\sigma(x_j^i) = s_j^i$ is derivable.

The language defined by G is $\{s | S(s) \text{ is derivable}\}$.
If the maximal arity of N is lower than k, G is a k-MCFG.

Other characterization of MCFLs.

The languages definable with MCFGs are Multiple Context Free Languages (MCFLs). MCFLs form an Abstract Family of Language (thus they are closed under rational transduction), and are exactly captured by many kinds of formalisms:

- Linear Context Free Rewriting Systems,

Other characterization of MCFLs.

The languages definable with MCFGs are Multiple Context Free Languages (MCFLs). MCFLs form an Abstract Family of Language (thus they are closed under rational transduction), and are exactly captured by many kinds of formalisms:

- Linear Context Free Rewriting Systems,
- string languages definable with Hyperedge Replacement Grammars,

Other characterization of MCFLs.

The languages definable with MCFGs are Multiple Context Free Languages (MCFLs). MCFLs form an Abstract Family of Language (thus they are closed under rational transduction), and are exactly captured by many kinds of formalisms:

- Linear Context Free Rewriting Systems,
- string languages definable with Hyperedge Replacement Grammars,
- multi component Tree Adjoining Grammars,

Other characterization of MCFLs.

The languages definable with MCFGs are Multiple Context Free Languages (MCFLs). MCFLs form an Abstract Family of Language (thus they are closed under rational transduction), and are exactly captured by many kinds of formalisms:

- Linear Context Free Rewriting Systems,
- string languages definable with Hyperedge Replacement Grammars,
- multi component Tree Adjoining Grammars,
- second order Abstract Categorial Grammars, or non-duplicating higher-order extended IO/OI grammars (third-order collapsible pushdown automata),

Other characterization of MCFLs.

The languages definable with MCFGs are Multiple Context Free Languages (MCFLs). MCFLs form an Abstract Family of Language (thus they are closed under rational transduction), and are exactly captured by many kinds of formalisms:

- Linear Context Free Rewriting Systems,
- string languages definable with Hyperedge Replacement Grammars,
- multi component Tree Adjoining Grammars,
- second order Abstract Categorial Grammars, or non-duplicating higher-order extended IO/OI grammars (third-order collapsible pushdown automata),
- yield of the output languages of finite copying transductions,

Other characterization of MCFLs.

The languages definable with MCFGs are Multiple Context Free Languages (MCFLs). MCFLs form an Abstract Family of Language (thus they are closed under rational transduction), and are exactly captured by many kinds of formalisms:

- Linear Context Free Rewriting Systems,
- string languages definable with Hyperedge Replacement Grammars,
- multi component Tree Adjoining Grammars,
- second order Abstract Categorial Grammars, or non-duplicating higher-order extended IO/OI grammars (third-order collapsible pushdown automata),
- yield of the output languages of finite copying transductions,
- etc...

Outline

A 2-MCFG for O_2

$$
\begin{array}{rl}
S(xy) & \text{:- } Inv(x, y) \\
\hline
Inv(x_1 y_1, y_2 x_2) & \text{:- } Inv(x_1, x_2), Inv(y_1, y_2) \\
Inv(x_1 x_2 y_1, y_2) & \text{:- } Inv(x_1, x_2), Inv(y_1, y_2) \\
Inv(y_1, x_1 x_2 y_2) & \text{:- } Inv(x_1, x_2), Inv(y_1, y_2) \\
Inv(y_1 x_1 x_2, y_2) & \text{:- } Inv(x_1, x_2), Inv(y_1, y_2) \\
Inv(y_1, y_2 x_1 x_2) & \text{:- } Inv(x_1, x_2), Inv(y_1, y_2) \\
\hline
Inv(x_1 y_1 x_2, y_2) & \text{:- } Inv(x_1, x_2), Inv(y_1, y_2) \\
Inv(x_1, y_1 x_2 y_2) & \text{:- } Inv(x_1, x_2), Inv(y_1, y_2) \\
\hline
Inv(\epsilon, \epsilon) & \text{:- } \\
Inv(\alpha, \overline{\alpha}) & \text{:- } \\
Inv(\overline{\alpha}, \alpha) & \text{:- }
\end{array}
$$

where $\alpha \in \{a; b\}$

A 2-MCFG for O_2

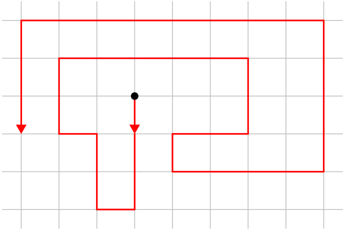
$$
\begin{array}{rl}
S(xy) & \text{:- } Inv(x, y) \\
\hline
Inv(x_1 y_1, y_2 x_2) & \text{:- } Inv(x_1, x_2), Inv(y_1, y_2) \\
Inv(x_1 x_2 y_1, y_2) & \text{:- } Inv(x_1, x_2), Inv(y_1, y_2) \\
Inv(y_1, x_1 x_2 y_2) & \text{:- } Inv(x_1, x_2), Inv(y_1, y_2) \\
Inv(y_1 x_1 x_2, y_2) & \text{:- } Inv(x_1, x_2), Inv(y_1, y_2) \\
Inv(y_1, y_2 x_1 x_2) & \text{:- } Inv(x_1, x_2), Inv(y_1, y_2) \\
\hline
Inv(x_1 y_1 x_2, y_2) & \text{:- } Inv(x_1, x_2), Inv(y_1, y_2) \\
Inv(x_1, y_1 x_2 y_2) & \text{:- } Inv(x_1, x_2), Inv(y_1, y_2) \\
\hline
Inv(\epsilon, \epsilon) & \text{:- } \\
Inv(\alpha, \overline{\alpha}) & \text{:- } \\
Inv(\overline{\alpha}, \alpha) & \text{:- }
\end{array}
$$

where $\alpha \in \{a; b\}$

Theorem: Given w_1 and w_2 such that $w_1 w_2 \in O_2$, $Inv(w_1, w_2)$ is derivable.

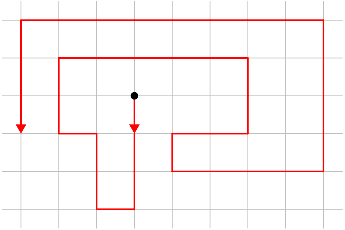
A graphical interpretation of O_2.

Graphical interpretation of the word
$\overline{aaa}\,\overline{b}aa\overline{b}aabbbbb\overline{aa}\,\overline{bb}\,\overline{a}bbbbaaaa\overline{bbbbbbbb}\,\overline{aaa}$:

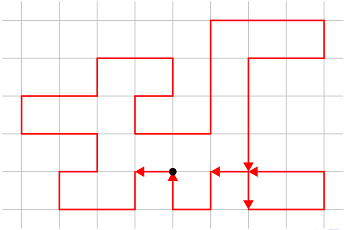


A graphical interpretation of O_2.

Graphical interpretation of the word
$\bar{a}\bar{a}\bar{a}\bar{b}aa\bar{b}aabbbbb\bar{a}\bar{a}\bar{b}\bar{b}\bar{a}bbbbaaaa\bar{b}\bar{b}\bar{b}\bar{b}\bar{b}\bar{b}\bar{b}\bar{b}\bar{a}\bar{a}\bar{a}$:

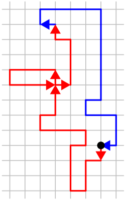


The words in O_2 are precisely the words that are represented as closed curves: $\bar{b}\bar{a}\bar{b}\bar{b}aba\bar{b}\bar{b}abbabb\bar{a}\bar{b}\bar{a}bbaaabbb\bar{a}\bar{b}\bar{b}\bar{a}\bar{a}\bar{a}\bar{a}bba\bar{b}\bar{b}\bar{b}\bar{a}\bar{b}a$

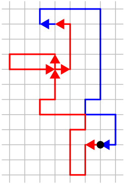


Parsing with the grammar

Rule $Inv(x_1 y_1 x_2, y_2) \text{ :- } Inv(x_1, x_2), Inv(y_1, y_2)$



$Inv(\color{red}{\bar{a}\bar{b}\bar{a}\bar{a}\bar{b}aaaba\bar{b}\bar{b}\bar{b}abaaa\bar{b}\bar{b}\bar{b}\bar{a}bbbbaaa\bar{b}a}, \color{blue}{\bar{b}abbbb\bar{a}\bar{a}\bar{a}\bar{a}\bar{a}\bar{a}\bar{b}\bar{a}bb\bar{a}\bar{a}\bar{b}})$

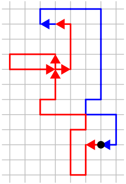


$Inv(\color{red}{\bar{b}\bar{a}\bar{a}\bar{b}aaaba\bar{b}\bar{b}\bar{b}abaaa\bar{b}\bar{b}\bar{b}\bar{a}bbbbaaa\bar{b}}, \color{blue}{\bar{b}abbbb\bar{a}\bar{a}\bar{a}\bar{a}\bar{a}\bar{a}\bar{b}\bar{a}bb\bar{a}\bar{a}\bar{b}})$

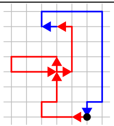
$Inv(\color{red}{\bar{a}}, \color{blue}{a})$

Parsing with the grammar

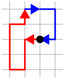
Rule: $Inv(x_1 y_1, y_2 x_2)$:-$Inv(x_1, x_2), Inv(y_1, y_2)$



$Inv(\overline{b}\overline{a}\overline{a}\overline{b}aaaba\overline{b}\overline{b}\overline{b}abaaa\overline{b}\overline{b}\overline{b}\overline{a}bbbbaaa\overline{b}, \overline{b}abbbb\overline{a}\overline{a}\overline{a}\overline{a}\overline{a}\overline{a}\overline{b}\overline{a}bb\overline{a}\overline{a}\overline{b})$



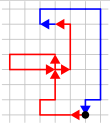
$Inv(\overline{b}\overline{b}\overline{b}abaaa\overline{b}\overline{b}\overline{b}\overline{a}bbbbaaa\overline{b}, \overline{b}abbbb\overline{a}\overline{a}\overline{a}\overline{a}\overline{a}\overline{a}\overline{b}\overline{a})$



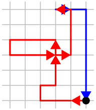
$Inv(\overline{b}\overline{a}\overline{a}\overline{b}aaaba, bb\overline{a}\overline{a}\overline{b})$

Parsing with the grammar

Rule $Inv(x_1, y_1x_2y_2)$:-$Inv(x_1, x_2), Inv(y_1, y_2)$



$Inv(\bar{b}\bar{b}\bar{b}abaaa\bar{b}\bar{b}\bar{b}\bar{a}bbbbaaa\bar{b}, \bar{b}abbbb\bar{a}\bar{a}\bar{a}\bar{a}\bar{a}\bar{a}\bar{b}\bar{a})$

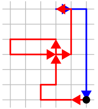


$Inv(\bar{b}\bar{b}\bar{b}abaaa\bar{b}\bar{b}\bar{b}\bar{a}bbbbaaa\bar{b}, bb\bar{a}\bar{a}\bar{a}\bar{a}\bar{a}\bar{a})$

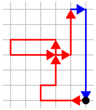
$Inv(\bar{b}abb, \bar{b}\bar{a})$

Parsing with the grammar

Rule: $Inv(x_1 y_1, y_2 x_2)$:-$Inv(x_1, x_2), Inv(y_1, y_2)$



$Inv(\overline{bbb}abaaa\overline{bbb}\overline{a}bbbbaaa\overline{b}, bb\overline{aaaaaa})$

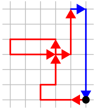


$Inv(\overline{bbb}abaaa\overline{bbb}\overline{a}bbbbaaa, b\overline{aaaaaa})$

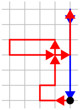
$Inv(\overline{b}, b)$

Parsing with the grammar

Rule: $Inv(x_1 y_1, y_2 x_2)$:-$Inv(x_1, x_2), Inv(y_1, y_2)$



$Inv(\overline{bbb}abaaa\overline{bbb}\overline{a}bbbbaaa, b\overline{aaaaaa})$

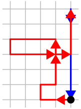


$Inv(\overline{bb}abaaa\overline{bbb}\overline{a}bbbbaaa, \overline{aaaaaa})$

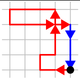
$Inv(\overline{b}, b)$

Parsing with the grammar

Rule: $Inv(x_1 y_1, y_2 x_2) \text{:-} Inv(x_1, x_2), Inv(y_1, y_2)$



$Inv(\overline{bb}abaaa\overline{bbb}\overline{a}bbbbaaa, \overline{aaaaaa})$

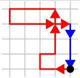


$Inv(\overline{bb}abaaa\overline{bbb}\overline{a}bbbb, \overline{aaa})$

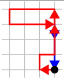
$Inv(aaa, \overline{aaa})$

Parsing with the grammar

Rule: $Inv(x_1 y_1 x_2, y_2)$:- $Inv(x_1, x_2), Inv(y_1, y_2)$



$Inv(\bar{b}\bar{b}aba aa\bar{b}\bar{b}\bar{b}\bar{a}bbbb, \bar{a}\bar{a}\bar{a})$



$Inv(\bar{b}abaaa\bar{b}\bar{b}\bar{b}\bar{a}bbb, \bar{a}\bar{a}\bar{a})$

$Inv(\bar{a}, a)$

The proof of the Theorem

Theorem: Given w_1 and w_2 such that $w_1 w_2 \in O_2$, $Inv(w_1, w_2)$ is derivable.

The proof is done by induction on the lexicographically ordered pairs $(|w_1 w_2|, \max(|w_1|, |w_2|))$.
There are five cases:

Case 1: w_1 or w_2 equal ϵ:

The proof of the Theorem

Theorem: Given w_1 and w_2 such that $w_1 w_2 \in O_2$, $Inv(w_1, w_2)$ is derivable.

The proof is done by induction on the lexicographically ordered pairs $(|w_1 w_2|, \max(|w_1|, |w_2|))$.
There are five cases:

Case 1: w_1 or w_2 equal ϵ:
w.l.o.g., $w_1 \neq \epsilon$, then by induction hypothesis, for any v_1 and v_2 different from ϵ such that $w_1 = v_1 v_2$, $Inv(v_1, v_2)$ is derivable then:

$$\frac{Inv(v_1, v_2) \quad Inv(\epsilon, \epsilon)}{Inv(v_1 v_2 = w_1, \epsilon)} \; Inv(x_1 x_2 y_1, y_2)\text{:-}Inv(x_1, x_2),\; Inv(y_1, y_2)$$

The proof of the Theorem

Theorem: Given w_1 and w_2 such that $w_1 w_2 \in O_2$, $Inv(w_1, w_2)$ is derivable.

The proof is done by induction on the lexicographically ordered pairs $(|w_1 w_2|, \max(|w_1|, |w_2|))$.
There are five cases:

Case 2: $w_1 = s_1 w_1' s_2$ and $w_2 = s_3 w_2' s_4$ and for $i, j \in \{1; 2; 3; 4\}$, s.t. $i \neq j$, $\{s_i; s_j\} \in \{\{a; \overline{a}\}; \{b; \overline{b}\}\}$:

The proof of the Theorem

Theorem: Given w_1 and w_2 such that $w_1 w_2 \in O_2$, $Inv(w_1, w_2)$ is derivable.

The proof is done by induction on the lexicographically ordered pairs $(|w_1 w_2|, \max(|w_1|, |w_2|))$.
There are five cases:

Case 2: $w_1 = s_1 w_1' s_2$ and $w_2 = s_3 w_2' s_4$ and for $i, j \in \{1; 2; 3; 4\}$, s.t. $i \neq j$, $\{s_i; s_j\} \in \{\{a; \overline{a}\}; \{b; \overline{b}\}\}$:
e.g., if $i = 1$, $j = 2$, $s_1 = a$ and $s_2 = \overline{a}$ then by induction hypothesis $Inv(w_1', w_2)$ is derivable and:

$$\frac{Inv(a, \overline{a}) \quad Inv(w_1', w_2)}{Inv(a w_1' \overline{a}, w_2)} \; Inv(x_1 y_1 x_2, y_2)\text{:-}Inv(x_1, x_2),\ Inv(y_1, y_2)$$

The proof of the Theorem

Theorem: Given w_1 and w_2 such that $w_1 w_2 \in O_2$, $Inv(w_1, w_2)$ is derivable.

The proof is done by induction on the lexicographically ordered pairs $(|w_1 w_2|, \max(|w_1|, |w_2|))$.
There are five cases:

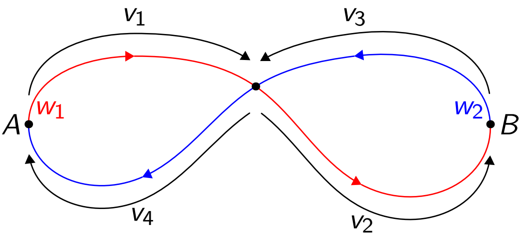
Case 3: the curves representing w_1 and w_2 have a non-trivial intersection point:

The proof of the Theorem

Theorem: Given w_1 and w_2 such that $w_1 w_2 \in O_2$, $Inv(w_1, w_2)$ is derivable.

The proof is done by induction on the lexicographically ordered pairs $(|w_1 w_2|, \max(|w_1|, |w_2|))$.
There are five cases:

Case 3: the curves representing w_1 and w_2 have a non-trivial intersection point:



$$\frac{Inv(v_1, v_4) \quad Inv(v_2, v_3)}{Inv(v_1 v_2 = w_1, v_3 v_4 = w_2)}$$

The proof of the Theorem

Theorem: Given w_1 and w_2 such that $w_1 w_2 \in O_2$, $Inv(w_1, w_2)$ is derivable.

The proof is done by induction on the lexicographically ordered pairs $(|w_1 w_2|, \max(|w_1|, |w_2|))$.
There are five cases:

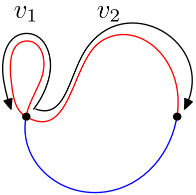
Case 4: the curve representing w_1 or w_2 starts or ends with a loop:

The proof of the Theorem

Theorem: Given w_1 and w_2 such that $w_1 w_2 \in O_2$, $Inv(w_1, w_2)$ is derivable.

The proof is done by induction on the lexicographically ordered pairs $(|w_1 w_2|, \max(|w_1|, |w_2|))$.
There are five cases:

Case 4: the curve representing w_1 or w_2 starts or ends with a loop:



$$\frac{Inv(v_1, \epsilon) \quad Inv(v_2, w_2)}{Inv(v_1 v_2 = w_1, w_2)}$$

The proof of the Theorem

Theorem: Given w_1 and w_2 such that $w_1 w_2 \in O_2$, $Inv(w_1, w_2)$ is derivable.

The proof is done by induction on the lexicographically ordered pairs $(|w_1 w_2|, \max(|w_1|, |w_2|))$.
There are five cases:

Case 5: w_1 and w_2 do not start or end with compatible letters, the curve representing then do not intersect and do not start or end with a loop.

Solving case 5: towards geometry

- w.l.o.g. we may assume that w_1 and w_2 start and end with a or b,

Solving case 5: towards geometry

- w.l.o.g. we may assume that w_1 and w_2 start and end with a or b,
- if we consider subwords w_1' and w_2' of w_1 and w_2 obtained by erasing factors of w_1 and w_2 that are in O_2, we have:

Solving case 5: towards geometry

- w.l.o.g. we may assume that w_1 and w_2 start and end with a or b,
- if we consider subwords w_1' and w_2' of w_1 and w_2 obtained by erasing factors of w_1 and w_2 that are in O_2, we have:
  - w_1' and w_2' start or end with a or b,

Solving case 5: towards geometry

- w.l.o.g. we may assume that w_1 and w_2 start and end with a or b,
- if we consider subwords w_1' and w_2' of w_1 and w_2 obtained by erasing factors of w_1 and w_2 that are in O_2, we have:
  - w_1' and w_2' start or end with a or b,
  - the curve that represents $w_1' w_2'$ is a Jordan curve,

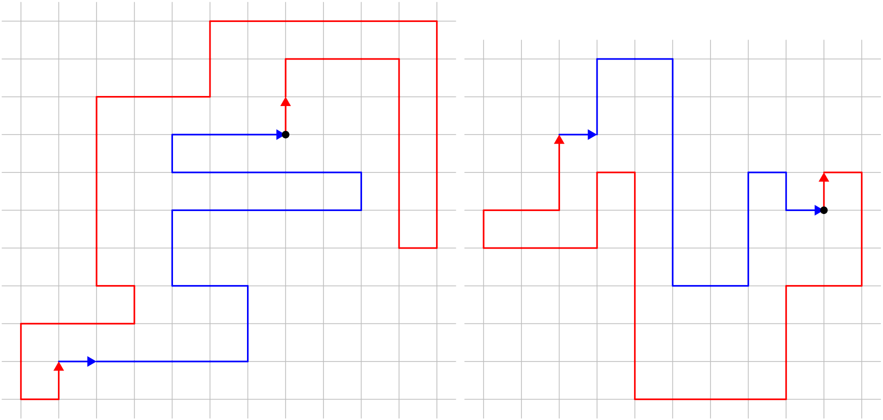
Solving case 5: towards geometry

- w.l.o.g. we may assume that w_1 and w_2 start and end with a or b,
- if we consider subwords w_1' and w_2' of w_1 and w_2 obtained by erasing factors of w_1 and w_2 that are in O_2, we have:
  - w_1' and w_2' start or end with a or b,
  - the curve that represents $w_1' w_2'$ is a Jordan curve,
  - if $w_1' = v_1' v_2' v_3'$ such that $v_1' v_3'$ and $v_2' w_2'$ are in O_2 then $w_1 = v_1 v_2 v_3$ so that $v_1 v_3$ and $v_2 w_2$ are in O_2.
  - if $w_2' = v_1' v_2' v_3'$ such that $v_1' v_3'$ and $w_1' v_2'$ are in O_2 then $w_2 = v_1 v_2 v_3$ so that $v_1 v_3$ and $w_1 v_2$ are in O_2.

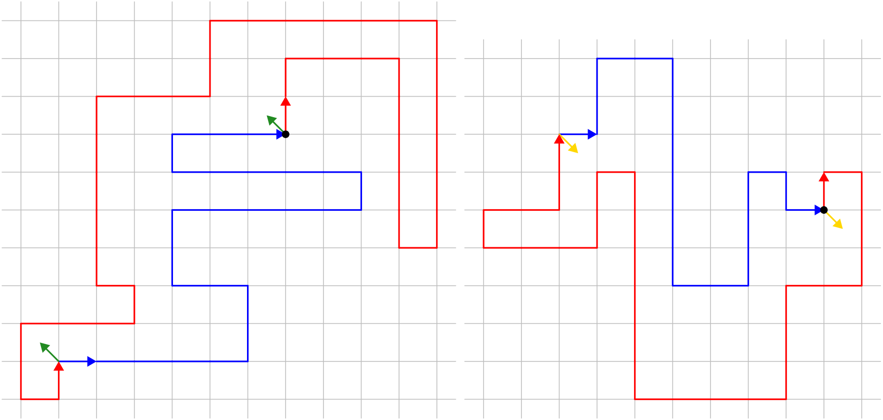
Solving case 5: towards geometry

- w.l.o.g. we may assume that w_1 and w_2 start and end with a or b,
- if we consider subwords w_1' and w_2' of w_1 and w_2 obtained by erasing factors of w_1 and w_2 that are in O_2, we have:
  - w_1' and w_2' start or end with a or b,
  - the curve that represents $w_1'w_2'$ is a Jordan curve,
  - if $w_1' = v_1'v_2'v_3'$ such that $v_1'v_3'$ and $v_2'w_2'$ are in O_2 then $w_1 = v_1v_2v_3$ so that v_1v_3 and v_2w_2 are in O_2.
  - if $w_2' = v_1'v_2'v_3'$ such that $v_1'v_3'$ and $w_1'v_2'$ are in O_2 then $w_2 = v_1v_2v_3$ so that v_1v_3 and w_1v_2 are in O_2.
  - we will prove the existence of such v_1', v_2' and v_3' for any such w_1' and w_2'.

Solving case 5: a geometrical invariant

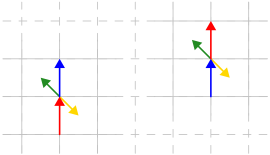
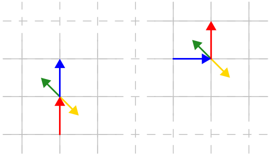
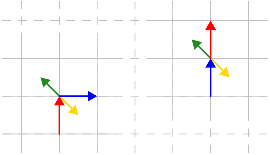
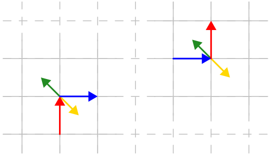


Solving case 5: a geometrical invariant



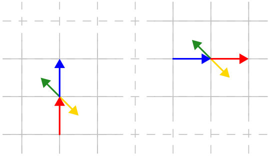
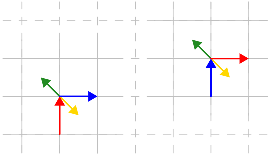
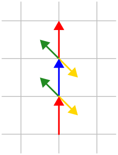
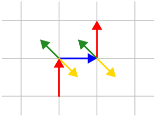
Solving case 5: a geometrical invariant

An invariant on the Jordan curve representing $w_1' w_2'$:

 $w_1' = a w_1'' a$ and $w_2' = a w_2'' a$	 $w_1' = a w_1'' a$ and $w_2' = a w_2'' b$
 $w_1' = a w_1'' a$ and $w_2' = b w_2'' a$	 $w_1' = a w_1'' a$ and $w_2' = b w_2'' b$

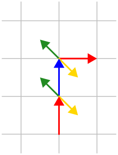
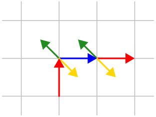
Solving case 5: a geometrical invariant

An invariant on the Jordan curve representing $w_1' w_2'$:

 $w_1' = a w_1'' b$ and $w_2' = a w_2'' b$	 $w_1' = a w_1'' b$ and $w_2' = b w_2'' a$
 $w_1' = a w_1'' a$ and $w_2' = a$	 $w_1' = a w_1'' a$ and $w_2' = b$

Solving case 5: a geometrical invariant

An invariant on the Jordan curve representing $w_1' w_2'$:

	
$w_1' = a w_1'' b$ and $w_2' = a$	$w_1' = a w_1'' b$ and $w_2' = b$

Outline

The *MIX* and O_2 languages

Multiple Context Free Grammars (MCFGs)

A grammar for O_2

A Theorem on Jordan curves

Conclusion and conjectures

On Jordan curves

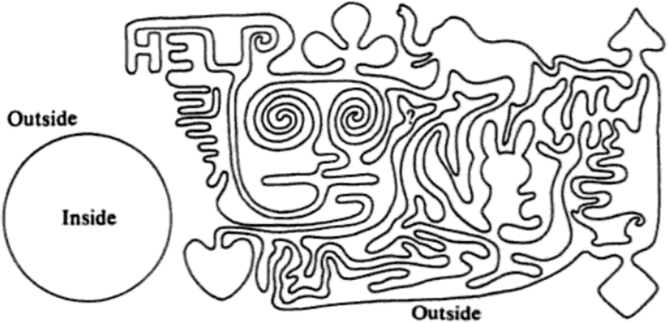


Figure 13.1 Two Jordan curves.

illustration from: A combinatorial introduction to topology by Michael Henle (Dover Publications).

On Jordan curves

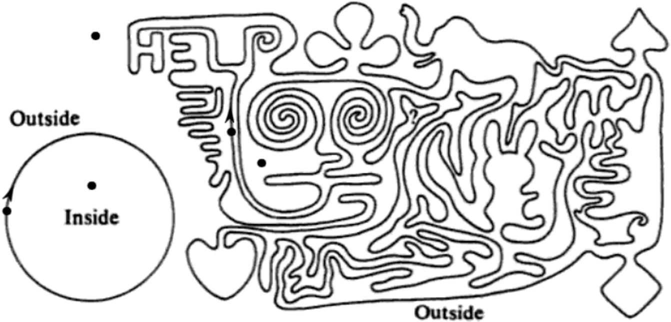


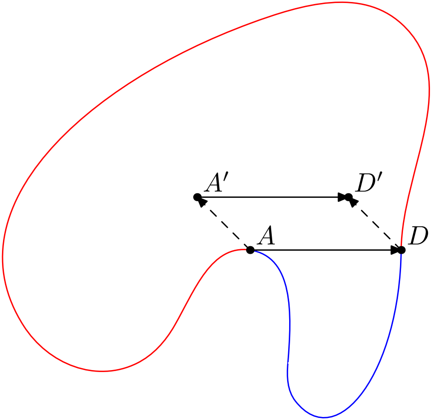
Figure 13.1 Two Jordan curves.

illustration from: A combinatorial introduction to topology by Michael Henle (Dover Publications).

Theorem: There is $k \in \{-1; 1\}$ such that the winding number of Jordan curve around a point in its interior is k, its winding number around a point in its exterior is 0.

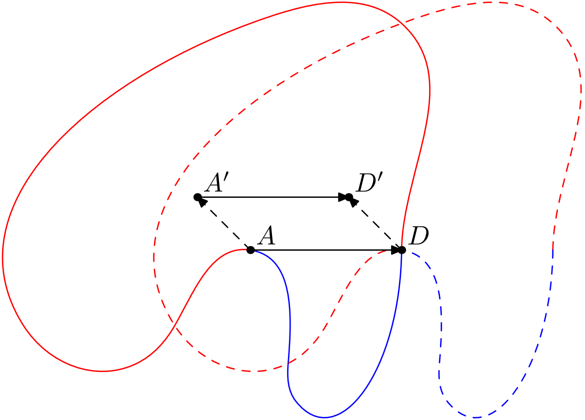
A theorem on Jordan curves

Theorem: If A and D are two points on a Jordan curve J such that there are two points A' and D' inside J such that $\overrightarrow{AD} = \overrightarrow{A'D'}$, then there are two points B and C pairwise distinct from A and D such that A, B, C, and D appear in that order on one of the arcs going from A to D and $\overrightarrow{AD} = \overrightarrow{BC}$.



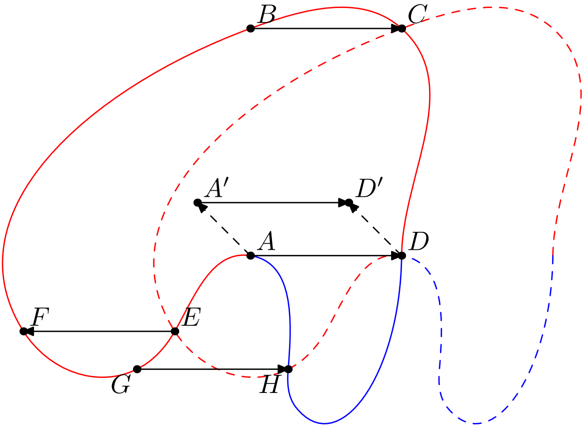
A theorem on Jordan curves

Theorem: If A and D are two points on a Jordan curve J such that there are two points A' and D' inside J such that $\overrightarrow{AD} = \overrightarrow{A'D'}$, then there are two points B and C pairwise distinct from A and D such that A, B, C, and D appear in that order on one of the arcs going from A to D and $\overrightarrow{AD} = \overrightarrow{BC}$.



A theorem on Jordan curves

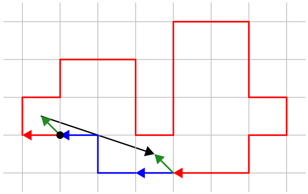
Theorem: If A and D are two points on a Jordan curve J such that there are two points A' and D' inside J such that $\overrightarrow{AD} = \overrightarrow{A'D'}$, then there are two points B and C pairwise distinct from A and D such that A, B, C, and D appear in that order on one of the arcs going from A to D and $\overrightarrow{AD} = \overrightarrow{BC}$.



A theorem on Jordan curves

Theorem: If A and D are two points on a Jordan curve J such that there are two points A' and D' inside J such that $\overrightarrow{AD} = \overrightarrow{A'D'}$, then there are two points B and C pairwise distinct from A and D such that A, B, C, and D appear in that order on one of the arcs going from A to D and $\overrightarrow{AD} = \overrightarrow{BC}$.

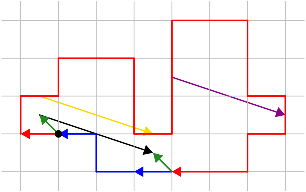
Applying this Theorem solves case 5.



A theorem on Jordan curves

Theorem: If A and D are two points on a Jordan curve J such that there are two points A' and D' inside J such that $\overrightarrow{AD} = \overrightarrow{A'D'}$, then there are two points B and C pairwise distinct from A and D such that A, B, C, and D appear in that order on one of the arcs going from A to D and $\overrightarrow{AD} = \overrightarrow{BC}$.

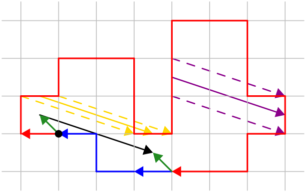
Applying this Theorem solves case 5.



A theorem on Jordan curves

Theorem: If A and D are two points on a Jordan curve J such that there are two points A' and D' inside J such that $\overrightarrow{AD} = \overrightarrow{A'D'}$, then there are two points B and C pairwise distinct from A and D such that A, B, C, and D appear in that order on one of the arcs going from A to D and $\overrightarrow{AD} = \overrightarrow{BC}$.

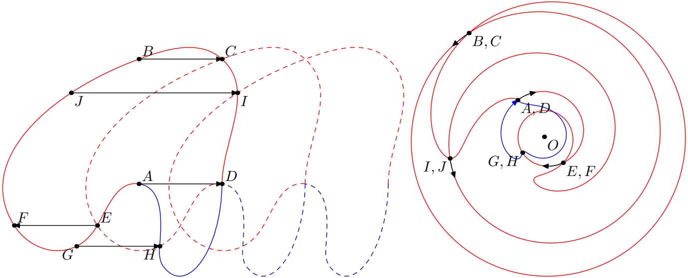
Applying this Theorem solves case 5.



Simple curves, translations, intersections and the complex exponential

Let's suppose that $D - A = 1$

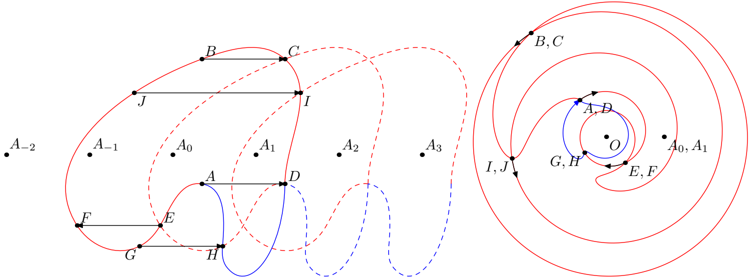
let $\varphi : \begin{cases} \mathbb{C} & \rightarrow & \mathbb{C} - \{0\} \\ z & \rightarrow & e^{2i\pi z} \end{cases}$.



φ transforms arcs performing translation of k into arc that have k as winding number around 0.

Simple curves, translations, intersections and the complex exponential

Let's suppose that $D - A = 1$ and that $A_0 = A' = 0$,
$A_1 = D' = 1, \ldots, A_k = k$
let $\varphi : \begin{cases} \mathbb{C} & \to & \mathbb{C} - \{0\} \\ z & \to & e^{2i\pi z} \end{cases}$.

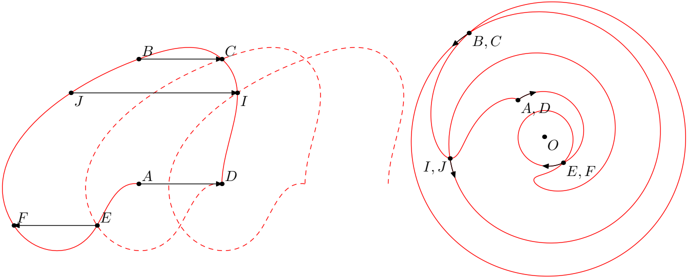


φ sums up the winding number of a Jordan curve around the A_i as the winding number around $\varphi(A_0) = \varphi(0) = 1$.

Simple curves, translations, intersections and the complex exponential

Let's suppose that $D - A = 1$ and that $A_0 = A' = 0$, $A_1 = D' = 1, \ldots, A_k = k$

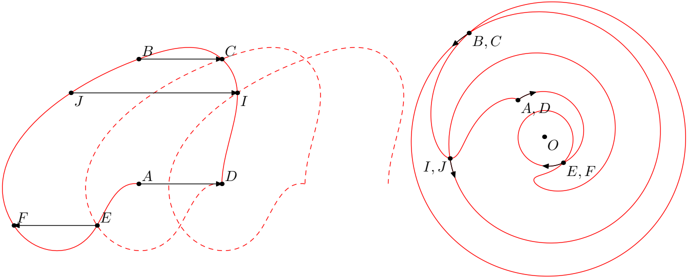
let $\varphi : \begin{cases} \mathbb{C} & \to & \mathbb{C} - \{0\} \\ z & \to & e^{2i\pi z} \end{cases}$.



Simple curves, translations, intersections and the complex exponential

Let's suppose that $D - A = 1$ and that $A_0 = A' = 0$, $A_1 = D' = 1, \ldots, A_k = k$

let $\varphi : \begin{cases} \mathbb{C} & \to & \mathbb{C} - \{0\} \\ z & \to & e^{2i\pi z} \end{cases}$.

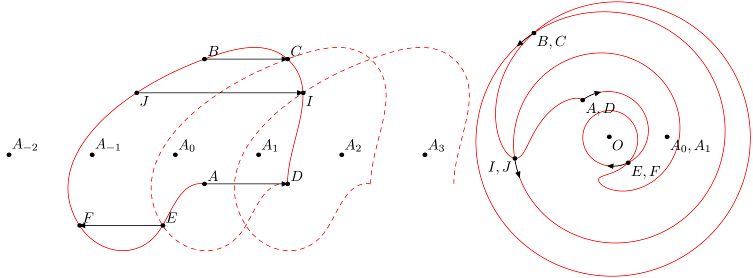


Lemma: a simple path J from A to D (*resp.* D to A) does not contain B and C as required in the Theorem iff $\varphi(J)$ is a simple curve of $\mathbb{C} - \{0\}$ that belong to the homotopy class 1 (*resp.* -1).

Simple curves, translations, intersections and the complex exponential

Let's suppose that $D - A = 1$ and that $A_0 = A' = 0$, $A_1 = D' = 1, \dots, A_k = k$

let $\varphi : \left\{ \begin{array}{ccc} \mathbb{C} & \to & \mathbb{C} - \{0\} \\ z & \to & e^{2i\pi z} \end{array} \right.$.



Lemma: a simple path J from A to D (*resp.* D to A) does not contain B and C as required in the Theorem iff $\varphi(J)$ is a simple curve of $\mathbb{C} - \{1\}$ that belong to the homotopy class 0 or 1 (*resp.* or -1).

Proving the Theorem

Let $wn(J, z)$ be the winding number of a closed curve around z.

Proving the Theorem

Let $wn(J, z)$ be the winding number of a closed curve around z.

Corollary: if J is a simple closed curve of $\mathbb{C}$ composed with two curves J_1 and J_2 respectively going from A to D and D to A which do not contain points B and C as required in the Theorem then $wn(\varphi(J), 1) = wn(\varphi(J_1), 1) + wn(\varphi(J_2), 1)$ is in $\{-1; 0; 1\}$.

Proving the Theorem

Let $wn(J, z)$ be the winding number of a closed curve around z.

Corollary: if J is a simple closed curve of $\mathbb{C}$ composed with two curves J_1 and J_2 respectively going from A to D and D to A which do not contain points B and C as required in the Theorem then $wn(\varphi(J), 1) = wn(\varphi(J_1), 1) + wn(\varphi(J_2), 1)$ is in $\{-1; 0; 1\}$.

Lemma: if J is a simple closed curve of $\mathbb{C}$ composed with two curves J_1 and J_2 respectively going from A to D and D to A such that 0 and 1 are in the interior of J, then either $wn(\varphi(J), 1) < -1$ or $wn(\varphi(J), 1) > 1$.

Proving the Theorem

Let $wn(J, z)$ be the winding number of a closed curve around z.

Corollary: if J is a simple closed curve of $\mathbb{C}$ composed with two curves J_1 and J_2 respectively going from A to D and D to A which do not contain points B and C as required in the Theorem then $wn(\varphi(J), 1) = wn(\varphi(J_1), 1) + wn(\varphi(J_2), 1)$ is in $\{-1; 0; 1\}$.

Lemma: if J is a simple closed curve of $\mathbb{C}$ composed with two curves J_1 and J_2 respectively going from A to D and D to A such that 0 and 1 are in the interior of J, then either $wn(\varphi(J), 1) < -1$ or $wn(\varphi(J), 1) > 1$.

The Theorem follows by contradiction.

Outline

Conclusion

- we have showed that O_2 is a 2-MCFL exhibiting the first non-virtually free group language that is proved to belong to an interesting class of language,
- this implies that contrary to the usual conjecture we have showed that MIX is a 2-MCFLs.

Conjectures

Well-nestedness:

$$\frac{\text{Well-nested} \quad Inv(y_1 x_1 x_2, y_2) \text{:-} Inv(x_1, x_2), Inv(y_1, y_2)}{\text{Not well-nested} \quad Inv(y_1 x_1 y_2, x_2) \text{:-} Inv(x_1, x_2), Inv(y_1, y_2)}$$

MCFG_{wn} are MCFGs with well-nested rules.

- MCFL_{wn} coincide with non-duplicating IO/OI,
- MCFL is incomparable with IO or OI.

Conjectures

Well-nestedness:

Well-nested
$Inv(y_1x_1x_2, y_2) \text{:-} Inv(x_1, x_2), Inv(y_1, y_2)$
Not well-nested
$Inv(y_1x_1y_2, x_2) \text{:-} Inv(x_1, x_2), Inv(y_1, y_2)$

MCFG_{wn} are MCFGs with well-nested rules.

- MCFL_{wn} coincide with non-duplicating IO/OI,
- MCFL is incomparable with IO or OI.

Thus the following conjectures:

- mildly context sensitive languages may well be, as advocated by Kanazawa, MCFL_{wn}

Conjectures

Well-nestedness:

$$\frac{\text{Well-nested} \quad Inv(y_1 x_1 x_2, y_2) \text{:-} Inv(x_1, x_2), Inv(y_1, y_2)}{\text{Not well-nested} \quad Inv(y_1 x_1 y_2, x_2) \text{:-} Inv(x_1, x_2), Inv(y_1, y_2)}$$

MCFG_{wn} are MCFGs with well-nested rules.

- MCFL_{wn} coincide with non-duplicating IO/OI,
- MCFL is incomparable with IO or OI.

Thus the following conjectures:

- mildly context sensitive languages may well be, as advocated by Kanazawa, MCFL_{wn}
- O_2 and MIX should not be a MCFL_{wn}

Conjectures

Well-nestedness:

$$\frac{\text{Well-nested} \quad Inv(y_1x_1x_2, y_2) \text{:-} Inv(x_1, x_2), Inv(y_1, y_2)}{\text{Not well-nested} \quad Inv(y_1x_1y_2, x_2) \text{:-} Inv(x_1, x_2), Inv(y_1, y_2)}$$

MCFG_{wn} are MCFGs with well-nested rules.

- MCFL_{wn} coincide with non-duplicating IO/OI,
- MCFL is incomparable with IO or OI.

Thus the following conjectures:

- mildly context sensitive languages may well be, as advocated by Kanazawa, MCFL_{wn}
- O_2 and MIX should not be a MCFL_{wn}
- semilinear rational cones included in OI should be included in MCFL_{wn}

Conjectures

Well-nestedness:

$$\frac{\text{Well-nested} \quad Inv(y_1 x_1 x_2, y_2) \text{:-} Inv(x_1, x_2), Inv(y_1, y_2)}{\text{Not well-nested} \quad Inv(y_1 x_1 y_2, x_2) \text{:-} Inv(x_1, x_2), Inv(y_1, y_2)}$$

MCFG$_{wn}$ are MCFGs with well-nested rules.

- MCFL$_{wn}$ coincide with non-duplicating IO/OI,
- MCFL is incomparable with IO or OI.

Thus the following conjectures:

- mildly context sensitive languages may well be, as advocated by Kanazawa, MCFL$_{wn}$
- O_2 and MIX should not be a MCFL$_{wn}$
- semilinear rational cones included in OI should be included in MCFL$_{wn}$
- O_2 and MIX should not be in OI.

Conjectures

Well-nestedness:

$$\frac{\begin{array}{c}\text{Well-nested}\\ Inv(y_1x_1x_2, y_2)\text{:-}Inv(x_1, x_2), Inv(y_1, y_2)\end{array}}{\begin{array}{c}\text{Not well-nested}\\ Inv(y_1x_1y_2, x_2)\text{:-}Inv(x_1, x_2), Inv(y_1, y_2)\end{array}}$$

MCFG_{wn} are MCFGs with well-nested rules.

- MCFL_{wn} coincide with non-duplicating IO/OI,
- MCFL is incomparable with IO or OI.

Thus the following conjectures:

- mildly context sensitive languages may well be, as advocated by Kanazawa, MCFL_{wn}
- O_2 and MIX should not be a MCFL_{wn}
- semilinear rational cones included in OI should be included in MCFL_{wn}
- O_2 and MIX should not be in OI.

Open question:

- Is O_3 an MCFL?