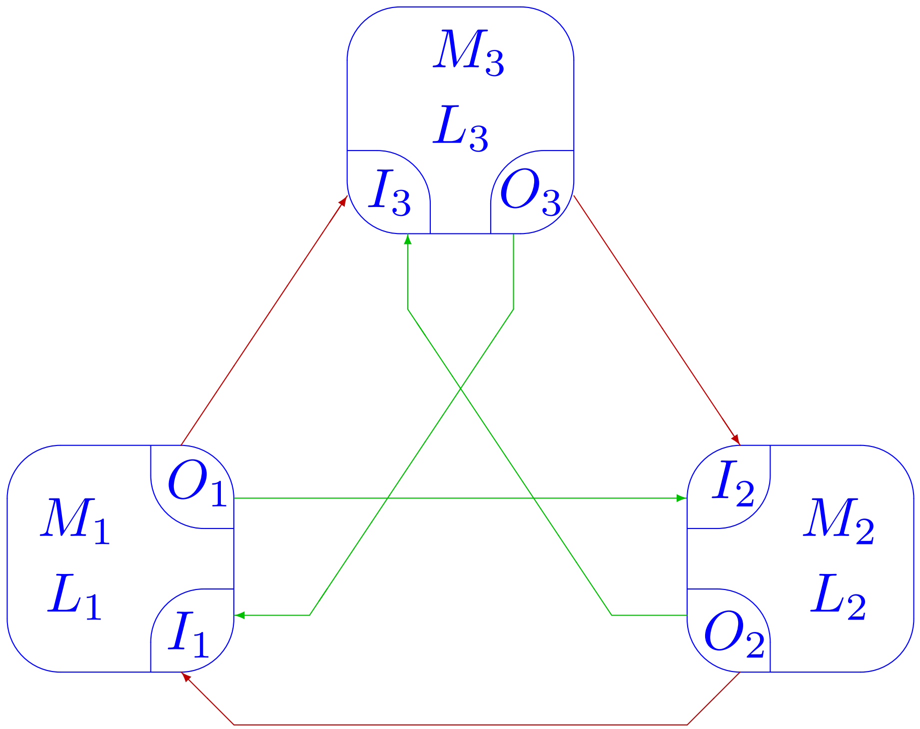
Target Based Accepting Networks of Evolutionary Processors with Regular Filters

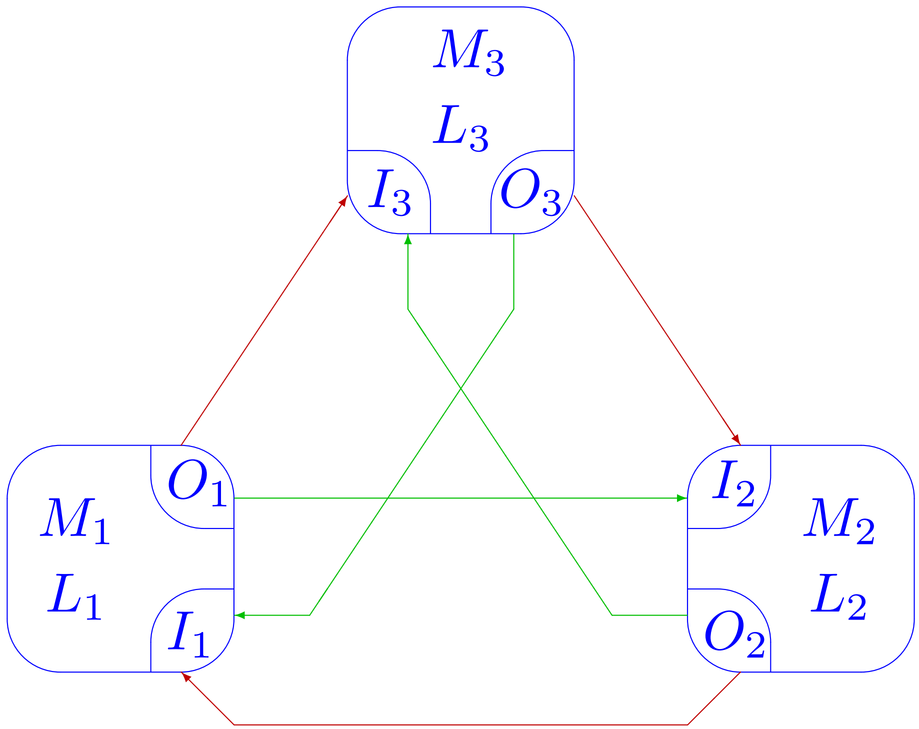
Bianca Truthe
Otto-von-Guericke-Universität Magdeburg, Germany
truthe@iws.cs.uni-magdeburg.de

Workshop on Non-Classical Models of Automata and Applications
August 31 – September 1, 2009, Wrocław, Poland

Introduction

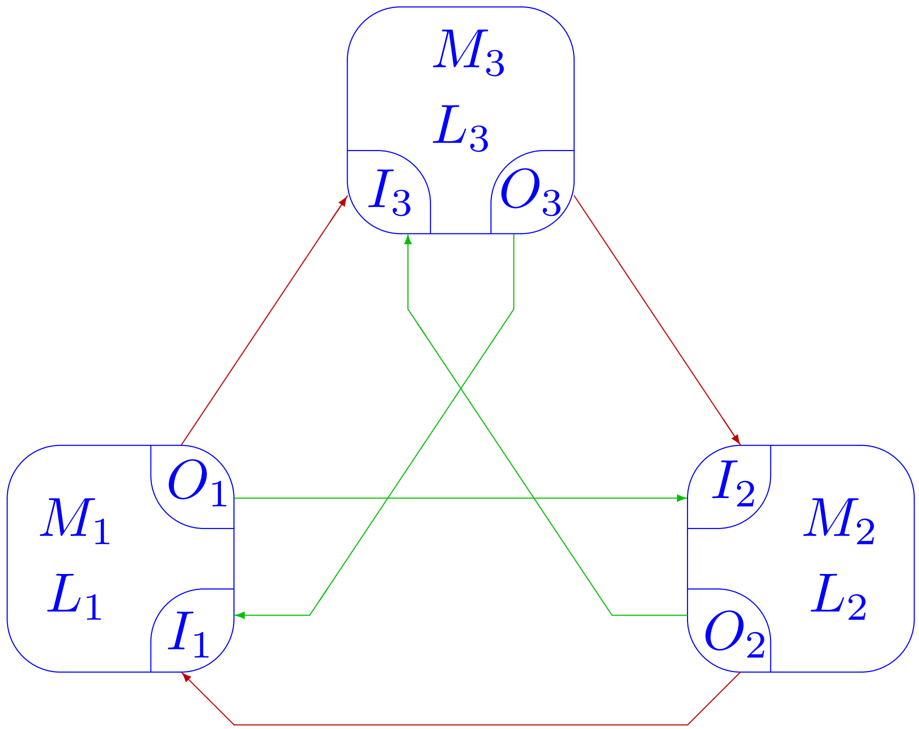


Introduction



- E. Csuhaj-Varjú, A. Salomaa: In *New Trends in Formal Languages*, 1997
- J. Castellanos, C. Martín-Vide, V. Mitrana, J. Sempere: In *LNCS 2084*, 2001

Introduction



- E. Csuhaj-Varjú, A. Salomaa: In *New Trends in Formal Languages*, 1997
- J. Castellanos, C. Martín-Vide, V. Mitrana, J. Sempere: In *LNCS 2084*, 2001
- A. Alhazov, J. Dassow, C. Martín-Vide, Y. Rogozhin, B. Truthe: Fundamenta Informaticae 91 (2009)
- J. Dassow, V. Mitrana: NCGT'08
- V. Mitrana, B. Truthe: LATA'09

Definitions

ANEP: $\mathcal{N} = (U, V, N_1, N_2, \ldots, N_n, E, j, O)$

Processor: $N_i = (M_i, I_i, O_i)$

substituting: $M_i \subseteq \{ a \to b \mid a, b \in V \}$
deleting: $M_i \subseteq \{ a \to \lambda \mid a \in V \}$
inserting: $M_i \subseteq \{ \lambda \to b \mid b \in V \}$

Definitions

ANEP: $\mathcal{N} = (U, V, N_1, N_2, \ldots, N_n, E, j, O)$

Processor: $N_i = (M_i, I_i, O_i)$

- substituting: $M_i \subseteq \{ a \to b \mid a, b \in V \}$
- deleting: $M_i \subseteq \{ a \to \lambda \mid a \in V \}$
- inserting: $M_i \subseteq \{ \lambda \to b \mid b \in V \}$

Configuration: $C_t^w = (C_t^w(1), C_t^w(2), \ldots, C_t^w(n))$ $[C_0^w(j) = \{w\},\ C_0^w(i) = \emptyset]$

Definitions

ANEP: $\mathcal{N} = (U, V, N_1, N_2, \ldots, N_n, E, j, O)$

Processor: $N_i = (M_i, I_i, O_i)$

- substituting: $M_i \subseteq \{ a \to b \mid a, b \in V \}$
- deleting: $M_i \subseteq \{ a \to \lambda \mid a \in V \}$
- inserting: $M_i \subseteq \{ \lambda \to b \mid b \in V \}$

Configuration: $C_t^w = (C_t^w(1), C_t^w(2), \ldots, C_t^w(n))$ $[C_0^w(j) = \{w\},\ C_0^w(i) = \emptyset]$

Evolution: $C_{2t}^w(i) \Longrightarrow^{M_i} C_{2t+1}^w(i)$

Communication: $C_{2t+2}^w(i) = C_{2t+1}^w(i) \setminus O_i \cup \bigcup_{(k,i) \in E} C_{2t+1}^w(k) \cap O_k \cap I_i$

Definitions

ANEP: $\mathcal{N} = (U, V, N_1, N_2, \ldots, N_n, E, j, O)$

Processor: $N_i = (M_i, I_i, O_i)$

- substituting: $M_i \subseteq \{\, a \to b \mid a, b \in V \,\}$
- deleting: $M_i \subseteq \{\, a \to \lambda \mid a \in V \,\}$
- inserting: $M_i \subseteq \{\, \lambda \to b \mid b \in V \,\}$

Configuration: $C_t^w = (C_t^w(1), C_t^w(2), \ldots, C_t^w(n))$ $[C_0^w(j) = \{w\},\ C_0^w(i) = \emptyset]$

Evolution: $C_{2t}^w(i) \Longrightarrow^{M_i} C_{2t+1}^w(i)$

Communication: $C_{2t+2}^w(i) = C_{2t+1}^w(i) \setminus O_i \cup \bigcup_{(k,i) \in E} C_{2t+1}^w(k) \cap O_k \cap I_i$

Computation: $C_0 \Longrightarrow C_1 \vdash C_2 \Longrightarrow C_3 \vdash \cdots$

Definitions

ANEP: $\mathcal{N} = (U, V, N_1, N_2, \ldots, N_n, E, j, O)$

Processor: $N_i = (M_i, I_i, O_i)$

- substituting: $M_i \subseteq \{ a \to b \mid a, b \in V \}$
- deleting: $M_i \subseteq \{ a \to \lambda \mid a \in V \}$
- inserting: $M_i \subseteq \{ \lambda \to b \mid b \in V \}$

Configuration: $C_t^w = (C_t^w(1), C_t^w(2), \ldots, C_t^w(n))$ $[C_0^w(j) = \{w\},\ C_0^w(i) = \emptyset]$

Evolution: $C_{2t}^w(i) \Longrightarrow^{M_i} C_{2t+1}^w(i)$

Communication: $C_{2t+2}^w(i) = C_{2t+1}^w(i) \setminus O_i \cup \bigcup_{(k,i)\in E} C_{2t+1}^w(k) \cap O_k \cap I_i$

Computation: $C_0 \Longrightarrow C_1 \vdash C_2 \Longrightarrow C_3 \vdash \cdots$

Language accepted: $L(\mathcal{N}) = \{ w \in U^* \mid \exists t \geq 0\, \exists o \in O : C_t^w(o) \neq \emptyset \}$

Previous Work

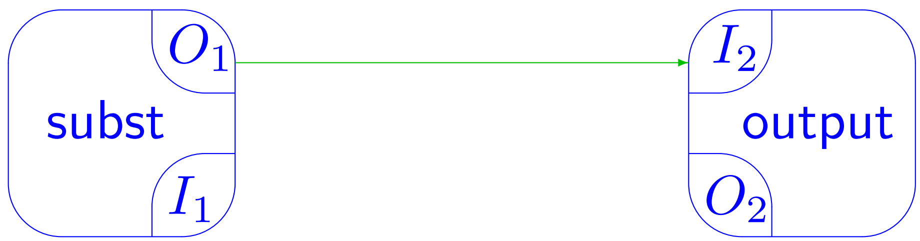
Theorem: *For accepting any context-sensitive language, one substituting processor and one output processor are sufficient.*

- B. Truthe, V. Mitrana: In LATA 2009, *LNCS 5457*, 2009

Previous Work

Theorem: *For accepting any context-sensitive language, one substituting processor and one output processor are sufficient.*

Proof: NEP with 1 substituting node and 1 output node simulating backwards a CS grammar in Kuroda normal form



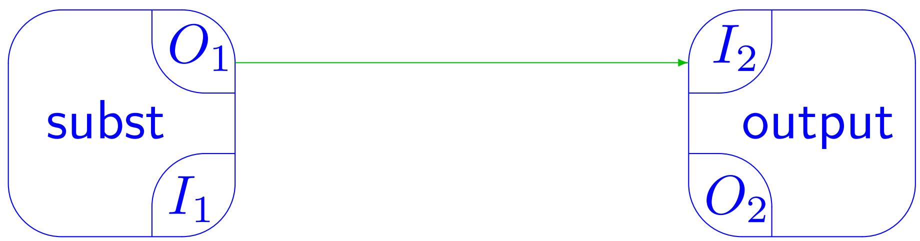
$A \rightarrow BC$

- B. Truthe, V. Mitrana: In LATA 2009, *LNCS 5457*, 2009

Previous Work

Theorem: *For accepting any context-sensitive language, one substituting processor and one output processor are sufficient.*

Proof: NEP with 1 substituting node and 1 output node simulating backwards a CS grammar in Kuroda normal form



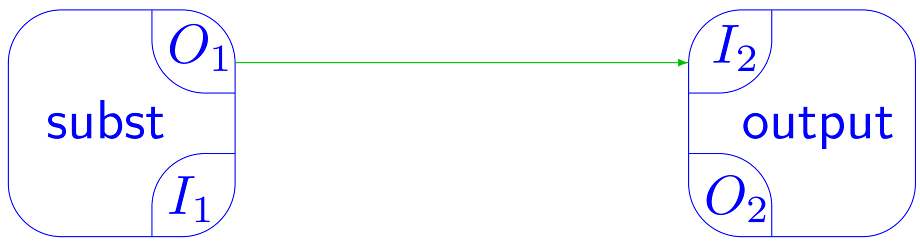
$A \rightarrow BC$

- B. Truthe, V. Mitrana: In LATA 2009, *LNCS 5457*, 2009

Previous Work

Theorem: *For accepting any context-sensitive language, one substituting processor and one output processor are sufficient.*

Proof: NEP with 1 substituting node and 1 output node simulating backwards a CS grammar in Kuroda normal form



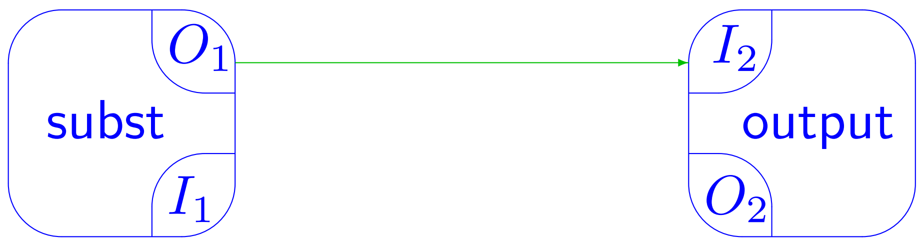
$A \to BC$: $u\,B\,C\,v$

- B. Truthe, V. Mitrana: In LATA 2009, *LNCS 5457*, 2009

Previous Work

Theorem: *For accepting any context-sensitive language, one substituting processor and one output processor are sufficient.*

Proof: NEP with 1 substituting node and 1 output node simulating backwards a CS grammar in Kuroda normal form



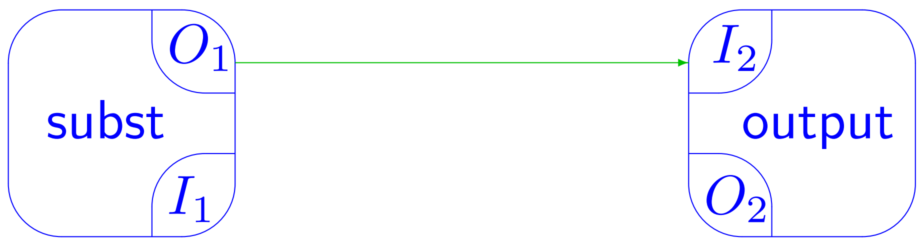
$A \rightarrow BC$: up_1Cv

- B. Truthe, V. Mitrana: In LATA 2009, *LNCS 5457*, 2009

Previous Work

Theorem: *For accepting any context-sensitive language, one substituting processor and one output processor are sufficient.*

Proof: NEP with 1 substituting node and 1 output node simulating backwards a CS grammar in Kuroda normal form



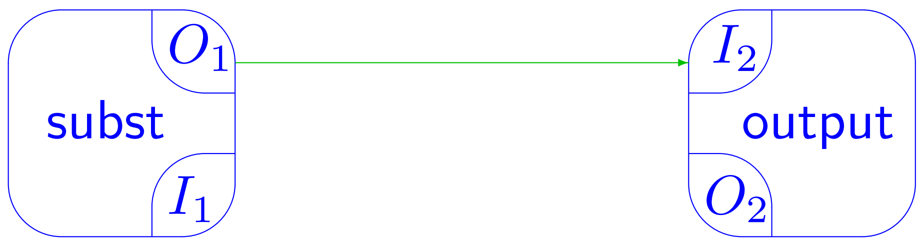
$A \to BC$: $u p_1 C v$

- B. Truthe, V. Mitrana: In LATA 2009, *LNCS 5457*, 2009

Previous Work

Theorem: *For accepting any context-sensitive language, one substituting processor and one output processor are sufficient.*

Proof: NEP with 1 substituting node and 1 output node simulating backwards a CS grammar in Kuroda normal form



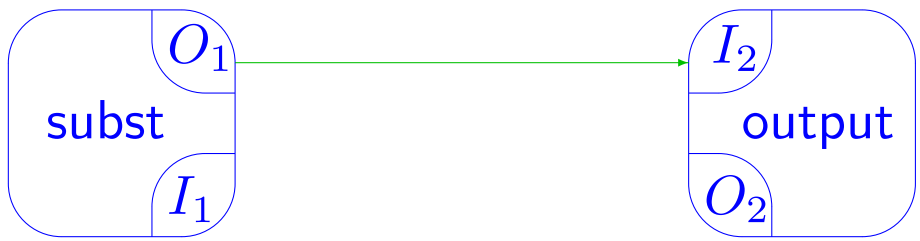
$A \to BC$: up_1p_2v

- B. Truthe, V. Mitrana: In LATA 2009, *LNCS 5457*, 2009

Previous Work

Theorem: *For accepting any context-sensitive language, one substituting processor and one output processor are sufficient.*

Proof: NEP with 1 substituting node and 1 output node simulating backwards a CS grammar in Kuroda normal form



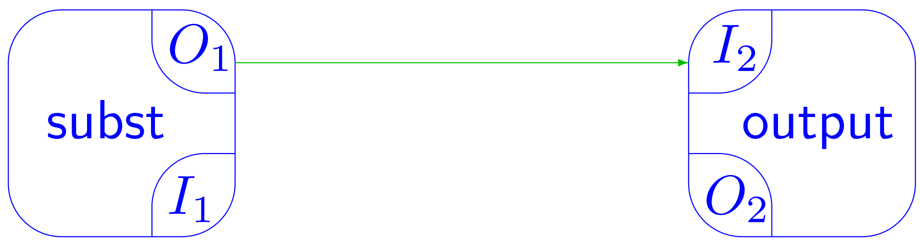
$A \to BC$: $u p_1 p_2 v$

- B. Truthe, V. Mitrana: In LATA 2009, *LNCS 5457*, 2009

Previous Work

Theorem: *For accepting any context-sensitive language, one substituting processor and one output processor are sufficient.*

Proof: NEP with 1 substituting node and 1 output node simulating backwards a CS grammar in Kuroda normal form



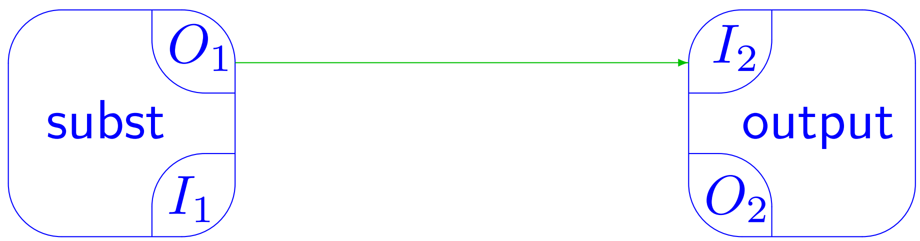
$A \to BC$: $u p_3 p_2 v$

- B. Truthe, V. Mitrana: In LATA 2009, *LNCS 5457*, 2009

Previous Work

Theorem: *For accepting any context-sensitive language, one substituting processor and one output processor are sufficient.*

Proof: NEP with 1 substituting node and 1 output node simulating backwards a CS grammar in Kuroda normal form



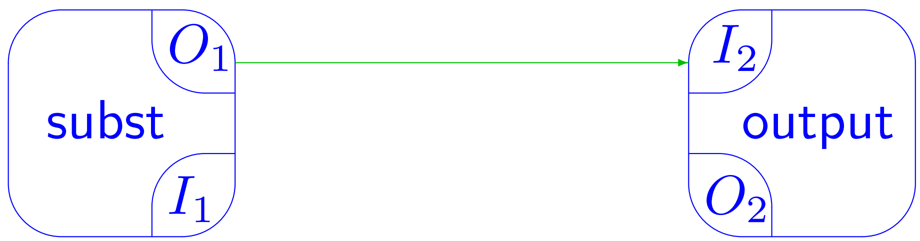
$A \to BC$: up_3p_2v

- B. Truthe, V. Mitrana: In LATA 2009, *LNCS 5457*, 2009

Previous Work

Theorem: *For accepting any context-sensitive language, one substituting processor and one output processor are sufficient.*

Proof: NEP with 1 substituting node and 1 output node simulating backwards a CS grammar in Kuroda normal form



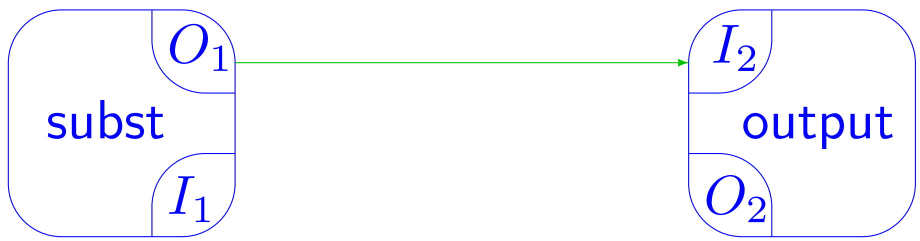
$A \to BC$: up_3p_4v

- B. Truthe, V. Mitrana: In LATA 2009, *LNCS 5457*, 2009

Previous Work

Theorem: *For accepting any context-sensitive language, one substituting processor and one output processor are sufficient.*

Proof: NEP with 1 substituting node and 1 output node simulating backwards a CS grammar in Kuroda normal form



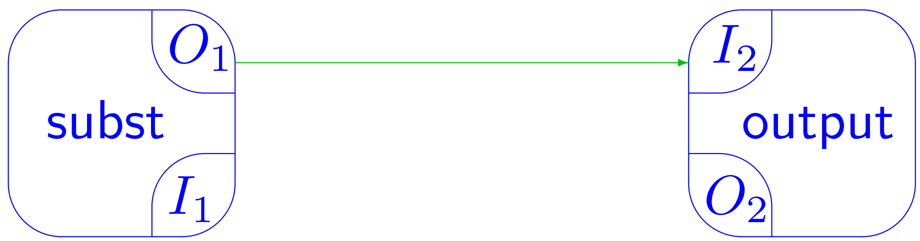
$A \to BC$: $u p_3 p_4 v$

- B. Truthe, V. Mitrana: In LATA 2009, *LNCS 5457*, 2009

Previous Work

Theorem: *For accepting any context-sensitive language, one substituting processor and one output processor are sufficient.*

Proof: NEP with 1 substituting node and 1 output node simulating backwards a CS grammar in Kuroda normal form



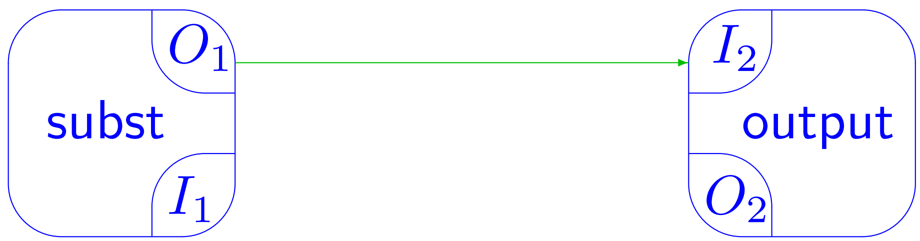
$A \to BC$: uAp_4v

- B. Truthe, V. Mitrana: In LATA 2009, *LNCS 5457*, 2009

Previous Work

Theorem: *For accepting any context-sensitive language, one substituting processor and one output processor are sufficient.*

Proof: NEP with 1 substituting node and 1 output node simulating backwards a CS grammar in Kuroda normal form



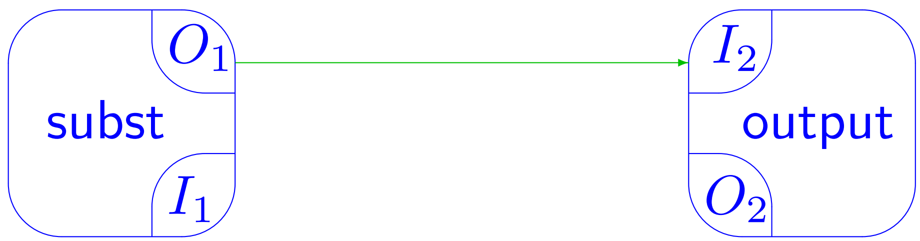
$A \to BC$: $uA\,p_4 v$

- B. Truthe, V. Mitrana: In LATA 2009, *LNCS 5457*, 2009

Previous Work

Theorem: *For accepting any context-sensitive language, one substituting processor and one output processor are sufficient.*

Proof: NEP with 1 substituting node and 1 output node simulating backwards a CS grammar in Kuroda normal form



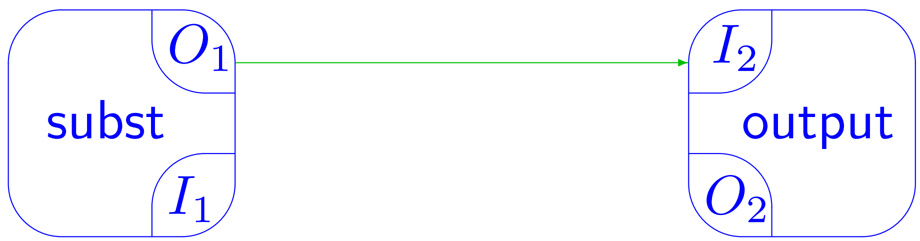
$A \rightarrow BC$: uA_v

- B. Truthe, V. Mitrana: In LATA 2009, *LNCS 5457*, 2009

Previous Work

Theorem: *For accepting any context-sensitive language, one substituting processor and one output processor are sufficient.*

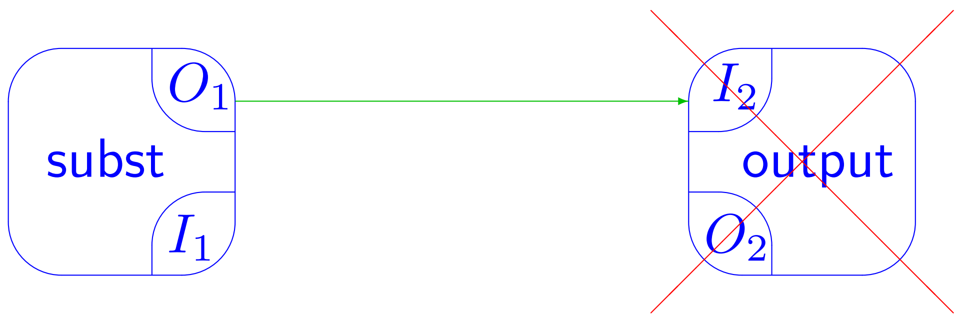
Proof: NEP with 1 substituting node and 1 output node simulating backwards a CS grammar in Kuroda normal form



$$I_2 = \{\_\}^*\{S\}\{\_\}^*$$

- B. Truthe, V. Mitrana: In LATA 2009, *LNCS 5457*, 2009

New Idea



$$B = \{\_\}^*\{S\}\{\_\}^*$$

Substitution and Insertion

Theorem: *Any recursively enumerable language can be accepted by a network of one substituting processor and one inserting processor.*

Substitution and Insertion

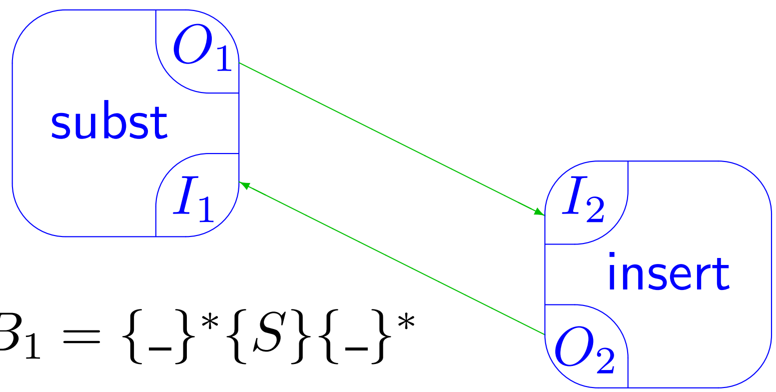
Theorem: *Any recursively enumerable language can be accepted by a network of one substituting processor and one inserting processor.*

Proof: NEP with 1 substituting node, 1 inserting node and 1 output node simulating backwards an RE grammar in Kuroda normal form

$A \to x$

$A \to BC$

$AB \to CD$



$$B_1 = \{\_\}^*\{S\}\{\_\}^*$$

$$A \to \lambda$$

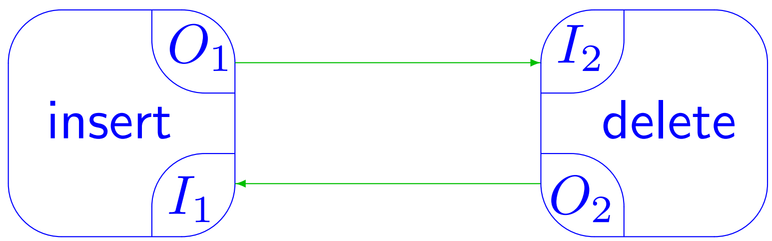
Insertion and Deletion

Theorem: *Any recursively enumerable language can be accepted by a network of one inserting processors and one deleting processor.*

Insertion and Deletion

Theorem: *Any recursively enumerable language can be accepted by a network of one inserting processors and one deleting processor.*

Proof: NEP with 1 inserting node and 1 deleting node simulating backwards an RE grammar in Kuroda normal form



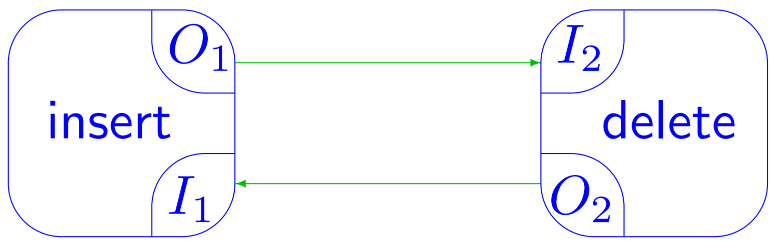
$B_2 = \{S_-\}$

$\alpha \to \beta$: $u\beta v$

Insertion and Deletion

Theorem: *Any recursively enumerable language can be accepted by a network of one inserting processors and one deleting processor.*

Proof: NEP with 1 inserting node and 1 deleting node simulating backwards an RE grammar in Kuroda normal form



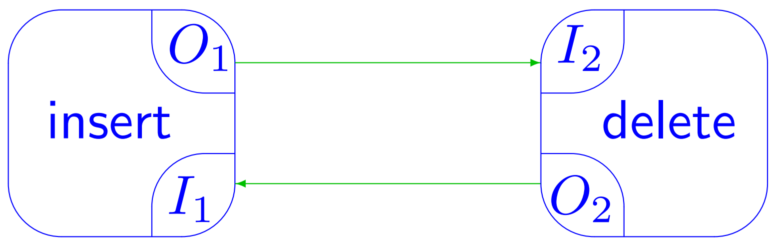
$B_2 = \{S_-\}$

$\alpha \to \beta$: $up_1p_3\beta p_2p_4v$

Insertion and Deletion

Theorem: *Any recursively enumerable language can be accepted by a network of one inserting processors and one deleting processor.*

Proof: NEP with 1 inserting node and 1 deleting node simulating backwards an RE grammar in Kuroda normal form



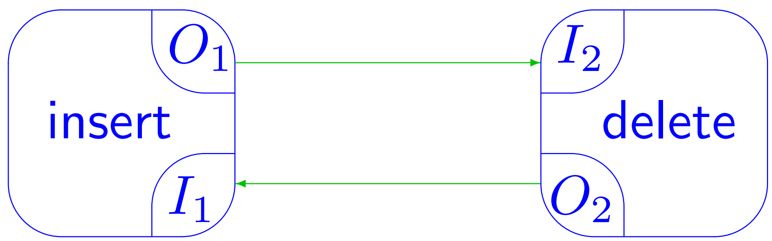
$B_2 = \{S_-\}$

$\alpha \to \beta$: $up_1p_3\alpha\beta p_2p_4v$

Insertion and Deletion

Theorem: *Any recursively enumerable language can be accepted by a network of one inserting processors and one deleting processor.*

Proof: NEP with 1 inserting node and 1 deleting node simulating backwards an RE grammar in Kuroda normal form



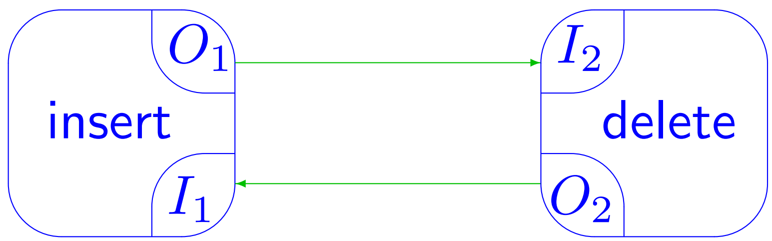
$B_2 = \{S_-\}$

$\alpha \rightarrow \beta$: $up_1p_3\alpha\beta p_2p_4v$

Insertion and Deletion

Theorem: *Any recursively enumerable language can be accepted by a network of one inserting processors and one deleting processor.*

Proof: NEP with 1 inserting node and 1 deleting node simulating backwards an RE grammar in Kuroda normal form



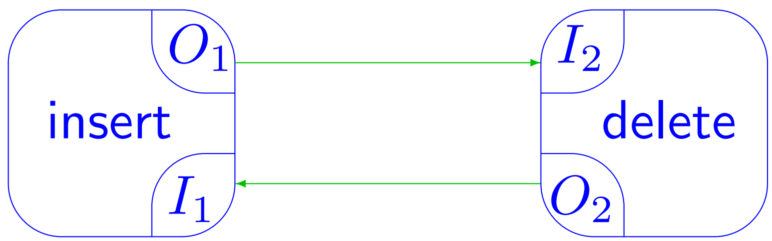
$B_2 = \{S_-\}$

$\alpha \to \beta$: $up_1p_3\alpha\beta p_2p_4v$

Insertion and Deletion

Theorem: *Any recursively enumerable language can be accepted by a network of one inserting processors and one deleting processor.*

Proof: NEP with 1 inserting node and 1 deleting node simulating backwards an RE grammar in Kuroda normal form



$B_2 = \{S_-\}$

$\alpha \rightarrow \beta$: $u\alpha v$

Definition of Target Based ANEPs

TB-ANEP: $\mathcal{N} = (U, V, N_1, N_2, \ldots, N_n, E, j)$

Processor: $N_i = (M_i, I_i, O_i, B_i)$

$B_i \subseteq V^*$ is called target set

Definition of Target Based ANEPs

TB-ANEP: $\mathcal{N} = (U, V, N_1, N_2, \ldots, N_n, E, j)$

Processor: $N_i = (M_i, I_i, O_i, B_i)$

$B_i \subseteq V^*$ is called target set

Language accepted:

$L(\mathcal{N}) = \{\, w \in U^* \mid \exists t \geq 0\, \exists o : 1 \leq o \leq n$ and $C_t^w(o) \cap B_o \neq \emptyset \,\}$

Definition of Target Based ANEPs

TB-ANEP: $\mathcal{N} = (U, V, N_1, N_2, \ldots, N_n, E, j)$

Processor: $N_i = (M_i, I_i, O_i, B_i)$

$B_i \subseteq V^*$ is called target set

Language accepted:

$L(\mathcal{N}) = \{\, w \in U^* \mid \exists t \geq 0\, \exists o : 1 \leq o \leq n$ and $C_t^w(o) \cap B_o \neq \emptyset \,\}$

Theorem: Every conventional network can be transformed into a target based network that accepts the same language.

Equivalence

Theorem: Every target based network can be transformed into a conventional network that accepts the same language.

Equivalence

Theorem: Every target based network can be transformed into a conventional network that accepts the same language.

TB-ANEP is called *acceptance uniform* if all nodes $N_o = (M_o, I_o, O_o, B_o)$ with $B_o \neq \emptyset$ satisfy $M_o = \emptyset$.

Equivalence

Theorem: Every target based network can be transformed into a conventional network that accepts the same language.

TB-ANEP is called *acceptance uniform* if all nodes $N_o = (M_o, I_o, O_o, B_o)$ with $B_o \neq \emptyset$ satisfy $M_o = \emptyset$.

Lemma: Every target based network can be transformed into an acceptance uniform network that accepts the same language.

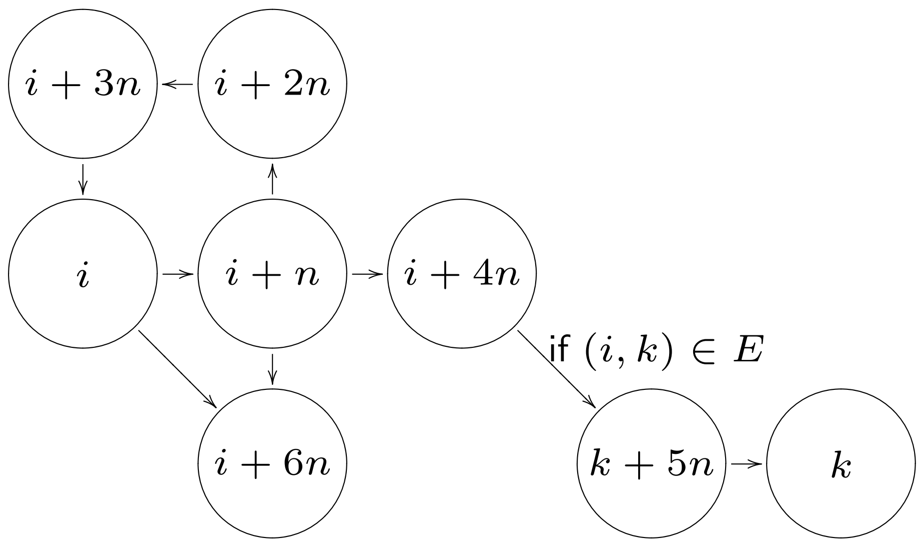
Acceptance Uniform TB-ANEP

Lemma: Every target based network can be transformed into an acceptance uniform network that accepts the same language.

Acceptance Uniform TB-ANEP

Lemma: Every target based network can be transformed into an acceptance uniform network that accepts the same language.

$$
\begin{aligned}
N'_i &= (\emptyset, V^*, V^*, \emptyset),\\
N'_{i+n} &= (M_i, V^*, V^*, \emptyset),\\
N'_{i+2n} &= (\emptyset, V^* \setminus O_i, V^*, \emptyset),\\
N'_{i+3n} &= (\emptyset, V^*, V^*, \emptyset),\\
N'_{i+4n} &= (\emptyset, O_i, V^*, \emptyset),\\
N'_{i+5n} &= (\emptyset, I_i, V^*, \emptyset),\\
N'_{i+6n} &= (\emptyset, B_i, V^*, B_i).
\end{aligned}
$$

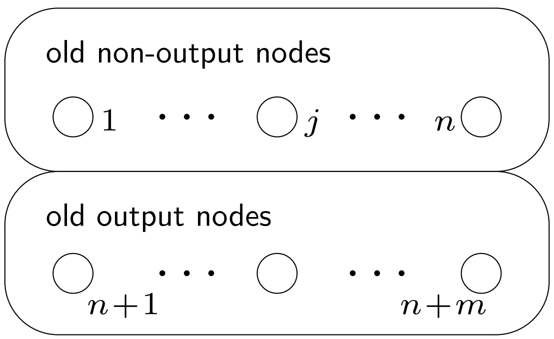


Conventional ANEP

Lemma: Every acceptance uniform network can be transformed into a conventional network that accepts the same language.

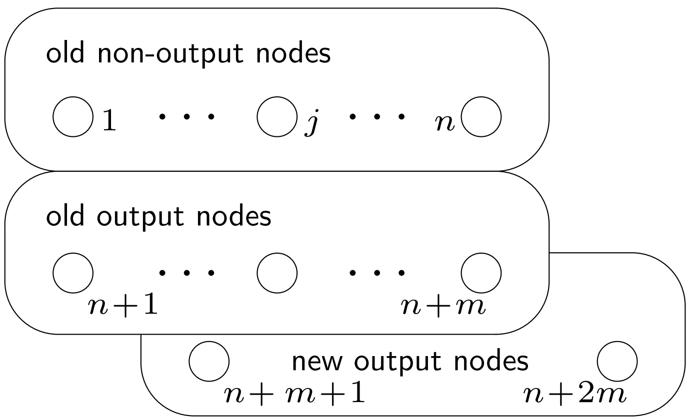
Conventional ANEP

Lemma: Every acceptance uniform network can be transformed into a conventional network that accepts the same language.



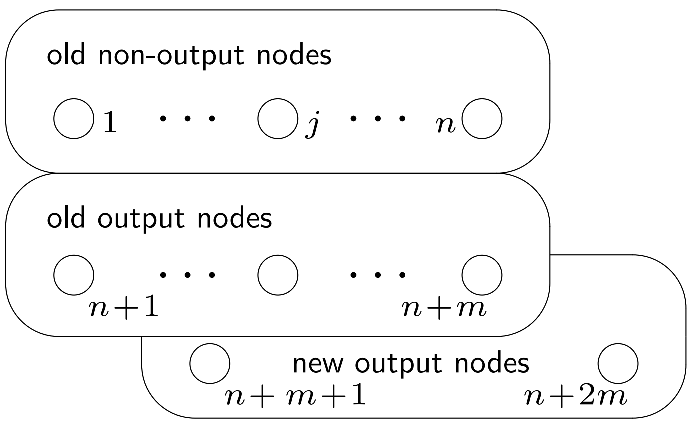
Conventional ANEP

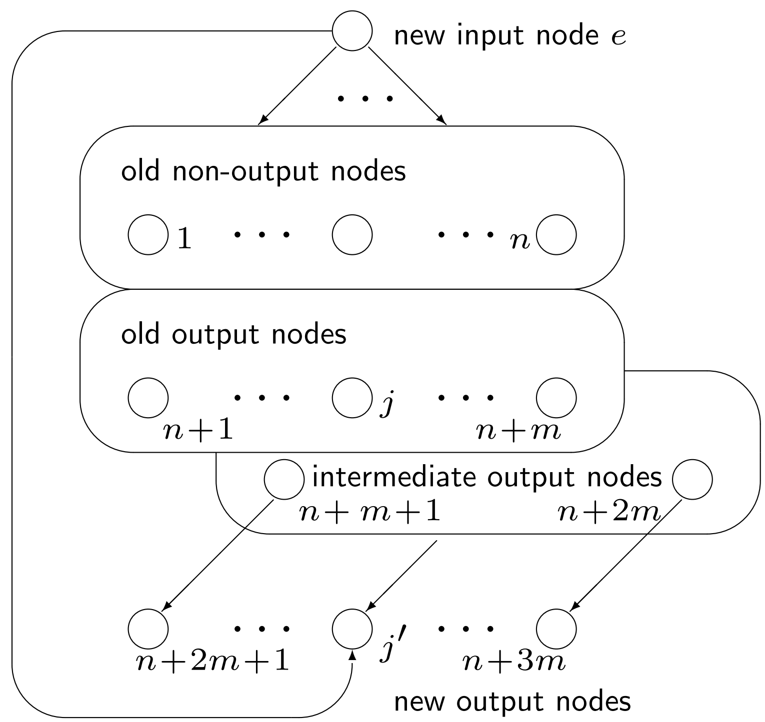
Lemma: Every acceptance uniform network can be transformed into a conventional network that accepts the same language.



Conventional ANEP

Lemma: Every acceptance uniform network can be transformed into a conventional network that accepts the same language.





Summary

Target based accepting networks and conventional ones have the same computational power.

The number of processors a target based network needs for accepting a language is not higher than the number of processors that a conventional network needs for the same language.