Strictly Locally Testable and Resources Restricted Control Languages in Tree-Controlled Grammars

Bianca Truthe
Justus-Liebig-Universität Giessen

Theorietag, Kaiserslautern, October 4 – 6, 2023

Introduction

- In the area of formal languages and automata theory, regular languages and finite automata are widely studied.

Introduction

- In the area of formal languages and automata theory, regular languages and finite automata are widely studied.
- Several classes of specific finite automata and their accepted languages have been investigated separately.

Introduction

- In the area of formal languages and automata theory, regular languages and finite automata are widely studied.
- Several classes of specific finite automata and their accepted languages have been investigated separately.
- Many papers have been published by different authors on subregular families of languages.

Introduction

- In the area of formal languages and automata theory, regular languages and finite automata are widely studied.
- Several classes of specific finite automata and their accepted languages have been investigated separately.
- Many papers have been published by different authors on subregular families of languages.
- Focus is often on the decrease of descriptional or computational complexity when going from arbitrary regular languages to special ones.

Introduction

- In the area of formal languages and automata theory, regular languages and finite automata are widely studied.
- Several classes of specific finite automata and their accepted languages have been investigated separately.
- Many papers have been published by different authors on subregular families of languages.
- Focus is often on the decrease of descriptional or computational complexity when going from arbitrary regular languages to special ones.
- Here, the generative capacity of tree-controlled grammars with special regular control languages is considered.

Tree-Controlled Grammars

Tree-Controlled Grammars

- introduced by Karel Čulik II and Hermann A. Maurer in 1977
- formal model for generating languages,
- enhance a context-free grammar by a mechanism to control the derivation,
- words on each level of the derivation tree must belong to a given regular language.

Karel Čulik II and Hermann A. Maurer: Tree controlled grammars.
Computing 19 (1977), 129–139.

Tree-Controlled Grammars

Tree-Controlled Grammars

- introduced by Karel Čulik II and Hermann A. Maurer in 1977
- formal model for generating languages,
- enhance a context-free grammar by a mechanism to control the derivation,
- words on each level of the derivation tree must belong to a given regular language.

Gheorghe Păun (1979): $\mathcal{TC}(REG) = CS$ and $\mathcal{TC}_\lambda(REG) = RE$

Karel Čulik II and Hermann A. Maurer: Tree controlled grammars.
Computing 19 (1977), 129–139.

Gheorghe Păun: On the generative capacity of tree controlled grammars.
Computing 21 (1979), 213–220.

Tree-Controlled Grammars

Tree-Controlled Grammars

– introduced by Karel Čulik II and Hermann A. Maurer in 1977
– formal model for generating languages,
– enhance a context-free grammar by a mechanism to control the derivation,
– words on each level of the derivation tree must belong to a given regular language.

Gheorghe Păun (1979): $\mathcal{TC}(REG) = CS$ and $\mathcal{TC}_\lambda(REG) = RE$

Search for families X with $CF \subset \mathcal{TC}(X) \subset CS$

Karel Čulik II and Hermann A. Maurer: Tree controlled grammars.
Computing 19 (1977), 129–139.

Gheorghe Păun: On the generative capacity of tree controlled grammars.
Computing 21 (1979), 213–220.

Definitions

Tree-controlled grammar with control in $\mathcal{F}$: $G = (N, T, P, S, R)$ where

- (N, T, P, S) is a context-free grammar with
  - a set N of non-terminal symbols,
  - a set T of terminal symbols,
  - a set P of context-free non-erasing rules (except $S \to \lambda$),
  - an axiom $S \in N$,
- a regular set $R \in \mathcal{F}$ over $N \cup T$.

Definitions

Tree-controlled grammar with control in $\mathcal{F}$: $G = (N, T, P, S, R)$ where

- (N, T, P, S) is a context-free grammar with
  - a set N of non-terminal symbols,
  - a set T of terminal symbols,
  - a set P of context-free non-erasing rules (except $S \to \lambda$),
  - an axiom $S \in N$,
- a regular set $R \in \mathcal{F}$ over $N \cup T$.

Generated language:

$$L(G) = \{\ z \in T^* \mid z \text{ has a derivation tree } t \text{ where all but the last level belong to } R\ \}$$

Definitions

Tree-controlled grammar with control in $\mathcal{F}$: $G = (N, T, P, S, R)$ where

- (N, T, P, S) is a context-free grammar with
  - a set N of non-terminal symbols,
  - a set T of terminal symbols,
  - a set P of context-free non-erasing rules (except $S \to \lambda$),
  - an axiom $S \in N$,
- a regular set $R \in \mathcal{F}$ over $N \cup T$.

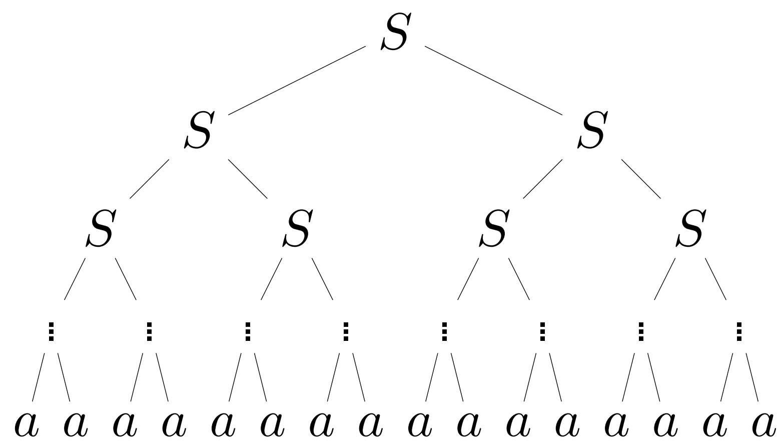
Generated language:

$$L(G) = \{\, z \in T^* \mid z \text{ has a derivation tree } t \text{ where all but the last level belong to } R \,\}$$

Language family: $\mathcal{TC}(\mathcal{F}) = \{\, L(G) \mid G = (N, T, P, S, R),\ R \in \mathcal{F} \,\}$

Example

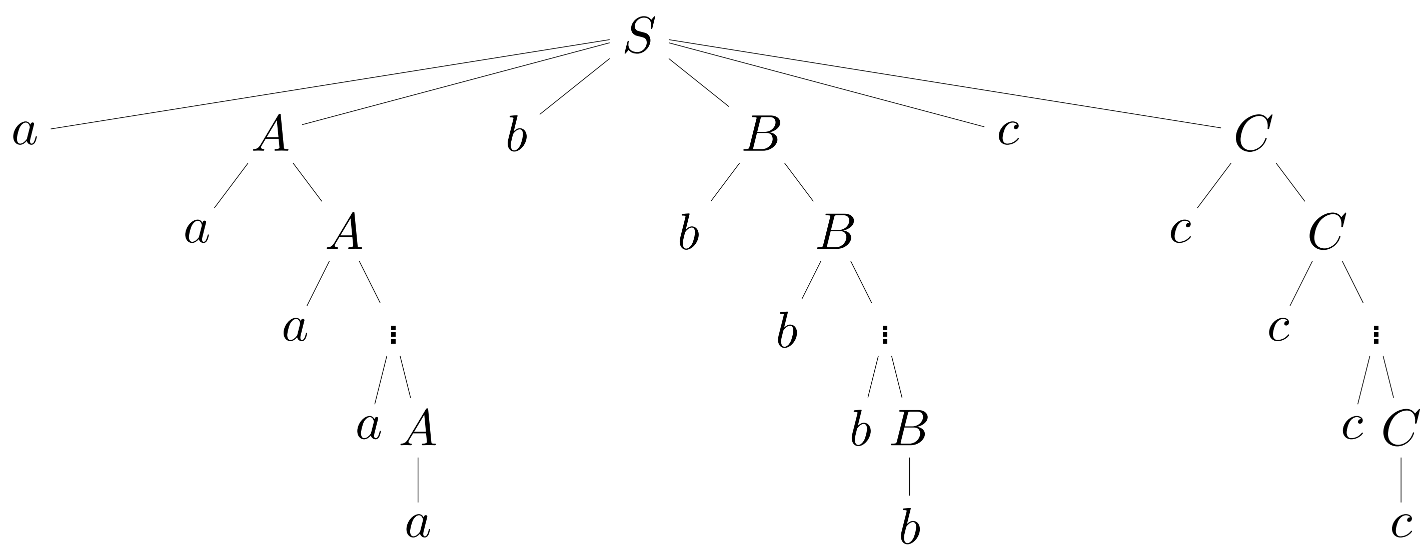
$$G_1 = (\{S\}, \{a\}, \{S \to SS, S \to a\}, S, \{S\}^*)$$



$$\rightsquigarrow L(G_1) = \{\, a^{2^n} \mid n \geq 0 \,\}$$

Another Example

$G_2 = (\{S, A, B, C\}, \{a, b, c\}, P, S, \{S, aAbBcC\})$ with
$P = \{S \to aAbBcC, A \to aA, B \to bB, C \to cC, A \to a, B \to b, C \to c\}$



$\rightsquigarrow L(G_2) = \{ a^n b^n c^n \mid n \geq 2 \}$

Subregular Language Families

Let L be a language over an alphabet V.
We say that the language L – with respect to the alphabet V – is

Subregular Language Families

Let L be a language over an alphabet V.
We say that the language L – with respect to the alphabet V – is

- *monoidal* if and only if $L = V^*$,

Subregular Language Families

Let L be a language over an alphabet V.
We say that the language L – with respect to the alphabet V – is

- *monoidal* if and only if $L = V^*$,
- *nilpotent* if and only if it is finite or its complement $V^* \setminus L$ is finite,

Subregular Language Families

Let L be a language over an alphabet V.
We say that the language L – with respect to the alphabet V – is

- *monoidal* if and only if $L = V^*$,
- *nilpotent* if and only if it is finite or its complement $V^* \setminus L$ is finite,
- *combinational* if and only if it has the form $L = V^*A$ for some subset $A \subseteq V$,

Subregular Language Families

Let L be a language over an alphabet V.
We say that the language L – with respect to the alphabet V – is

- *monoidal* if and only if $L = V^*$,
- *nilpotent* if and only if it is finite or its complement $V^* \setminus L$ is finite,
- *combinational* if and only if it has the form $L = V^*A$ for some subset $A \subseteq V$,
- *definite* if and only if it can be represented in the form $L = A \cup V^*B$ where A and B are finite subsets of V^*,

Subregular Language Families

Let L be a language over an alphabet V.
We say that the language L – with respect to the alphabet V – is

- *monoidal* if and only if $L = V^*$,
- *nilpotent* if and only if it is finite or its complement $V^* \setminus L$ is finite,
- *combinational* if and only if it has the form $L = V^*A$ for some subset $A \subseteq V$,
- *definite* if and only if it can be represented in the form $L = A \cup V^*B$ where A and B are finite subsets of V^*,
- *suffix-closed* (or *fully initial* or *multiple-entry* language) if and only if, for any words $x \in V^*$ and $y \in V^*$, the relation $xy \in L$ implies $y \in L$,

Subregular Language Families

Let L be a language over an alphabet V.
We say that the language L – with respect to the alphabet V – is

- *ordered* if and only if the language is accepted by some deterministic finite automaton $\mathcal{A} = (V, Z, z_0, F, \delta)$ where $(Z, \preceq)$ is a totally ordered set and, for any $a \in V$, the relation $z \preceq z'$ implies $\delta(z, a) \preceq \delta(z', a)$,

Subregular Language Families

Let L be a language over an alphabet V.
We say that the language L – with respect to the alphabet V – is

- *ordered* if and only if the language is accepted by some deterministic finite automaton $\mathcal{A} = (V, Z, z_0, F, \delta)$ where $(Z, \preceq)$ is a totally ordered set and, for any $a \in V$, the relation $z \preceq z'$ implies $\delta(z, a) \preceq \delta(z', a)$,
- *commutative* if and only if it contains with each word also all permutations of this word,

Subregular Language Families

Let L be a language over an alphabet V.
We say that the language L – with respect to the alphabet V – is

- *ordered* if and only if the language is accepted by some deterministic finite automaton $\mathcal{A} = (V, Z, z_0, F, \delta)$ where $(Z, \preceq)$ is a totally ordered set and, for any $a \in V$, the relation $z \preceq z'$ implies $\delta(z, a) \preceq \delta(z', a)$,
- *commutative* if and only if it contains with each word also all permutations of this word,
- *circular* if and only if it contains with each word also all circular shifts of this word,

Subregular Language Families

Let L be a language over an alphabet V.
We say that the language L – with respect to the alphabet V – is

- *ordered* if and only if the language is accepted by some deterministic finite automaton $\mathcal{A} = (V, Z, z_0, F, \delta)$ where $(Z, \preceq)$ is a totally ordered set and, for any $a \in V$, the relation $z \preceq z'$ implies $\delta(z, a) \preceq \delta(z', a)$,
- *commutative* if and only if it contains with each word also all permutations of this word,
- *circular* if and only if it contains with each word also all circular shifts of this word,
- *non-counting* (or *star-free*) if and only if there is a natural number $k \geq 1$ such that, for any words $x \in V^*$, $y \in V^*$, and $z \in V^*$, it holds $xy^kz \in L$ if and only if $xy^{k+1}z \in L$,

Subregular Language Families

Let L be a language over an alphabet V.
We say that the language L – with respect to the alphabet V – is

- *power-separating* if and only if, there is a natural number $m \geq 1$ such that for any $x \in V^*$, either $J_x^m \cap L = \emptyset$ or $J_x^m \subseteq L$ where $J_x^m = \{x^n \mid n \geq m\}$,

Subregular Language Families

Let L be a language over an alphabet V.
We say that the language L – with respect to the alphabet V – is

- *power-separating* if and only if, there is a natural number $m \geq 1$ such that for any $x \in V^*$, either $J_x^m \cap L = \emptyset$ or $J_x^m \subseteq L$ where $J_x^m = \{x^n \mid n \geq m\}$,
- *union-free* if and only if L can be described by a regular expression which is only built by product and star,

Subregular Language Families

Let L be a language over an alphabet V.
We say that the language L – with respect to the alphabet V – is

- *power-separating* if and only if, there is a natural number $m \geq 1$ such that for any $x \in V^*$, either $J_x^m \cap L = \emptyset$ or $J_x^m \subseteq L$ where $J_x^m = \{x^n \mid n \geq m\}$,
- *union-free* if and only if L can be described by a regular expression which is only built by product and star,
- *strictly locally k-testable* if and only if there are three subsets B, I, and E of V^k such that any word $a_1a_2\ldots a_n$ with $n \geq k$ and $a_i \in V$ for $1 \leq i \leq n$ belongs to the language L if and only if $a_1a_2\ldots a_k \in B$, $a_{j+1}a_{j+2}\ldots a_{j+k} \in I$ $(1 \leq j \leq n-k-1)$, and $a_{n-k+1}a_{n-k+2}\ldots a_n \in E$,

Subregular Language Families

Let L be a language over an alphabet V.
We say that the language L – with respect to the alphabet V – is

- *power-separating* if and only if, there is a natural number $m \geq 1$ such that for any $x \in V^*$, either $J_x^m \cap L = \emptyset$ or $J_x^m \subseteq L$ where $J_x^m = \{x^n \mid n \geq m\}$,
- *union-free* if and only if L can be described by a regular expression which is only built by product and star,
- *strictly locally k-testable* if and only if there are three subsets B, I, and E of V^k such that any word $a_1 a_2 \ldots a_n$ with $n \geq k$ and $a_i \in V$ for $1 \leq i \leq n$ belongs to the language L if and only if $a_1 a_2 \ldots a_k \in B$, $a_{j+1} a_{j+2} \ldots a_{j+k} \in I$ $(1 \leq j \leq n-k-1)$, and $a_{n-k+1} a_{n-k+2} \ldots a_n \in E$,
- *strictly locally testable* if and only if it is strictly locally k-testable for some natural number k.

Further Subregular Language Families

$$REG_n^Z = \{ L \mid L \in REG \text{ with } State(L) \leq n \},$$

where

$$State(L) = \min \{ State(A) \mid A \text{ is a det. finite automaton accepting } L \},$$

with

$$State(A) = |Z|$$

Further Subregular Language Families

$$REG_n^Z = \{ L \mid L \in REG \text{ with } State(L) \leq n \},$$
$$RL_n^V = \{ L \mid L \in REG \text{ with } Var_{RL}(L) \leq n \},$$

where

$$State(L) = \min \{ State(A) \mid A \text{ is a det. finite automaton accepting } L \},$$
$$Var_{RL}(L) = \min \{ Var(G) \mid G \text{ is a right-linear grammar generating } L \},$$

with

$State(A) = |Z|$, $Var(G) = |N|$

Further Subregular Language Families

$$REG_n^Z = \{ L \mid L \in REG \text{ with } State(L) \leq n \},$$
$$RL_n^V = \{ L \mid L \in REG \text{ with } Var_{RL}(L) \leq n \},$$
$$RL_n^P = \{ L \mid L \in REG \text{ with } Prod_{RL}(L) \leq n \},$$

where

$$State(L) = \min \{ State(A) \mid A \text{ is a det. finite automaton accepting } L \},$$
$$Var_{RL}(L) = \min \{ Var(G) \mid G \text{ is a right-linear grammar generating } L \},$$
$$Prod_{RL}(L) = \min \{ Prod(G) \mid G \text{ is a right-linear grammar generating } L \}.$$

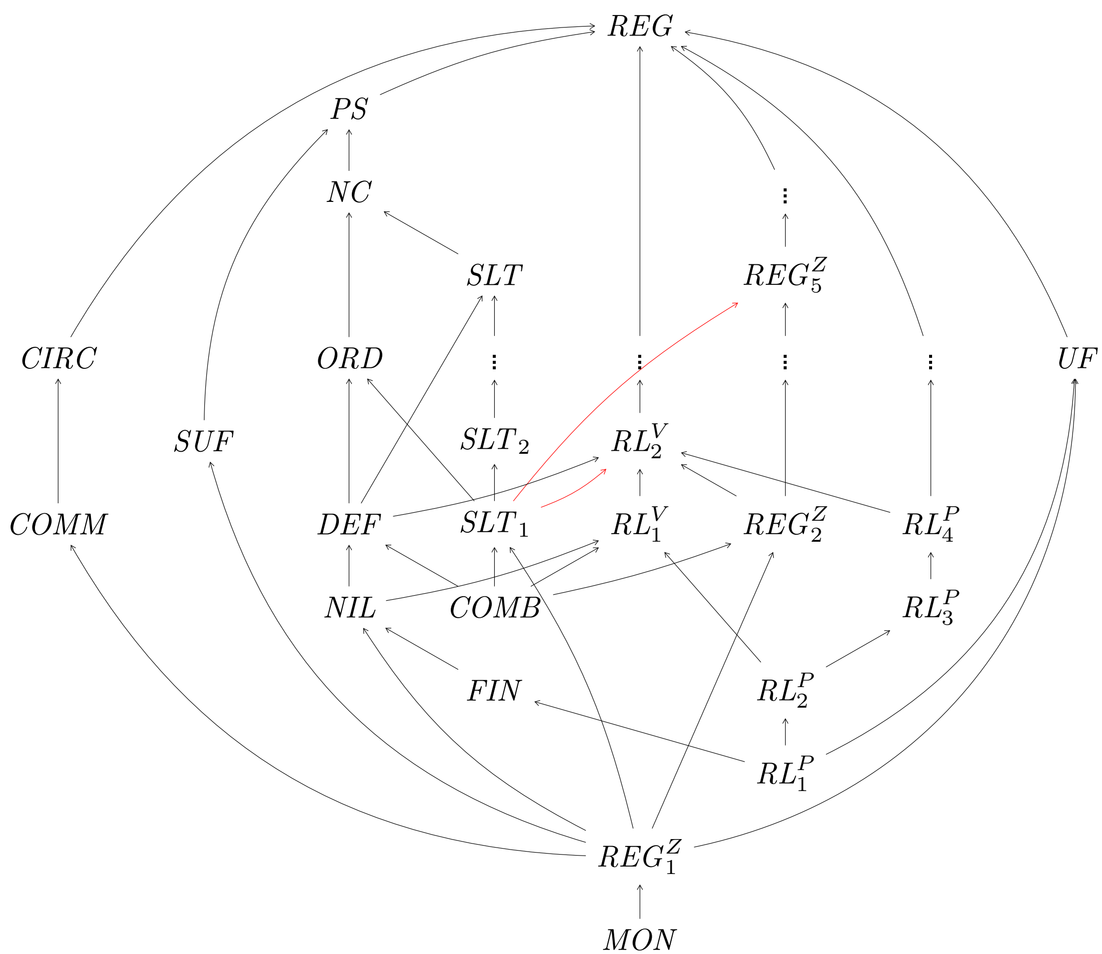
with

$$State(A) = |Z|, Var(G) = |N|, Prod(G) = |P|.$$

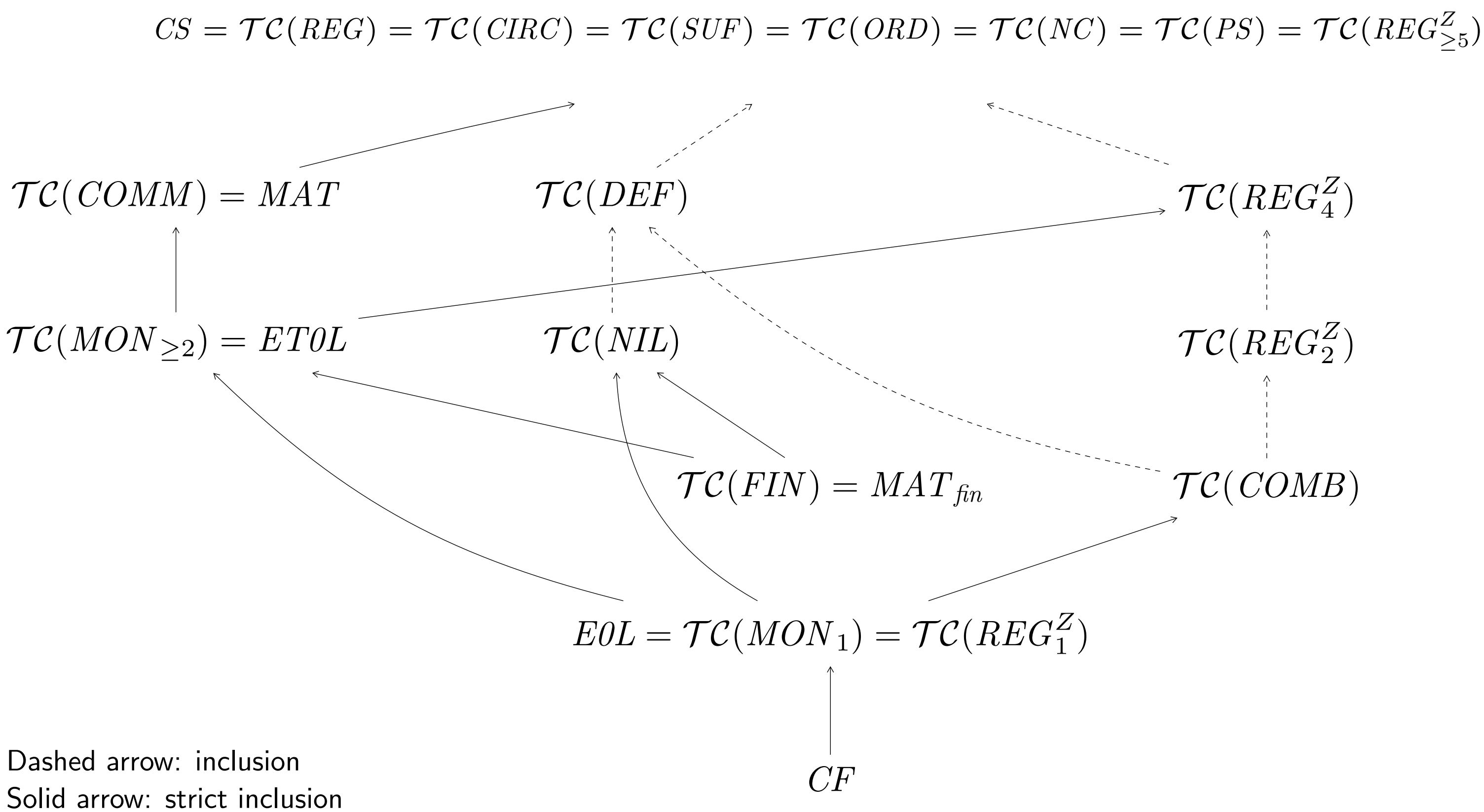
Families Under Consideration

$$\mathcal{F} \in \{FIN, MON, NIL, COMB, DEF, SUF, ORD\}$$
$$\cup \{COMM, CIRC, NC, PS, UF, REG\}$$
$$\cup \{\, SLT_k \mid k \geq 1 \,\} \cup \{SLT\}$$
$$\cup \{\, REG_n^Z \mid n \geq 1 \,\} \cup \{\, RL_n^V \mid n \geq 1 \,\} \cup \{\, RL_n^P \mid n \geq 1 \,\}$$

Hierarchy of Subregular Language Families

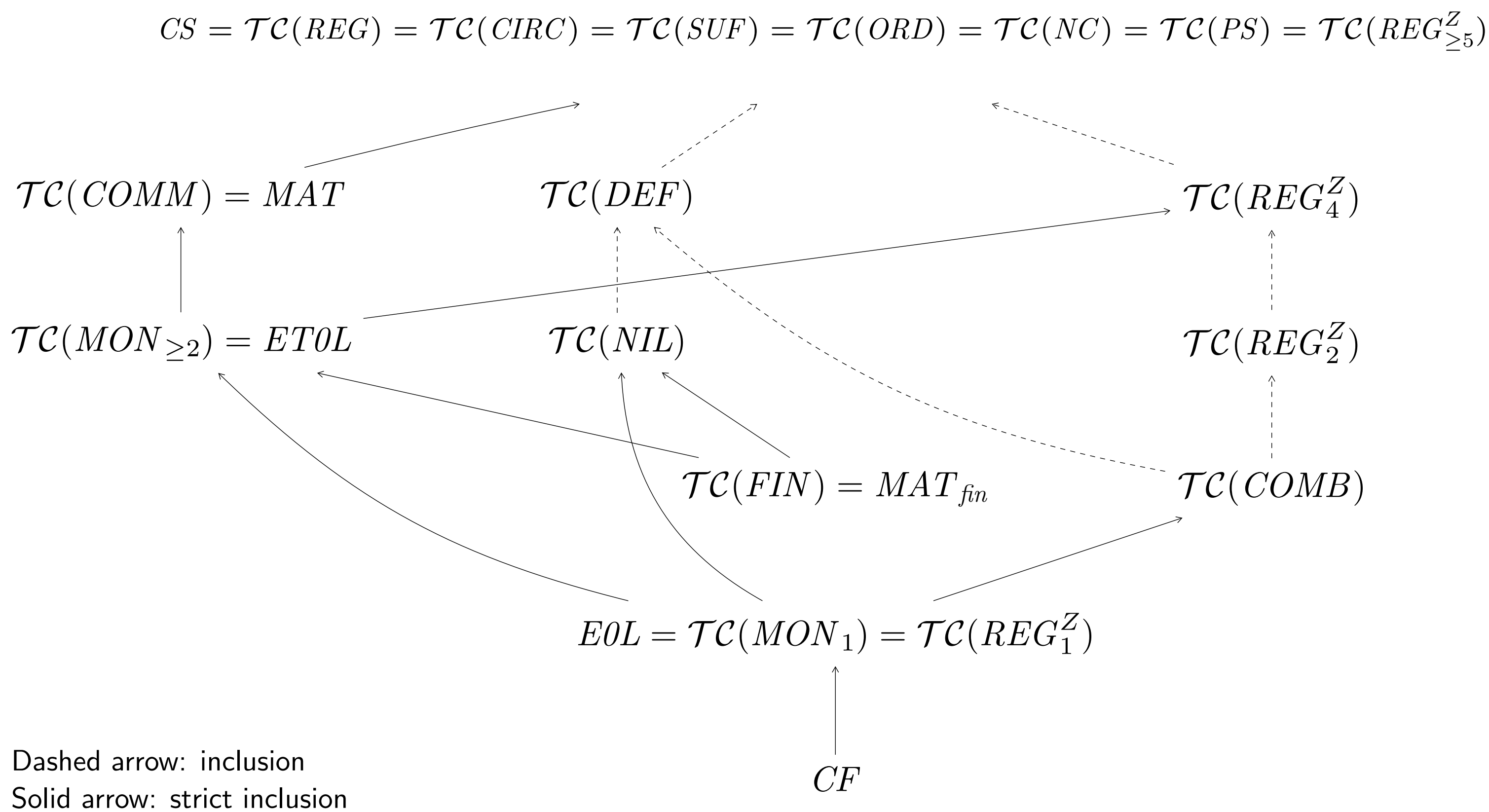


Previous Work

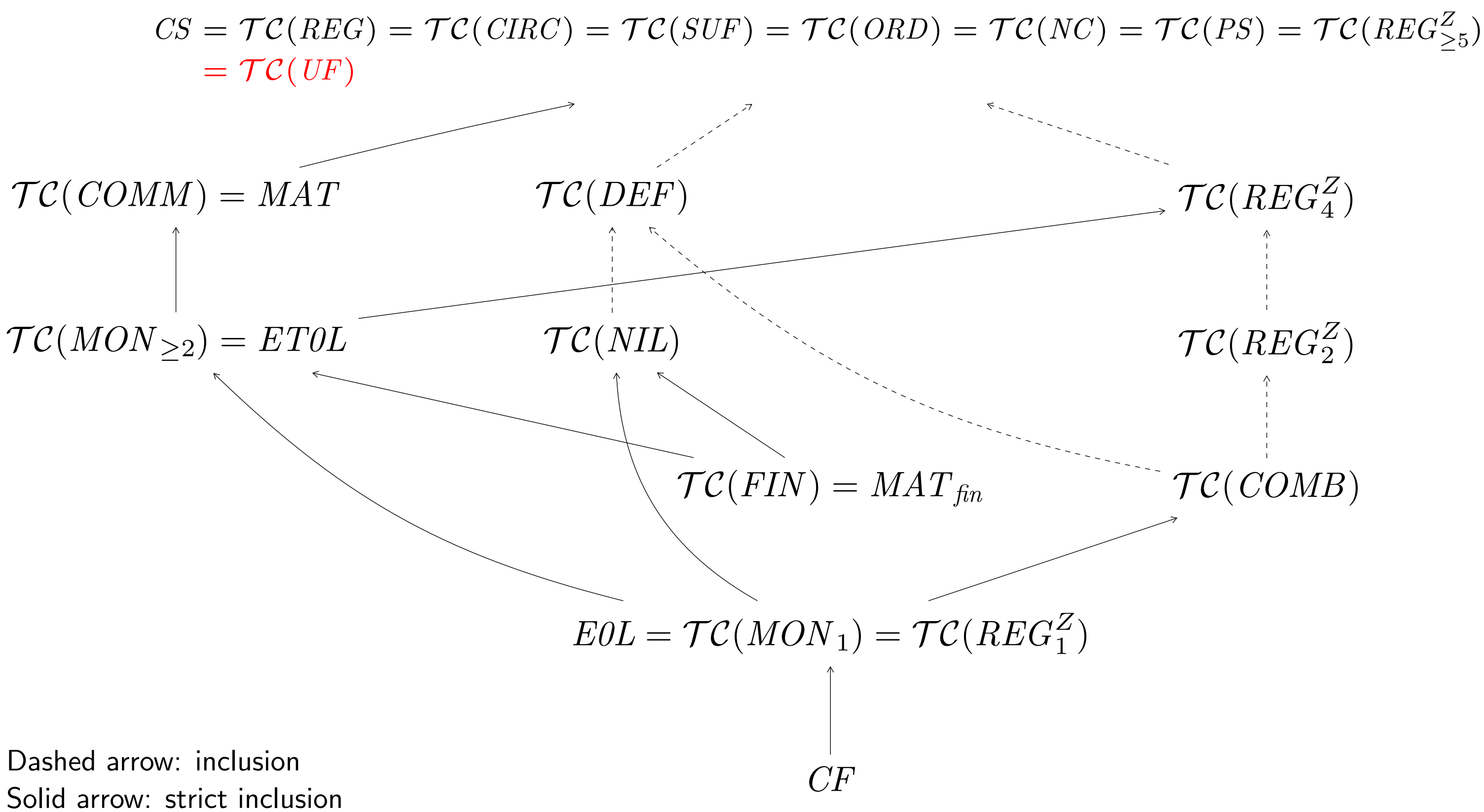


Dashed arrow: inclusion
Solid arrow: strict inclusion

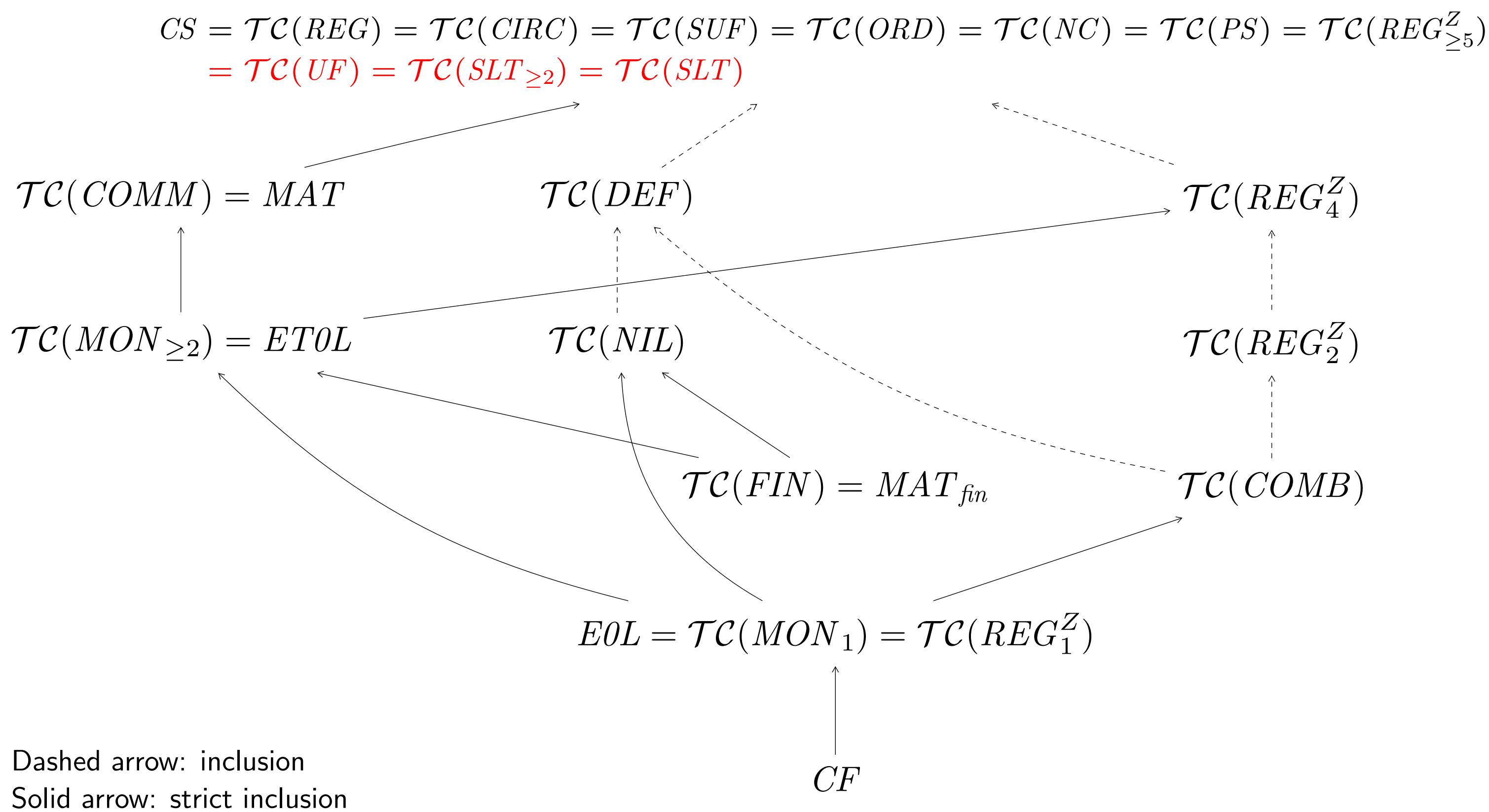
Results


$CS = \mathcal{TC}(REG) = \mathcal{TC}(CIRC) = \mathcal{TC}(SUF) = \mathcal{TC}(ORD) = \mathcal{TC}(NC) = \mathcal{TC}(PS) = \mathcal{TC}(REG^Z_{\geq 5})$
$\mathcal{TC}(COMM) = MAT$
$\mathcal{TC}(DEF)$
$\mathcal{TC}(REG^Z_4)$
$\mathcal{TC}(MON_{\geq 2}) = ET0L$
$\mathcal{TC}(NIL)$
$\mathcal{TC}(REG^Z_2)$
$\mathcal{TC}(FIN) = MAT_{fin}$
$\mathcal{TC}(COMB)$
$E0L = \mathcal{TC}(MON_1) = \mathcal{TC}(REG^Z_1)$
CF
Dashed arrow: inclusion
Solid arrow: strict inclusion

Results


$CS = \mathcal{TC}(REG) = \mathcal{TC}(CIRC) = \mathcal{TC}(SUF) = \mathcal{TC}(ORD) = \mathcal{TC}(NC) = \mathcal{TC}(PS) = \mathcal{TC}(REG^Z_{\geq 5})$
$= \mathcal{TC}(UF)$
$\mathcal{TC}(COMM) = MAT$
$\mathcal{TC}(DEF)$
$\mathcal{TC}(REG^Z_4)$
$\mathcal{TC}(MON_{\geq 2}) = ET0L$
$\mathcal{TC}(NIL)$
$\mathcal{TC}(REG^Z_2)$
$\mathcal{TC}(FIN) = MAT_{fin}$
$\mathcal{TC}(COMB)$
$E0L = \mathcal{TC}(MON_1) = \mathcal{TC}(REG^Z_1)$
CF
Dashed arrow: inclusion
Solid arrow: strict inclusion

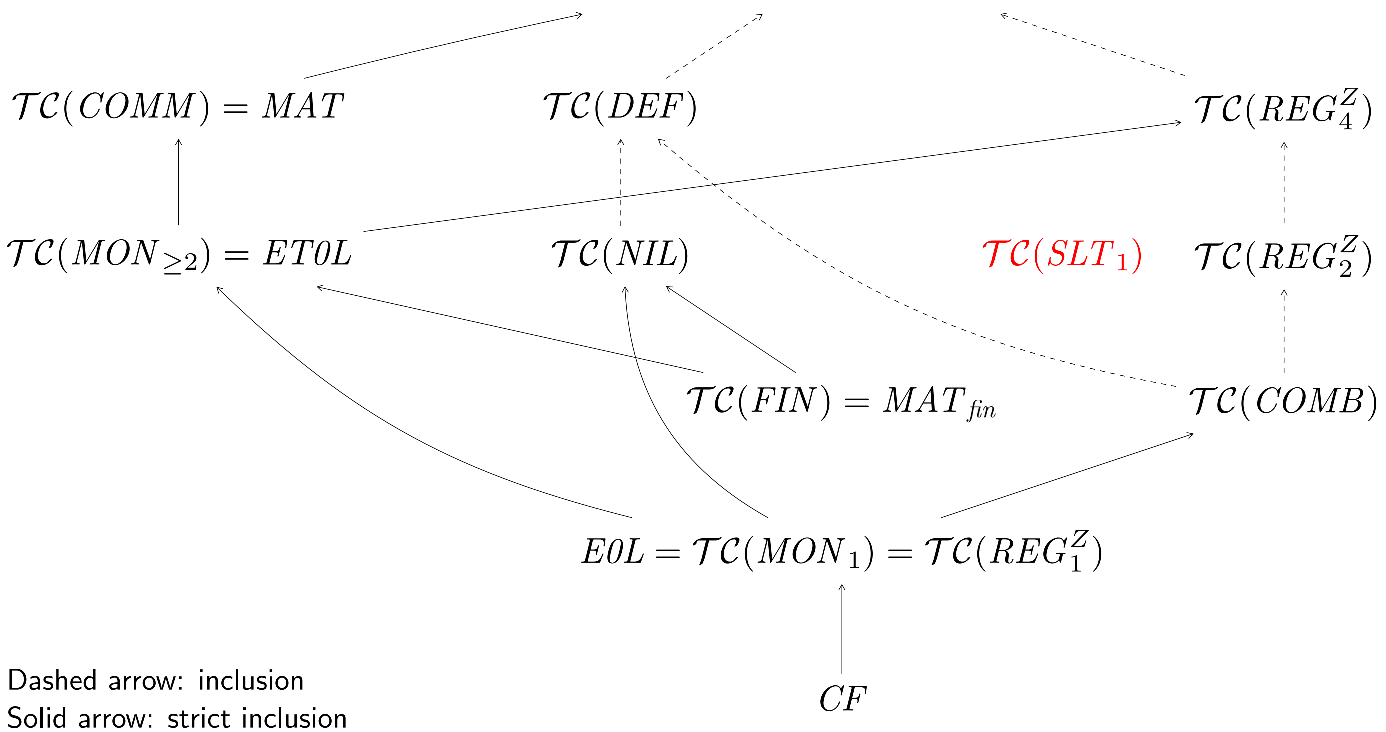
Results



Dashed arrow: inclusion
Solid arrow: strict inclusion

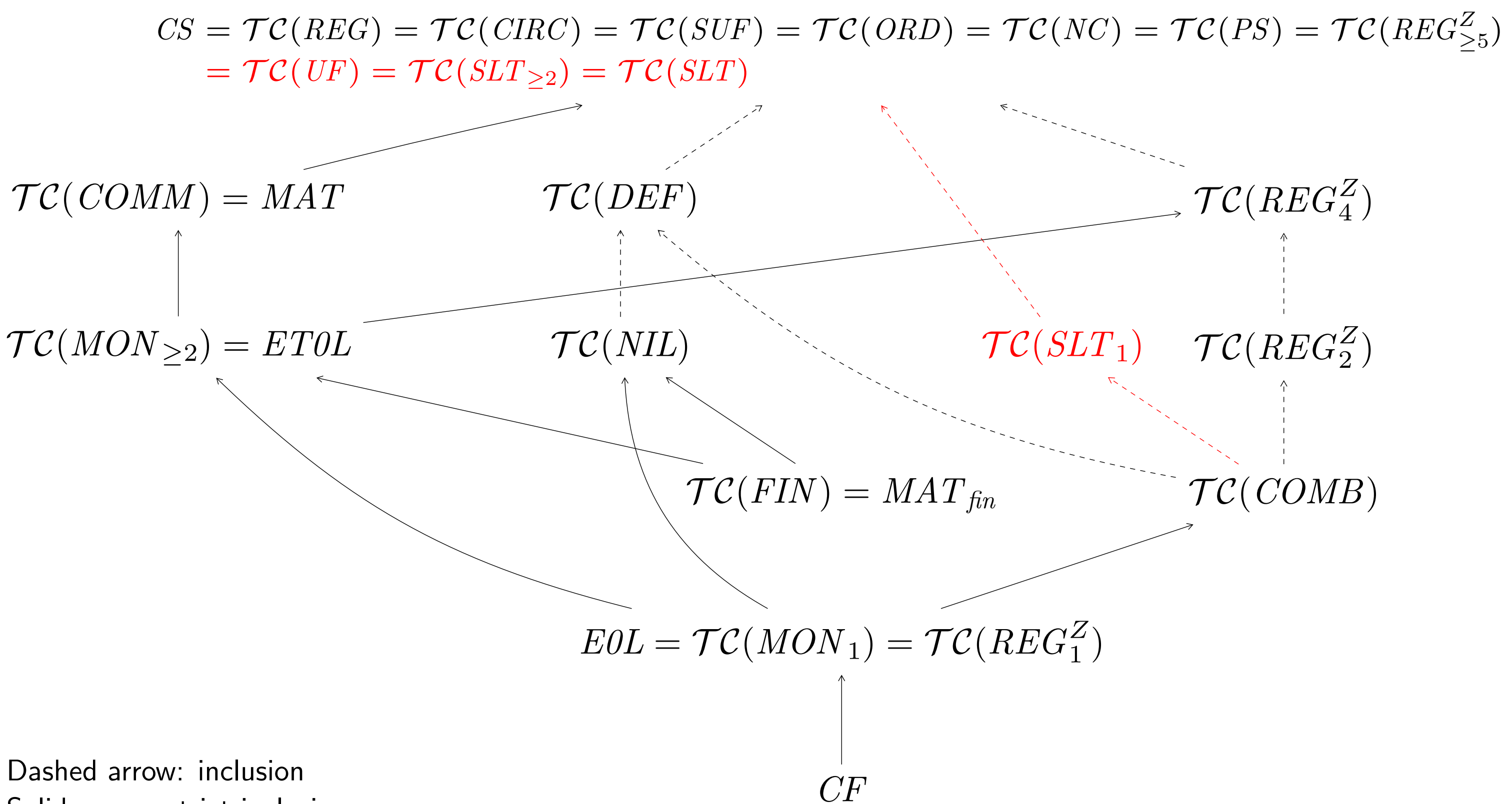
Results

$$CS = \mathcal{TC}(REG) = \mathcal{TC}(CIRC) = \mathcal{TC}(SUF) = \mathcal{TC}(ORD) = \mathcal{TC}(NC) = \mathcal{TC}(PS) = \mathcal{TC}(REG^Z_{\geq 5})$$
$$= \mathcal{TC}(UF) = \mathcal{TC}(SLT_{\geq 2}) = \mathcal{TC}(SLT)$$



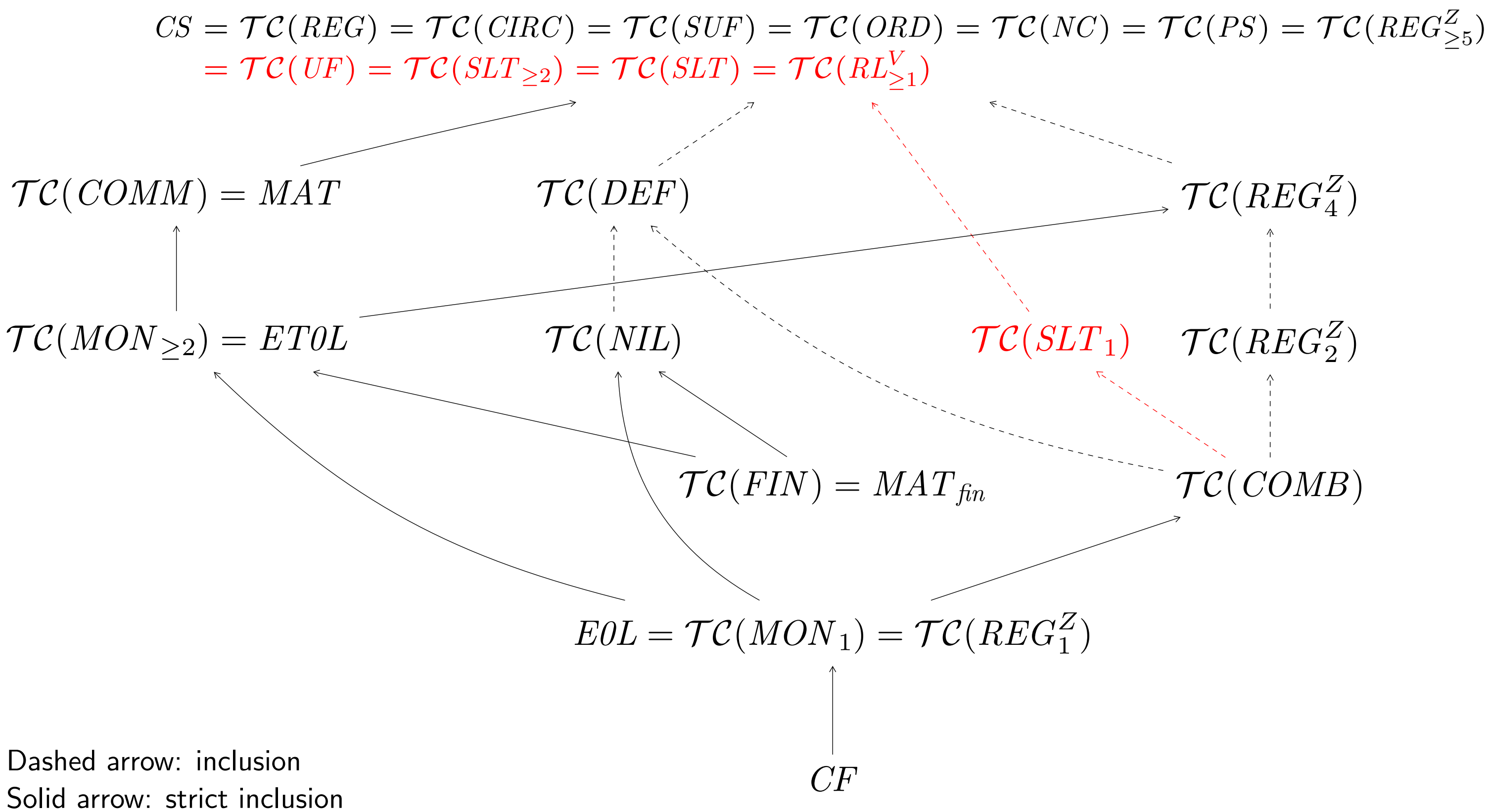
Dashed arrow: inclusion
Solid arrow: strict inclusion

Results

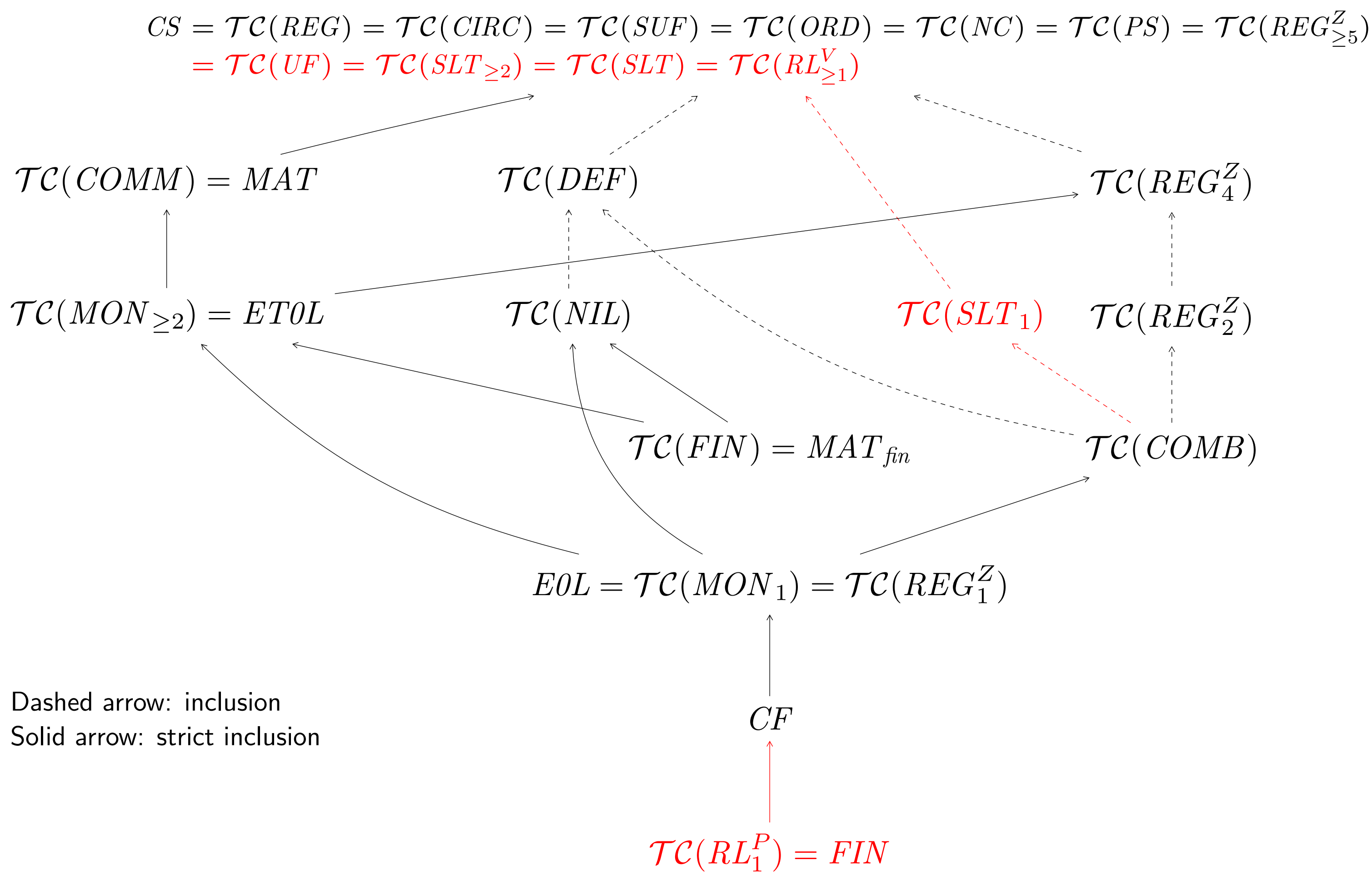


Dashed arrow: inclusion
Solid arrow: strict inclusion

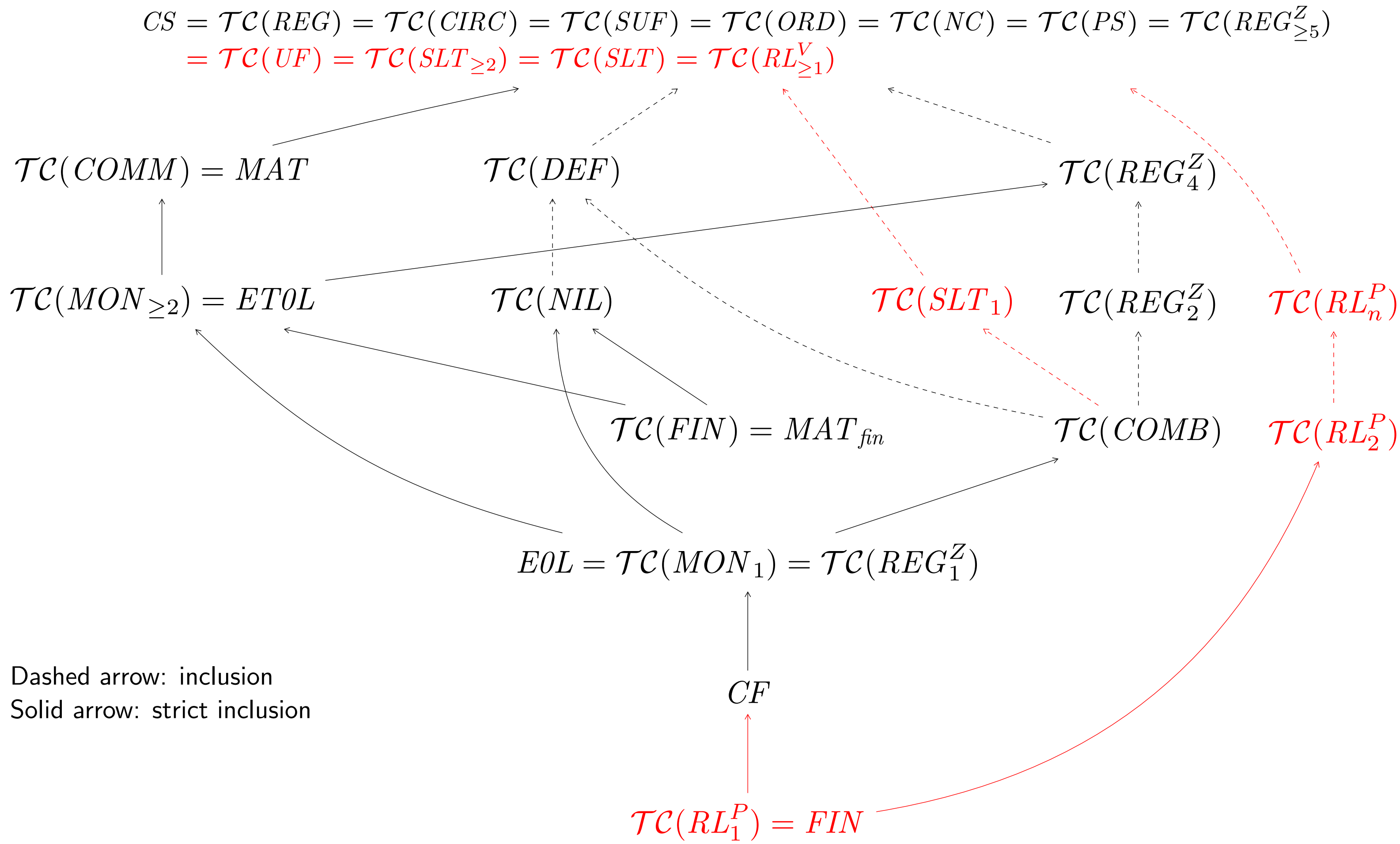
Results


$CS = \mathcal{TC}(REG) = \mathcal{TC}(CIRC) = \mathcal{TC}(SUF) = \mathcal{TC}(ORD) = \mathcal{TC}(NC) = \mathcal{TC}(PS) = \mathcal{TC}(REG^Z_{\geq 5})$
$= \mathcal{TC}(UF) = \mathcal{TC}(SLT_{\geq 2}) = \mathcal{TC}(SLT) = \mathcal{TC}(RL^V_{\geq 1})$
$\mathcal{TC}(COMM) = MAT$
$\mathcal{TC}(DEF)$
$\mathcal{TC}(REG^Z_4)$
$\mathcal{TC}(MON_{\geq 2}) = ET0L$
$\mathcal{TC}(NIL)$
$\mathcal{TC}(SLT_1)$
$\mathcal{TC}(REG^Z_2)$
$\mathcal{TC}(FIN) = MAT_{fin}$
$\mathcal{TC}(COMB)$
$E0L = \mathcal{TC}(MON_1) = \mathcal{TC}(REG^Z_1)$
CF
Dashed arrow: inclusion
Solid arrow: strict inclusion

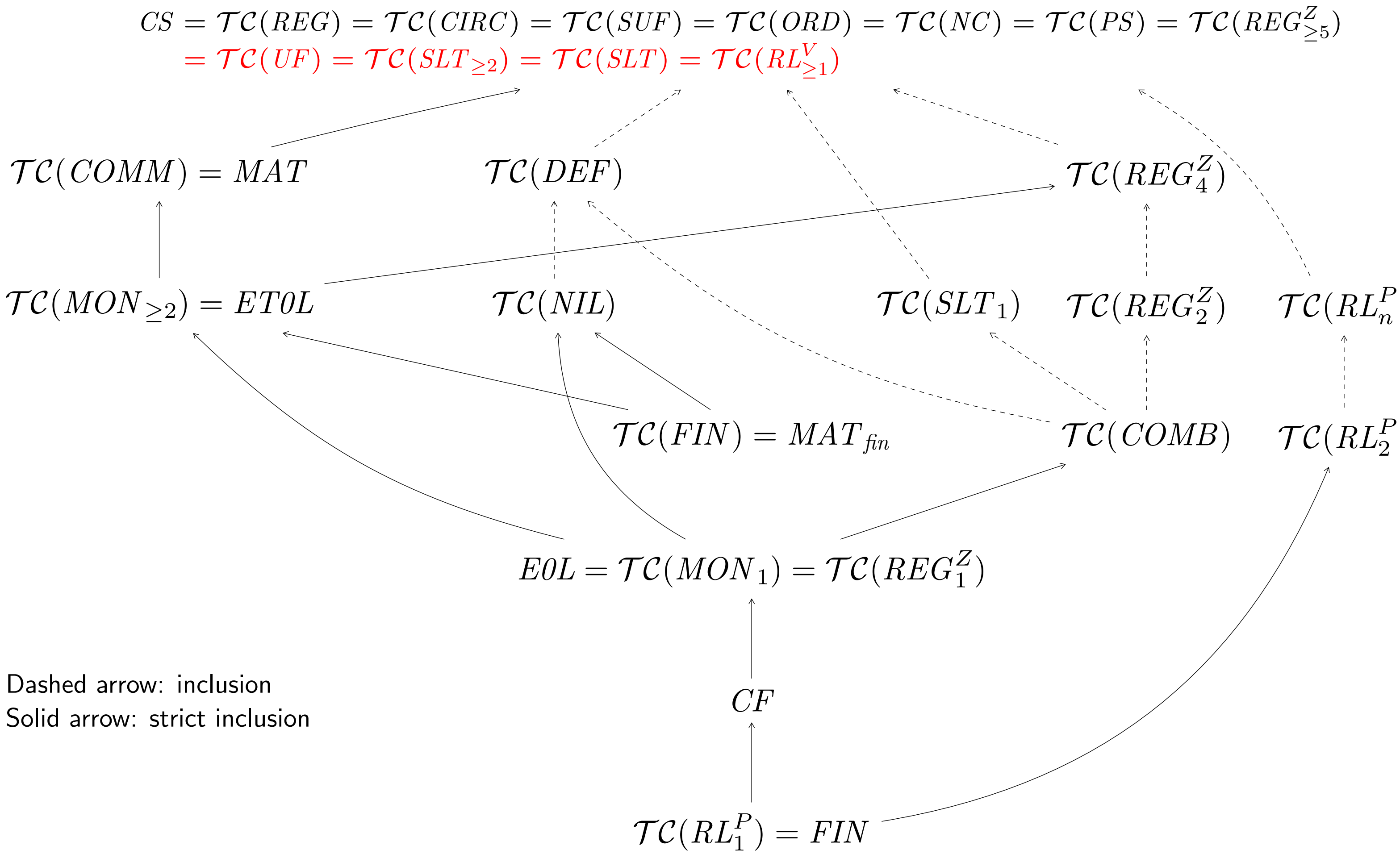
Results


$CS = \mathcal{TC}(REG) = \mathcal{TC}(CIRC) = \mathcal{TC}(SUF) = \mathcal{TC}(ORD) = \mathcal{TC}(NC) = \mathcal{TC}(PS) = \mathcal{TC}(REG^Z_{\geq 5})$
$= \mathcal{TC}(UF) = \mathcal{TC}(SLT_{\geq 2}) = \mathcal{TC}(SLT) = \mathcal{TC}(RL^V_{\geq 1})$
$\mathcal{TC}(COMM) = MAT$
$\mathcal{TC}(DEF)$
$\mathcal{TC}(REG^Z_4)$
$\mathcal{TC}(MON_{\geq 2}) = ET0L$
$\mathcal{TC}(NIL)$
$\mathcal{TC}(SLT_1)$
$\mathcal{TC}(REG^Z_2)$
$\mathcal{TC}(FIN) = MAT_{fin}$
$\mathcal{TC}(COMB)$
$E0L = \mathcal{TC}(MON_1) = \mathcal{TC}(REG^Z_1)$
CF
$\mathcal{TC}(RL^P_1) = FIN$
Dashed arrow: inclusion
Solid arrow: strict inclusion

Results


$CS = \mathcal{TC}(REG) = \mathcal{TC}(CIRC) = \mathcal{TC}(SUF) = \mathcal{TC}(ORD) = \mathcal{TC}(NC) = \mathcal{TC}(PS) = \mathcal{TC}(REG^{Z}_{\geq 5})$
$= \mathcal{TC}(UF) = \mathcal{TC}(SLT_{\geq 2}) = \mathcal{TC}(SLT) = \mathcal{TC}(RL^{V}_{\geq 1})$
$\mathcal{TC}(COMM) = MAT$
$\mathcal{TC}(DEF)$
$\mathcal{TC}(REG^{Z}_{4})$
$\mathcal{TC}(MON_{\geq 2}) = ET0L$
$\mathcal{TC}(NIL)$
$\mathcal{TC}(SLT_1)$
$\mathcal{TC}(REG^{Z}_{2})$
$\mathcal{TC}(RL^{P}_{n})$
$\mathcal{TC}(FIN) = MAT_{fin}$
$\mathcal{TC}(COMB)$
$\mathcal{TC}(RL^{P}_{2})$
$E0L = \mathcal{TC}(MON_1) = \mathcal{TC}(REG^{Z}_{1})$
CF
$\mathcal{TC}(RL^{P}_{1}) = FIN$
Dashed arrow: inclusion
Solid arrow: strict inclusion

Family of the Context-Sensitive Languages



Simulating a Context-Sensitive Grammar

Kuroda normal form: $G = (N, T, P, S)$ where each rule has the form

- $A \to BC$, $A \to B$, or $A \to a$ with $A, B, C \in N$ and $a \in T$, or
- $AB \to CD$ with $A, B, C, D \in N$

Simulating a Context-Sensitive Grammar

Kuroda normal form: $G = (N, T, P, S)$ where each rule has the form

- $A \to BC$, $A \to B$, or $A \to a$ with $A, B, C \in N$ and $a \in T$, or
- $AB \to CD$ with $A, B, C, D \in N$

Rules of the form $A \to BC$ or $A \to B$ can be simulated by themselves.

Simulating a Context-Sensitive Grammar

Kuroda normal form: $G = (N, T, P, S)$ where each rule has the form

- $A \to BC$, $A \to B$, or $A \to a$ with $A, B, C \in N$ and $a \in T$, or
- $AB \to CD$ with $A, B, C, D \in N$

Rules of the form $A \to BC$ or $A \to B$ can be simulated by themselves.

Context-sensitive rules have to be simulated by context-free rules together with some control.

Simulating a Context-Sensitive Grammar

Kuroda normal form: $G = (N, T, P, S)$ where each rule has the form

- $A \to BC$, $A \to B$, or $A \to a$ with $A, B, C \in N$ and $a \in T$, or
- $AB \to CD$ with $A, B, C, D \in N$

Rules of the form $A \to BC$ or $A \to B$ can be simulated by themselves.

Context-sensitive rules have to be simulated by context-free rules together with some control.

$p : AB \to CD$ will be simulated by the rules

$$A \to A_{p,1}, \quad B \to B_{p,2}, \quad A_{p,1} \to C, \quad B_{p,2} \to D$$

Simulating a Context-Sensitive Grammar

Kuroda normal form: $G = (N, T, P, S)$ where each rule has the form

– $A \to BC$, $A \to B$, or $A \to a$ with $A, B, C \in N$ and $a \in T$, or
– $AB \to CD$ with $A, B, C, D \in N$

Rules of the form $A \to BC$ or $A \to B$ can be simulated by themselves.

Rules of the form $A \to a$ should be modified to $A \to \hat{a}$ with a non-terminal placeholder $\hat{a}$.

Context-sensitive rules have to be simulated by context-free rules together with some control.

$p : AB \to CD$ will be simulated by the rules

$$A \to A_{p,1}, \quad B \to B_{p,2}, \quad A_{p,1} \to C, \quad B_{p,2} \to D$$

Simulating a Context-Sensitive Grammar

$G = (N, T, P, S)$ as above $\rightsquigarrow$ tc-grammar $G_{\mathrm{tc}} = (N_{\mathrm{tc}}, T, P_{\mathrm{tc}}, S, R_{\mathrm{tc}})$:

Simulating a Context-Sensitive Grammar

$G = (N, T, P, S)$ as above $\rightsquigarrow$ tc-grammar $G_{\mathrm{tc}} = (N_{\mathrm{tc}}, T, P_{\mathrm{tc}}, S, R_{\mathrm{tc}})$:

$$P_{\mathrm{cf}} = P \cap (\{\, A \to BC \mid A, B, C \in N \,\} \cup \{\, A \to B \mid A, B \in N \,\})$$

Simulating a Context-Sensitive Grammar

$G = (N, T, P, S)$ as above $\rightsquigarrow$ tc-grammar $G_{\mathrm{tc}} = (N_{\mathrm{tc}}, T, P_{\mathrm{tc}}, S, R_{\mathrm{tc}})$:

$$P_{\mathrm{cf}} = P \cap (\{\, A \to BC \mid A, B, C \in N \,\} \cup \{\, A \to B \mid A, B \in N \,\}),$$

$$P_{\mathrm{t}} = \{\, A \to \hat{a} \mid A \in N, a \in T, A \to a \in P \,\} \cup \{\, \hat{a} \to a \mid a \in T \,\}$$

Simulating a Context-Sensitive Grammar

$G = (N, T, P, S)$ as above $\rightsquigarrow$ tc-grammar $G_{\mathrm{tc}} = (N_{\mathrm{tc}}, T, P_{\mathrm{tc}}, S, R_{\mathrm{tc}})$:

$$P_{\mathrm{cf}} = P \cap (\{\, A \to BC \mid A, B, C \in N \,\} \cup \{\, A \to B \mid A, B \in N \,\}),$$

$$P_{\mathrm{t}} = \{\, A \to \hat{a} \mid A \in N, a \in T, A \to a \in P \,\} \cup \{\, \hat{a} \to a \mid a \in T \,\},$$

$$P_{\mathrm{d}} = \{\, A \to A \mid A \in N \,\} \cup \{\, \hat{a} \to \hat{a} \mid a \in T \,\}$$

Simulating a Context-Sensitive Grammar

$G = (N, T, P, S)$ as above $\rightsquigarrow$ tc-grammar $G_{\mathrm{tc}} = (N_{\mathrm{tc}}, T, P_{\mathrm{tc}}, S, R_{\mathrm{tc}})$:

$$P_{\mathrm{cf}} = P \cap (\{\, A \to BC \mid A, B, C \in N \,\} \cup \{\, A \to B \mid A, B \in N \,\}),$$

$$P_{\mathrm{t}} = \{\, A \to \hat{a} \mid A \in N, a \in T, A \to a \in P \,\} \cup \{\, \hat{a} \to a \mid a \in T \,\},$$

$$P_{\mathrm{d}} = \{\, A \to A \mid A \in N \,\} \cup \{\, \hat{a} \to \hat{a} \mid a \in T \,\},$$

$$P_{\mathrm{cs}} = \bigcup_{p:AB \to CD \in P} \{A \to A_{p,1},\ B \to B_{p,2},\ A_{p,1} \to C,\ B_{p,2} \to D\}$$

Simulating a Context-Sensitive Grammar

$G = (N, T, P, S)$ as above $\rightsquigarrow$ tc-grammar $G_{\text{tc}} = (N_{\text{tc}}, T, P_{\text{tc}}, S, R_{\text{tc}})$:

$$P_{\text{cf}} = P \cap (\{\, A \to BC \mid A, B, C \in N \,\} \cup \{\, A \to B \mid A, B \in N \,\}),$$

$$P_{\text{t}} = \{\, A \to \hat{a} \mid A \in N, a \in T, A \to a \in P \,\} \cup \{\, \hat{a} \to a \mid a \in T \,\},$$

$$P_{\text{d}} = \{\, A \to A \mid A \in N \,\} \cup \{\, \hat{a} \to \hat{a} \mid a \in T \,\},$$

$$P_{\text{cs}} = \bigcup_{p: AB \to CD \in P} \{A \to A_{p,1},\ B \to B_{p,2},\ A_{p,1} \to C,\ B_{p,2} \to D\},$$

$$P_{\text{tc}} = P_{\text{cf}} \cup P_{\text{t}} \cup P_{\text{d}} \cup P_{\text{cs}}$$

Simulating a Context-Sensitive Grammar

$G = (N, T, P, S)$ as above $\rightsquigarrow$ tc-grammar $G_{\mathrm{tc}} = (N_{\mathrm{tc}}, T, P_{\mathrm{tc}}, S, R_{\mathrm{tc}})$:

$$P_{\mathrm{cf}} = P \cap (\{ A \to BC \mid A, B, C \in N \} \cup \{ A \to B \mid A, B \in N \}),$$

$$P_{\mathrm{t}} = \{ A \to \hat{a} \mid A \in N, a \in T, A \to a \in P \} \cup \{ \hat{a} \to a \mid a \in T \},$$

$$P_{\mathrm{d}} = \{ A \to A \mid A \in N \} \cup \{ \hat{a} \to \hat{a} \mid a \in T \},$$

$$P_{\mathrm{cs}} = \bigcup_{p:AB \to CD \in P} \{A \to A_{p,1},\ B \to B_{p,2},\ A_{p,1} \to C,\ B_{p,2} \to D\},$$

$$P_{\mathrm{tc}} = P_{\mathrm{cf}} \cup P_{\mathrm{t}} \cup P_{\mathrm{d}} \cup P_{\mathrm{cs}},$$

$$N_{\mathrm{cf}} = N \cup \{ \hat{a} \mid a \in T \}$$

Simulating a Context-Sensitive Grammar

$G = (N, T, P, S)$ as above $\rightsquigarrow$ tc-grammar $G_{\mathrm{tc}} = (N_{\mathrm{tc}}, T, P_{\mathrm{tc}}, S, R_{\mathrm{tc}})$:

$$P_{\mathrm{cf}} = P \cap (\{\, A \to BC \mid A, B, C \in N \,\} \cup \{\, A \to B \mid A, B \in N \,\}),$$

$$P_{\mathrm{t}} = \{\, A \to \hat{a} \mid A \in N, a \in T, A \to a \in P \,\} \cup \{\, \hat{a} \to a \mid a \in T \,\},$$

$$P_{\mathrm{d}} = \{\, A \to A \mid A \in N \,\} \cup \{\, \hat{a} \to \hat{a} \mid a \in T \,\},$$

$$P_{\mathrm{cs}} = \bigcup_{p:AB \to CD \in P} \{A \to A_{p,1},\ B \to B_{p,2},\ A_{p,1} \to C,\ B_{p,2} \to D\},$$

$$P_{\mathrm{tc}} = P_{\mathrm{cf}} \cup P_{\mathrm{t}} \cup P_{\mathrm{d}} \cup P_{\mathrm{cs}},$$

$$N_{\mathrm{cf}} = N \cup \{\, \hat{a} \mid a \in T \,\},$$

$$N_1 = \{\, A_{p,1} \mid p : AB \to CD \in P \,\},\ N_2 = \{\, B_{p,2} \mid p : AB \to CD \in P \,\}$$

Simulating a Context-Sensitive Grammar

$G = (N, T, P, S)$ as above $\rightsquigarrow$ tc-grammar $G_{\text{tc}} = (N_{\text{tc}}, T, P_{\text{tc}}, S, R_{\text{tc}})$:

$$P_{\text{cf}} = P \cap (\{ A \to BC \mid A, B, C \in N \} \cup \{ A \to B \mid A, B \in N \}),$$

$$P_{\text{t}} = \{ A \to \hat{a} \mid A \in N, a \in T, A \to a \in P \} \cup \{ \hat{a} \to a \mid a \in T \},$$

$$P_{\text{d}} = \{ A \to A \mid A \in N \} \cup \{ \hat{a} \to \hat{a} \mid a \in T \},$$

$$P_{\text{cs}} = \bigcup_{p: AB \to CD \in P} \{A \to A_{p,1},\ B \to B_{p,2},\ A_{p,1} \to C,\ B_{p,2} \to D\},$$

$$P_{\text{tc}} = P_{\text{cf}} \cup P_{\text{t}} \cup P_{\text{d}} \cup P_{\text{cs}},$$

$$N_{\text{cf}} = N \cup \{ \hat{a} \mid a \in T \},$$

$$N_1 = \{ A_{p,1} \mid p : AB \to CD \in P \},\ N_2 = \{ B_{p,2} \mid p : AB \to CD \in P \},$$

$$N_{\text{tc}} = N_{\text{cf}} \cup N_1 \cup N_2$$

Simulating a Context-Sensitive Grammar

$G = (N, T, P, S)$ as above $\rightsquigarrow$ tc-grammar $G_{\mathrm{tc}} = (N_{\mathrm{tc}}, T, P_{\mathrm{tc}}, S, R_{\mathrm{tc}})$:

$$P_{\mathrm{cf}} = P \cap (\{\, A \to BC \mid A, B, C \in N \,\} \cup \{\, A \to B \mid A, B \in N \,\}),$$

$$P_{\mathrm{t}} = \{\, A \to \hat{a} \mid A \in N, a \in T, A \to a \in P \,\} \cup \{\, \hat{a} \to a \mid a \in T \,\},$$

$$P_{\mathrm{d}} = \{\, A \to A \mid A \in N \,\} \cup \{\, \hat{a} \to \hat{a} \mid a \in T \,\},$$

$$P_{\mathrm{cs}} = \bigcup_{p:AB \to CD \in P} \{A \to A_{p,1},\ B \to B_{p,2},\ A_{p,1} \to C,\ B_{p,2} \to D\},$$

$$P_{\mathrm{tc}} = P_{\mathrm{cf}} \cup P_{\mathrm{t}} \cup P_{\mathrm{d}} \cup P_{\mathrm{cs}},$$

$$N_{\mathrm{cf}} = N \cup \{\, \hat{a} \mid a \in T \,\},$$

$$N_1 = \{\, A_{p,1} \mid p : AB \to CD \in P \,\},\ N_2 = \{\, B_{p,2} \mid p : AB \to CD \in P \,\},$$

$$N_{\mathrm{tc}} = N_{\mathrm{cf}} \cup N_1 \cup N_2,$$

$$N_{12} = \{\, A_{p,1}B_{p,2} \mid p : AB \to CD \in P \,\}$$

Simulating a Context-Sensitive Grammar

$G = (N, T, P, S)$ as above $\rightsquigarrow$ tc-grammar $G_{\text{tc}} = (N_{\text{tc}}, T, P_{\text{tc}}, S, R_{\text{tc}})$:

$$P_{\text{cf}} = P \cap (\{ A \to BC \mid A, B, C \in N \} \cup \{ A \to B \mid A, B \in N \}),$$

$$P_{\text{t}} = \{ A \to \hat{a} \mid A \in N, a \in T, A \to a \in P \} \cup \{ \hat{a} \to a \mid a \in T \},$$

$$P_{\text{d}} = \{ A \to A \mid A \in N \} \cup \{ \hat{a} \to \hat{a} \mid a \in T \},$$

$$P_{\text{cs}} = \bigcup_{p: AB \to CD \in P} \{A \to A_{p,1},\ B \to B_{p,2},\ A_{p,1} \to C,\ B_{p,2} \to D\},$$

$$P_{\text{tc}} = P_{\text{cf}} \cup P_{\text{t}} \cup P_{\text{d}} \cup P_{\text{cs}},$$

$$N_{\text{cf}} = N \cup \{ \hat{a} \mid a \in T \},$$

$$N_1 = \{ A_{p,1} \mid p : AB \to CD \in P \},\ N_2 = \{ B_{p,2} \mid p : AB \to CD \in P \},$$

$$N_{\text{tc}} = N_{\text{cf}} \cup N_1 \cup N_2,$$

$$N_{12} = \{ A_{p,1} B_{p,2} \mid p : AB \to CD \in P \}, \quad R_{\text{tc}} = (N_{\text{cf}} \cup N_{12})^*$$

Properties of the Control Language

$R_{\mathrm{tc}} = (N_{\mathrm{cf}} \cup N_{12})^*$ with

$$N_{\mathrm{cf}} = N \cup \{\, \hat{a} \mid a \in T \,\},\ N_{12} = \{\, A_{p,1}B_{p,2} \mid p : AB \to CD \in P \,\}$$

Properties of the Control Language

$R_{\mathrm{tc}} = (N_{\mathrm{cf}} \cup N_{12})^*$ with

$$N_{\mathrm{cf}} = N \cup \{\ \hat{a} \mid a \in T\ \},\ N_{12} = \{\ A_{p,1}B_{p,2} \mid p : AB \to CD \in P\ \}$$

Strictly locally 2-testable: $R_{\mathrm{tc}} = [B, I, E, F]$ with

$$B = N_{\mathrm{cf}}^2 \cup N_{\mathrm{cf}}N_1 \cup N_{12}, \quad I = N_{\mathrm{cf}}^2 \cup N_{\mathrm{cf}}N_1 \cup N_{12} \cup N_2N_{\mathrm{cf}} \cup N_2N_1,$$
$$E = N_{\mathrm{cf}}^2 \cup N_{12} \cup N_2N_{\mathrm{cf}}, \quad F = N_{\mathrm{cf}} \cup \{\lambda\}.$$

Properties of the Control Language

$R_{\text{tc}} = (N_{\text{cf}} \cup N_{12})^*$ with

$$N_{\text{cf}} = N \cup \{\, \hat{a} \mid a \in T \,\},\ N_{12} = \{\, A_{p,1}B_{p,2} \mid p : AB \to CD \in P \,\}$$

Strictly locally 2-testable: $R_{\text{tc}} = [B, I, E, F]$ with

$$B = N_{\text{cf}}^2 \cup N_{\text{cf}}N_1 \cup N_{12}, \quad I = N_{\text{cf}}^2 \cup N_{\text{cf}}N_1 \cup N_{12} \cup N_2N_{\text{cf}} \cup N_2N_1,$$
$$E = N_{\text{cf}}^2 \cup N_{12} \cup N_2N_{\text{cf}}, \quad F = N_{\text{cf}} \cup \{\lambda\}.$$

$\leadsto CS \subseteq \mathcal{TC}(SLT_2) \subseteq \mathcal{TC}(SLT_k) \subseteq \mathcal{TC}(SLT) \subseteq CS$ for $k \geq 3$.

Properties of the Control Language

$R_{\mathrm{tc}} = (N_{\mathrm{cf}} \cup N_{12})^*$ with

$$N_{\mathrm{cf}} = N \cup \{\, \hat{a} \mid a \in T \,\},\ N_{12} = \{\, A_{p,1}B_{p,2} \mid p : AB \to CD \in P \,\}$$

Strictly locally 2-testable: $R_{\mathrm{tc}} = [B, I, E, F]$ with

$$B = N_{\mathrm{cf}}^2 \cup N_{\mathrm{cf}}N_1 \cup N_{12}, \quad I = N_{\mathrm{cf}}^2 \cup N_{\mathrm{cf}}N_1 \cup N_{12} \cup N_2N_{\mathrm{cf}} \cup N_2N_1,$$
$$E = N_{\mathrm{cf}}^2 \cup N_{12} \cup N_2N_{\mathrm{cf}}, \quad F = N_{\mathrm{cf}} \cup \{\lambda\}.$$

$\rightsquigarrow CS \subseteq \mathcal{TC}(SLT_2) \subseteq \mathcal{TC}(SLT_k) \subseteq \mathcal{TC}(SLT) \subseteq CS$ for $k \geq 3$.

Thus, $\mathcal{TC}(SLT_k) = CS$ for $k \geq 2$ and $\mathcal{TC}(SLT) = CS$.

Properties of the Control Language

$R_{\mathrm{tc}} = (N_{\mathrm{cf}} \cup N_{12})^*$ with

$$N_{\mathrm{cf}} = N \cup \{\, \hat{a} \mid a \in T \,\},\ N_{12} = \{\, A_{p,1}B_{p,2} \mid p : AB \to CD \in P \,\}$$

$R_{\mathrm{tc}} \in RL_1^V$

Properties of the Control Language

$R_{\mathrm{tc}} = (N_{\mathrm{cf}} \cup N_{12})^*$ with

$$N_{\mathrm{cf}} = N \cup \{\, \hat{a} \mid a \in T \,\},\ N_{12} = \{\, A_{p,1}B_{p,2} \mid p : AB \to CD \in P \,\}$$

$R_{\mathrm{tc}} \in RL_1^V$: generated by a right-linear grammar $G' = (\{S'\}, N_{\mathrm{tc}}, P', S')$ where

$$P' = \{\, S' \to xS' \mid x \in N_{\mathrm{cf}} \cup N_{12} \,\} \cup \{\, S' \to x \mid x \in N_{\mathrm{cf}} \cup N_{12} \,\}.$$

Properties of the Control Language

$R_{\mathrm{tc}} = (N_{\mathrm{cf}} \cup N_{12})^*$ with

$$N_{\mathrm{cf}} = N \cup \{\, \hat{a} \mid a \in T \,\},\ N_{12} = \{\, A_{p,1}B_{p,2} \mid p : AB \to CD \in P \,\}$$

$R_{\mathrm{tc}} \in RL_1^V$: generated by a right-linear grammar $G' = (\{S'\}, N_{\mathrm{tc}}, P', S')$ where

$$P' = \{\, S' \to xS' \mid x \in N_{\mathrm{cf}} \cup N_{12} \,\} \cup \{\, S' \to x \mid x \in N_{\mathrm{cf}} \cup N_{12} \,\}.$$

$\rightsquigarrow$ $CS \subseteq \mathcal{TC}(RL_1^V) \subseteq \mathcal{TC}(RL_n^V) \subseteq CS$ for $n \geq 2$.

Properties of the Control Language

$R_{\text{tc}} = (N_{\text{cf}} \cup N_{12})^*$ with

$$N_{\text{cf}} = N \cup \{\, \hat{a} \mid a \in T \,\},\ N_{12} = \{\, A_{p,1}B_{p,2} \mid p : AB \to CD \in P \,\}$$

$R_{\text{tc}} \in RL_1^V$: generated by a right-linear grammar $G' = (\{S'\}, N_{\text{tc}}, P', S')$ where

$$P' = \{\, S' \to xS' \mid x \in N_{\text{cf}} \cup N_{12} \,\} \cup \{\, S' \to x \mid x \in N_{\text{cf}} \cup N_{12} \,\}.$$

$\rightsquigarrow CS \subseteq \mathcal{TC}(RL_1^V) \subseteq \mathcal{TC}(RL_n^V) \subseteq CS$ for $n \geq 2$.

Thus, $\mathcal{TC}(RL_n^V) = CS$ for $n \geq 1$.

Properties of the Control Language

$R_{\mathrm{tc}} = (N_{\mathrm{cf}} \cup N_{12})^*$ with

$$N_{\mathrm{cf}} = N \cup \{\, \hat{a} \mid a \in T \,\},\ N_{12} = \{\, A_{p,1}B_{p,2} \mid p : AB \to CD \in P \,\}$$

$R_{\mathrm{tc}} \in UF$

Properties of the Control Language

$R_{\mathrm{tc}} = (N_{\mathrm{cf}} \cup N_{12})^*$ with

$$N_{\mathrm{cf}} = N \cup \{\ \hat{a} \mid a \in T\ \},\ N_{12} = \{\ A_{p,1}B_{p,2} \mid p : AB \to CD \in P\ \}$$

$R_{\mathrm{tc}} \in UF$: for a finite language $L = \{w_1, w_2, \ldots, w_n\}$, the Kleene-closure is $L^* = (\{w_1\}^*\{w_2\}^* \cdots \{w_n\}^*)^*$ and therefore union-free.

Properties of the Control Language

$R_{\mathrm{tc}} = (N_{\mathrm{cf}} \cup N_{12})^*$ with

$$N_{\mathrm{cf}} = N \cup \{\, \hat{a} \mid a \in T \,\},\ N_{12} = \{\, A_{p,1}B_{p,2} \mid p : AB \to CD \in P \,\}$$

$R_{\mathrm{tc}} \in UF$: for a finite language $L = \{w_1, w_2, \ldots, w_n\}$, the Kleene-closure is $L^* = (\{w_1\}^*\{w_2\}^* \cdots \{w_n\}^*)^*$ and therefore union-free.

$\rightsquigarrow CS \subseteq \mathcal{TC}(UF) \subseteq CS$.

Properties of the Control Language

$R_{\mathrm{tc}} = (N_{\mathrm{cf}} \cup N_{12})^*$ with

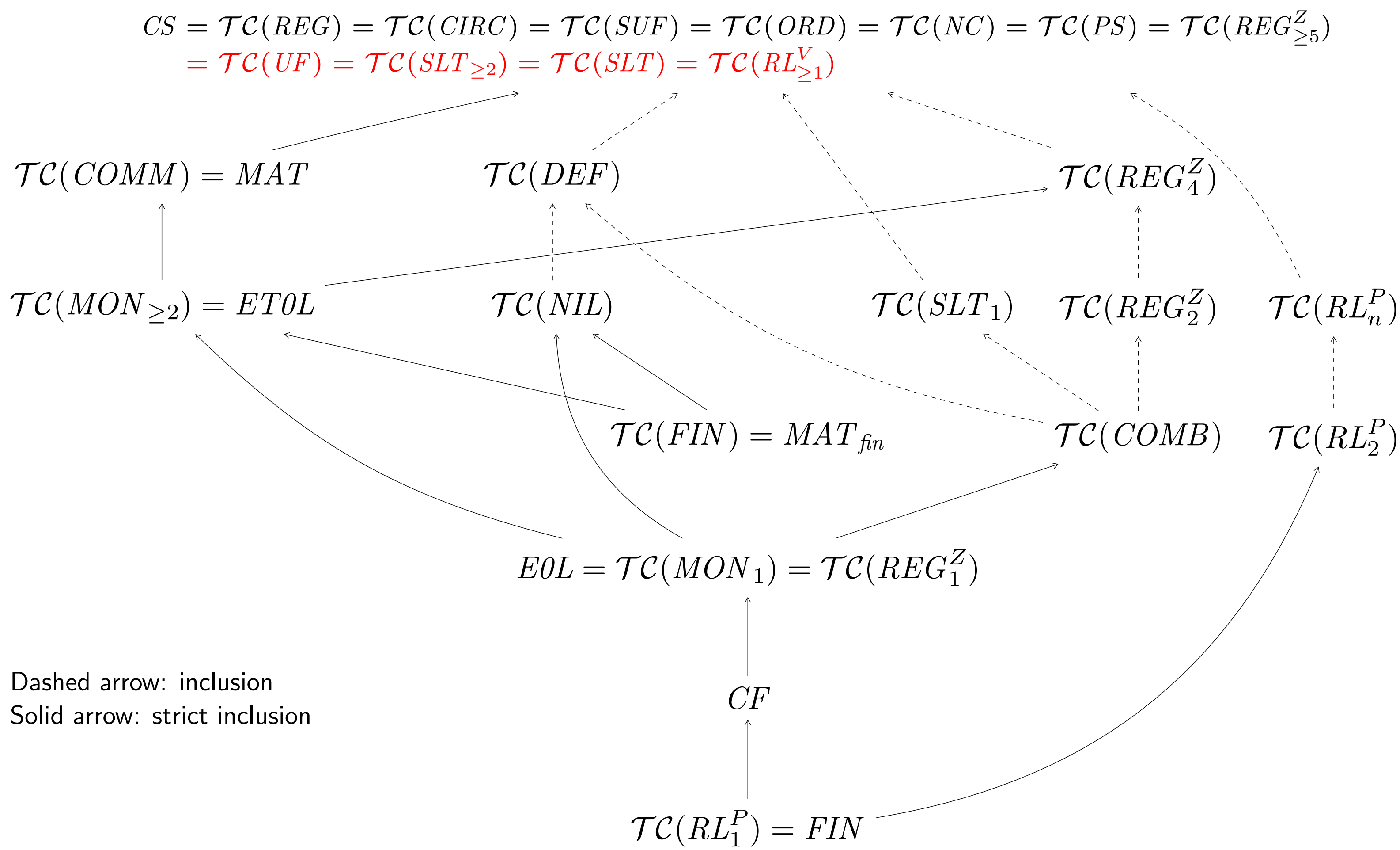
$$N_{\mathrm{cf}} = N \cup \{\, \hat{a} \mid a \in T \,\},\ N_{12} = \{\, A_{p,1}B_{p,2} \mid p : AB \to CD \in P \,\}$$

$R_{\mathrm{tc}} \in UF$: for a finite language $L = \{w_1, w_2, \ldots, w_n\}$, the Kleene-closure is $L^* = (\{w_1\}^*\{w_2\}^* \cdots \{w_n\}^*)^*$ and therefore union-free.

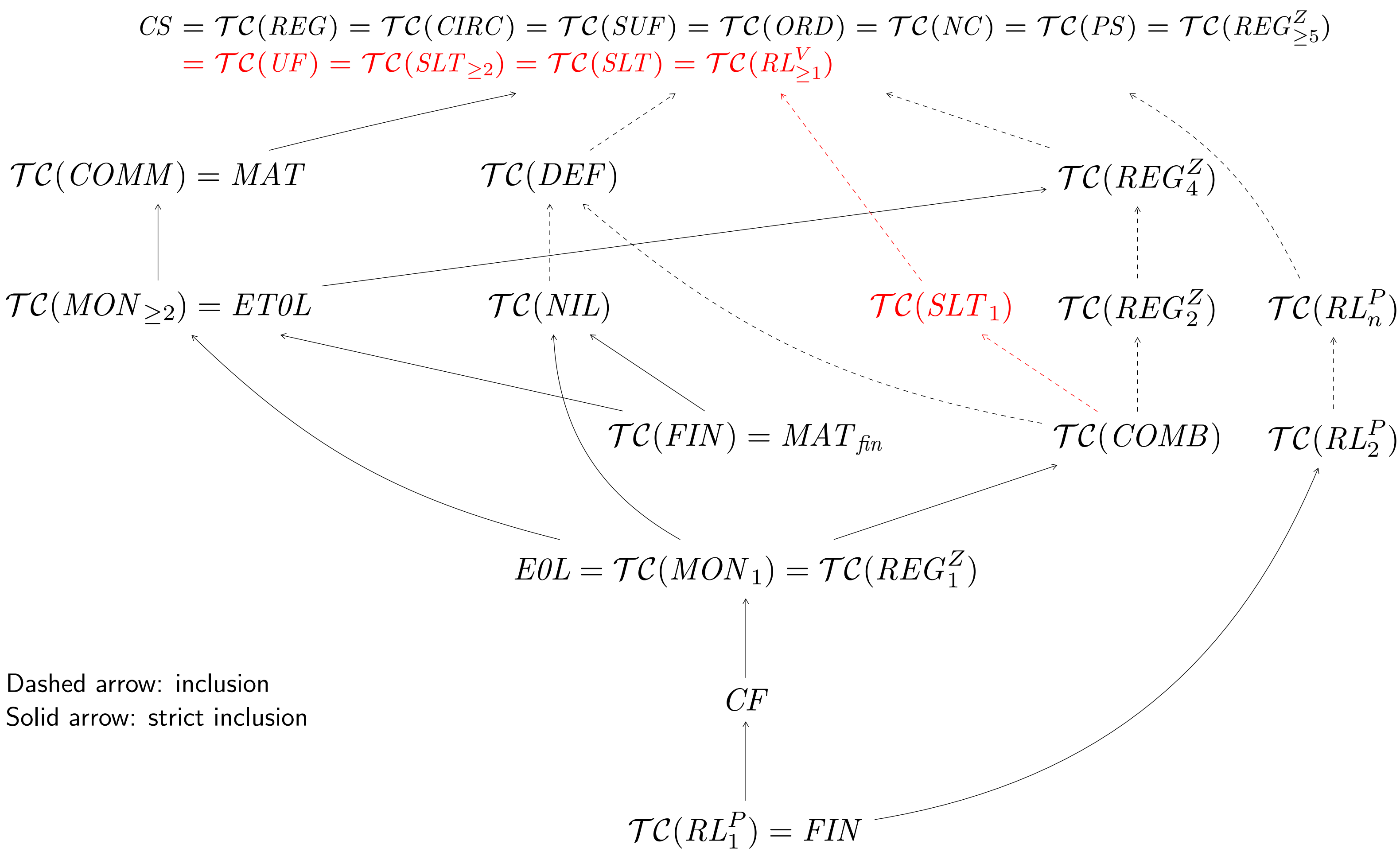
$\rightsquigarrow CS \subseteq \mathcal{TC}(UF) \subseteq CS$.

Thus, $\mathcal{TC}(UF) = CS$.

Family of the Context-Sensitive Languages

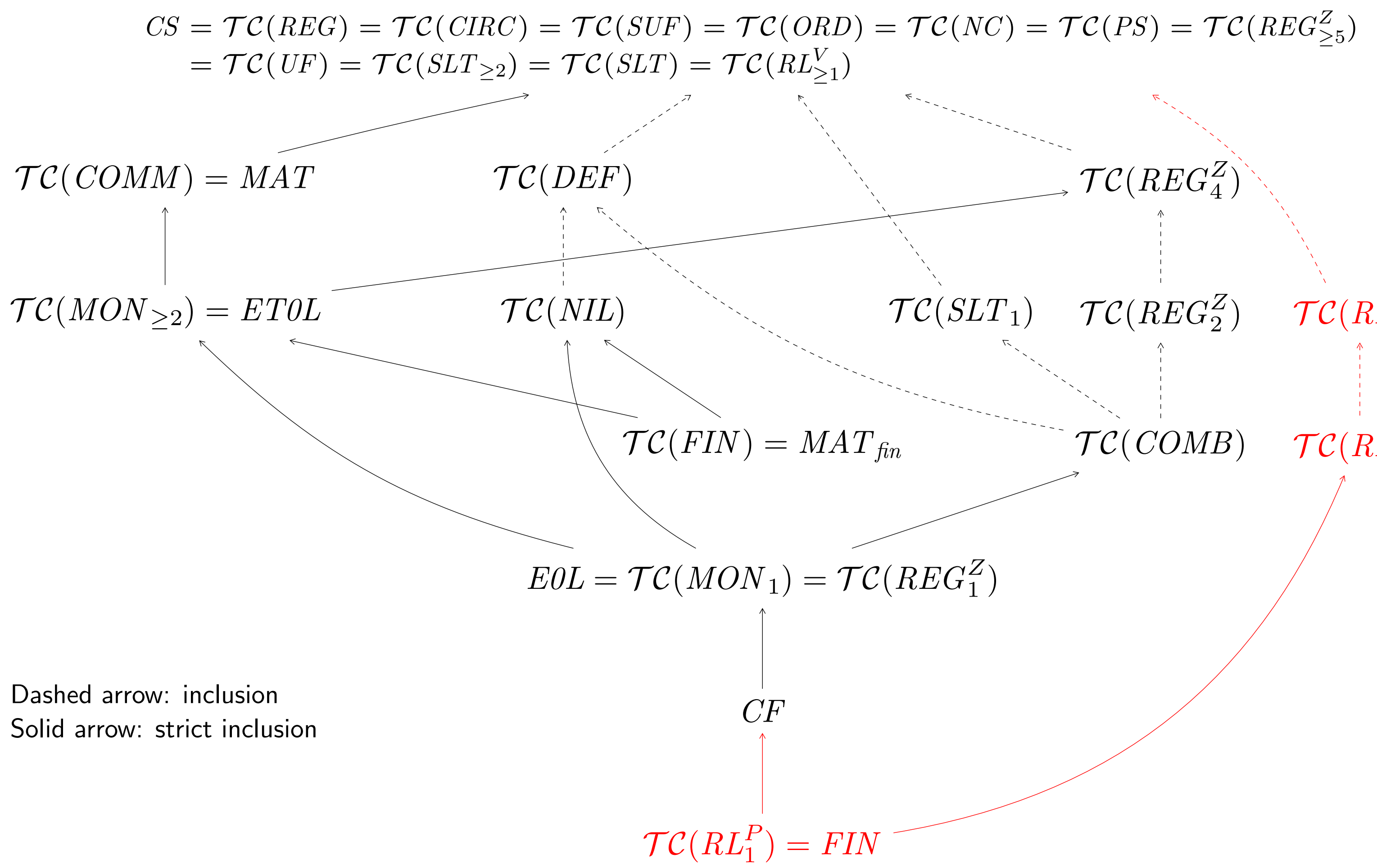


Family of the Context-Sensitive Languages



Dashed arrow: inclusion
Solid arrow: strict inclusion

Family of the Context-Sensitive Languages



Control Languages generated by one Rule

Let $G = (N, T, P, S, R)$ be a tc-grammar with $R \in RL_1^P$.

Control Languages generated by one Rule

Let $G = (N, T, P, S, R)$ be a tc-grammar with $R \in RL_1^P$.

$\leadsto |R| < 1$

Control Languages generated by one Rule

Let $G = (N, T, P, S, R)$ be a tc-grammar with $R \in RL_1^P$.

$\rightsquigarrow |R| < 1,$

$$\rightsquigarrow L(G) = \begin{cases} \{\, w \mid w \in T^* \text{ and } S \to w \in P \,\}, & \text{if } R = \{S\}, \\ \emptyset, & \text{otherwise} \end{cases}$$

Control Languages generated by one Rule

Let $G = (N, T, P, S, R)$ be a tc-grammar with $R \in RL_1^P$.

$\rightsquigarrow |R| < 1$,

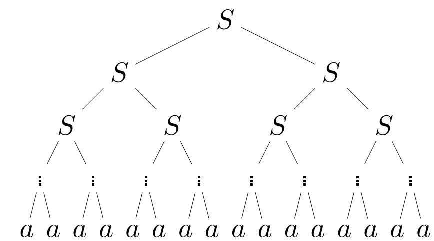
$$\rightsquigarrow L(G) = \begin{cases} \{\, w \mid w \in T^* \text{ and } S \to w \in P \,\}, & \text{if } R = \{S\}, \\ \emptyset, & \text{otherwise}, \end{cases}$$

$\rightsquigarrow |L(G)| < \infty$.

Thus, $\mathcal{TC}(RL_1^P) = FIN \subset CF$.

Control Languages generated by two Rules

Examples from the beginning:



$$G_1 = (\{S\}, \{a\}, \{S \to SS, S \to a\}, S, \{S\}^*)$$

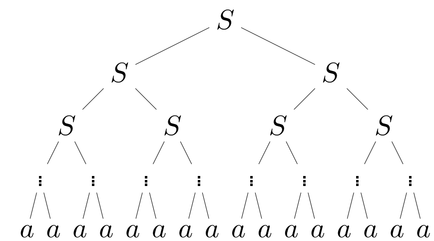
$$\rightsquigarrow L(G_1) = \{\, a^{2^n} \mid n \geq 0 \,\}$$

Control Languages generated by two Rules

Examples from the beginning:

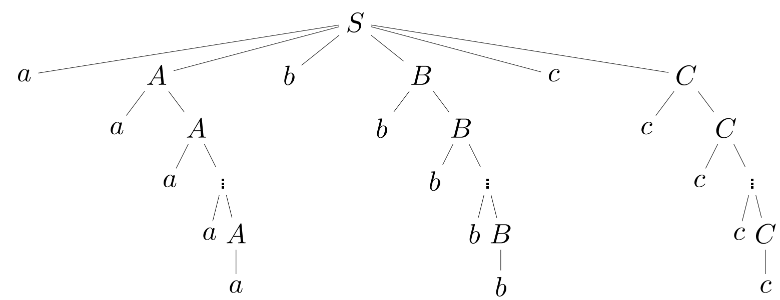
$$G_1 = (\{S\}, \{a\}, \{S \to SS, S \to a\}, S, \{S\}^*)$$

$$\rightsquigarrow L(G_1) = \{\, a^{2^n} \mid n \geq 0 \,\}$$



$G_2 = (\{S, A, B, C\}, \{a, b, c\}, P, S, \{S, aAbBcC\})$ with
$P = \{S \to aAbBcC, A \to aA, B \to bB, C \to cC, A \to a, B \to b, C \to c\}$

$$\rightsquigarrow L(G_2) = \{\, a^n b^n c^n \mid n \geq 2 \,\}$$

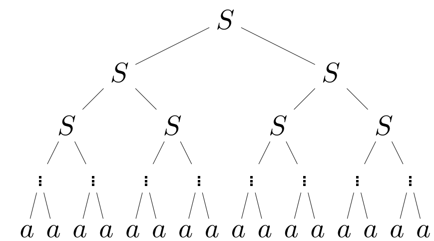


Control Languages generated by two Rules

Examples from the beginning:

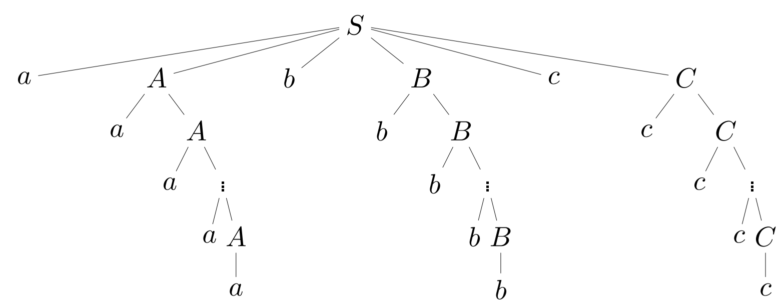
$$G_1 = (\{S\}, \{a\}, \{S \to SS, S \to a\}, S, \{S\}^*)$$

$$\rightsquigarrow L(G_1) = \{\, a^{2^n} \mid n \geq 0 \,\}$$



$G_2 = (\{S, A, B, C\}, \{a, b, c\}, P, S, \{S, aAbBcC\})$ with
$P = \{S \to aAbBcC, A \to aA, B \to bB, C \to cC, A \to a, B \to b, C \to c\}$

$$\rightsquigarrow L(G_2) = \{\, a^n b^n c^n \mid n \geq 2 \,\}$$



$\rightsquigarrow L(G_i) \in \mathcal{TC}(RL_2^P) \setminus CF$ for $i \in \{1, 2\}$
Thus, $\mathcal{TC}(RL_1^P) \subset \mathcal{TC}(RL_2^P)$.

Future Work

- In many cases, the strictness of the inclusion remains open.
- The incomparability of families is open as well in many cases.
- Consider tree-controlled grammars with erasing rules.
- Consider other subregular control languages.
- Relate the families of languages generated by tree-controlled grammars to language families obtained by other grammars/systems with regulated rewriting.