# Decidability Questions for Insertion Systems

Andreas Malcher

Institut für Informatik, Universität Giessen,
Arndtstr. 2, 35392 Giessen, Germany
email: malcher@informatik.uni-giessen.de

NCMA 2017, Prague, Czech Republic

# Insertion Systems

# Insertion Systems

- Introduced in 1981 by Galiukschov as semicontextual grammars.

# Insertion Systems

- Introduced in 1981 by Galiukschov as semicontextual grammars.
- Idea: Start with a finite set of axioms and iteratively apply insertion rules which insert strings subject to certain contexts.

# Insertion Systems

- Introduced in 1981 by Galiukschov as semicontextual grammars.
- Idea: Start with a finite set of axioms and iteratively apply insertion rules which insert strings subject to certain contexts.
- Insertion Systems are an intermediate model between Chomsky grammars and contextual grammars.

# Insertion Systems

- Introduced in 1981 by Galiukschov as semicontextual grammars.
- Idea: Start with a finite set of axioms and iteratively apply insertion rules which insert strings subject to certain contexts.
- Insertion Systems are an intermediate model between Chomsky grammars and contextual grammars.
- Strings are only inserted as in contextual grammars and not rewritten as in Chomsky grammars.

# Insertion Systems

- Introduced in 1981 by Galiukschov as semicontextual grammars.
- Idea: Start with a finite set of axioms and iteratively apply insertion rules which insert strings subject to certain contexts.
- Insertion Systems are an intermediate model between Chomsky grammars and contextual grammars.
- Strings are only inserted as in contextual grammars and not rewritten as in Chomsky grammars.
- The insertion is controlled by contexts as is done in context-sensitive (Chomsky) grammars.

# Insertion Systems

- Introduced in 1981 by Galiukschov as semicontextual grammars.
- Idea: Start with a finite set of axioms and iteratively apply insertion rules which insert strings subject to certain contexts.
- Insertion Systems are an intermediate model between Chomsky grammars and contextual grammars.
- Strings are only inserted as in contextual grammars and not rewritten as in Chomsky grammars.
- The insertion is controlled by contexts as is done in context-sensitive (Chomsky) grammars.
- Insertion systems are a special case of insertion-deletion systems which have extensively been investigated.

# Insertion Systems

An insertion system $S$ is a triple $S = \langle T, A, I \rangle$, where

- ➜ $T$ is an alphabet,
- ➜ $A \subseteq T^*$ is a finite set of axioms, and
- ➜ $I \subseteq T^* \times T^+ \times T^*$ is a finite set of insertion rules.

# Insertion Systems

An insertion system $S$ is a triple $S = \langle T, A, I \rangle$, where

- ➜ $T$ is an alphabet,
- ➜ $A \subseteq T^*$ is a finite set of axioms, and
- ➜ $I \subseteq T^* \times T^+ \times T^*$ is a finite set of insertion rules.

For $x, y \in T^*$ we write $x \Rightarrow y$, if $x = x_1 uv x_2$ and $y = x_1 u\alpha v x_2$ for $(u, \alpha, v) \in I$ and $x_1, x_2 \in T^*$.

# Insertion Systems

An insertion system $S$ is a triple $S = \langle T, A, I \rangle$, where

- ➜ $T$ is an alphabet,
- ➜ $A \subseteq T^*$ is a finite set of axioms, and
- ➜ $I \subseteq T^* \times T^+ \times T^*$ is a finite set of insertion rules.

For $x, y \in T^*$ we write $x \Rightarrow y$, if $x = x_1 uv x_2$ and $y = x_1 u\alpha v x_2$ for $(u, \alpha, v) \in I$ and $x_1, x_2 \in T^*$.

The generated language is $L(S) = \{\, w \mid x \Rightarrow^* w \text{ for some } x \in A \,\}$.

# Insertion Systems

An insertion system $S$ is a triple $S = \langle T, A, I \rangle$, where

- $T$ is an alphabet,
- $A \subseteq T^*$ is a finite set of axioms, and
- $I \subseteq T^* \times T^+ \times T^*$ is a finite set of insertion rules.

For $x, y \in T^*$ we write $x \Rightarrow y$, if $x = x_1 uv x_2$ and $y = x_1 u\alpha v x_2$ for $(u, \alpha, v) \in I$ and $x_1, x_2 \in T^*$.

The generated language is $L(S) = \{\, w \mid x \Rightarrow^* w \text{ for some } x \in A \,\}$.

The size of an insertion system is defined by the vector $(n, l, r)$, where the integers $n = \max\{\, |\alpha| \mid (u, \alpha, v) \in I \,\}$, $l = \max\{\, |u| \mid (u, \alpha, v) \in I \,\}$, and $r = \max\{\, |v| \mid (u, \alpha, v) \in I \,\}$.

# Insertion Systems

An insertion system $S$ is a triple $S = \langle T, A, I \rangle$, where

- $T$ is an alphabet,
- $A \subseteq T^*$ is a finite set of axioms, and
- $I \subseteq T^* \times T^+ \times T^*$ is a finite set of insertion rules.

For $x, y \in T^*$ we write $x \Rightarrow y$, if $x = x_1 u v x_2$ and $y = x_1 u \alpha v x_2$ for $(u, \alpha, v) \in I$ and $x_1, x_2 \in T^*$.

The generated language is $L(S) = \{ w \mid x \Rightarrow^* w \text{ for some } x \in A \}$.

The size of an insertion system is defined by the vector $(n, l, r)$, where the integers $n = \max\{ |\alpha| \mid (u, \alpha, v) \in I \}$, $l = \max\{ |u| \mid (u, \alpha, v) \in I \}$, and $r = \max\{ |v| \mid (u, \alpha, v) \in I \}$.

The family of all systems of size $(n, l, r)$ is denoted by $\mathrm{INS}_n^{l,r}$.

# Insertion Systems

## Example 1

Consider $S_1 = \langle \{a, b\}, \{ab\}, \{(a, ab, b)\} \rangle \in \mathsf{INS}_2^{1,1}$.

# Insertion Systems

## Example 1

Consider $S_1 = \langle \{a, b\}, \{ab\}, \{(a, ab, b)\} \rangle \in \mathsf{INS}_2^{1,1}$.

We obtain that $L(S_1) = \{\, a^m b^m \mid m \geq 1 \,\}$.

# Insertion Systems

## Example 1

Consider $S_1 = \langle \{a, b\}, \{ab\}, \{(a, ab, b)\} \rangle \in \mathsf{INS}_2^{1,1}$.

We obtain that $L(S_1) = \{\, a^m b^m \mid m \geq 1 \,\}$.
For example, $ab$

# Insertion Systems

**Example 1**

Consider $S_1 = \langle \{a, b\}, \{ab\}, \{(a, ab, b)\} \rangle \in \mathsf{INS}_2^{1,1}$.

We obtain that $L(S_1) = \{ a^m b^m \mid m \geq 1 \}$.
For example, $ab \Rightarrow aabb$

# Insertion Systems

## Example 1

Consider $S_1 = \langle \{a, b\}, \{ab\}, \{(a, ab, b)\} \rangle \in \mathsf{INS}_2^{1,1}$.

We obtain that $L(S_1) = \{ a^m b^m \mid m \geq 1 \}$.
For example, $ab \Rightarrow aabb \Rightarrow aaabbb$

# Insertion Systems

## Example 1

Consider $S_1 = \langle \{a, b\}, \{ab\}, \{(a, ab, b)\} \rangle \in \mathsf{INS}_2^{1,1}$.

We obtain that $L(S_1) = \{ a^m b^m \mid m \geq 1 \}$.
For example, $ab \Rightarrow aabb \Rightarrow aaabbb \Rightarrow aaaabbbb$.

# Insertion Systems

**Example 1**

Consider $S_1 = \langle \{a, b\}, \{ab\}, \{(a, ab, b)\} \rangle \in \mathsf{INS}_2^{1,1}$.

We obtain that $L(S_1) = \{ a^m b^m \mid m \geq 1 \}$.
For example, $ab \Rightarrow aabb \Rightarrow aaabbb \Rightarrow aaaabbbb$.

**Example 2**

Consider $S_2 = \langle \{a, b\}, \{ab, ba\}, \{(\lambda, ab, \lambda), (\lambda, ba, \lambda)\} \rangle \in \mathsf{INS}_2^{0,0}$.

# Insertion Systems

**Example 1**

Consider $S_1 = \langle \{a, b\}, \{ab\}, \{(a, ab, b)\} \rangle \in \mathsf{INS}_2^{1,1}$.

We obtain that $L(S_1) = \{ a^m b^m \mid m \geq 1 \}$.
For example, $ab \Rightarrow aabb \Rightarrow aaabbb \Rightarrow aaaabbbb$.

**Example 2**

Consider $S_2 = \langle \{a, b\}, \{ab, ba\}, \{(\lambda, ab, \lambda), (\lambda, ba, \lambda)\} \rangle \in \mathsf{INS}_2^{0,0}$.

We obtain that $L(S_2) = \{ w \in \{a, b\}^+ \mid |w|_a = |w|_b \}$.

# Insertion Systems

**Example 1**

Consider $S_1 = \langle \{a, b\}, \{ab\}, \{(a, ab, b)\} \rangle \in \mathsf{INS}_2^{1,1}$.

We obtain that $L(S_1) = \{ a^m b^m \mid m \geq 1 \}$.
For example, $ab \Rightarrow aabb \Rightarrow aaabbb \Rightarrow aaaabbbb$.

**Example 2**

Consider $S_2 = \langle \{a, b\}, \{ab, ba\}, \{(\lambda, ab, \lambda), (\lambda, ba, \lambda)\} \rangle \in \mathsf{INS}_2^{0,0}$.

We obtain that $L(S_2) = \{ w \in \{a, b\}^+ \mid |w|_a = |w|_b \}$.
For example, $ab$

# Insertion Systems

**Example 1**

Consider $S_1 = \langle \{a, b\}, \{ab\}, \{(a, ab, b)\} \rangle \in \mathsf{INS}_2^{1,1}$.

We obtain that $L(S_1) = \{ \, a^m b^m \mid m \geq 1 \, \}$.
For example, $ab \Rightarrow aabb \Rightarrow aaabbb \Rightarrow aaaabbbb$.

**Example 2**

Consider $S_2 = \langle \{a, b\}, \{ab, ba\}, \{(\lambda, ab, \lambda), (\lambda, ba, \lambda)\} \rangle \in \mathsf{INS}_2^{0,0}$.

We obtain that $L(S_2) = \{ \, w \in \{a, b\}^+ \mid |w|_a = |w|_b \, \}$.
For example, $ab \Rightarrow aabb$

# Insertion Systems

## Example 1

Consider $S_1 = \langle \{a, b\}, \{ab\}, \{(a, ab, b)\} \rangle \in \mathsf{INS}_2^{1,1}$.

We obtain that $L(S_1) = \{ a^m b^m \mid m \geq 1 \}$.
For example, $ab \Rightarrow aabb \Rightarrow aaabbb \Rightarrow aaaabbbb$.

## Example 2

Consider $S_2 = \langle \{a, b\}, \{ab, ba\}, \{(\lambda, ab, \lambda), (\lambda, ba, \lambda)\} \rangle \in \mathsf{INS}_2^{0,0}$.

We obtain that $L(S_2) = \{ w \in \{a, b\}^+ \mid |w|_a = |w|_b \}$.
For example, $ab \Rightarrow aabb \Rightarrow aababb$

# Insertion Systems

## Example 1

Consider $S_1 = \langle \{a, b\}, \{ab\}, \{(a, ab, b)\} \rangle \in \mathsf{INS}_2^{1,1}$.

We obtain that $L(S_1) = \{ a^m b^m \mid m \geq 1 \}$.
For example, $ab \Rightarrow aabb \Rightarrow aaabbb \Rightarrow aaaabbbb$.

## Example 2

Consider $S_2 = \langle \{a, b\}, \{ab, ba\}, \{(\lambda, ab, \lambda), (\lambda, ba, \lambda)\} \rangle \in \mathsf{INS}_2^{0,0}$.

We obtain that $L(S_2) = \{ w \in \{a, b\}^+ \mid |w|_a = |w|_b \}$.
For example, $ab \Rightarrow aabb \Rightarrow aababb \Rightarrow aabababb$.

# Insertion Systems

## Example 3

$S_3 = \langle \{a, b, c, d\}, \{ab\}, \{(a, c, b), (c, d, b), (c, a, d), (a, b, d)\}\rangle$ belongs to $\mathsf{INS}_1^{1,1}$.

# Insertion Systems

## Example 3

$S_3 = \langle\{a, b, c, d\}, \{ab\}, \{(a, c, b), (c, d, b), (c, a, d), (a, b, d)\}\rangle$ belongs to $\mathsf{INS}_1^{1,1}$.

We obtain that $L(S_3) = \{\, (ac)^m x(db)^m \mid m \geq 0, x \in \{ab, acb\} \,\} \cup \{\, (ac)^m x(db)^m \mid m \geq 1, x \in \{\lambda, a\} \,\}$ is not regular.

# Insertion Systems

## Example 3

$S_3 = \langle\{a, b, c, d\}, \{ab\}, \{(a, c, b), (c, d, b), (c, a, d), (a, b, d)\}\rangle$ belongs to $\mathsf{INS}_1^{1,1}$.

We obtain that $L(S_3) = \{ (ac)^m x(db)^m \mid m \geq 0, x \in \{ab, acb\} \} \cup \{ (ac)^m x(db)^m \mid m \geq 1, x \in \{\lambda, a\} \}$ is not regular.

For example, $ab$

# Insertion Systems

## Example 3

$S_3 = \langle \{a, b, c, d\}, \{ab\}, \{(a, c, b), (c, d, b), (c, a, d), (a, b, d)\} \rangle$ belongs to $\mathsf{INS}_1^{1,1}$.

We obtain that $L(S_3) = \{ \, (ac)^m x (db)^m \mid m \geq 0, x \in \{ab, acb\} \, \} \cup \{ \, (ac)^m x (db)^m \mid m \geq 1, x \in \{\lambda, a\} \, \}$ is not regular.

For example, $ab \Rightarrow acb$

# Insertion Systems

## Example 3

$S_3 = \langle \{a,b,c,d\}, \{ab\}, \{(a,c,b),(c,d,b),(c,a,d),(a,b,d)\} \rangle$ belongs to $\mathsf{INS}_1^{1,1}$.

We obtain that $L(S_3) = \{ (ac)^m x (db)^m \mid m \geq 0, x \in \{ab, acb\} \} \cup \{ (ac)^m x (db)^m \mid m \geq 1, x \in \{\lambda, a\} \}$ is not regular.

For example, $ab \Rightarrow acb \Rightarrow acdb$

# Insertion Systems

## Example 3

$S_3 = \langle \{a, b, c, d\}, \{ab\}, \{(a, c, b), (c, d, b), (c, a, d), (a, b, d)\} \rangle$ belongs to $\mathsf{INS}_1^{1,1}$.

We obtain that $L(S_3) = \{ (ac)^m x(db)^m \mid m \geq 0, x \in \{ab, acb\} \} \cup \{ (ac)^m x(db)^m \mid m \geq 1, x \in \{\lambda, a\} \}$ is not regular.

For example, $ab \Rightarrow acb \Rightarrow acdb \Rightarrow acadb$

# Insertion Systems

## Example 3

$S_3 = \langle \{a,b,c,d\}, \{ab\}, \{(a,c,b),(c,d,b),(c,a,d),(a,b,d)\} \rangle$ belongs to $\mathsf{INS}_1^{1,1}$.

We obtain that $L(S_3) = \{\, (ac)^m x (db)^m \mid m \geq 0, x \in \{ab, acb\} \,\} \cup \{\, (ac)^m x (db)^m \mid m \geq 1, x \in \{\lambda, a\} \,\}$ is not regular.

For example, $ab \Rightarrow acb \Rightarrow acdb \Rightarrow acadb \Rightarrow acabdb$.

# Insertion Systems

## Theorem

# Insertion Systems

## Theorem

(1) For every $\mathrm{INS}_*^{m,n}$ an equivalent context-sensitive grammar can be constructed for all $m, n \geq 0$.

# Insertion Systems

## Theorem

(1) For every $\mathrm{INS}_{*}^{m,n}$ an equivalent context-sensitive grammar can be constructed for all $m, n \geq 0$.

(2) $\mathscr{L}(\mathrm{INS}_{*}^{0,0}) \subset \mathscr{L}(\mathrm{INS}_{*}^{1,1}) \subset \mathscr{L}(\mathrm{INS}_{*}^{2,2}) \ldots \subset \mathscr{L}(\mathrm{INS}_{*}^{*})$.

# Insertion Systems

## Theorem

(1) For every $\mathrm{INS}_*^{m,n}$ an equivalent context-sensitive grammar can be constructed for all $m, n \geq 0$.

(2) $\mathscr{L}(\mathrm{INS}_*^{0,0}) \subset \mathscr{L}(\mathrm{INS}_*^{1,1}) \subset \mathscr{L}(\mathrm{INS}_*^{2,2}) \ldots \subset \mathscr{L}(\mathrm{INS}_*^{*})$.

(3) $\mathrm{INS}_*^{m,n}$ is incomparable with $\mathrm{REG}$ for all $m, n \geq 0$.

# Insertion Systems

## Theorem

(1) For every $\mathrm{INS}_*^{m,n}$ an equivalent context-sensitive grammar can be constructed for all $m, n \geq 0$.

(2) $\mathscr{L}(\mathrm{INS}_*^{0,0}) \subset \mathscr{L}(\mathrm{INS}_*^{1,1}) \subset \mathscr{L}(\mathrm{INS}_*^{2,2}) \ldots \subset \mathscr{L}(\mathrm{INS}_*^{*})$.

(3) $\mathrm{INS}_*^{m,n}$ is incomparable with $\mathrm{REG}$ for all $m, n \geq 0$.

(4) For every $\mathrm{INS}_*^{1,1}$ an equivalent context-free grammar can be constructed.

# Insertion Systems

**Theorem**

(1) For every $\mathrm{INS}_*^{m,n}$ an equivalent context-sensitive grammar can be constructed for all $m, n \geq 0$.

(2) $\mathscr{L}(\mathrm{INS}_*^{0,0}) \subset \mathscr{L}(\mathrm{INS}_*^{1,1}) \subset \mathscr{L}(\mathrm{INS}_*^{2,2}) \ldots \subset \mathscr{L}(\mathrm{INS}_*^{*})$.

(3) $\mathrm{INS}_*^{m,n}$ is incomparable with $\mathrm{REG}$ for all $m, n \geq 0$.

(4) For every $\mathrm{INS}_*^{1,1}$ an equivalent context-free grammar can be constructed.

(5) $\mathscr{L}(\mathrm{INS}_2^{2,2})$ contains a non-semilinear language.

# Insertion Systems

## Theorem

(1) For every $\mathrm{INS}_*^{m,n}$ an equivalent context-sensitive grammar can be constructed for all $m, n \geq 0$.

(2) $\mathscr{L}(\mathrm{INS}_*^{0,0}) \subset \mathscr{L}(\mathrm{INS}_*^{1,1}) \subset \mathscr{L}(\mathrm{INS}_*^{2,2}) \ldots \subset \mathscr{L}(\mathrm{INS}_*^{*})$.

(3) $\mathrm{INS}_*^{m,n}$ is incomparable with $\mathrm{REG}$ for all $m, n \geq 0$.

(4) For every $\mathrm{INS}_*^{1,1}$ an equivalent context-free grammar can be constructed.

(5) $\mathscr{L}(\mathrm{INS}_2^{2,2})$ contains a non-semilinear language.

(6) $\mathscr{L}(\mathrm{INS}_*^{m,n})$ is incomparable with $\mathrm{CFL}$ for all $m, n \geq 2$.

# Insertion Systems

## Theorem

(1) For every $\mathrm{INS}_*^{m,n}$ an equivalent context-sensitive grammar can be constructed for all $m, n \geq 0$.

(2) $\mathscr{L}(\mathrm{INS}_*^{0,0}) \subset \mathscr{L}(\mathrm{INS}_*^{1,1}) \subset \mathscr{L}(\mathrm{INS}_*^{2,2}) \ldots \subset \mathscr{L}(\mathrm{INS}_*^{*})$.

(3) $\mathrm{INS}_*^{m,n}$ is incomparable with $\mathrm{REG}$ for all $m, n \geq 0$.

(4) For every $\mathrm{INS}_*^{1,1}$ an equivalent context-free grammar can be constructed.

(5) $\mathscr{L}(\mathrm{INS}_2^{2,2})$ contains a non-semilinear language.

(6) $\mathscr{L}(\mathrm{INS}_*^{m,n})$ is incomparable with $\mathrm{CFL}$ for all $m, n \geq 2$.

(7) $\mathscr{L}(\mathrm{INS}_*^{m,n})$ is an anti-AFL for all $m, n \geq 0$.

# Decidability Questions for Insertion Systems

# Decidability Questions for Insertion Systems

➜ Almost no decidability results are known.

# Decidability Questions for Insertion Systems

- Almost no decidability results are known.

**Theorem**

Let $S$ be an $\mathsf{INS}_n^{l,r}$ with $n \geq 1$ and $l, r \geq 0$. Then, it is decidable in deterministic polynomial time whether or not $L(S)$ is empty.

# Decidability Questions for Insertion Systems

- Almost no decidability results are known.

**Theorem**

Let $S$ be an $\mathsf{INS}_n^{l,r}$ with $n \geq 1$ and $l, r \geq 0$. Then, it is decidable in deterministic polynomial time whether or not $L(S)$ is empty.

**Theorem**

The fixed membership problem for $\mathscr{L}(\mathsf{INS}_n^{l,r})$ having $n \geq 1$ and $l, r \geq 0$ is solvable in deterministic polynomial time.

# Decidability Questions for Insertion Systems

➜ Almost no decidability results are known.

**Theorem**

Let $S$ be an $\mathsf{INS}_n^{l,r}$ with $n \geq 1$ and $l, r \geq 0$. Then, it is decidable in deterministic polynomial time whether or not $L(S)$ is empty.

**Theorem**

The fixed membership problem for $\mathscr{L}(\mathsf{INS}_n^{l,r})$ having $n \geq 1$ and $l, r \geq 0$ is solvable in deterministic polynomial time.

**Theorem**

Let $S$ be an $\mathsf{INS}_n^{l,r}$ with $n \geq 1$ and $l, r \geq 0$. Then, it is decidable whether or not $L(S)$ is equal to $T^*$.

# Decidability Questions for Insertion Systems

➜ No undecidability results are known.

# Decidability Questions for Insertion Systems

- No undecidability results are known.
- Consider the related concept of pure context-sensitive grammars (Maurer, Salomaa, Wood [1980]).

# Decidability Questions for Insertion Systems

- ➜ No undecidability results are known.

- ➜ Consider the related concept of pure context-sensitive grammars (Maurer, Salomaa, Wood [1980]).
- ➜ Consider the related concept of sentential forms languages (Harju, Penttonen [1979]).

# Decidability Questions for Insertion Systems

- No undecidability results are known.

- Consider the related concept of pure context-sensitive grammars (Maurer, Salomaa, Wood [1980]).
- Consider the related concept of sentential forms languages (Harju, Penttonen [1979]).
- Problem: Both concepts use the rewriting of strings instead of only inserting strings.

# Decidability Questions for Insertion Systems

- No undecidability results are known.
- Consider the related concept of pure context-sensitive grammars (Maurer, Salomaa, Wood [1980]).
- Consider the related concept of sentential forms languages (Harju, Penttonen [1979]).
- Problem: Both concepts use the rewriting of strings instead of only inserting strings.
- However, some ideas of Harju and Penttonen [1979] can be refined to work for insertion systems as well.

# Undecidability Results for Insertion Systems

**Theorem**

Let $S$ be an $\mathsf{INS}_5^{2,2}$ and $R$ be a regular language. Then, it is undecidable whether or not $R \subseteq L(S)$.

# Undecidability Results for Insertion Systems

**Theorem**

Let $S$ be an $\mathsf{INS}_5^{2,2}$ and $R$ be a regular language. Then, it is undecidable whether or not $R \subseteq L(S)$.

- Reduction from Post's Correspondence Problem.

# Undecidability Results for Insertion Systems

**Theorem**

Let $S$ be an $\mathsf{INS}_5^{2,2}$ and $R$ be a regular language. Then, it is undecidable whether or not $R \subseteq L(S)$.

- Reduction from Post's Correspondence Problem.
- Use a refinement of the construction given by Harju and Penttonen [1979].

# Undecidability Results for Insertion Systems

## Theorem

Let $S$ be an $\mathsf{INS}_5^{2,2}$ and $R$ be a regular language. Then, it is undecidable whether or not $R \subseteq L(S)$.

- Reduction from Post's Correspondence Problem.
- Use a refinement of the construction given by Harju and Penttonen [1979].
- Use the recent result (Halava, Harju, Hirvensalo, Karhumäki [2008]) that the length of the strings occurring in an instance can be bounded by 2.

# Undecidability Results for Insertion Systems

**Theorem**

Let $S$ be an $\mathsf{INS}_5^{2,2}$ and $R$ be a regular language. Then, it is undecidable whether or not $R \subseteq L(S)$.

- Reduction from Post's Correspondence Problem.
- Use a refinement of the construction given by Harju and Penttonen [1979].
- Use the recent result (Halava, Harju, Hirvensalo, Karhumäki [2008]) that the length of the strings occurring in an instance can be bounded by 2.

**Corollary**

Let $S$ and $S'$ be two insertion systems from $\mathsf{INS}_5^{2,2}$. Then, it is undecidable whether or not $L(S) \subseteq L(S')$.

# Undecidability Results for Insertion Systems

## Theorem

Let $S$ and $S'$ be two insertion systems from $\mathsf{INS}_5^{1,1}$. Then, it is undecidable whether or not $L(S) = L(S')$.

# Undecidability Results for Insertion Systems

## Theorem

Let $S$ and $S'$ be two insertion systems from $\mathsf{INS}_5^{1,1}$. Then, it is undecidable whether or not $L(S) = L(S')$.

## Corollary

Let $S$ and $S'$ be two insertion systems from $\mathsf{INS}_5^{1,1}$. Then, it is undecidable whether or not $L(S) \subseteq L(S')$.

# Undecidability Results for Insertion Systems

**Theorem**

Let $S$ and $S'$ be two insertion systems from $\mathsf{INS}_5^{1,1}$. Then, it is undecidable whether or not $L(S) = L(S')$.

**Corollary**

Let $S$ and $S'$ be two insertion systems from $\mathsf{INS}_5^{1,1}$. Then, it is undecidable whether or not $L(S) \subseteq L(S')$.

**Theorem**

Let $S$ and $S'$ be two insertion systems from $\mathsf{INS}_3^{1,1}$. Then, it is undecidable whether or not $L(S) \cap L(S')$ is empty.

# Undecidability of Finiteness

# Undecidability of Finiteness

- Use a reduction from the halting problem for two-counter machines on empty input.

# Undecidability of Finiteness

- Use a reduction from the halting problem for two-counter machines on empty input.
- Encode configurations of counter machines in strings generated by insertion systems.

# Undecidability of Finiteness

- Use a reduction from the halting problem for two-counter machines on empty input.
- Encode configurations of counter machines in strings generated by insertion systems.
- Implement the increase or decrease by inserting strings.

# Undecidability of Finiteness

- Use a reduction from the halting problem for two-counter machines on empty input.
- Encode configurations of counter machines in strings generated by insertion systems.
- Implement the increase or decrease by inserting strings.
- An essential ingredient is the construction of signals.

# Undecidability of Finiteness

- Use a reduction from the halting problem for two-counter machines on empty input.
- Encode configurations of counter machines in strings generated by insertion systems.
- Implement the increase or decrease by inserting strings.
- An essential ingredient is the construction of signals.

**Lemma**

It is possible to construct an insertion system from $\mathrm{INS}_2^{4,4}$ that realizes a signal from left to right.

# Undecidability of Finiteness

- Use a reduction from the halting problem for two-counter machines on empty input.
- Encode configurations of counter machines in strings generated by insertion systems.
- Implement the increase or decrease by inserting strings.
- An essential ingredient is the construction of signals.

**Lemma**

It is possible to construct an insertion system from $\mathrm{INS}_2^{4,4}$ that realizes a signal from left to right.

$$\triangleright \mathtt{rr} abc \triangleleft$$
$$\triangleright \mathtt{rr} a \mathtt{rr} bc \triangleleft$$
$$\triangleright \mathtt{rr} a \mathtt{rr} b \mathtt{rr} c \triangleleft$$
$$\triangleright \mathtt{r\$r} a \mathtt{rr} b \mathtt{rr} c \triangleleft$$
$$\triangleright \mathtt{r\$r} a \mathtt{r\$r} b \mathtt{rr} c \triangleleft$$
$$\triangleright \mathtt{r\$r} a \mathtt{r\$r} b \mathtt{rr} c \mathtt{rr} \triangleleft$$
$$\triangleright \mathtt{r\$r} a \mathtt{r\$r} b \mathtt{r\$r} c \mathtt{rr} \triangleleft$$

# Undecidability of Finiteness

## Lemma

It is possible to construct an insertion system from $\mathrm{INS}_2^{4,4}$ that realizes a signal from left to right followed by a signal from right to left.

# Undecidability of Finiteness

**Lemma**

It is possible to construct an insertion system from $\mathrm{INS}_2^{4,4}$ that realizes a signal from left to right followed by a signal from right to left.

**Lemma**

It is possible to construct an insertion system from $\mathrm{INS}_2^{4,4}$ that realizes signals bouncing arbitrarily often between the left and the right.

# Undecidability of Finiteness

**Lemma**

It is possible to construct an insertion system from $\mathsf{INS}_2^{4,4}$ that realizes a signal from left to right followed by a signal from right to left.

**Lemma**

It is possible to construct an insertion system from $\mathsf{INS}_2^{4,4}$ that realizes signals bouncing arbitrarily often between the left and the right.

**Lemma**

It is possible to construct an insertion system from $\mathsf{INS}_2^{4,4}$ that realizes a signal $\mathtt{rr}$ from left to right that is changed to a signal $\mathtt{r}'\mathtt{r}'$ if a certain symbol is found on its way from left to right. Moreover, such a signal may insert some symbol and continue as signal $\mathtt{r}'\mathtt{r}'$.

$\triangleright\mathtt{r\$r}a\mathtt{r\$r}b\mathtt{r\$r}c\mathtt{rr}\triangleleft$

$\triangleright\mathtt{r\$r}a\mathtt{r\$r}b\mathtt{r\$r}c\mathtt{rllr}\triangleleft$

$\triangleright\mathtt{r\$r}a\mathtt{r\$r}b\mathtt{r\$r}c\mathtt{llrllr}\triangleleft$

$\triangleright\mathtt{r\$r}a\mathtt{r\$r}b\mathtt{r\$r}c\mathtt{llrl\&lr}\triangleleft$

$\triangleright\mathtt{r\$r}a\mathtt{r\$r}b\mathtt{r\$rll}c\mathtt{llrl\&lr}\triangleleft$

$\triangleright\mathtt{r\$r}a\mathtt{r\$r}b\mathtt{r\$rll}c\mathtt{l\&lrl\&lr}\triangleleft$

$\triangleright\mathtt{r\$r}a\mathtt{r\$r}b\mathtt{r\$llrll}c\mathtt{l\&lrl\&lr}\triangleleft$

$\triangleright\mathtt{r\$r}a\mathtt{r\$r}b\mathtt{r\$llrl\&l}c\mathtt{l\&lrl\&lr}\triangleleft$

$\triangleright\mathtt{r\$r}a\mathtt{r\$r}b\mathtt{rll\$llrl\&l}c\mathtt{l\&lrl\&lr}\triangleleft$

$\triangleright\mathtt{r\$r}a\mathtt{r\$r}b\mathtt{rll\$l\&lrl\&l}c\mathtt{l\&lrl\&lr}\triangleleft$

$\triangleright\mathtt{r\$r}a\mathtt{r\$r}b\mathtt{llrll\$l\&lrl\&l}c\mathtt{l\&lrl\&lr}\triangleleft$

$\triangleright\mathtt{r\$r}a\mathtt{r\$r}b\mathtt{llrl\&l\$l\&lrl\&l}c\mathtt{l\&lrl\&lr}\triangleleft$

$\triangleright\mathtt{r\$r}a\mathtt{r\$rll}b\mathtt{llrl\&l\$l\&lrl\&l}c\mathtt{l\&lrl\&lr}\triangleleft$

$\triangleright\mathtt{r\$r}a\mathtt{r\$rll}b\mathtt{l\&lrl\&l\$l\&lrl\&l}c\mathtt{l\&lrl\&lr}\triangleleft$

$\triangleright\mathtt{r\$r}a\mathtt{r\$llrll}b\mathtt{l\&lrl\&l\$l\&lrl\&l}c\mathtt{l\&lrl\&lr}\triangleleft$

$\triangleright\mathtt{r\$r}a\mathtt{r\$llrl\&l}b\mathtt{l\&lrl\&l\$l\&lrl\&l}c\mathtt{l\&lrl\&lr}\triangleleft$

$\triangleright\mathtt{r\$r}a\mathtt{rll\$llrl\&l}b\mathtt{l\&lrl\&l\$l\&lrl\&l}c\mathtt{l\&lrl\&lr}\triangleleft$

...

# Undecidability of Finiteness

## Theorem

Let $S$ be an insertion system from $\mathsf{INS}_3^{4,4}$. Then, it is undecidable whether or not $L(S)$ is finite.

# Undecidability of Finiteness

**Theorem**

Let $S$ be an insertion system from $\mathsf{INS}_3^{4,4}$. Then, it is undecidable whether or not $L(S)$ is finite.

- Configurations are words consisting of a center part containing the current state and the information whether or not Counter 1 and Counter 2 are zero,

# Undecidability of Finiteness

**Theorem**

Let $S$ be an insertion system from $\mathsf{INS}_3^{4,4}$. Then, it is undecidable whether or not $L(S)$ is finite.

- Configurations are words consisting of a center part containing the current state and the information whether or not Counter 1 and Counter 2 are zero,
- a right part that contains the current value of Counter 1, and

# Undecidability of Finiteness

**Theorem**

Let $S$ be an insertion system from $\mathsf{INS}_3^{4,4}$. Then, it is undecidable whether or not $L(S)$ is finite.

- Configurations are words consisting of a center part containing the current state and the information whether or not Counter 1 and Counter 2 are zero,
- a right part that contains the current value of Counter 1, and
- a left part that contains the current value of Counter 2.

# Undecidability of Finiteness

**Theorem**

Let $S$ be an insertion system from $\mathsf{INS}_3^{4,4}$. Then, it is undecidable whether or not $L(S)$ is finite.

- Configurations are words consisting of a center part containing the current state and the information whether or not Counter 1 and Counter 2 are zero,
- a right part that contains the current value of Counter 1, and
- a left part that contains the current value of Counter 2.
- The axiom is $\triangleright{=}_2 s_0 {=}_1 \triangleleft$.

# Undecidability of Finiteness

**Theorem**

Let $S$ be an insertion system from $\mathsf{INS}_3^{4,4}$. Then, it is undecidable whether or not $L(S)$ is finite.

- Configurations are words consisting of a center part containing the current state and the information whether or not Counter 1 and Counter 2 are zero,
- a right part that contains the current value of Counter 1, and
- a left part that contains the current value of Counter 2.
- The axiom is $\triangleright =_2 s_0 =_1 \triangleleft$.
- Knowing the center part $=_2 s_0 =_1$ the next state can be determined according to the transition function and the current value of Counter 1 or Counter 2 has to be changed.

# Undecidability of Finiteness

**Theorem**

Let $S$ be an insertion system from $\mathsf{INS}_3^{4,4}$. Then, it is undecidable whether or not $L(S)$ is finite.

- Configurations are words consisting of a center part containing the current state and the information whether or not Counter 1 and Counter 2 are zero,
- a right part that contains the current value of Counter 1, and
- a left part that contains the current value of Counter 2.
- The axiom is $\triangleright =_2 s_0 =_1 \triangleleft$.
- Knowing the center part $=_2 s_0 =_1$ the next state can be determined according to the transition function and the current value of Counter 1 or Counter 2 has to be changed.
- The current value $m$ of a counter will be encoded by $m$ symbols 1 in the left or right part of the word.

# Undecidability of Finiteness

### Theorem

Let $S$ be an insertion system from $\mathsf{INS}_3^{4,4}$. Then, it is undecidable whether or not $L(S)$ is finite.

# Undecidability of Finiteness

**Theorem**

Let $S$ be an insertion system from $\mathsf{INS}_3^{4,4}$. Then, it is undecidable whether or not $L(S)$ is finite.

- To increase a counter we basically insert a symbol 1 to the left or right and update the center.

# Undecidability of Finiteness

**Theorem**

Let $S$ be an insertion system from $\mathsf{INS}_3^{4,4}$. Then, it is undecidable whether or not $L(S)$ is finite.

- ➜ To increase a counter we basically insert a symbol 1 to the left or right and update the center.

Example:

$$\triangleright =_2 s_0 =_1 \triangleleft$$
$$\triangleright =_2 s_0 \overline{s}_1 > 1 =_1 \triangleleft$$
$$\triangleright =_2 s_0 =_2 s_1 > \overline{s}_1 > 1 =_1 \triangleleft$$
$$\triangleright =_2 s_0 =_2 s_1 \overline{s}_2 > 1 > \overline{s}_1 > 1 =_1 \triangleleft$$
$$\triangleright =_2 s_0 =_2 s_1 =_2 s_2 > \overline{s}_2 > 1 > \overline{s}_1 > 1 =_1 \triangleleft$$

# Undecidability of Finiteness

## Theorem

Let $S$ be an insertion system from $\mathsf{INS}_3^{4,4}$. Then, it is undecidable whether or not $L(S)$ is finite.

# Undecidability of Finiteness

**Theorem**

Let $S$ be an insertion system from $\mathsf{INS}_3^{4,4}$. Then, it is undecidable whether or not $L(S)$ is finite.

- To decrease a counter we have to make a valid 1 invalid.

# Undecidability of Finiteness

**Theorem**

Let $S$ be an insertion system from $\mathsf{INS}_3^{4,4}$. Then, it is undecidable whether or not $L(S)$ is finite.

- To decrease a counter we have to make a valid 1 invalid.
- Symbol $\overline{1}$ has the meaning that a subsequent 1 is invalid and does not count for the current value of the counter.

# Undecidability of Finiteness

**Theorem**

Let $S$ be an insertion system from $\mathsf{INS}_3^{4,4}$. Then, it is undecidable whether or not $L(S)$ is finite.

- To decrease a counter we have to make a valid 1 invalid.
- Symbol $\overline{1}$ has the meaning that a subsequent 1 is invalid and does not count for the current value of the counter.
- Send a signal which checks whether there is a valid 1,

# Undecidability of Finiteness

**Theorem**

Let $S$ be an insertion system from $\mathsf{INS}_3^{4,4}$. Then, it is undecidable whether or not $L(S)$ is finite.

- To decrease a counter we have to make a valid 1 invalid.
- Symbol $\overline{1}$ has the meaning that a subsequent 1 is invalid and does not count for the current value of the counter.
- Send a signal which checks whether there is a valid 1,
- make this 1 invalid by inserting $\overline{1}$ in front of it,

# Undecidability of Finiteness

**Theorem**

Let $S$ be an insertion system from $\mathrm{INS}_3^{4,4}$. Then, it is undecidable whether or not $L(S)$ is finite.

- To decrease a counter we have to make a valid 1 invalid.
- Symbol $\overline{1}$ has the meaning that a subsequent 1 is invalid and does not count for the current value of the counter.
- Send a signal which checks whether there is a valid 1,
- make this 1 invalid by inserting $\overline{1}$ in front of it,
- check whether there is another valid 1. If so, the counter is non-zero and it is zero otherwise.

# Undecidability of Finiteness

**Theorem**

Let $S$ be an insertion system from $\mathrm{INS}_3^{4,4}$. Then, it is undecidable whether or not $L(S)$ is finite.

- To decrease a counter we have to make a valid 1 invalid.
- Symbol $\overline{1}$ has the meaning that a subsequent 1 is invalid and does not count for the current value of the counter.
- Send a signal which checks whether there is a valid 1,
- make this 1 invalid by inserting $\overline{1}$ in front of it,
- check whether there is another valid 1. If so, the counter is non-zero and it is zero otherwise.
- Send this information back to the center and update the state and the status of the counter.

# Undecidability of Finiteness

**Theorem**

Let $S$ be an insertion system from $\mathsf{INS}_3^{4,4}$. Then, it is undecidable whether or not $L(S)$ is finite.

- To decrease a counter we have to make a valid 1 invalid.
- Symbol $\overline{1}$ has the meaning that a subsequent 1 is invalid and does not count for the current value of the counter.
- Send a signal which checks whether there is a valid 1,
- make this 1 invalid by inserting $\overline{1}$ in front of it,
- check whether there is another valid 1. If so, the counter is non-zero and it is zero otherwise.
- Send this information back to the center and update the state and the status of the counter.
- All tasks can be realized with the constructions provided.

# Undecidability Results

### Theorem

Let $S$ be an insertion system from $\mathsf{INS}_3^{4,4}$. Then, it is undecidable whether or not $L(S)$ is finite.

# Undecidability Results

**Theorem**

Let $S$ be an insertion system from $\mathsf{INS}_3^{4,4}$. Then, it is undecidable whether or not $L(S)$ is finite.

- The insertion system generates finitely many words if and only if the two-counter machine halts.

# Undecidability Results

**Theorem**

Let $S$ be an insertion system from $\mathsf{INS}_3^{4,4}$. Then, it is undecidable whether or not $L(S)$ is finite.

- The insertion system generates finitely many words if and only if the two-counter machine halts.

**Theorem**

Let $S$ be an insertion system from $\mathsf{INS}_3^{4,4}$ and $R$ be a regular language. Then, it is undecidable whether or not $L(S) \subseteq R$.

# Undecidability Results

**Theorem**

Let $S$ be an insertion system from $\mathsf{INS}_3^{4,4}$. Then, it is undecidable whether or not $L(S)$ is finite.

- The insertion system generates finitely many words if and only if the two-counter machine halts.

**Theorem**

Let $S$ be an insertion system from $\mathsf{INS}_3^{4,4}$ and $R$ be a regular language. Then, it is undecidable whether or not $L(S) \subseteq R$.

**Theorem**

Let $S$ be an insertion system from $\mathsf{INS}_4^{4,4}$ Then, it is undecidable whether or not $L(S)$ is a regular language.

# Conclusions

# Conclusions

- Emptiness and universality are decidable for any insertion system.

# Conclusions

- Emptiness and universality are decidable for any insertion system.
- Finiteness and inclusion in a regular language is undecidable for all insertion systems having a size of at least $(3, 4, 4)$.

# Conclusions

- Emptiness and universality are decidable for any insertion system.
- Finiteness and inclusion in a regular language is undecidable for all insertion systems having a size of at least $(3, 4, 4)$.
- Equality and inclusion is undecidable for all insertion systems having a size of at least $(5, 1, 1)$.

# Conclusions

- Emptiness and universality are decidable for any insertion system.
- Finiteness and inclusion in a regular language is undecidable for all insertion systems having a size of at least $(3, 4, 4)$.
- Equality and inclusion is undecidable for all insertion systems having a size of at least $(5, 1, 1)$.
- Inclusion of a regular language is undecidable for all insertion systems having a size of at least $(3, 4, 4)$.

# Conclusions

- Emptiness and universality are decidable for any insertion system.
- Finiteness and inclusion in a regular language is undecidable for all insertion systems having a size of at least $(3, 4, 4)$.
- Equality and inclusion is undecidable for all insertion systems having a size of at least $(5, 1, 1)$.
- Inclusion of a regular language is undecidable for all insertion systems having a size of at least $(3, 4, 4)$.
- Regularity is undecidable for all insertion systems having a size of at least $(4, 4, 4)$.

# Conclusions

- Emptiness and universality are decidable for any insertion system.
- Finiteness and inclusion in a regular language is undecidable for all insertion systems having a size of at least $(3, 4, 4)$.
- Equality and inclusion is undecidable for all insertion systems having a size of at least $(5, 1, 1)$.
- Inclusion of a regular language is undecidable for all insertion systems having a size of at least $(3, 4, 4)$.
- Regularity is undecidable for all insertion systems having a size of at least $(4, 4, 4)$.
- The decidability status for insertion systems having a size which is not covered by the above cases is unknown.

# Conclusions

- Emptiness and universality are decidable for any insertion system.
- Finiteness and inclusion in a regular language is undecidable for all insertion systems having a size of at least $(3, 4, 4)$.
- Equality and inclusion is undecidable for all insertion systems having a size of at least $(5, 1, 1)$.
- Inclusion of a regular language is undecidable for all insertion systems having a size of at least $(3, 4, 4)$.
- Regularity is undecidable for all insertion systems having a size of at least $(4, 4, 4)$.
- The decidability status for insertion systems having a size which is not covered by the above cases is unknown.
- Study generalized systems such as graph-controlled insertion systems or matrix insertion grammars of small size which are known to be computationally incomplete.