Descriptional Complexity of Graph-Controlled Insertion-Deletion Systems

Henning Fernau [1] Lakshmanan Kuppusamy [2]
Indhumathi Raman[3]

Fachbereich 4 - Abteilung Informatikwissenschaften, Universität Trier, Germany.

School of Computing Science and Engineering, VIT University, Vellore, India.

School of Information Technology and Engineering, VIT University, Vellore, India.

07^{th} JULY 2016

Outline of the Talk

1. Insertion-Deletion (Ins-del) Systems
2. Graph-Controlled ins-del (*GCID*) Systems with examples
3. Results connecting with computational completeness using *GCID* systems.
4. Results connecting with Linear and Metalinear using *GCID* systems
5. Conclusion and Further Research Problems

Theoretical meaning of ins-del

- Insertion (Deletion) means appending (removing) a (sub)string to (from) a given string in a specified left and right context.
- If a string α is inserted between two parts w_1 and w_2 of a string w_1w_2 to get $w_1\alpha w_2$, the operation is *insertion*.
- Notation: $(w_1, \alpha, w_2)_{ins}$: means $(w_1w_2 \Longrightarrow w_1\alpha w_2)$
- If a substring β is deleted from a string $w_1\beta w_2$ to get w_1w_2, the operation is *deletion*.
- Notation: $(w_1, \beta, w_2)_{del}$: means $(w_1\beta w_2 \Longrightarrow w_1w_2\ )$
- Suffixes of w_1 and prefixes of w_2 are called the left and right context of α or β.
- Starting with axioms and iterating the ins-del operations, we get a set of terminal strings (language of ins-del system).

Variants of ins-del system

- Ins-del P systems by Alhazov *et al* (2011)
- Tissue P systems with ins-del rules by L Kuppusamy and Raghavan R(2003)
- Context-free ins-del systems by Margenstern *et al* (2005)
- Matrix ins-del systems by L Kuppusamy and A Mahendran (2011) and independently by I Petre and S Verlan (2012)
- Graph-controlled ins-del systems by R Freund, *et al* (2010).

Common objective: To characterize the recursively enumerable languages using an ins-del system with as minimal size as possible.

Example of GCID system

- $\Pi_1 = (3, \{\#, \$, a, b\}, \{a, b\}, \{\#\$\}, \{r_{11}, r_{12}, r_{13}, r_{21}, r_{22}, r_{31}\}, 1, 1, R)$

Component 1

$r_{11} : (1, (\#, a, \lambda)_{ins}, 2)$
$r_{12} : (1, (\#, b, \lambda)_{ins}, 3)$
$r_{13} : (1, (\lambda, \$, \lambda)_{del}, 2)$

Component 2

$r_{21} : (2, (\$, a, \lambda)_{ins}, 1)$
$r_{22} : (2, (\lambda, \#, \lambda)_{del}, 1)$

Component 3

$r_{31} : (3, (\$, b, \lambda)_{ins}, 1)$

Example of GCID system

- $\Pi_1 = (3, \{\#, \$, a, b\}, \{a, b\}, \{\#\$\}, \{r_{11}, r_{12}, r_{13}, r_{21}, r_{22}, r_{31}\}, 1, 1, R)$

Component 1

$r_{11} : (1, (\#, a, \lambda)_{ins}, 2)$
$r_{12} : (1, (\#, b, \lambda)_{ins}, 3)$
$r_{13} : (1, (\lambda, \$, \lambda)_{del}, 2)$

Component 2

$r_{21} : (2, (\$, a, \lambda)_{ins}, 1)$
$r_{22} : (2, (\lambda, \#, \lambda)_{del}, 1)$

Component 3

$r_{31} : (3, (\$, b, \lambda)_{ins}, 1)$

- The generated language $L(\Pi) = \{ww \mid w \in \{a, b\}^*\}$.
- Size of Π_1 is $(3; 1, 1, 0; 1, 0, 0)$.
- Underlying graph of the system is:

Example of GCID system

- $\Pi_1 = (3, \{\#, \$, a, b\}, \{a, b\}, \{\#\$\}, \{r_{11}, r_{12}, r_{13}, r_{21}, r_{22}, r_{31}\}, 1, 1, R)$

Component 1

$r_{11} : (1, (\#, a, \lambda)_{ins}, 2)$
$r_{12} : (1, (\#, b, \lambda)_{ins}, 3)$
$r_{13} : (1, (\lambda, \$, \lambda)_{del}, 2)$

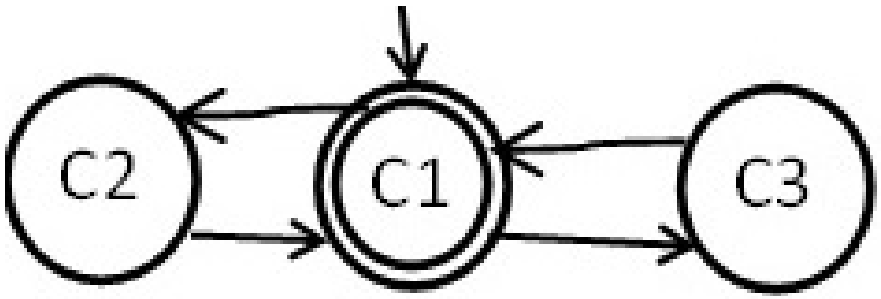
Component 2

$r_{21} : (2, (\$, a, \lambda)_{ins}, 1)$
$r_{22} : (2, (\lambda, \#, \lambda)_{del}, 1)$

Component 3

$r_{31} : (3, (\$, b, \lambda)_{ins}, 1)$

- The generated language $L(\Pi) = \{ww \mid w \in \{a, b\}^*\}$.
- Size of Π_1 is $(3; 1, 1, 0; 1, 0, 0)$.
- Underlying graph of the system is:



Another Example

- $L_2 = \{a^n b^n \mid n \geq 0\}$ is generated by:
- $\Pi_2 = (2, \{\#, \$, a, b\}, \{a, b\}, \{\#\$\}, \{r_{11}, r_{12}, r_{21}, r_{22}\}, 1, 1, R)$.

Component 1

$r_{11} : (1, (\#, a, \lambda)_{ins}, 2)$
$r_{12} : (1, (\lambda, \$, \lambda)_{del}, 2)$

Component 2

$r_{21} : (2, (\$, b, \lambda)_{ins}, 1)$
$r_{22} : (2, (\lambda, \#, \lambda)_{del}, 1)$

Another Example

- $L_2 = \{a^n b^n \mid n \geq 0\}$ is generated by:
- $\Pi_2 = (2, \{\#, \$, a, b\}, \{a, b\}, \{\#\$\}, \{r_{11}, r_{12}, r_{21}, r_{22}\}, 1, 1, R)$.

Component 1

$r_{11} : (1, (\#, a, \lambda)_{ins}, 2)$
$r_{12} : (1, (\lambda, \$, \lambda)_{del}, 2)$

Component 2

$r_{21} : (2, (\$, b, \lambda)_{ins}, 1)$
$r_{22} : (2, (\lambda, \#, \lambda)_{del}, 1)$

- Size of Π_2 is $(2; 1, 1, 0; 1, 0, 0)$.

Another Example

- $L_2 = \{a^n b^n \mid n \geq 0\}$ is generated by:
- $\Pi_2 = (2, \{\#, \$, a, b\}, \{a, b\}, \{\#\$\}, \{r_{11}, r_{12}, r_{21}, r_{22}\}, 1, 1, R)$.

Component 1

$r_{11} : (1, (\#, a, \lambda)_{ins}, 2)$
$r_{12} : (1, (\lambda, \$, \lambda)_{del}, 2)$

Component 2

$r_{21} : (2, (\$, b, \lambda)_{ins}, 1)$
$r_{22} : (2, (\lambda, \#, \lambda)_{del}, 1)$

- Size of Π_2 is $(2; 1, 1, 0; 1, 0, 0)$.
- Better simulated by $GCID(2; 1, 1, 0; 0, 0, 0)$ with axiom $\{\lambda, ab\}$ and the following rules

Component 1

$r_1 : (1, (a, a, \lambda)_{ins}, 2)$

Component 2

$r_2 : (2, (b, b, \lambda)_{ins}, 1)$

Another Example

- $L_2 = \{a^n b^n \mid n \geq 0\}$ is generated by:
- $\Pi_2 = (2, \{\#, \$, a, b\}, \{a, b\}, \{\#\$\}, \{r_{11}, r_{12}, r_{21}, r_{22}\}, 1, 1, R)$.

Component 1

$r_{11} : (1, (\#, a, \lambda)_{ins}, 2)$
$r_{12} : (1, (\lambda, \$, \lambda)_{del}, 2)$

Component 2

$r_{21} : (2, (\$, b, \lambda)_{ins}, 1)$
$r_{22} : (2, (\lambda, \#, \lambda)_{del}, 1)$

- Size of Π_2 is $(2; 1, 1, 0; 1, 0, 0)$.
- Better simulated by $GCID(2; 1, 1, 0; 0, 0, 0)$ with axiom $\{\lambda, ab\}$ and the following rules

Component 1

$r_1 : (1, (a, a, \lambda)_{ins}, 2)$

Component 2

$r_2 : (2, (b, b, \lambda)_{ins}, 1)$

Note: When $\alpha = \beta = \lambda$, then
$L(\Pi') = \{w \in \{a, b\}^* : |w|_a = |w|_b\}$

Defintion of GCID

In a GCID system, the ins-del rules are divided into components and the string move across the components.

- A graph-controlled insertion-deletion (GCID) system is

$$\Pi = (k, V, T, A, H, i_0, i_f, R)$$

- V is an alphabet, $T \subseteq V$, A is an axiom set, H is a label set.
- i_0 is the initial component and i_f is the final component.
- R is a set of k components each containing set of ins-del rules.
- A rule in R is of the form $l : (i, (w_1, \alpha, w_2)_t, j)$, $t \in \{ins, del\}$.
  - $l \in H$ is a label for the ins-del rule,
  - i denotes the current component (C_i),
  - j denotes the target component (C_j) to which the strings moves after applying the ins-del rule.
  - $L(\Pi) = \{x \in T^* : (w)_{i_0} \Rightarrow^* (x)_{i_f},\ w \in A\}$.

Size of GCID

The descriptional complexity of a GCID system is given by the size $(k; n, i', i''; m, j', j'')$ where

1. k is Number of components ($k \geq 1$)
2. n is the maximal length of the insertion string
3. i' is the maximal length of the left context used in insertion rules
4. i'' is the maximal length of the right context used in insertion rules
5. m is the maximal length of the deletion string
6. j' is the maximal length of the left context used in deletion rules
7. j'' is the maximal length of the right context used in deletion rules

Objective

- What is the generative power of the *GCID* systems w.r.t small sizes?
- What is the size required for the system to simulate RE lang.?
- A type-0 grammar $G = (N, T, P, S)$ is in SGNF if
  - N decomposes as $N = N' \cup N''$, where $N'' = \{A, B, C, D\}$ and N' contains at least the two non-terminals S and S',
  - The rules in P are of the form :
  - $X \to Yb$, $X \to bY$, $AB \to \lambda$, $CD \to \lambda$, $S' \to \lambda$. where $X, Y \in N'$, $X \neq Y$, $b \in T \cup N''$ [called Linear rules].
  - only linear rules are applied in Phase I and is completed by applying $S' \to \lambda$.
  - Then $AB \to \lambda$, $CD \to \lambda$ rules are applied.
- Can a GCID system of less size simulate weaker class of languages?

Auxiliary Result

- $GCID(k; n, i', i''; m, j', j'') = [GCID(k; n, i'', i'; m, j'', j')]^R$

Proof.

For every ins-del rule $(x, y, z)_\mu$ with $\mu \in \{ins, del\}$, we associate the reversed rule $\rho(r) = (z^R, y^R, x^R)_\mu$.
Map a rule $l: (i, r, j) \in \Pi$ to $l: (i, \rho(r), j)$ in $\rho(R)$. Define $\Pi^R = (k, V, T, A^R, H, i_0, i_f, \rho(R))$
Now, using induction, it is easy to see that

$$L(\Pi^R) = (L(\Pi))^R$$

and if $L(\Pi) = L$, then $L(\Pi)^R = L^R$. □

Further Additional Results

- Let $\mathcal{L}$ be a language class that is closed under reversal. Then,
  1. $\mathcal{L} = GCID(k; n, i', i''; m, j', j'')$ if and only if $\mathcal{L} = GCID(k; n, i'', i'; m, j'', j')$.
  2. $\mathcal{L} \subseteq GCID(k; n, i', i''; m, j', j'')$ if and only if $\mathcal{L} \subseteq GCID(k; n, i'', i'; m, j'', j')$.

Further Additional Results

- Let $\mathcal{L}$ be a language class that is closed under reversal. Then,
  1. $\mathcal{L} = GCID(k; n, i', i''; m, j', j'')$ if and only if $\mathcal{L} = GCID(k; n, i'', i'; m, j'', j')$.
  2. $\mathcal{L} \subseteq GCID(k; n, i', i''; m, j', j'')$ if and only if $\mathcal{L} \subseteq GCID(k; n, i'', i'; m, j'', j')$.
- Open problem: $\mathcal{L} = GCID(k; n, i', i''; m, j', j'')$ if and only if $\mathcal{L} = GCID(k; m, j', j''; n, i', i'')$

Seems not. At least for the cases when $m = 0$. Specifically,

$$GCID(k; n, x, y; 0, x', y') \neq GCID(k; 0, x', y'; n, x, y)$$

for $k \geq 1, n \geq 1,\ x, x', y, y' \geq 0$.

Existing vs Our Results

GCID systems of following sizes simulate the class of RE languages (are computationally complete)

Existing Results (on *RE*)

- Krassovitskiy *et al* (2011) $(4;1,1,0;2,0,0)$, $(4;2,0,0;1,1,0)$, $(4;1,1,0;1,1,0)$, $(4;1,1,0;1,0,1)$
- Ivanov & Verlan (2015) $(3;1,2,0;1,1,0)$, $(3;1,1,0;1,2,0)$
- Takahara & Yokomori (2003) $(1;1,1,1;1,1,1)$.

Our Results on *RE* (DCFS '16)

- $(3;1,1,1;1,1,0)$, $(3;1,1,1;1,0,1)$
- $(2;2,0,0;1,1,1)$

Our Results on Linear and Metalinear (DCFS '16)

GCID systems of following sizes simulate the class of (meta)linear languages

Simulating Linear languages

- $(2; 2, 1, 0; 1, 0, 0)$
- $(2; 2, 0, 1; 1, 0, 0)$
- $(3; 1, 1, 0; 1, 0, 0)$
- $(3; 1, 0, 1; 1, 0, 0)$

Simulating Metalinear languages

- $(4; 2, 1, 0; 1, 0, 0)$
- $(4; 2, 0, 1; 1, 0, 0)$
- $(5; 1, 1, 0; 1, 0, 0)$
- $(5; 1, 0, 1; 1, 0, 0)$

GCID(3;1,1,1;1,1,0) = RE

Consider the grammar $G = (N, T, P, S)$ in SGNF. Then we need to construct a GCID grammar Π s.t $L(G) \subseteq L(\Pi)$. The rules in P are injectively labelled from $[1 \ldots |P|]$.

GCID(3;1,1,1;1,1,0) = RE

Consider the grammar $G = (N, T, P, S)$ in SGNF. Then we need to construct a GCID grammar Π s.t $L(G) \subseteq L(\Pi)$. The rules in P are injectively labelled from $[1 \ldots |P|]$.

Simulate the rule p: $X \to bY$ by the following GCID rules:

Component 1

$p1.1 : (1, (\lambda, p, X)_{ins}, 2)$

Component 2

$p2.1 : (2, (\lambda, b, p)_{ins}, 3)$
$p2.2 : (2, (Y, X, \lambda)_{del}, 1)$

Component 3

$p3.1 : (3, (b, Y, p)_{ins}, 3)$
$p3.2 : (3, (Y, p, \lambda)_{del}, 2)$

GCID(3;1,1,1;1,1,0) = RE

Consider the grammar $G = (N, T, P, S)$ in SGNF. Then we need to construct a GCID grammar Π s.t $L(G) \subseteq L(\Pi)$. The rules in P are injectively labelled from $[1 \ldots |P|]$.

Simulate the rule p: $X \to bY$ by the following GCID rules:

Component 1

$p1.1 : (1, (\lambda, p, X)_{ins}, 2)$

Component 2

$p2.1 : (2, (\lambda, b, p)_{ins}, 3)$
$p2.2 : (2, (Y, X, \lambda)_{del}, 1)$

Component 3

$p3.1 : (3, (b, Y, p)_{ins}, 3)$
$p3.2 : (3, (Y, p, \lambda)_{del}, 2)$

$$(\alpha X \beta)_1 \Longrightarrow_{p1.1} (\alpha p X \beta)_2 \Longrightarrow_{p2.1} (\alpha b p X \beta)_3 \Longrightarrow_{p3.1} (\alpha b Y p X \beta)_3$$
$$\Longrightarrow_{p3.2} (\alpha b Y X \beta)_2 \Longrightarrow_{p2.2} (\alpha b Y \beta)_1.$$

GCID(3;1,1,1;1,1,0) = RE

Consider the grammar $G = (N, T, P, S)$ in SGNF. Then we need to construct a GCID grammar Π s.t $L(G) \subseteq L(\Pi)$. The rules in P are injectively labelled from $[1 \ldots |P|]$.

Simulate the rule p: $X \to bY$ by the following GCID rules:

Component 1

$p1.1 : (1, (\lambda, p, X)_{ins}, 2)$

Component 2

$p2.1 : (2, (\lambda, b, p)_{ins}, 3)$
$p2.2 : (2, (Y, X, \lambda)_{del}, 1)$

Component 3

$p3.1 : (3, (b, Y, p)_{ins}, 3)$
$p3.2 : (3, (Y, p, \lambda)_{del}, 2)$

$$(\alpha X \beta)_1 \Longrightarrow_{p1.1} (\alpha p X \beta)_2 \Longrightarrow_{p2.1} (\alpha b p X \beta)_3 \Longrightarrow_{p3.1} (\alpha b Y p X \beta)_3$$
$$\Longrightarrow_{p3.2} (\alpha b Y X \beta)_2 \Longrightarrow_{p2.2} (\alpha b Y \beta)_1.$$

Note: $p2.1 > p2.2$ and $p3.1 > p3.2$.

GCID(3;1,1,1;1,1,0) = RE

Simulate the rule q: $X \to Yb$ by the following GCID rules:

Component 1

$q1.1 : (1, (\lambda, q, X)_{ins}, 2)$

Component 2

$q2.1 : (2, (\lambda, Y, q)_{ins}, 3)$
$q2.2 : (2, (\lambda, q, \lambda)_{del}, 1)$

Component 3

$q3.1 : (3, (q, b, X)_{ins}, 3)$
$q3.2 : (3, (b, X, \lambda)_{del}, 2)$

- Working is more like the p rule.
- $q2.1 = q2.2$ and $q3.1 > q.3.2$.
- There is only one $X \in N'$ in the derivation at any point of time. Hence the rule $q3.2$ is used uniquely.
- This guarantees that the correct X is deleted.

GCID(3;1,1,1;1,1,0) = RE

Simulate the rule f: $AB \to \lambda$ by the following GCID rules:

Component 1

$f1.1: \ (1, (\lambda, f, A)_{ins}, 2)$

Component 2

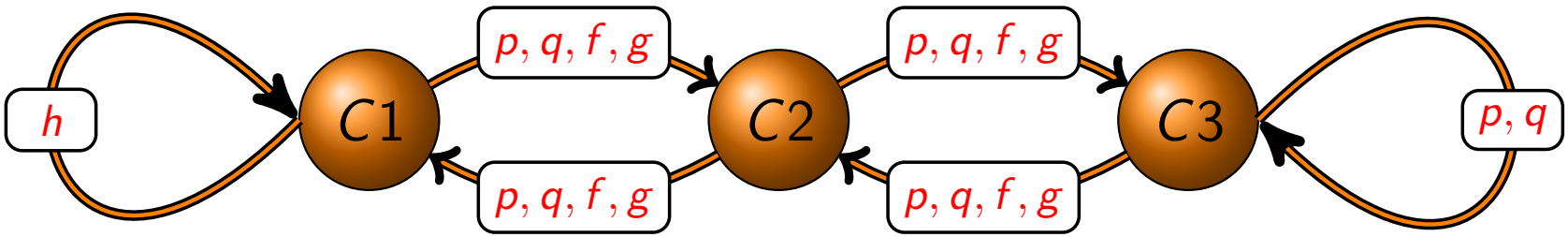
$f2.1: \ (2, (\lambda, f, \lambda)_{del}, 1)$
$f2.2: \ (2, (f, A, \lambda)_{del}, 3)$

Component 3

$f3.1: \ (3, (f, B, \lambda)_{del}, 2)$

$$(\alpha AB\beta)_1 \underset{\Longleftarrow f2.1}{\overset{\Longrightarrow f1.1}{}} (\alpha fAB\beta)_2 \Longrightarrow_{f2.2} (\alpha fB\beta)_3 \Longrightarrow_{f3.1}$$
$$(\alpha f\beta)_2 \Longrightarrow_{f2.1} (\alpha\lambda\beta)_1$$

Simulate the rule g: $CD \to \lambda$ by a similar (to f) set of GCID rules.
Simulate the rule $h: \ S' \to \lambda$ by $h1.1: (1, (\lambda, S', \lambda)_{del}, 1)$.


h
C1
p, q, f, g
p, q, f, g
C2
p, q, f, g
p, q, f, g
C3
p, q

GCID(2;2,0,0;1,1,1) = RE

- We simulate the rules $p : X \to bY$, $q : X \to Yb$, $f : AB \to \lambda$ by the following GCID rules:

Component 1

$p1.1 : (1, (\lambda, bY, \lambda)_{ins}, 2)$
$q1.1 : (1, (\lambda, Yb, \lambda)_{ins}, 2)$
$f1.1 : (1, (\lambda, ff', \lambda)_{ins}, 2)$
$f1.2 : (1, (A, f, f')_{del}, 1)$
$f1.3 : (1, (\lambda, A, f')_{del}, 2)$

Component 2

$p2.1 : (2, (Y, X, \lambda)_{del}, 1)$
$q2.1 : (2, (\lambda, X, Y)_{del}, 1)$
$f2.1 : (2, (f', B, \lambda)_{del}, 1)$
$f2.2 : (2, (\lambda, f', \lambda)_{del}, 1)$

- Simulation of $g : CD \to \lambda$ is similar to f rule.
- The rule $h : S' \to \lambda$ is simulated by $h1.1 : (1, (\lambda, S', \lambda)_{del}, 1)$
- In the f rule, what happens if multiple ff' are inserted? Then, nonterminals will remain in the derivation.
- multiple insertions do not correspond to multiple deletions.

$LIN \subsetneq GCID(2;2,1,0;1,0,0)$

- We simulate the rules $p : X \to aY$, $q : X \to Ya$, $f : X \to a$ by the following GCID rules:

Component 1

$p1.1 : (1, (X, aY, \lambda)_{ins}, 2)$
$q1.1 : (1, (X, Ya, \lambda)_{ins}, 2)$
$f1.1 : (1, (X, a, \lambda)_{ins}, 2)$

Component 2

$p2.1 : (2, (\lambda, X, \lambda)_{del}, 1)$
$q2.1 : (2, (\lambda, X, \lambda)_{del}, 1)$
$f2.1 : (2, (\lambda, X, \lambda)_{del}, 1)$

- Is the inclusion strict?
- (Smaller!) $GCID(2;1,0,0;0,0,0)$ generates the non-linear language $\{|w|_a = |w|_b : w \in \{a,b\}^*\}$ with axiom λ and

Component 1

$r1 : (1, (\lambda, a, \lambda)_{ins}, 2)$

Component 2

$r2 : (2, (\lambda, b, \lambda)_{ins}, 1)$

Metalinear Languages

- Smallest class containing linear languages and closed under concatenation. Note: LIN is not closed under concatenation.
- $MLIN = L_1 L_2 \dots L_k,\ k \geq 1, L_i \in LIN$
- $L(G) = L(G_1)L(G_2)\dots L(G_k)$ where G_i is a linear grammar.
- Axiom of G is $S \rightarrow S_1 S_2'$.
- Conventional LIN rules of G:
  - $X \rightarrow Ya$, $X \rightarrow aY$, $X \rightarrow a$
- Additional CF rules of G:
  - $S_i' \rightarrow S_i S_{i+1}'$ for $2 \leq i \leq k-1$
  - $S_k' \rightarrow S_k$.

Simulating the Transitions

- Start with axiom $S_1 S_2'$.
- Starting with S_1, simulate the rules of G_1 using the rules of $GCID(2; 2, 1, 0; 1, 0, 0)$.
- As G_1 generates a terminal word using the rule $f2.1$, transit to $C3$ instead of $C1$ i.e., $f2.1' : (2, (\lambda, X, \lambda)_{del}, 3)$.
- There is ut most only one $X \in G_i$ is present in the derivation.
- From $C3$, we simulate the rule $S_2' \to S_2 S_3'$ as follows:
  - $(3, (S_2', r_{1\to 2}, \lambda)_{ins}, 4)$, $(3, (r_{1\to 2}, S_2 S_3', \lambda)_{ins}, 2)$
  - $(4, (\lambda, S_2', \lambda)_{del}, 3)$
  - $(2, (\lambda, r_{1\to 2}, \lambda)_{del}, 1)$ (new rule in $C2$)
- Starting at $S_2 S_3'$, we generate L_2 and proceed to generate L_3 using $S_3' \to S_3 S_4'$.

Process continues...

- The process continues for $k-1$ times and at last L_k is generated. Above rules are deterministic.
- In addition to simulating LIN using $GCID(2;2,1,0;1,0,0)$, we have used two extra components to simulate metalinear languages. Therefore, $MLIN \subseteq GCID(4;2,1,0;1,0,0)$.

Process continues...

- The process continues for $k-1$ times and at last L_k is generated. Above rules are deterministic.
- In addition to simulating LIN using $GCID(2;2,1,0;1,0,0)$, we have used two extra components to simulate metalinear languages. Therefore, $MLIN \subseteq GCID(4;2,1,0;1,0,0)$.
- We can also show that $MLIN \subseteq GCID(3;2,1,0;1,0,1)$ using the following rules:
  - $(3, S_2', r_{1\to 2}, \lambda)_{ins}, 2)$ and $(3, (r_{1\to 2}, S_2S_3', \lambda)_{ins}, 2)$
  - $(2, (\lambda, S_2', r_{1\to 2})_{del}, 3)$ and $(2, (\lambda, r_{1\to 2}, \lambda)_{del}, 1)$.
  - The other details remain same.

Strictness of $MLIN \subsetneq GCID(4;2,1,0;1,0,0)$

- Is the inclusion strict?
- (Smaller!) $GCID(3;1,1,0;1,0,0)$ generates the non-metalinear language $\{a^n b^n : n \geq 0\}^*$ with axiom SS' and

Component 1

$r1.1 : (1,(S,a,\lambda)_{ins},2)$
$r1.2 : (1,(\lambda,S',\lambda)_{del},3)$

Component 2

$r2.1 : (2,(S',b,\lambda)_{ins},1)$

Component 3

$r3.1 : (3,(S,S',\lambda)_{ins},1)$
$r3.2 : (3,(\lambda,S,\lambda)_{del},1)$

- $MLIN \subsetneq GCID(4;2,1,0;1,0,0) \cap GCID(3;2,1,0;1,0,1)$.

$LIN \subsetneq GCID(3;1,1,0;1,0,0)$

We simulate the linear rules $p : X \to aY$, $q : X \to Ya$ and $f : X \to a$ by the following GCID rules:

Component 1

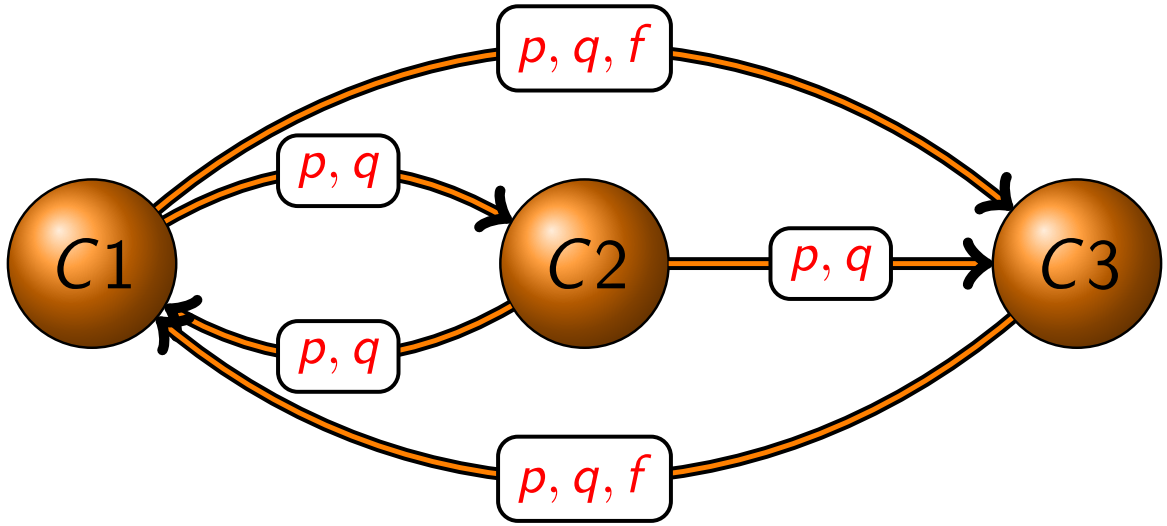
$p1.1 : (1, (X, p, \lambda)_{ins}, 3)$
$p1.2: (1, (p, a, \lambda)_{ins}, 2)$
$p1.3: (1, (p', Y, \lambda)_{ins}, 2)$
$q1.1 : (1, (X, q, \lambda)_{ins}, 3)$
$q1.2 : (1, (q, q', \lambda)_{ins}, 2)$
$q1.3 : (1, (q', Y, \lambda)_{ins}, 2)$
$f1.1 : (1, (X, a, \lambda)_{ins}, 3)$

Component 2

$p2.1 : (2, (p, p', \lambda)_{ins}, 3)$
$p2.2 : (2, (\lambda, p', \lambda)_{del}, 1)$
$q2.1 : (2, (q, a, \lambda)_{ins}, 3)$
$q2.2 : (2, (\lambda, q', \lambda)_{del}, 1)$

Component 3

$p3.1 : (3, (\lambda, X, \lambda)_{del}, 1)$
$p3.2 : (3, (\lambda, p, \lambda)_{del}, 1)$
$q3.1 : (3, (\lambda, X, \lambda)_{del}, 1)$
$q3.2 : (3, (\lambda, q, \lambda)_{del}, 1)$
$f3.1 : (3, (\lambda, X, \lambda)_{del}, 1)$


p, q, f
p, q
C1
C2
p, q
C3
p, q
p, q, f

More *LIN* and *MLIN* Results

As *LIN* is known to be closed under reversal, we have:

Theorem

$LIN \subsetneq GCID(3;1,0,1;1,0,0)$.

More *LIN* and *MLIN* Results

As *LIN* is known to be closed under reversal, we have:

Theorem

$LIN \subsetneq GCID(3;1,0,1;1,0,0)$.

Inheriting the proof idea of *MLIN*, We can also prove that

Theorem

$MLIN \in GCID(5;1,1,0;1,0,0) \cap GCID(5;1,0,1;1,0,0)$.

Conclusion

- We have considered the generative power of small sizes of *GCID* systems and proved results connecting with *RE*, *LIN* and *MLIN* with small sizes.
- Fixing the component size (we are interested with 3 components) and making other ins-del measures to be 0 or 1, we can have 64 different classes of *GCID* system.

  Some Open Questions:

Conclusion

- We have considered the generative power of small sizes of *GCID* systems and proved results connecting with *RE*, *LIN* and *MLIN* with small sizes.
- Fixing the component size (we are interested with 3 components) and making other ins-del measures to be 0 or 1, we can have 64 different classes of *GCID* system.

  Some Open Questions:

  1. What is their power and which classes of systems are equivalent/not-equivalent?
  2. what are the results known to us? and what is open?
  3. What are the canonical results to be proved so that the results follow immediately?

Conclusion Contd.

- Canonical result:
  Say, proving $GCID(3;1,0,0;1,1,1) = RE$ means

$$GCID(3;1,1,0;1,1,1) = RE = GCID(3;1,1,1;1,1,1)$$

- Some immediate conclusions that we can make are for $k \geq 1; x, y, x', y', m \geq 0$.

$$(i)FIN = GCID(k;0,x,y;m,x',y')$$

and

$$FIN \subsetneq GCID(k;1,x,y;m,x',y')$$

as $\{a^n \mid n \geq 0\} \in GCID(1;1,0,0;0,0,0)\ [(\lambda, a, \lambda)_{ins}, \Sigma = \{a\}]$

My special thanks to

1. Prof. Dr. Henning Fernau to support for my travel to/till Germany (and visit to Trier).
2. Prof. Dr. Radu Gramatovici for his hospitality support and all.

My special thanks to

1. Prof. Dr. Henning Fernau to support for my travel to/till Germany (and visit to Trier).
2. Prof. Dr. Radu Gramatovici for his hospitality support and all.

THANK YOU ALL

My special thanks to

1. Prof. Dr. Henning Fernau to support for my travel to/till Germany (and visit to Trier).
2. Prof. Dr. Radu Gramatovici for his hospitality support and all.

THANK YOU ALL

Any questions ??