Masterarbeit

Logics for Context-Free Languages

Michael Thomas

14. Juli 2006

Prüfer:
Prof. Dr. Heribert Vollmer
Prof. Dr. Kurt Schneider

Institut für Theoretische Informatik
Gottfried Wilhelm Leibniz Universität Hannover

Erklärung

Hiermit versichere ich, dass ich die vorliegende Masterarbeit selbstständig verfasst und keine anderen Quellen und Hilfsmittel als die angegebenen benutzt habe.

Michael Thomas

Contents

1 Introduction

In 1960, Büchi showed that a set is a regular language if and only if it is the set of models of a sentence in monadic second-order logic with successor. Since then the relation of formal language classes to first- and second-order logic has been investigated. Their connection to boolean circuits led to further interesting results: For example, all regular sets definable in first-order logic with arbitrary numerical predicates are already sets of models of sentences in first-order logic with order and the ternary modulo-operator only, the so-called regular numerical predicates. In 1994, Straubing surveyed that area in his book *Finite Automata, Formal Logic, and Circuit Complexity* [Str74], summarizing the connection of the regular languages to first-order logics, the circuit classes below NC^1 and finite semigroup theory.

Yet, the context-free languages—the class containing the sets of syntactically correct computer programs in common programming languages—have solely been related to formal logics in a semantic approach by Lautemann, Schwentick and Thérien [LST95].

The present thesis investigates the relationship of context-free languages to an extension of first-order logic with a non-regular numerical predicate, the addition. At first, chapter 2 will provide the necessary framework of formal languages and logic. Chapter 3 will start with a motivation for the choice of that logic called FO[+]. Then, examples of definable and indefinable languages in FO[+] will be given and a general upper bound for the definability will be derived. Furthermore, a subset of the context-free languages, the class of linear recursive languages, is shown to be captured by that logic; and as separation from the boolean closure of the linear recursive languages fails, equality is conjectured. That conjecture is then sustained by the fact that the class of context-free languages recognizable

with finite-turn pushdown automata, a proper superclass of the linear recursive languages, is shown not to capture FO[+] on the level of context-free languages; therefore, it is incomparable to FO[+]. Finally, Chapter 4 concludes by briefly summarizing the results and pointing out the open questions.

2 Preliminaries

The mathematical preliminaries used in this thesis mostly concern formal logic, formal languages and finite automata. It is the intention of this chapter to shortly recall the necessary terminology.

2.1 Words and Languages

An *alphabet* is a non-empty finite set of symbols. Throughout this thesis, alphabets will usually be denoted by uppercase Greek letters Σ, Δ, Γ. The elements of an alphabet are called *letters*. A *word* (or *string*) over an alphabet Σ is a finite sequence of elements of Σ. A word w will be written in a concatenated fashion $w = a_1 a_2 \cdots a_n$, where $a_1, a_2, \ldots, a_n \in \Sigma$ are the successive elements of the sequence. The length $|w|$ of a word w is the number of elements in the sequence w. The number of occurrences of the letter $a \in \Sigma$ in w will be denoted by $|w|_a$. The empty sequence, called the *empty word*, will be denoted by ε. For words $u = u_1 \cdots u_n, v = v_1 \cdots v_m$ the concatenation $u \cdot v$ or uv is defined as $uv = u_1 \cdots u_n v_1 \cdots v_m$. The set of all strings over an alphabet Σ is denoted by the Kleene closure $\Sigma^\star$; furthermore, let $\Sigma^+ = \Sigma^\star \setminus \{\varepsilon\}$. A *language* is a subset of $\Sigma^\star$ and will usually be denoted by L.

2.2 Formal Logic

In order to describe properties of words in formal logic, *first-order* and *monadic second-order formulas* will be used, with the latter being second-order formulas wherein quantification of second-order variables is allowed over unary second-order variables, i. e. set variables, only.

A *vocabulary* is a finite collection of constant, relation (or predicate) and function symbols. A *structure* over a vocabulary σ consists of a set $\mathbb{U}$, called *universe*, together with an interpretation $\sigma^{\mathfrak{A}}$ of

- each constant symbol c in σ as an element $c^{\mathfrak{A}}$ in the universe $\mathbb{U}$,
- each k-ary relation symbol R from σ as a k-ary relation $R^{\mathfrak{A}} \subseteq \mathbb{U}^k$, and
- each k-ary function symbol f from σ as a function $f^{\mathfrak{A}} \colon \mathbb{U}^k \to \mathbb{U}$.

A structure like the above will be denoted by $\mathfrak{A} = \langle \mathbb{U}, \sigma^{\mathfrak{A}} \rangle$, where $\sigma^{\mathfrak{A}}$ may be replaced by the symbols contained, for convenience. Further on, elements of the universe $a \in \mathbb{U}$ will also be referred to as elements of the structure $a \in \mathfrak{A}$; and the superscript distinguishing a symbol from its interpretation is going to be omitted from now on—the meaning will be clear from the context.

Let σ be a vocabulary. An *atomic formula* over σ is of the form $P(x_1, \ldots, x_k)$, where $x_1, x_2, \ldots, x_k$ are variables and P is a k-ary predicate symbol in σ. For example, for a variable x and a letter b, $Q_{\mathrm{b}}(x)$ is an atomic formula. *First-order formulas* over σ are build from atomic formulas in the usual way, using the connectives and ($\wedge$), or ($\vee$), negation ($\neg$) and universal ($\forall x$) and existential ($\exists x$) quantifiers. *Second-order formulas* are build analogously, additionally using second-order quantification over relations ($\exists X$, $\forall X$). Variable occurrences that are quantified will be called *bound*; occurrences of unbound variables also will be called *free*. If $x_1, \ldots, x_k$ are all the free first-order variables and $X_1, \ldots, X_l$ are all the free second-order variables of a formula φ, then it will be denoted by $\varphi(x_1, \ldots, x_k, X_1, \ldots, X_l)$. A *sentence* is a formula without free first- and second-order variables. A structure $\mathfrak{A}$ over the vocabulary σ is said to model a sentence φ over σ, written $\mathfrak{A} \models \varphi$, if φ is satisfied by interpreting each symbol in σ by its interpretation $\sigma^{\mathfrak{A}}$. Two formulas φ_1, φ_2 are said to be *(logically) equivalent* if $\mathfrak{A} \models \varphi_1 \iff \mathfrak{A} \models \varphi_2$ for all structures $\mathfrak{A}$; this will be written $\varphi_1 \equiv \varphi_2$.

Moreover, in first-order and monadic second-order logic, every occurrence of a k-ary function $f\colon \mathbb{U}^k \to \mathbb{U}$ can be substituted by a $k+1$-ary predicate $R_f = \{(x, f(x)) \mid x \in \mathbb{U}^k\}$ so that the original and modified formula is equivalent, hence vocabularies may be restricted to constant and predicate symbols, the *relational vocabularies*.

Let Σ be an alphabet. Each word $w = a_1 a_2 \cdots a_n \in \Sigma^\star$ is identified with a structure over a relational vocabulary σ, called the *word model* $\underline{w}$, that consists of the universe $\mathbb{U} = \{1, \ldots, n\}$ (roughly speaking the set of positions in w) and an interpretation $\sigma^{\underline{w}}$ of the symbols in σ. If $w = w_1 \cdots w_n$ is a word over an alphabet Σ, then the vocabulary σ of the word model $\underline{w}$ will always contain the unary predicate symbols Q_a, $a \in \Sigma$, with the fixed interpretation

$$\underline{w} \models Q_a(x) \iff w_x = a$$

for $x \in \{1, \ldots, n\}$, and the binary predicate symbol $=$ with the usual interpretation

$$\underline{w} \models x = y \iff x = y$$

for $x, y \in \{1, \ldots, n\}$. For example, over $\Sigma = \{\mathrm{a}, \mathrm{b}\}$, the word ababb corresponds to the word model $\underline{\mathrm{ababb}} = \langle \{1,2,3,4,5\}, =, Q_\mathrm{a}, Q_\mathrm{b} \rangle$, where $Q_\mathrm{a} = \{1,3\}$ and $Q_\mathrm{b} = \{2,4,5\}$. Since the main interest of this thesis is almost exclusively on word models, the predicate symbols Q_a, $a \in \Sigma$, and $=$ will be omitted when vocabularies are written out. Hence, the above word model over a vocabulary σ containing the symbols $Q_\mathrm{a}, <, +$ and $=$ will be denoted by $\langle \{1,2,3,4,5\}, +, < \rangle$.

A special kind of structures will be defined in order to ease handling of formulas with free variables: Let $\underline{w}$ be a word model over the vocabulary σ and V a structure with the same universe over a vocabulary σ', distinct from σ, consisting only of constant symbols. A $(\underline{w}, V)$-*structure* is a structure over symbols of both vocabularies, a vocabulary denoted by $\sigma \cup \sigma'$, in which every symbol from σ is interpreted as in $\underline{w}$ and every constant in σ' in interpreted as in V. Then, for any formula $\varphi(x_1, \ldots, x_k)$ over σ there exists a formula ϕ over an vocabulary $\sigma \cup \sigma'$ with k new constant symbols

$c_1, \ldots, c_k$, that is built from φ by replacing each free variable $x_i, 1 \leq i \leq k$, by the constant symbol c_i, such that for all $a_1, \ldots, a_k \in \{1, \ldots, |w|\}$

$$\underline{w} \models \varphi(a_1, \ldots, a_k) \iff (\underline{w}, V) \models \phi,$$

with $c_i^V = a_i, 1 \leq i \leq k$; for the sake of clarity, $(\underline{w}, V)$ will also be written with as $(\underline{w}, \vec{c})$. Then $(\underline{w}, V)$ can be interpreted as a word

$$(w_1, V_1) \cdots (w_n, V_n)$$

over the alphabet $\Sigma \times \mathfrak{P}(\{c_i \mid 1 \leq i \leq k\})$, where $w = w_1 \cdots w_n$ and $V_1 \cup \cdots \cup V_n$ is the partitioning of $\{c_i | 1 \leq i \leq k\}$ that satisfies $c_i^V = a_i \Rightarrow c_i \in V_{a_i}$. This notation will prove useful when defining numerical relations through automata.

To improve legibility, the predicate symbols possessing arithmetic equivalents will be written in the natural fashion. For example, the predicate symbols $<$, $=$ will be written in infix notation; the successor predicate will be denoted by $x + 1 = y$, for variables x, y.

Let φ be a sentence. The *language defined by* φ is

$$L(\varphi) = \left\{ w \in \Sigma^* \,\middle|\, \underline{w} \models \varphi \right\}.$$

The class of all languages definable by a first-order formula over a vocabulary σ is denoted by FO[σ]. Analogously, the class of all languages definable in monadic second-order will be denoted by MSO[σ]. Occasionally FO[σ] and MSO[σ] will also be used to refer to the underlying logic.

2.3 Ehrenfeucht-Fraïssé Games

Ehrenfeucht-Fraïssé games are a powerful tool to gain inexpressibility results for first-order logic. The game is played in rounds by two players, the spoiler and the duplicator. Given a pair of structures $(\mathfrak{A}, \mathfrak{B})$, the spoiler will try to show these structures are different while the duplicator will try to show they are the same. In an r-round game each round i, $1 \leq i \leq r$, consists of the following steps:

1. The spoiler chooses a structure ($\mathfrak{A}$ or $\mathfrak{B}$) and picks an element $a_i \in \mathfrak{A}$ or $b_i \in \mathfrak{B}$.

2. The Duplicator responds by picking an element in the other structure, $b_i \in \mathfrak{B}$ or $a_i \in \mathfrak{A}$, respectively.

Over strings one can interpret a player's move as the placement of a pebble on that position. To define the winner of a game, the notation of partial isomorphisms is required:

Definition 2.1. *Given two structure* $\mathfrak{A}, \mathfrak{B}$ *over the same relational vocabulary* σ *and tuples* $\vec{a} = (a_1, \cdots a_r)$ *in* $\mathfrak{A}$ *and* $\vec{b} = (b_1, \cdots b_r)$ *in* $\mathfrak{B}$, $(\vec{a}, \vec{b})$ *defines a* partial isomorphism *between* $\mathfrak{A}$ *and* $\mathfrak{B}$ *if*

- *for all* $1 \leq i, j \leq r$, $a_i = a_j$ *iff* $b_i = b_j$,
- *for every constant symbol* c *from* σ *and all* $1 \leq i \leq r$, $a_i = c_i^{\mathfrak{A}}$ *iff* $b_i = c_i^{\mathfrak{B}}$, *and*
- *for every* k*-ary predicate symbol* P *from* σ *and all* $i_j \in [1, r]$, $1 \leq j \leq k$,
$$(a_{i_1}, \ldots, a_{i_k}) \in P^{\mathfrak{A}} \text{ iff } (b_{i_1}, \ldots, b_{i_k}) \in P^{\mathfrak{B}}.$$

Returning to Ehrenfeucht-Fraïssé game, the duplicator wins the r-round game with moves $\vec{a}$ in $\mathfrak{A}$ and moves $\vec{b} \in \mathfrak{B}$ if and only if $((\vec{a}, \vec{c^{\mathfrak{A}}}), (\vec{b}, \vec{c^{\mathfrak{B}}}))$ is a partial isomorphism between the structures $\mathfrak{A}$ and $\mathfrak{B}$, where $\vec{c} = (c_1, \ldots, c_k)$ is the tuple of all constant symbols in σ, the vocabulary of the structures. If the duplicator can win the r-round game on the pair of σ-structures $(\mathfrak{A}, \mathfrak{B})$, he is said to have a *winning strategy* (in the σ-game on $(\mathfrak{A}, \mathfrak{B})$); this will be denoted by $\mathfrak{A} \sim_r^{\sigma} \mathfrak{B}$. If the vocabulary is clear from the context, the superscript σ will be omitted.

Informally an Ehrenfeucht-Fraïssé game is won by the duplicator if he can respond to the moves of the spoiler in such a way that, after each round, the pebbles placed in the first structure and the pebbles placed in the second structure behave in exactly the same way with respect to the predicates in the vocabulary.

Let the *quantifier rank* of a formula φ denote the maximum depth of quantifier nesting in φ. Two structures $\mathfrak{A}$ and $\mathfrak{B}$ are said to *agree* on a set $\mathcal{S}$ of formulas if for all $\varphi \in \mathcal{S} : \mathfrak{A} \models \varphi \iff \mathfrak{B} \models \varphi$.

Proposition 2.2 (Ehrenfeucht-Fraïssé, cf. [Lib04]). *Let $\mathfrak{A}$ and $\mathfrak{B}$ be structures over the vocabulary σ. Then $\mathfrak{A} \sim_r^\sigma \mathfrak{B}$ if and only if $\mathfrak{A}$ and $\mathfrak{B}$ agree on all $\varphi \in \mathrm{FO}[\sigma]$ with quantifier rank $\leq r$.*

Hence, $\sim_r^\sigma$ can also be interpreted as an equivalence relation on structures. Within this interpretation each move in a game corresponds to an operation introducing a new constant symbol: Given a pair of structures $(\mathfrak{A}, \mathfrak{B})$ and the player's moves $\vec{a}$ and $\vec{b}$, an r-round Ehrenfeucht-Fraïssé game generates a family of structures $((\mathfrak{A}_i, \mathfrak{B}_i))_{0 \leq i \leq r}$, where $(\mathfrak{A}_0, \mathfrak{B}_0) = (\mathfrak{A}, \mathfrak{B})$ and, for $0 \leq i < r$, $(\mathfrak{A}_{i+1}, \mathfrak{B}_{i+1})$ is obtained from $(\mathfrak{A}_i, \mathfrak{B}_i)$ by inserting the constant symbol c_i into the vocabulary and the interpretations $c_i^{\mathfrak{A}} = a_i$, $c_i^{\mathfrak{B}} = b_i$ into the structures. With this, the winning condition from above may be reformulated: the duplicator wins a r-round Ehrenfeucht-Fraïssé game, if $\mathfrak{A}_r \sim_0 \mathfrak{B}_r$.

To finally relate the inexpressibility of languages in first-order logic to Ehrenfeucht-Fraïssé games, the following corollary is stated.

Corollary 2.3. *Let $L \subseteq \Sigma^\star$ be a language. Then $L \notin \mathrm{FO}[\sigma]$ iff, for all $k \in \mathbb{N}$, there exist strings $u \in L, v \notin L$ such that $\underline{u} \sim_k^\sigma \underline{v}$.*

Proof. Assume there exists a $k \in \mathbb{N}$ so that for all strings $u \in L, v \notin L$, $\underline{u} \nsim_k^\sigma \underline{v}$ holds. Hence, for every pair $u \in L, v \notin L$, there is a formula $\varphi_{u,v}$ with quantifier rank $\leq k$ such that $\underline{u} \models \varphi_{u,v} \iff \underline{v} \not\models \varphi_{u,v}$. There are only finitely many $\mathrm{FO}[\sigma]$-formulas of quantifier rank $\leq k$. This leads to $L(\varphi) = L$ for the conjunction of all such formulas $\varphi_{u,v}$. The opposite gives the direction from left to right.

For the direction from right to left, let $k \in N$ be an arbitrary number. Let $u \in L, v \notin L$ be the strings with $u \sim_k^\sigma v$. The latter enforces $\underline{u} \models \varphi \iff \underline{v} \models \varphi$ for every formula φ of quantifier rank $\leq k$. Hence, no first-order formula over σ defines L, for k can be chosen arbitrary. □

Henceforth, Ehrenfeucht-Fraïssé games will be considered over word models mostly. On account for this, the natural extension of $\sim_k^\sigma$ on strings is introduced: let Σ be an alphabet and let $u, v \in \Sigma^\star$, then $u \sim_k^\sigma v \iff \underline{u} \sim_k^\sigma \underline{v}$.

2.4 Grammars and Automata

Grammars are a constructive method to describing languages. A *grammar* is a quadruple $G = (V, \Sigma, P, S)$, where

- V is a finite set of *variables* (or *nonterminal symbols*),
- Σ is an alphabet, w. l. o. g. $\Sigma \cap V = \emptyset$,
- $P \subseteq (\Sigma \cup V)^+ \times (\Sigma \cup V)^\star$ is a finite set of *productions* (or *rules*), and
- $S \in N$ is the starting symbol.

Grammars are arranged hierarchically according to the difficulty of their productions. The classes of that hierarchy, the Chomsky hierarchy, will be briefly recalled: A grammar is called *context-sensitive* if $\beta \in (\Sigma \cup V \setminus \{S\})^\star$ and either $|\alpha| \leq |\beta|$ or $\alpha = S$ for all $(\alpha, \beta) \in P$. A context-sensitive grammar is called *context-free* if $\alpha \in V$ for all $(\alpha, \beta) \in P$; that is $P \subseteq (V \times (\Sigma \cup V \setminus \{S\})^+) \cup \{(S, \varepsilon)\}$. Finally, a context-free grammar is called *regular* if $\beta \in \Sigma \cup \Sigma \cdot V$ or $P \subseteq (V \times (\Sigma \cup (\Sigma \cdot V))) \cup \{(S, \varepsilon)\}$, respectively. We will also write $\alpha \to \beta \in P$ instead of $(\alpha, \beta) \in P$.

To define the language generated by a grammar, let $\alpha, \beta \in (V \cup \Sigma)^\star$. β can be *derived* from α, written $\alpha \Rightarrow_G^1 \beta$ or $\alpha \Rightarrow_G \beta$, if $\alpha = \alpha_1 A \alpha_2$, $\beta = \alpha_1 \gamma \alpha_2$ and $A \to \gamma \in P$ for some $\alpha_1, \alpha_2 \in (V \cup \Sigma)^\star$. Now, with $\alpha \Rightarrow_G^0 \alpha$ for all $\alpha \in (\Sigma \cup V)^\star$, the relations $\Rightarrow_G^{i+1}$, $i > 1$, and $\Rightarrow_G^\star$ can defined as

$$\alpha \Rightarrow_G^{i+1} \beta \iff \exists \gamma \in (\Sigma \cup V)^\star \colon \alpha \Rightarrow_G^i \gamma \wedge \gamma \Rightarrow_G \beta$$

and the reflexive transitive closure of $\Rightarrow_G$, respectively. The language generated by a grammar $G = (V, \Sigma, P, S)$ is defined as

$$L(G) = \{w \in \Sigma^\star \mid S \Rightarrow_G^\star w\}.$$

A language will be called *context-sensitive/context-free/regular* if it admits a context-sensitive/context-free/regular grammar. The set of all context-sensitive/context-free/regular languages will be denoted by *CSL*/*CFL*/*REG*, the class of all context-free grammars by *CFG*.

Aside from grammars there are formalisms to describe the language classes with simplified computer models. A *finite automaton* (or *FA*) is a quintuple $A = (Q, \Sigma, \delta, q_0, F)$, where

- Q is the finite set of *states*,
- Σ is the finite set of the *input alphabet*,
- $\delta\colon Q \times \Sigma \to \mathfrak{P}(Q)$ is the *transition function*,
- $q_0 \in Q$ is the *starting state*, and
- $F \subseteq Q$ is the set of *final states*.

The extended transition function $\hat{\delta}\colon Q \times \Sigma^\star \to \mathfrak{P}(Q)$ is defined inductively by $\hat{\delta}(q, a) = \delta(q, a)$ for $q \in Q, a \in \Sigma$ and $\hat{\delta}(q, aw) = \hat{\delta}(\delta(q, a), w)$ for $q \in Q, a \in \Sigma, w \in \Sigma^\star$. The language accepted by a FA A is defined as

$$L(A) = \{w \in \Sigma^\star \mid \hat{\delta}(q_0, w) \cap F \neq \emptyset\}.$$

The set of all languages accepted by finite automata equals the set of regular languages.

A *pushdown automata* (or *PDA*) is a septuple $P = (Q, \Sigma, \Gamma, \delta, q_0, \bot, F)$, where

- Q is the finite set of *states*,
- Σ is the finite set of the *input alphabet*,

- Γ is the finite set of the *stack alphabet*,
- $\delta\colon Q \times (\Sigma \times \{\varepsilon\}) \times \Gamma \to \mathfrak{P}(Q \times \Gamma^\star)$, with $|\delta(q, a, A)|$ finite for all $q \in Q, a \in \Sigma, A \in \Gamma$, is the *transition function*,
- $q_0 \in Q$ is the *starting state*,
- $\bot \in \Gamma$ is the *initial stack symbol*, and
- $F \subseteq Q$ is the set of *final states*.

Just like finite automata, pushdown automata are used to decide membership in languages in the sense that accepted words belong to it. But contrary to finite automata, pushdown automata have the ability to store information on a stack. Since this information influences their behavior on strings, a formalism beyond $\hat{\delta}$ is required that takes the stack into consideration.

A *configuration* of a PDA P is a triple $(q, w, \alpha) \in Q \times \Sigma^\star \times \Gamma^\star$, where q is the state that P is in, w is the rest of the input word and $\alpha = A_1 \cdots A_n \in \Gamma^\star$ is the contents of the stack with A_1 being on top. The behavior of P is then described using the *transition relation* $\vdash_P$ induced by δ: Let $(q, aw, A\alpha)$ and $(q', w, \beta\alpha)$ be configurations of P, where $a \in \Sigma \cup \{\varepsilon\}, w \in \Sigma^\star, A \in \Gamma, \alpha, \beta \in \Gamma^\star$, then

$$(q, aw, A\alpha) \vdash_P (q', w, \beta\alpha) \iff (q', \beta) \in \delta(q, a, A).$$

The reflexive transitive closure of $\vdash_P$ will be denoted by $\vdash_P^\star$.

Given a PDA P and a word w, called *input word*, P is initialized in configuration $(q_0, w, \bot)$; w is *accepted* by P if and only if $(q_0, w, \bot) \vdash_P^\star (q_0, \varepsilon, \gamma)$ for some $\gamma \in \Gamma^\star$. The *language accepted* by P is the set

$$L(P) = \{w \in \Sigma^\star \mid (q_0, w, \bot) \vdash_P^\star (q_0, \varepsilon, \gamma), \gamma \in \Gamma^\star\}.$$

It holds that the class of languages acceptable with PDAs is exactly CFL.

If a PDA P satisfies

$$|\delta(q, a, A)| + |\delta(q, \varepsilon, A)| \leq 1,$$

for all $q \in Q, a \in \Sigma, A \in \Gamma$, then P is said to be *deterministic*. The set of languages accepted by deterministic PDAs will be denoted DCFL, the set of all deterministic context-free languages.

Let $q, q' \in Q, a \in \Sigma, A, B_1, \cdots, B_k \in \Gamma$. A transition $(q', B_1 \cdots B_k) \in \delta(q, a, A)$ with $k > 1$ is equivalent to the stack operation $push(B_1 \cdots B_k)$ (i. e. an operation, in which the string $B_1 \cdots B_k$ is put on top of a stack); a transition $(q', \varepsilon) \in \delta(q, a, A)$ is equivalent to the stack operation $pop(A)$ (i. e. an operation that removes the topmost symbol). A *turn* in the operation of a PDA P is a transition $c' \vdash\!\!\!-_P c''$ such that $c \vdash\!\!\!-_P c' \vdash\!\!\!-_P c''$, where $c \vdash\!\!\!-_P c'$ is a push-operation and $c' \vdash\!\!\!-_P c''$ is a pop-operation, or vice versa.

3 Definability of Languages in FO[+]

Context-free languages have been shown to coincide with the class of languages defined by sentences of the form $\exists M\varphi$, where $\varphi \in \mathrm{FO}[<]$ and M is a binary second order variable restricted to the class of matchings [LST95].

Since this approach yields a semantic restriction on the quantification, it is reasonable to investigate which context-free languages can be captured by first-order logic without semantic restrictions when extended with a numerical predicate that can relate positions in a similar way as matchings do. Proper candidates are numerical predicates definable in CFL; this preserves the possibility of showing equality of the extended first-order logic with the subclass of CFL to identify later on.

3.1 Numerical Predicates Expressible in CFL

Let Σ be an alphabet and R be a k-ary numerical relation. R is said to be *definable* in REG (CFL resp.) if there is a FA A (PDA P resp.) that accepts the set of all $(\underline{w}, \vec{c})$-structures satisfying R. That is in the case of REG

$$L(A) = \left\{(\underline{w}, \vec{c}) \in \Sigma^\star \times \mathfrak{P}(\{c_i \mid 1 \leq i \leq k\}) \;\middle|\; \underline{w} \models R(c_1, \ldots, c_k)\right\}.$$

The atomic formulas $x = y$, $Q_a(x)$ for all $a \in \Sigma$, the natural order-predicate $x < y$ and the natural modulo-predicate $x \equiv y \pmod z$ are definable in REG (e. g. figure 3.1; for more information, cf. [Str74]), whereas the natural addition $x + y = z$ is not definable in REG:

Assume $x + y = z$ is definable by a formula $\varphi(x, y, z) \in \mathrm{FO}[<]$ and let

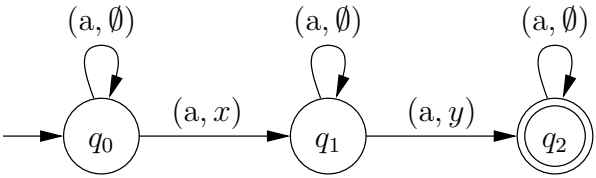

Figure 3.1: A finite automaton defining the numerical relation $x < y$ over $\Sigma = \{\mathrm{a}\}$.

$\underline{w}$ be a word model. Then the formula

$$\phi(x) \equiv \exists z \left(\forall y(y \leq z) \wedge \varphi(x, x, z)\right) \tag{3.1}$$

is true for the position $x = \frac{|w|}{2}$ if $|w|$ is even; $\phi(x)$ is inconsistent otherwise. Thus the sentence

$$\exists x \left(\phi(x) \wedge \forall y(y \leq x \rightarrow Q_{\mathrm{a}}(x)) \wedge \forall y(y > x \rightarrow Q_{\mathrm{b}}(x))\right)$$

defines the language $\{\mathrm{a}^n\mathrm{b}^n \mid n > 0\} \notin \mathrm{REG}$, what contradicts the assumption that addition is definable in REG. Nevertheless, the relation $x + y = z$ is definable in CFL:

Let $\Sigma = \{\mathrm{a}\}$, $w = w_1 \cdots w_n \in \Sigma^\star$ and V be a structure containing the constant symbols x, y and z. Define a PDA P over the alphabet $\Sigma \times \mathfrak{P}(\{x, y, z\})$ such that $(w_1, V_1) \cdots (w_n, V_n)$ is accepted if and only if $(V_i)_{i \in \{1,\ldots,n\}}$ is a partition of $\{x, y, z\}$ and $x \in V_i, y \in V_j, z \in V_k \Longrightarrow i + j = k$:

$$P = \big(\{q_0, q_1, q_2, q_3\}, \Sigma \times \mathfrak{P}(\{x, y, z\}), \{0, \bot\}, \delta, q_0, \bot, \{q_3\}\big),$$

where

$$\begin{aligned}
\delta(q_0, (\mathrm{a}, \emptyset), \bot) &= (q_0, 0\bot), & \delta(q_1, (\mathrm{a}, \emptyset), 0) &= (q_1, 0),\\
\delta(q_0, (\mathrm{a}, \{x\}), \bot) &= (q_1, 0\bot), & \delta(q_1, (\mathrm{a}, \{y\}), 0) &= (q_2, \varepsilon),\\
\delta(q_0, (\mathrm{a}, \{x, y\}), \bot) &= (q_2, \bot), & \delta(q_2, (\mathrm{a}, \emptyset), 0) &= (q_2, \varepsilon),\\
\delta(q_0, (\mathrm{a}, \emptyset), 0) &= (q_0, 00), & \delta(q_2, (\mathrm{a}, \{z\}), \bot) &= (q_3, \bot),\\
\delta(q_0, (\mathrm{a}, \{x\}), 0) &= (q_1, 00), & \delta(q_3, (\mathrm{a}, \emptyset), \bot) &= (q_3, \bot),\\
\delta(q_0, (\mathrm{a}, \{x, y\}), 0) &= (q_2, 0).
\end{aligned}$$

On the other hand, first-order logic extended with addition over integers yields the expressive power to capture languages in CFL. For example, formula (3.1), with $\varphi(x, x, z)$ replaced by $x + x = z$, defines $\{\mathrm{a}^n\mathrm{b}^n \mid n > 0\} \in \mathrm{CFL} \setminus \mathrm{REG}$. Therefore, addition is a sensible candidate for further investigation.

3.2 Introducing First-Order Logic with Addition

First-order logic with addition, FO[+], refers to the class of languages definable by first-order formulas over the vocabulary $+$, $=$ and Q_a, for all $a \in \Sigma$, where the interpretation of $=$ and Q_a is as determined in section 2.2 and the interpretation of $+$ is the natural addition: let $w \in \Sigma^\star$ and x, y, z be constant symbols, then

$$(\underline{w}, x, y, z) \models x + y = z \iff x^{(\underline{w},x,y,z)} + y^{(\underline{w},x,y,z)} = z^{(\underline{w},x,y,z)}.$$

In order to relate FO[+] to other first-order logics, a brief survey of the elementary results connecting languages and logics will be given.

3.2.1 Logics for Regular Languages

The logics to be surveyed are the first- and monadic second-order logic with successor, FO[+1] and MSO[+1], and first-order logic together with the natural ordering, FO[<]. The successor predicate $+1$ as well as the order predicate $<$ are hereby interpreted as

$$(\underline{w}, x, y) \models x + 1 = y \iff x^{(\underline{w},x,y)} + 1 = y^{(\underline{w},x,y)}$$

and

$$(\underline{w}, x, y) \models x < y \iff x^{(\underline{w},x,y)} < y^{(\underline{w},x,y)}.$$

They have been of quite an interest due to their connection to regular languages and boolean circuit classes. To present the results, some definitions have yet to be made.

Definition 3.1. *A* star-free regular expression *over an alphabet Σ is an expression built from the symbols $\emptyset$ and a, for each $a \in \Sigma$, using the operations concatenation $(\cdot)$, union $(+)$ and complement $(^-)$.*

The language $L(\alpha)$ defined by a star-free regular expression α is defined as $L(\emptyset) = \emptyset$, $L(a) = \{a\}$, for all $a \in \Sigma$, $L(\alpha_1 + \alpha_2) = L(\alpha_1) \cup L(\alpha_2)$, $L(\alpha_1 \cdot \alpha_2) = L(\alpha_1) \cdot L(\alpha_2)$, and $L(\overline{\alpha}) = \overline{L(\alpha)}$.

A language defined by a star-free regular expression will be denoted as a star-free language *and the set of all such languages as* star-free REG.

Now the results crucial to this context are summarized by the following theorem. Their proofs will be omitted, instead the ideas will be sketched and references be given.

Theorem 3.2. *1. The set of strings of even length is not definable in* FO[<].

2. $\mathrm{FO}[+1] \subsetneq \mathrm{FO}[<] = \text{star-free REG} \subsetneq \mathrm{REG} = \mathrm{MSO}[+1]$.

Proof sketch. 1. Let Σ be an alphabet. The set of strings of even length is the language $L = \left\{ w \in \Sigma^\star \,\middle|\, |w| \equiv 0 \pmod 2 \right\}$. Assume $L = L(\varphi)$ for some $\varphi \in \mathrm{FO}[<]$ with quantifier rank r, and consider the words a^{2^r} and a^{2^r+1}, for some $a \in \Sigma$. There is a winning-strategy for the duplicator in the r-round Ehrenfeucht-Fraïssé game on (a^{2^r}, a^{2^r+1}). Hence $\underline{a^{2^r}} \models \varphi \iff \underline{a^{2^r+1}} \models \varphi$, what gives a contradiction.

This winning strategy of the duplicator ensures that, after each round l, the distances between corresponding pebbles in the two word models are large enough not be be distinguished by another $r - l$ rounds. In particular, let $(z_1, \ldots, z_l)$ and $(z'_1, \ldots, z'_l)$ be the moves in a^{2^r} and a^{2^r+1}, respectively; and let $z_{-1} = 1$, $z'_{-1} = 1$, $z_0 = 2^r$, $z'_0 = 2^r + 1$ be the minimal and maximal positions in the word models. Then, for all $-1 \leq i, j \leq l$,

- if $|z_i - z_j| < 2^{r-l}$ or $|z'_i - z'_j| < 2^{r-l}$, then $|z_i - z_j| = |z'_i - z'_j|$,
- $|z_i - z_j| \geq 2^{r-l}$ iff $|z'_i - z'_j| \geq 2^{r-l}$,

- $z_i \leq z_j$ iff $z'_i \leq z'_j$.

This claim can be proven by induction on r and is carried out in [Lib04, Theorem 3.6]. Another proof is shown in [Str74, Theorem 2.1].

2. "FO[+1] $\subsetneq$ FO[<]": The successor predicate is definable in FO[<] via the formula $\varphi(x, y) \equiv x < y \wedge \neg\exists z(x < z \wedge z < y)$, thus FO[+1] $\subseteq$ FO[<]. To separate these classes from each other, an equivalence relation $\approx_r^k$ on $\Sigma^\star$ is defined: $w_1 \approx_r^k w_2$ if and only if w_1 and w_2 have the same prefix and suffix of length $k-1$ and every substring v of length $|v| \leq k$ does occurs either in both words $\geq r$-times or the same number of times.

   The languages definable in FO[+1] coincide with unions of equivalence classes of $\approx_r^k$. But since $\mathrm{a}^k\mathrm{ba}^k\mathrm{ca}^k \approx_r^k \mathrm{a}^k\mathrm{ca}^k\mathrm{ba}^k$ for all $r > 0$ and k chosen large enough, the language $L(\mathrm{a}^\star\mathrm{ba}^\star\mathrm{ca}^\star) \in$ FO[<] is not definable in FO[+1]. [Str74, chapter IV.3] contains the necessary details.

   "FO[<] = star-free REG": Both directions are proven by induction on the composition of either star-free regular expressions or first-order formulas. For example, $\emptyset$ is defined by the formula $\forall x(x \neq x)$; for $a \in \Sigma$, $\{a\}$ is defined by $\exists x(\forall y(y = x) \wedge Q_a(x))$; and for star-free expressions e_1, e_2 with $L(e_i) = L(\varphi_i), \varphi_i \in$ FO[<] for $i \in \{1, 2\}$, $\overline{e_1}$ is defined by $\neg\varphi_1$, $e_1 + e_2$ is defined by $\varphi_1 \vee \varphi_2$ and $e_1 \cdot e_2$ is defined by $\exists x\varphi'_1 \wedge \varphi'_2$, where φ'_1 is constructed from φ_1 by replacing each quantifier $Qy\phi$, $Q \in \{\exists, \forall\}$ with $Qy(y \leq x \wedge \phi)$, and φ'_2 is constructed from φ_2 by replacing each quantifier $Qy\phi$ with $Qy(y > x \wedge \phi)$. See [Lib04, Theorem 7.26] for the complete proof.

   "star-free REG $\subsetneq$ REG": As seen in theorem 3.2-1 the language $L((\mathrm{aa})^\star)$ is not definable in FO[<], ergo not star-free. This ensues the claim.

   "REG = MSO[+1]": This result is due to Büchi. First let L be a

regular language recognized by the finite automaton A with states $\{q_0, \ldots, q_k\}$. L is defined by an MSO[+1]-formula that expresses the existence of a valid computation path on inputs belonging to L. That is, if A accepts input $w, |w| = n$, then

a) there exists a partitioning $X_0 \cup \cdots \cup X_{q-1}$ of $\{1, \ldots, n\}$ so that $j \in X_i$ if and only if A is in state i after reading the jth letter of w,

b) after reading the first letter, A enters the state q_k with $1 \in X_k$,

c) the transitions between different X_i's are consistent with the automaton, and

d) the automaton enters an accepting state after reading the last letter.

The conditions enumerated above can be checked in FO[<] by formulas $\varphi_i, i \in \{\mathrm{a}, \mathrm{b}, \mathrm{c}, \mathrm{d}\}$. Hence the formula $\varphi \in \mathrm{MSO}[+1]$ defining L is of the form

$$\exists X_0 \cdots \exists X_{q-1} \varphi_\mathrm{a} \wedge \varphi_\mathrm{b} \wedge \varphi_\mathrm{c} \wedge \varphi_\mathrm{d},$$

where $X_i, 0 \leq i < q$ are the set variables.

For the converse direction a finite automaton is constructed by induction on the composition of the defining formula. For details, consult [Str74, Theorem III.1.1] or [Lib04, Theorem 7.21].

□

Returning to first-order logic with addition, the relation of FO[<] to FO[+] will be determined next.

Theorem 3.3. $\mathrm{FO}[<] \subsetneq \mathrm{FO}[+]$.

Proof. The inclusion follows from the logical equivalence of the formulas $x < y$ and $\exists z(x + z = y)$ over initial segments of $\mathbb{N}$, i. e. over sets $\{1, \ldots, n\}$ for some arbitrary $n \in \mathbb{N}$.

Furthermore, strictness is gained from the definability of the set of even-length strings: over every alphabet Σ the formula

$$\varphi \equiv \exists z\big(\forall x(x \leq z) \wedge \exists y(y + y = z)\big)$$

defines the language $L(\varphi) = \{w \in \Sigma^\star \,|\, |w| \equiv 0 \pmod 2\}$. □

3.2.2 Languages in FO[+]

The fact that FO[+] exceeds FO[<] in terms of definable languages raises the question for examples. This subsection is devoted to presenting some interesting languages definable in FO[+].

Example 3.4. The language $L_1 = \{\mathrm{a}^n\mathrm{b}^n \,|\, n > 0\}$ is definable in FO[+] by the formula

$$\exists x \exists y\Big(x + x = y \wedge \forall z\big(z \leq y \wedge (z \leq x \to Q_\mathrm{a}(z)) \wedge (z > x \to Q_\mathrm{b}(z))\big)\Big).$$

L_1 lies in CFL $\setminus$ REG.

Example 3.5. The language $L_2 = \{\mathrm{a}^n\mathrm{b}^n\mathrm{c}^n \,|\, n > 0\}$ is definable in FO[+] by the formula

$$\begin{aligned}\exists x_1 \exists x_2 \exists y\big(x_1 + x_1 = x_2 \wedge x_1 + x_2 = y \wedge \forall z\big(z \leq y \wedge (z \leq x_1 \to Q_\mathrm{a}(z)) \wedge{}\\ (x_1 < z \leq x_2 \to Q_\mathrm{b}(z)) \wedge{}\\ (x_2 < z \to Q_\mathrm{c}(z))\big)\big).\end{aligned}$$

L_2 lies in $\mathfrak{B}(\mathrm{DCFL}) \setminus \mathrm{CFL}$: it is the intersection of $\{\mathrm{a}^n\mathrm{b}^n\mathrm{c}^m \,|\, n, m > 0\} \in \mathrm{DCFL}$ and $\{\mathrm{a}^n\mathrm{b}^m\mathrm{c}^m \,|\, n, m > 0\} \in \mathrm{DCFL}$; and $L_2 \notin \mathrm{CFL}$, as commonly known.

Example 3.6. The language $L_3 = \{ww \,|\, w \in \Sigma^\star\}$ is expressible in FO[+] by the formula

$$\begin{aligned}\exists x\Big(&\exists y\big(x + x = y \wedge \forall z(z \leq y)\big) \wedge\\ &\forall y\big(y \leq x \to \exists z\big(x + y = z \wedge \textstyle\bigwedge_{a\in\Sigma} Q_a(y) \leftrightarrow Q_a(z)\big)\big)\Big).\end{aligned}$$

L_3 lies in $\mathfrak{B}(\mathrm{CFL}) \setminus \mathrm{CFL}$: $\overline{L_3} \in \mathrm{CFL}$ and $L_3 \notin \mathrm{CFL}$.

3.3 Limitations to the Definability of FO[+]

As seen in the examples 3.5 and 3.6, the expressive power of FO[+] exceeds that of the context-free languages. But where are the bounds to the definability of FO[+]? And are there yet regular languages that can not be expressed in FO[+]? The latter question will also be tackled in section 3.5. Now indefinable languages and an upper bound for the languages definable in FO[+] will be presented.

3.3.1 Indefinable Languages

The extension of FO[<] with addition introduces positional argumentation. One can assume that in presence of a neutral letter the power gained by addition is rendered useless, as positions of letters may be shifted by insertion and deletion of the neutral letter. The generalization of the above assumption to arbitrary numerical predicates is the Crane Beach Conjecture [BIL$^+$01, Sch01]. It poses a central tool for indefinability results of languages in logics.

Definition 3.7. *Let $L \subseteq \Sigma^\star$ be a language. $e \in \Sigma$ is called* neutral letter *for L if for all $u, v \in \Sigma^\star$: $uv \in L \iff uev \in L$.*

Theorem 3.8 (Crane Beach Conjecture in the case of FO[+])**.** *Every language $L \in$ FO[+] that has a neutral letter is definable in* FO[<].

The actual proof given in [BIL$^+$01] follows from collapse results for first-order queries over finite databases given in [LS01]. A suitable reformulation of which will be given below preceding the proof of theorem 3.8. The results of [LS01] are based again on a result worked out in [Lyn82] stating that for each first-order formula φ with addition there is an infinite set $Q \subseteq \mathbb{N}$ such that φ can not distinguish between subsets of Q. Details of this proof will become important in the proof of theorem 3.13.

Let a mapping $q\colon \mathbb{N} \to \mathbb{N}$ be called *order-preserving* iff $i < j \iff q(i) \leq q(j)$ for all $i, j \in \mathbb{N}$.

Proposition 3.9 ([LS01], theorem 3)**.** *For every $k \in \mathbb{N}$ there exists a $r(k) \in \mathbb{N}$ and an order-preserving mapping $q\colon \mathbb{N} \to \mathbb{N}$ such that for every vocabulary σ and $n, m \in \mathbb{N}$: if*

$$\langle \{1, \cdots, n\}, \sigma^{\mathfrak{A}} \rangle \sim^{<}_{r(k)} \langle \{1, \cdots, m\}, \sigma^{\mathfrak{B}} \rangle,$$

then

$$\langle \{1, \cdots, q(n)\}, \sigma^{q,\mathfrak{A}} \rangle \sim^{+}_{k} \langle \{1, \cdots, q(m)\}, \sigma^{q,\mathfrak{B}} \rangle,$$

with $\sigma^{q,\mathfrak{A}}$ being a shorthand for the set of interpretations $R^{q,\mathfrak{A}} = \{q(i) \,|\, i \in R^{\mathfrak{A}}\}$ of all symbols R in σ.

Proof of theorem 3.8. Let $L \subseteq \Sigma^\star$ be a language with a neutral letter e. Assume that $L \notin \mathrm{FO}[<]$, then there exist strings $u = u_1 \cdots u_n, v = v_1 \cdots v_m \in \Sigma^\star$ with $u \in L$, $v \notin L$ and $u \sim^{<}_{r(k)} v$ for every $r(k) \in \mathbb{N}$.

Construct strings $u', v' \in \Sigma^\star$ from u, v by inserting the neutral letter e so that $u'_{q(i)} = u_i$ and $v'_{q(j)} = v_j$ for $1 \leq i \leq n$, $1 \leq j \leq m$, with q as in proposition 3.9. Since $\underline{u} = \langle \{1, \cdots, n\}, \sigma^{\mathfrak{A}} \rangle \sim^{<}_{r(k)} \langle \{1, \cdots, m\}, \sigma^{\mathfrak{B}} \rangle = \underline{v}$, an application of the proposition induces

$$\langle \{1, \ldots, q(n)\}, \sigma^{q,\mathfrak{A}} \rangle \sim^{+}_{k} \langle \{1, \ldots, q(m)\}, \sigma^{q,\mathfrak{B}} \rangle.$$

This gives a winning strategy for the duplicator in the k-round $+$-game on $(\underline{u'}, \underline{v'})$: every move of the players in $(\underline{u}, \underline{v})$ can be directly translated to the word models of u' and v'.

Choosing the vocabulary σ as the set of symbols $Q_a, a \in \Sigma$, $+$ and $=$, it follows $\underline{u'} \sim^{+}_{k} \underline{v'}$ and eventually $L \notin \mathrm{FO}[<] \Longrightarrow L \notin \mathrm{FO}[+]$. The converse results in the claim. □

With theorem 3.8 at hand a first language can be shown not to lie in FO[+].

Corollary 3.10. *Let $\Sigma = \{\mathrm{a}, \mathrm{b}\}$. The language $L = \left\{ w \in \Sigma^\star \,\middle|\, |w|_{\mathrm{a}} \equiv 0 \pmod 2 \right\}$ is not definable in* FO[+].

Proof. Let $\Sigma = \{\mathrm{a}, \mathrm{b}\}$ and $L = \big\{w \in \{\mathrm{a}, \mathrm{b}\}^\star \,\big|\, |w|_\mathrm{a} \equiv 0 \pmod 2\big\}$. Assume L is definable in FO[<] by a formula φ. Then $L(\phi) = \{\mathrm{a}^n \,|\, n \equiv 0 \pmod 2\}$ for the formula

$$\phi \equiv \varphi \wedge \forall x(Q_\mathrm{a}(x)).$$

This contradicts theorem 3.2-1: L is not definable in FO[<].

Furthermore, for every $u, v \in \Sigma^\star$: $uv \in L \iff ubv \in L$, because $|uv|_\mathrm{a} = |ubv|_\mathrm{a}$. So b is a neutral letter for L, what enforces $L \notin \mathrm{FO}[+]$ in connection with the Crane Beach Conjecture. □

Even though this language is indeed regular, it is likewise possible to show indefinability of proper context-free languages.

Lemma 3.11. FO[+] *is closed under length preserving homomorphisms.*

Proof. Let $\varphi \in \mathrm{FO}[+]$ define the language $L \subseteq \Sigma^\star$ and let $h\colon \Sigma^\star \to \Delta^\star$ be a length preserving homomorphism, i. e. h satisfies $h(xy) = h(x)h(y)$ and $|h(x)| = |x|$ for all $x, y \in \Sigma^\star$.

Construct $\varphi_h \in \mathrm{FO}[+]$ from φ by replacing $Q_a(x)$ with $Q_{h(a)}(x)$ for all $a \in \Sigma^\star$. Since $h(w) = h(w_1) \cdots h(w_n)$ for $w = w_1 \cdots w_n \in \Sigma^\star$, it holds

$$w \in L \iff \underline{w} \models \varphi \iff \underline{h(w)} \models \varphi_h \iff h(w) \in h(L),$$

where $h(L) = \{h(w) \,|\, w \in L\}$. Thus $h(L) \in \mathrm{FO}[+]$. □

Corollary 3.12. *Let Σ be an alphabet with $\{\mathrm{a}, \mathrm{b}\} \subsetneq \Sigma$ and $\mathrm{R} \in \{=, \neq, <, >\}$ be a binary relation. Then the language $L = \big\{w \in \Sigma^\star \,\big|\, |w|_\mathrm{a} \,\mathrm{R}\, |w|_\mathrm{b}\big\}$ is not definable in* FO[+].

Proof. Let $\mathrm{R} \in \{=, \neq, <, >\}$ and let $L = \big\{w \in \{\mathrm{a}, \mathrm{b}, \mathrm{c}\}^\star \,\big|\, |w|_\mathrm{a} \,\mathrm{R}\, |w|_\mathrm{b}\big\}$. With lemma 3.11, it suffices to show $L \notin \mathrm{FO}[+]$ over the alphabet $\Delta = \{\mathrm{a}, \mathrm{b}, \mathrm{c}\}$. The homomorphism $h\colon \Sigma \to \Delta = \{\mathrm{a}, \mathrm{b}, \mathrm{c}\}$ that maps every letter different from a and b to c then gives the desired result.

Assume L is definable in FO[<] via φ. Then the formula

$$\phi \equiv \varphi \wedge \exists x \forall y\big((y \leq x \to Q_\mathrm{a}(y)) \wedge (x < y \to Q_\mathrm{b}(y))\big)$$

defines the language $L(\phi) = \{\mathrm{a}^n\mathrm{b}^m \mid n \,\mathrm{R}\, m\} \notin \mathrm{REG}$. This is contradictory to $\mathrm{FO}[<] \subset \mathrm{REG}$ (theorem 3.2-2). It follows that $L \notin \mathrm{FO}[<]$. Moreover the letter c serves as neutral letter for L, and thus $L \notin \mathrm{FO}[+]$. □

Going more into the details of the proof of proposition 3.9, the set of languages on which the expressiveness of FO[<] and FO[+] coincide can be extended from languages with neutral letters to languages with neutral letters and certain restrictions placed on the length of their strings. Intuitively this should be clear, since the power of addition is rendered useless by neutral letters—any condition concerning lengths can not restore it, because strings may be arbitrary pumped or padded.

Theorem 3.13. *Let Σ be an alphabet with $\{\mathrm{a}\} \subsetneq \Sigma$. None of the languages $\left\{w \in \Sigma^\star \,\middle|\, |w|_\mathrm{a} \equiv h \pmod p, |w| \equiv l \pmod r\right\}$, $h, l, p, r \in \mathbb{N}$, $0 \leq h < p$, $0 \leq l < r$, is definable in* FO[+].

Proof. For $h, l, p, r \in \mathbb{N}$ and $0 \leq h < p$, $0 \leq l < r$, let $L_{(h,p)(l,r)} = \left\{w \in \Sigma^\star \,\middle|\, |w|_\mathrm{a} \equiv h \pmod p, |w| \equiv l \pmod r\right\}$. It suffices to show that $L_{(h,p)(l,r)}$ is not definable in FO[+] over the alphabet $\Sigma = \{\mathrm{a}, \mathrm{e}\}$; any letter in Σ different from a can be mapped to e via a length-preserving homomorphism.

Assume $L_{(h,p)(l,r)}$ is definable via $\varphi \in \mathrm{FO}[+]$ with quantifier rank $\leq k$. The order-preserving mapping $q\colon \mathbb{N} \to \mathbb{N}$ that is utilized to prove proposition 3.9 is defined as $q(i) = p_{i+1}$ for any sequence satisfying the conditions

$$\begin{aligned} &p_0 = 0, \\ &p_i \geq 2^{k+3} f(k)^3 p_{i-1} + 2g(k) f(k)^2 \text{ for all } i > 0, \text{ and} \\ &p_i \equiv p_j \pmod{f(k)!} \text{ for all } i, j > 0, \end{aligned}$$

where $f, g\colon \mathbb{N} \to \mathbb{N}$,

$$\begin{aligned} f(0) &= 1, \quad f(i+1) = 2f(i)^4, \\ g(0) &= 0, \quad g(i+1) = 2g(i)f(i)^2 + f(i)!, \end{aligned}$$

are strictly increasing functions that ensure indistinguishability with respect to all linear combinations of positions. As p_1 can be chosen arbitrary beyond

a certain threshold, it is possible to satisfy $f(k)! \,|\, p_i - l$; in other words $q(i-1) \equiv l \pmod{f(k)!}$, or $q(i-1) \equiv l \pmod{r}$.

With this choice of q it is possible to construct strings that are not distinguishable by first-order sentences with addition:

According to theorem 3.2-1, there exists $m(k) \in \mathbb{N}$ so that $\mathrm{a}^{2^{m(k)}} \sim^{<}_{m(k)} \mathrm{a}^{2^{m(k)}+1}$. Let $u = \mathrm{a}^{px+h}$ and $v = \mathrm{a}^{px+h+1}$, where $x > \lceil \frac{2^{m(k)}-h}{p} \rceil$. It follows that the strings $u', v' \in \Sigma^\star$, constructed from u, v by inserting the neutral letter e so that $u'_{q(i)} = u_i$ and $v'_{q(j)} = v_j$ for $1 \leq i \leq px+h$, $1 \leq j \leq px+h+1$ (cf. proof of theorem 3.8), are of length $l \pmod{r}$ with $u' \sim^+_k v'$. Now $|u'|_\mathrm{a} = |u|_\mathrm{a} \equiv h \pmod{p}$, $|v'|_\mathrm{a} = |v|_\mathrm{a} \equiv h+1 \not\equiv h \pmod{p}$ and $q(|u|) \equiv q(|v|) \equiv l \pmod{r}$ result in $u' \in L_{(h,p)(l,r)}, v' \notin L_{(h,p)(l,r)}$, which is contradictory to $u' \sim^+_k v'$. Hence $L_{(k,p)(l,r)} \notin \mathrm{FO}[+]$. □

Another consequence of the preceding is the theorem of Ginsburg and Spanier, cf. [GS66]:

Corollary 3.14 (FO[+]-sentences have semi-linear spectra)**.** *For every sentence* $\varphi \in \mathrm{FO}[+]$ *the set* $\mathrm{Spec}(\varphi) = \big\{ n \in \mathbb{N} \setminus \{0\} \,\big|\, \langle \{1, \ldots, n\}, + \rangle \models \varphi \big\}$ *is semi-linear, i. e. there exist* $n_0, p \in \mathbb{N}$ *such that*

$$n \in \mathrm{Spec}(\varphi) \iff n + p \in \mathrm{Spec}(\varphi)$$

for all $n > n_0$.

Proof. Construct strings u', v' as in the previous proof over a one letter alphabet $\Sigma = \{\mathrm{e}\}$. Then again it is $u' \sim_k v'$. Furthermore, $|u'| = q(|u|)$, $|v'| = q(|v|) = q(|u|+1)$, and thus $|v'| - |u'| = q(|u|+1) - q(|u|) = l \cdot f(k)!$ for some $l \in \mathbb{N}$; w. l. o. g. $l = 1$ can be assumed. Hence

$$\underline{u'} \models \varphi \iff \underline{v'} \models \varphi,$$

what implies

$$\langle \{1, \ldots, q(|u|)\}, + \rangle \sim^+_k \langle \{1, \ldots, q(|u|) + f(k)!\}, + \rangle,$$

because the symbol Q_e can be dropped from σ—it does not contain any information. Now setting $n = q(|u|)$ and $p = f(k)!$ yields

$$\langle \{1, \ldots, n\}, + \rangle \models \varphi \iff \langle \{1, \ldots, n+p\}, + \rangle \models \varphi.$$

Since $q(|u|) = p_{|u|+1}$ may be chosen arbitrary beyond $2g(k)f(k)^2$, this holds for all $n \geq n_0 = 2g(k)f(k)^2$. □

Corollary 3.14 gives a nice connection to relations of natural numbers resulting in the indefinability of the set of prime numbers in FO[+].

Proposition 3.15. *The set of prime numbers is not definable in* FO[+]*, i. e. let* Σ *be an alphabet and let* $\mathbb{P}$ *denote the set of all prime numbers, then* $\{w \in \Sigma^\star \mid |w| \in \mathbb{P}\} \notin \mathrm{FO}[+]$.

Proof. The set of prime numbers is not semi-linear: Assume the opposite with parameter p and choose an $n \in \mathbb{P}$ with $n > n_0, n > p$. It is $\gcd(n, p) = 1$ and $n \in \mathbb{P} \iff n + p \in \mathbb{P} \iff n + kp \in \mathbb{P}$ for all $k \in \mathbb{N}$. Thus $\{x \mid x \equiv kp \pmod{n}\} = \{0, \ldots, n-1\}$. So there exists a $k \in \mathbb{N}$ with $n|(n + kp)$, what contradicts the assumption. □

Proposition 3.15 also provides the insight that multiplication and the divisibility relations can not be definable in FO[+] (otherwise the set of primes could be defined using formulas for one of the latter sets). However, lets now turn to a famous set of conext-free languages, the Dyck languages.

Definition 3.16. *Let* Σ *be an alphabet and let* $\Delta = \Sigma \cup \overline{\Sigma}$ *with* $\overline{\Sigma} = \{\overline{a} \mid a \in \Sigma\}$*. The* Dyck language $D^\star_\Sigma$ *is defined as the equivalence class* $[\varepsilon]$ *with respect to the equivalence relation implied by* $a\overline{a} = \varepsilon$ *for* $a \in \Sigma$*.*

For any given alphabet Σ, the Dyck language $D^\star_\Sigma$ is generated by the grammar $G = (\{S, T\}, \Sigma, P, S)$ with the rules $P = \{S \to \varepsilon, S \to TS\} \cup \{T \to aS\overline{a} \mid a \in \Sigma\}$. G is equivalent to a context-free grammar. Hence $D^\star_\Sigma \in \mathrm{CFL}$.

It is folklore that $D^\star_\Sigma \in \mathrm{DCFL} \setminus \mathrm{REG}$, what involves that $D^\star_\Sigma$ supplemented with a neutral letter is not definable in FO[+]. Accordingly also

Greibach's hardest context-free language $H_\Sigma \in \mathrm{CFL} \setminus \mathrm{DCFL}$ [ABB97, pp. 27–28] supplemented with a neutral letter can not be definable in FO[+]. This is particularly interesting in the context of groupodial based language classes, since H_Σ is complete for the class LOGCFL (cf. [LMSV01]).

Due to the preceding results and the closure under length-preserving homomorphisms, it can be conjectured that $D^\star_\Sigma \notin \mathrm{FO}[+]$. But the problem with proving this is that, for every $w \in D^\star_\Sigma$, there is an even number of positions between two corresponding letters; whereas the construction in theorem 3.8 relies heavily on the fact that, between adjacent positions in u, an odd number of neutral letters is being inserted when constructing u' ($p_i \equiv p_j \pmod{f(k)!}$ for all $i, j > 0$, and $f(k)!$ is even for all $k > 1$). Nevertheless, a small variation of the grammar G is indefinable:

Let $\Sigma' = \Sigma \cup \{\mathrm{e}\}$ for some $\mathrm{e} \notin \Sigma$ and consider the grammar $G = (\{S, T\}, \Sigma, P, S)$ with the rules

$$P = \{S \to \mathrm{e}, S \to TS\} \cup \{T \to aS\overline{a} \mid a \in \Sigma\}.$$

Obviously G is a context-free grammar. $L(G)$ is a variant of the Dyck language that extends the equivalence relation $a\overline{a} = \varepsilon$ with the rule $\mathrm{e} = \varepsilon$. Let $D^\star_{\Sigma,\mathrm{e}}$ denote the language generated by G.

Theorem 3.17. *Let Σ be an alphabet and let $\mathrm{e} \notin \Sigma$, then $D^\star_{\Sigma,\mathrm{e}} \notin \mathrm{FO}[+]$.*

Proof. Let $D^\star_{\Sigma,\mathrm{e}}$ be over $\Delta = \Sigma \cup \overline{\Sigma} \cup \{\mathrm{e}\}$, where $\overline{\Sigma} = \{\overline{a} \mid a \in \Sigma\}$. Assume $D^\star_{\Sigma,\mathrm{e}} = L(\varphi)$ for $\varphi \in \mathrm{FO}[+]$ with quantifier rank $\leq k$. For $c \in \Sigma$, consider the length preserving homomorphism $f\colon \Delta^\star \to \{\mathrm{a}, \mathrm{b}\}^\star$,

$$f(x) = \begin{cases} \mathrm{a}, & \text{if } x = c \text{ or } x = \overline{c}, \\ \mathrm{b}, & \text{otherwise.} \end{cases}$$

The language $f(D^\star_\Sigma)$ is the set of strings having

- an even length,
- an even number of positions with the letter a, and

- an odd number of b's between two adjacent positions with the letter a.

Fix $k \in \mathbb{N}$ and let $q\colon \mathbb{N} \to \mathbb{N}$ denote an order-preserving mapping conforming to the conditions of the proof of theorem 3.13 that additionally satisfies $q(1) \equiv 0 \pmod 2$. According to theorem 3.2-1, there exists an $r(k)$ so that $\mathrm{a}^{2^{r(k)}} \sim^{<}_{r(k)} \mathrm{a}^{2^{r(k)}+1}$.

Construct once more strings u', v' from $u = \mathrm{a}^{2^{r(k)}}, v = \mathrm{a}^{2^{r(k)}+1}$ by inserting the letter b so that $u'_q(i) = u_i$ for all $1 \leq i \leq 2^{r(k)}$ and $v'_q(j) = v_j$ for all $1 \leq j \leq 2^{r(k)}+1$. The inserted blocks of (the letter b) in u' and v' are of odd length, because q maps all positions in u and v to even positions in u' and v': $q(i) \equiv q(j) \pmod{f(k)!}$ for all $i, j > 0$ and $f(k)!$ is even for $k > 1$. In particular $q(|u|) \equiv q(|v|) \equiv 0 \pmod 2$. Concluding, it holds $u' \sim^{+}_{k} v'$, but $|u|_{\mathrm{a}} \equiv 0 \not\equiv 1 \equiv |v|_{\mathrm{a}} \pmod 2$, therefore $u \in f(D^{\star}_{\Sigma}), v \notin f(D^{\star}_{\Sigma})$. Hence $D^{\star}_{\Sigma,\mathrm{e}} \notin \mathrm{FO}[+]$. □

3.3.2 A general upper bound

So far, several languages have been shown not to lie in FO[+]. Yet an upper bound for the expressive power of first-order with addition is lacking.

An answer to this question will be gathered from first-order logics supplemented with "counting quantifiers", as investigated in [Str74].

Definition 3.18. *Let $\mathfrak{A} = \langle \mathbb{U}, \sigma^{\mathfrak{A}} \rangle$ be a structure and let $\varphi(x)$ be a first-order formula over σ. A* counting quantifier $\exists^{r \bmod q}$ *is defined as a quantor with the semantics*

$$\mathfrak{A} \models \exists^{r \bmod q} x \varphi(x) \quad \textit{iff} \quad |\{x \in \mathbb{U} \mid \mathfrak{A} \models \varphi(x)\}| \equiv r \bmod q.$$

Denote by (FO+MOD)[σ] *the extension of first-order logic over the vocabulary σ with the counting quantifiers $\exists^{r \bmod q}$ for all $q > 1$ and $0 \leq r < q$.*

That is, $\exists^{r \bmod q} x \varphi(x)$ holds for formula $\varphi(x)$ if the number of positions in $\mathbb{U}$ satisfying $\varphi(x)$ is congruent to r modulo q. It is obvious that counting quantifiers add expressiveness to FO[+]:

Example 3.19. Reconsider the language investigated in corollary 3.10, $L = \left\{ w \in \Sigma^\star \,\middle|\, |w|_\mathrm{a} \equiv 0 \bmod 2 \right\}$. $L \notin \mathrm{FO}[+]$ is definable in (FO+MOD)[+] by the short formula

$$\exists^{0 \bmod 2} x Q_\mathrm{a}(x).$$

Therefore, $\mathrm{FO}[+] \subsetneq (\mathrm{FO}+\mathrm{MOD})[+] \subseteq (\mathrm{FO}+\mathrm{MOD})[+,\times]$, where $\times$ is interpreted as the natural multiplication. Even more is known about the last inclusion: in [Ruh99], Ruhl separated $(\mathrm{FO}+\mathrm{MOD})[+]$ from $(\mathrm{FO}+\mathrm{MOD})[+,\times]$ by showing that deterministic transitive closure can not be expressed in (FO+MOD)[+]. Let AC^0 denote the set of languages definable by polynomial-size, constant-depth boolean circuits with AND-, OR- and NOT-gates; and let TC^0 denote AC^0 supplemented with MOD-gates. Together with $(\mathrm{FO}+\mathrm{MOD})[+,\times] = \mathrm{TC}^0$ (cf. [BIS90]), the following bound for the expressiveness of FO[+] can be given:

Theorem 3.20. *Let* $\mathrm{NSPACE}(s)$ *denote the set of all languages definable by Turing machines in space* $\leq cs$ *for some* $c \in \mathbb{N}$ *and let* $\mathrm{L} = \mathrm{NSPACE}(\log n)$. *Then*

$$\mathrm{FO}[+] \subsetneq (\mathrm{FO}+\mathrm{MOD})[+] \subsetneq (\mathrm{FO}+\mathrm{MOD})[+,\times] = \mathrm{TC}^0 \subseteq \mathrm{L} \subsetneq \mathrm{CSL}.$$

Proof. The inclusions $\mathrm{FO}[+] \subseteq (\mathrm{FO}+\mathrm{MOD})[+] \subseteq (\mathrm{FO}+\mathrm{MOD})[+,\times]$ are obviously true. The separation $\mathrm{FO}[+] \subsetneq (\mathrm{FO}+\mathrm{MOD})[+]$ follows from example 3.19. For the separation of (FO+MOD)[+] from (FO+MOD)[+, ×], consult [Ruh99]. The equality $(\mathrm{FO}+\mathrm{MOD})[+,\times] = \mathrm{TC}^0$ is shown in [BIS90, theorem 11.2] and is a straight-forward translation between sentences of $(\mathrm{FO}+\mathrm{MOD})[+,\times]$ and constant-depth, polynomial-size circuits. Finally $\mathrm{TC}^0 \subseteq \mathrm{L} = \mathrm{SPACE}(\log(n)) \subsetneq \mathrm{NSPACE}(n) = \mathrm{CSL}$ follows directly from the definitions of TC^0, L and CSL as well as the existence of languages in $\mathrm{SPACE}(n) \setminus \mathrm{SPACE}(\log n)$. □

The resulting relation of FO[+] to the classes of the Chomsky hierarchy is shown in figure 3.2; there $\exists \mathit{Matching}\mathrm{FO}[<]$ denotes class of languages definable by sentences $\exists M \varphi$, where M is a matching and $\varphi \in \mathrm{FO}[<]$.

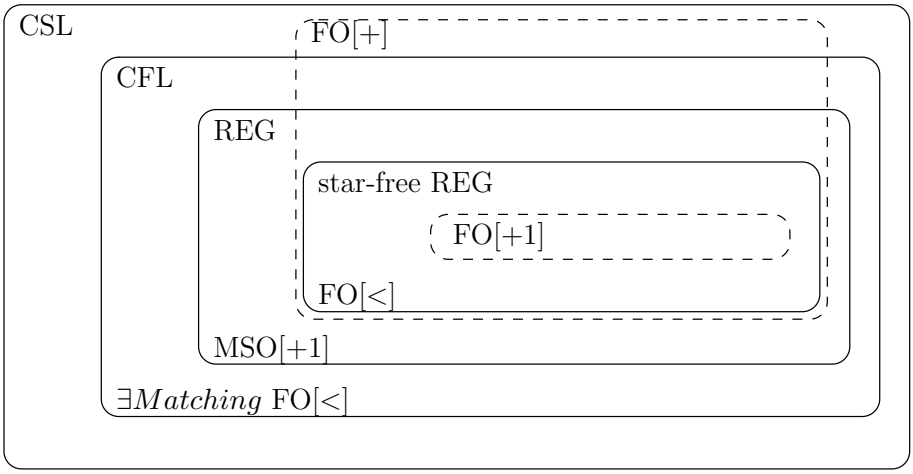

Figure 3.2: FO[+] related to the Chomsky hierarchy.

3.4 Context-Free Languages and FO[+]

The previous section has been devoted to explore the limitations of FO[+], and it has been shown that even though FO[+] captures some languages beyond CFL, it does not entirely capture CFL. This section investigates whether a nice characterization of the fragment of CFL being captured by FO[+] can be given.

3.4.1 FO[+] and linear recursive grammars

Let $G = (V, \Sigma, P, S)$ be a context-free grammar. A *loop* is a sequence of productions $(p_i)_{1 \leq i \leq n}$, $n \in \mathbb{N}$, so that $p_i \equiv A_i \to \alpha_{i_1} A_{i+1} \alpha_{i_2} \in P$ for $1 \leq i < n$ and $p_n \equiv A_n \to \alpha_{n_1} A_1 \alpha_{n_2} \in P$, with $\alpha_{i_1}, \alpha_{i_2} \in (\Sigma \cup V)^\star$, $A_i \in V$, $1 \leq i \leq n$. A production $p \in P$ contained in a loop will be denoted as *loop production*. The *length* of a loop is the number of productions in its sequence. A loop is called *linear* if $\alpha_{i_1}, \alpha_{i_2} \in \Sigma^\star$ for all $1 \leq i \leq n$. The notion of linear loops is inspired by linear context-free grammars, whose

rules contain at most one nonterminal on every production's right-side. The languages generated by these grammars are recognized by 1-turn PDAs, i. e. PDAs that alter their stack operation only once, from push to pop or vice versa (cf. [Har78]).

Let $G = (V, \Sigma, P, S)$ be a context-free grammar and let $(p_i)_{1 \leq i \leq n}$ be a loop in G. Then

$$\mathrm{Var}((p_i)_{1 \leq i \leq n}) = \{A \in V \mid p_i \equiv A \to \alpha, 1 \leq i \leq n\}$$

is the set of variables on the left-hand sides of productions in $(p_i)_{1 \leq i \leq n}$.

Definition 3.21. *Let $G = (V, \Sigma, P, S)$ be a context-free grammar. G is called* linear recursive *if every loop is linear and there is a partitioning $V = V_1 \cup V_2 \cup \cdots \cup V_k$ such that the left-hand nonterminals of every loop lie in one partition and different loops have distinct sets of left-hand nonterminals.*

That is, for loops $(p_i)_{1 \leq i \leq n}, (q_j)_{1 \leq j \leq m}$

$$Var((p_i)_{1 \leq i \leq n}) \subseteq V_l,$$

for some $1 \leq l \leq k$, and

$$Var((p_i)_{1 \leq i \leq n}) \cap Var((q_i)_{1 \leq i \leq m}) = \emptyset.$$

Note that this implies the following partial order on V: $A < B \iff \min\left\{k \,\middle|\, S \Rightarrow_G^k \alpha A \beta, \alpha, \beta \in (\Sigma \cup V)^\star\right\} < \min\left\{k \,\middle|\, S \Rightarrow_G^k \alpha B \beta, \alpha, \beta \in (\Sigma \cup V)^\star\right\}$ for $A, B \in V$.

However, the set of languages generated by linear recursive grammars is neither a subclass nor a superclass of the languages generated by linear grammars. For the first direction consider the language $L = \{a^n \overline{a}^n b^m \overline{b}^m \mid n, m > 0\}$. L is generated by a grammar with starting symbol S and rules

$$S \to AB, A \to aA\overline{a}, A \to a\overline{a}, B \to bB\overline{b}, B \to b\overline{b}.$$

L can not be recognized by a 1-turn PDA, since —informally speaking— the push and pop operation has to change more than once to compare the lengths in the first and in the second parts. For the other direction the language $L = \left\{w \in \Sigma^\star \,\middle|\, |w|_\mathrm{a} \equiv 0 \pmod 2\right\}$ from corollary 3.10 is linear, but not linear recursive: $L(G) = L$ for the linear CFG $G = (\{S, U, G\}, \Sigma, P, S)$ with the rules

$$\begin{aligned} P &= \left\{S \to \varepsilon, S \to \mathrm{a}U, U \to \mathrm{a}G, G \to \mathrm{a}U\right\} \\ &\cup \left\{X \to bX \,\middle|\, b \in \Sigma \setminus \{\mathrm{a}\}, X \in \{G, U\}\right\}. \end{aligned}$$

The fact that L is not linear recursive will be covered in example 3.22.

This discrepancy emerges since restrictions are placed on the productions belonging to some loop only. Linear recursive grammars may rather be thought of as a concatenation and nesting of linear grammars not containing "entangled" loops.

Example 3.22.

- Consider the language $L = \{\mathrm{a}^n\mathrm{b}^n \,|\, n \in \mathbb{N}\}$. L is generated by a linear recursive CFG $G = (\{S\}, \{\mathrm{a}, \mathrm{b}\}, P, S)$ with the productions $P = \{S \to \mathrm{a}S\mathrm{b},\ S \to \mathrm{ab}\}$.

- There is no linear recursive CFG generating the language $L = \left\{w \in \{a, b\}^\star \,\middle|\, |w|_a \equiv 0 \pmod 2\right\} \notin \mathrm{FO}[+]$:

  Assume $L = L(G)$ for the linear recursive CFG G. To generate L, G has to contain a loop; w. l. o. g. the loop productions can be assumed to generate one terminal per production only. Hence the number of a's modulo 2 has to be remembered in the nonterminals of the loop. Since counting modulo 2 requires at least two nonterminals to distinguish the different cosets and the production of a single b does not change the number of a's, loops that differ in more than terminals have to exist. Thus, G can not be linear recursive.

- The Dyck language $D_\Sigma^\star$ admits no linear recursive CFG.

  Assume $L(G) = D_\Sigma^\star$ for the linear recursive CFG $G = (V, \Sigma, P, S)$. G admits a partial order on P, viz the order mentioned succeeding

definition 3.21. Therefore, every non-loop-production may only be used once in a derivation.

Let $x \in D_\Sigma^\star$. The number of partitions $x = uav\overline{a}w$ with $u, v, w \in (\Sigma \cup \overline{\Sigma})^\star$, $a \in \Sigma$, $\overline{a} \in \overline{\Sigma}$ being a's corresponding character and $ua^n v\overline{a}^n w \in D_\Sigma^\star$ for all $n \in \mathbb{N}$ is bounded by the number of nonterminals on the right-hand side of all non-loop productions. Let k be the maximal number of partitions over all x derivable from G. Then the word $(a^l\overline{a}^l)^{k+1}$ with $l > \max\{m \mid A \to \alpha \in P, |\alpha| = m\}$ can not be derivable from G, and $L(G) \subsetneq D_\Sigma^\star$ follows.

Lemma 3.23. *Let $G = (V, \Sigma, P, S)$ be a linear recursive CFG. Then there exists a linear recursive CFG $G' = (V', \Sigma, P', S)$ and a partitioning $V_1' \cup \cdots \cup V_n'$ of V' such that $L(G) = L(G')$, every loop in G' is of length one and, for all loops $(p_i)_{1\leq i\leq k}$ in G,*

$$Var((p_i)_{1\leq i\leq k}) = V_l,$$

for some $1 \leq l \leq n$.

Proof. Let $G = (V, \Sigma, P, S)$ be a linear recursive CFG. There exists a partitioning $V_1 \cup V_2 \cup \cdots \cup V_n = V$ such that for each pair of loops $(p_i)_{1\leq i\leq k_1}, (q_j)_{1\leq i\leq k_2}$ the following holds:

$$\mathrm{Var}((p_i)_{1\leq i\leq k_1}) \subseteq V_l,$$

for some $1 \leq l \leq n$, and

$$\mathrm{Var}((p_i)_{1\leq i\leq k_1}) \cap \mathrm{Var}((q_j)_{1\leq j\leq k_2}) = \emptyset.$$

Let $P = P_1 \cup P_2 \cup \cdots \cup P_n$ be the partitioning of P implied by the left-hand sides of the productions, viz.

$$P_l = \{p \in P \mid p \equiv A \to \alpha, A \in V_l, \alpha \in (V \cup \Sigma)^\star\},$$

for $1 \leq k \leq l$. Let $(p_i)_{1\leq i\leq m(l)}$ be a loop in one of the P_l,

$$\begin{aligned} p_i &\equiv A_i \to \alpha_{i_1} A_{i+1} \alpha_{i_2}, && 1 \leq i < m(l), \\ p_{m(l)} &\equiv A_{m(l)} \to \alpha_{m_1(l)} A_1 \alpha_{m_2(l)}, \end{aligned}$$

where $A_i \in V_l, \alpha_{i_1}, \alpha_{i_2} \in \Sigma^\star$ for $1 \leq i \leq m(l)$. For each nonterminal in $\{A_i \,|\, p_i \equiv A_i \to \beta_{i_1} B \beta_{i_2}, B \notin V_l, \beta_{i_1}, \beta_{i_2} \in \Sigma^\star, 1 \leq i \leq m(l)\}$, introduce a new nonterminal A_i' and set $V_{l_i} = \{A_i'\}$,

$$\begin{aligned} P_{l_i} &= \{A_i' \to \alpha_{i_1} \cdots \alpha_{m_1(l)} \alpha_1 \cdots \alpha_{i_1 - 1} A_i' \alpha_{i_2} \alpha_{i_2 - 1} \cdots \alpha_1 \alpha_{m_2(l)} \cdots \alpha_{i_2}\} \\ &\cup \{A_i' \to \beta_{i_1} B \beta_{i_2} \in P, B \notin V_l\}. \end{aligned}$$

If P_l is loop-free, set $P_{l_i} = V_{l_i} = \emptyset$. As a consequence, for all $1 \leq l \leq n$, the set $P_l' = P_l \setminus \{p_{m(l)} \,|\, (p_i)_{1\leq i\leq m(l)} \text{ is a loop in } P_l\}$ is loop-free, each loop in $\bigcup_{1\leq i\leq m(l)} P_{l_i}$ is of length one and for every loop $(p_i)_{1\leq i\leq m}$ there is some $1 \leq j \leq m(l)$ so that $\mathrm{Var}((p_i)_{1\leq i\leq m}) = V_{l_j}$. Thus define $G' = (V', \Sigma, P', S)$, where

$$\begin{aligned} V' &= V \cup \bigcup_{\substack{1\leq l\leq n \\ 1\leq i\leq m(l)}} V_{l_i}, \\ P' &= (P \setminus \{p_m \,|\, (p_i)_{1\leq i\leq m} \text{ is a loop in } P\}) \\ &\cup \bigcup_{\substack{1\leq l\leq n \\ 1\leq i\leq m(l)}} (P_{l_i} \cup \{A_i \to A_i' \,|\, A_i' \in V_{l_i}\}). \end{aligned}$$

Next $L(G) = L(G')$ will be shown. Therefore let $w \in \Sigma^\star$. If $S \Rightarrow_G^\star w$ and w does not contain any loops, then $S \Rightarrow_{G'}^\star w$ by the same sequence of productions. If there exists $A \in V$ such that

$$S \Rightarrow_G^\star \alpha_1 A \alpha_2 \Rightarrow_G^k \alpha_1 \beta_1 A \beta_2 \alpha_2 \Rightarrow_G \alpha_1 \beta_1 \gamma_1 B \gamma_2 \beta_2 \alpha_2 \Rightarrow_G^\star w$$

for some $k \geq 1$, $\alpha_1, \alpha_2, \beta_1, \beta_2, \gamma_1, \gamma_2 \in (V \cup \Sigma)^\star$, $A \neq B \in V$, then

$$S \Rightarrow_{G'}^\star \alpha_1 A \alpha_2 \Rightarrow_{G'} \alpha_1 A' \alpha_2 \Rightarrow_{G'}^l \alpha_1 \beta_1 A' \beta_2 \alpha_2 \Rightarrow_{G'} \alpha_1 \beta_1 \gamma_1 B \gamma_2 \beta_2 \alpha_2 \Rightarrow_{G'}^\star w$$

for $l = \frac{k}{x}$, where x is the length of the loop starting in nonterminal $A \in V$. The opposite direction holds with the identical argumentation. Hence $L(G) = L(G')$. □

Now a fragment of CFL being captured by FO[+] can be specified. Let *linear recursive CFL* denote the set of all languages generated by linear recursive grammars, i. e. $\{L \mid \text{there exists a linear recursive CFG with } L(G) = L\}$.

Theorem 3.24. Linear recursive CFL $\subseteq$ FO[+].

Proof. Let $G = (V, \Sigma, P, S)$ be a linear recursive CFG. According to to the above lemma, G can be assumed to contain only linear loops of length one. A formula $\phi \in$ FO[+] with $L(\phi) = L(G)$ can be constructed as follows:

For $u, w = w_1 \dots w_n \in \Sigma^\star$, let $\varphi^{\cdot\cdot}_u(x, y)$ be a formula with

$$\underline{w} \models \varphi^{\cdot\cdot}_u(x, y) \iff w_{x+1} \cdots w_{y-1} = u$$

if $x + 1 < y$, and $\varphi^{\cdot\cdot}_u(x, y) \equiv \mathit{true}$ otherwise. Furthermore, let $\varphi^{*\cdot}_u(x, y)$ and $\varphi^{\cdot *}_u(x, y)$ be formulas with

$$\begin{aligned} \underline{w} \models \varphi^{*\cdot}_u(x, y) &\iff w_x \cdots w_{y-1} = u, \\ \underline{w} \models \varphi^{\cdot *}_u(x, y) &\iff w_{x+1} \cdots w_y = u \end{aligned}$$

if $x < y$, and $\varphi^{\cdot *}_u(x, y) \equiv \varphi^{\cdot *}_u(x, y) \equiv \mathit{true}$ otherwise. Analogously define $\varphi^{**}_u(x, y)$ to hold if the string beginning in position x and ending in position y equals u. Clearly $\varphi^{\cdot\cdot}_u(x, y), \varphi^{*\cdot}_u(x, y), \varphi^{\cdot *}_u(x, y), \varphi^{**}_u(x, y) \in$ FO[+].

Now define a formula $\chi_X(x, y)$ that holds if and only if the string beginning in position x and ending in y can be derived from a variable X: for every non-loop production $p \equiv X \to \alpha$, $\alpha = v_0 X_1 v_1 \cdots v_{s-1} X_s v_s$, $X_1, \dots, X_s \in V \setminus \{X\}$, $v_0, \dots, v_s \in \Sigma^\star$, $s \geq 1$, define

$$\begin{aligned} \chi_p(x, y) \equiv \exists x_1 \exists y_1 \cdots \exists x_s \exists y_s \big(&(x \leq x_1 < y_1 \leq x_2 < \cdots \leq x_s < y_s \leq y) \wedge \\ &(\varphi^{*\cdot}_{v_0}(x, x_1) \wedge \varphi^{\cdot\cdot}_{v_1}(y_1, x_2) \wedge \cdots \wedge \varphi^{\cdot *}_{v_s}(y_s, y)) \wedge \\ &(\chi_{X_1}(x_1, y_1) \wedge \chi_{X_2}(x_2, y_2) \wedge \cdots \wedge \chi_{X_s}(x_s, y_s)) \big), \end{aligned} \tag{3.2}$$

and $\chi_p(x,y) = \varphi^{**}_{v_0}(x,y)$ if $s = 0$ (i. e. $p \equiv X \to v_0 \in \Sigma^\star$). Equation 3.2 expresses that there is a partitioning of the string such that all substrings $w_{x_i} \cdots w_{y_i}$, $1 \le i \le s$, can be derived from the corresponding X_i and the remaining substrings of terminals between them are matched by the terminals in the production.

Those production formulas are grouped by their left-hand nonterminals: for every variable $X \in V$ not occurring on the left-hand side of any loop production define

$$\chi_X(x,y) \equiv \bigvee_{\substack{p \in P \\ p \equiv X \to \alpha}} \chi_p(x,y),$$

where $\alpha \in (V \cup \Sigma)^\star$. For every other variable $X \in V$ there is exactly one loop production $X \to v_0 X v_1 \in P$, where $v_0, v_1 \in \Sigma^\star$. Thus define

$$\begin{aligned}\chi_X(x,y) \equiv \exists x_t \exists y_t \Big(& x_t \le y_t \wedge x_t - x \bmod |v_0| \equiv 0 \wedge y - y_t \bmod |v_1| \equiv 0 \ \wedge \\ & \forall z (x \le z < x_t \wedge z - x \bmod |v_0| \equiv 0 \to \varphi^{*\cdot}_v(z, z+|v_0|)) \wedge \\ & \forall z (y_t < z \le y \wedge y - z \bmod |v_1| \equiv 0 \to \varphi^{\cdot *}_v(z, z+|v_1|)) \wedge \\ & \exists d_x \exists d_y \big(x + d_x = x_t \wedge y_t + d_y = y \wedge \vartheta_{|v_0|,|v_1|}(d_x, d_y) \big) \wedge \\ & \big(\bigvee_{\substack{p \equiv X \to \gamma \in P \\ \gamma \in (V \setminus \{X\} \cup \Sigma)^\star}} \chi_p(x_t, y_t) \big) \Big),\end{aligned} \tag{3.3}$$

where $\vartheta_{k_0,k_1}(x,y)$ is a formula that holds if $\frac{x}{k_0} = \frac{y}{k_1}$. This is expressible in FO[+], since k_0, k_1 are constants depending on the production:

$$\begin{aligned}\vartheta_{k_1,k_2}(x,y) \equiv \exists d_1 \cdots \exists d_{k_0} \exists e_1 \cdots \exists e_{k_1} \big(& \textstyle\bigwedge_{1 < i \le |k_0|} d_i = d_i + d_1 \wedge \\ & \textstyle\bigwedge_{1 < i \le |k_1|} e_i = e_i + e_1 \wedge \\ & e_1 = d_1 \wedge d_{k_0} = x \wedge e_{k_1} = y \big).\end{aligned}$$

Altogether, for a variable $X \in V$ occurring on the left-hand side of some loop production and a string $w \in \Sigma^\star$, equation (3.3) expresses that the substring beginning in position x and ending in y can be split into 3 parts $s_1 = w_x \cdots w_{x_t - 1}$, $s_2 = w_{x_t} \cdots w_{y_t}$, $s_3 = w_{y_t + 1} \cdots w_y$, where

- $|s_1|$ and $|s_3|$ are a multiples of $|v_0|$ and $|v_1|$ respectively,
- s_1 consists of subsequent occurrences of v_0,
- s_3 consists of subsequent occurrences of v_1,
- $\frac{|s_1|}{|v_0|} = \frac{|s_3|}{|v_1|}$, and
- s_2 can be derived from some variable that "exits the loop".

Due to the existence of a partial order on V, the formula $\chi_X(x, y)$ is well-defined for all $X \in V$. And finally the formula

$$\phi \equiv \exists x \exists y \left(\chi_S(x, y) \wedge \forall z(x \leq z \wedge z \leq y)\right)$$

holds for any given word model $\underline{w}$ if and only if $S \Rightarrow_G^\star w$. □

Corollary 3.25. $\mathfrak{B}$(linear recursive CFL) $\subseteq$ FO[+].

Unfortunately the converse of theorem 3.24 does not hold, since it is possible to construct a non-linear recursive CFLs definable in FO[+].

For example, take the language of all strings over $\{\mathrm{a}, \mathrm{b}\}$ in which each substring matching $\mathrm{ba}^\star\mathrm{b}$ contains an even number of a's, i. e. $L = \big\{w \in \Sigma^\star \,\big|\, \forall u, v \in \{\mathrm{a}, \mathrm{b}\}^\star, w = u\mathrm{ba}^n\mathrm{b}v : n \equiv 0 \pmod 2\big\}$. L is definable in FO[+] via the formula

$$\forall x \forall y \big(Q_\mathrm{b}(x) \wedge Q_\mathrm{b}(y) \wedge \forall z(x \leq z \leq y \rightarrow Q_\mathrm{a}(z))\big) \rightarrow \exists d \big(x + d = y \wedge \varphi(d)\big),$$

where $\varphi(d)$ holds if and only if d is an even number.

Although $L \in$ CFL, it does not admit a linear recursive grammar, since the number of substrings matching $\mathrm{a}^\star$ is unbounded and every rule has to be of finite length. Assume $L(G) = L$ for a linear recursive grammar $G = (V, \{\mathrm{a}, \mathrm{b}\}, P, S)$. Then P has to contain a rule that inserts a b at the end of the current derivate and starts the next block of a's; accordingly, it is of the form $A_1 \rightarrow \alpha \mathrm{b} A_2$, with $A_1, A_2 \in V$, $\alpha \in (\{\mathrm{a}, \mathrm{b}\} \cup V)^\star$. From the definition of linear recursiveness and the fact that a word in L may contain

arbitrary many blocks of a's, it follows that $\alpha \in \{\mathrm{a}, \mathrm{b}\}^\star$ and that there exists a sequence of rules such that $A_2 \Rightarrow_G^\star \beta_1 A_1 \beta_2$ for some $\beta_1, \beta_2 \in (\{\mathrm{a}, \mathrm{b}\} \cup V)^\star$. Since every loop of G must be primitive, $\beta_1, \beta_2 \in \{\mathrm{a}, \mathrm{b}\}^\star$ has to hold. This contravenes the arbitrary size of the blocks of a's. Thus $L(G) \subsetneq L$, a contradiction.

Corollary 3.26. Linear recursive CFL $\subsetneq$ FO[+].

As $L \in$ CFL, the above actually shows linear recursive $\subsetneq \mathrm{FO}[+] \cap \mathrm{CFL}$. However, the separation of $\mathfrak{B}$(linear recursive CFL) from FO[+] fails with this particular L: $\overline{L}$ is generated by a linear-recursive grammar with the rules

$$S \to \mathrm{a}S\mathrm{a},\ S \to \mathrm{a}S\mathrm{b},\ S \to \mathrm{b}S\mathrm{a},\ S \to \mathrm{b}S\mathrm{b},\ S \to \mathrm{b}T\mathrm{b},\ T \to \mathrm{a}T\mathrm{a},\ T \to \mathrm{a}$$

and starting symbol S.

Inherently, the boolean closure adds expressiveness to linear recursive CFL. And yet separation of these two classes fails with every suitable language mentioned before.

Conjecture 3.27. $\mathfrak{B}$(linear recursive CFL) = FO[+].

Another approach to gaining a formal description of the fragment of CFL definable in FO[+] is the relaxation of the restrictions on grammars. However, this approach does not work in a non-trivial way: Dropping the restriction that each two loops have to lie in distinct partitions results in the possibility to specify a grammar for the language $L = \left\{ w \in \{a, b\}^\star \,\middle|\, |w|_a \equiv 0 \pmod 2 \right\} \notin \mathrm{FO}[+]$. For example, $L = L(G)$ for a grammar G with starting symbol S and rules

$$\begin{array}{lll} S \to \varepsilon, & G \to aU, & U \to aG, \\ S \to G, & G \to bG, & U \to bU, \\ & G \to b, & U \to a. \end{array}$$

Likewise dropping the restriction that disallows sharing of nonterminals standing on left-hand sides of loop productions among loops admits a

grammar with starting symbol S and rules

$$S \to \varepsilon,\ S \to T,\ T \to aTa,\ T \to bT,\ T \to Tb,$$

defining the above language L, too.

3.4.2 FO[+] and finite-turn PDAs

In the following the relationship of FO[+] to a superclass of linear recursive grammars will be investigated. This superclass is the set of all languages recognizable by PDAs that change their stack operation only finitely often from push to pop, or vice versa, while working. The idea to investigate such PDAs raises from linear grammars being recognizable by 1-turn PDAs and that concatenation and nesting of linear grammars with primitive loops only results in linear recursive grammars. For an in-depth survey on the connection of finite-turn PDAs to linear grammars, see [Har78].

Definition 3.28. *A* finite-turn PDA $P = (Q, \Sigma, \Gamma, \delta, q_0, \bot, F)$ *is a PDA that turns its stack operation finitely often from push to pop, or vice versa, for all* $w \in L(P)$.

Furthermore, let finite-turn CFL *denote the set of all context-free languages recognizable by finite-turn PDAs.*

Lemma 3.29. *Every linear recursive CFL is recognizable by a finite-turn PDA.*

Proof. Let $G = (V, \Sigma, P, X_0)$, $V = \{X_i \mid 0 \leq i \leq n\}$, be a linear recursive grammar, in which w. l. o. g. every loop is of length at most one. The rules of P are in the form $X_{i_0} \to v_0 X_{i_1} v_1 \cdots v_{s-1} X_{i_s} v_s$, where $X_{i_k} \in V$, $0 \leq k \leq n$, and $v_0, \ldots, v_s \in \Sigma^\star$ with either (1) $i_k \neq i_0$ for all $k > 0$ or (2) $s = 1$. Since G admits a partial order on V, $i_k \geq i_0$ can be assumed for all $k > 0$. Let n_i, $1 \leq i \leq n$, denote the number of productions in P with nonterminal X_i of the left-hand side and let $n_{i,j}$, $1 \leq j \leq n_i$, denote the length of their corresponding right-hand sides.

Furthermore, for $1 \leq i \leq n$, let

$$P_i = \{X_i \to \alpha_{i,j,0} \cdots \alpha_{i,j,n_{i,j}} \,|\, \alpha_{i,j,0}, \cdots, \alpha_{i,j,n_{i,j}} \in \Sigma \cup V, 1 \leq j \leq n_i\} \subseteq P$$

denote all the rules with X_i on the left-hand side.

In the following a PDA $P = (Q, \Sigma, \Gamma, \delta, q_{X_0}, \bot, F)$ with $L(P) = L(G)$ will be defined. The set of states is given as the union of two sets,

$$Q = \{q_{X_i} \,|\, 0 \leq i \leq n\} \cup \{q_{i,j,k} \,|\, 1 \leq i \leq n, 1 \leq j \leq n_i, 1 \leq k \leq n_{i,j}\},$$

where a state in the first set expresses that P now expects a string matching a rule with X_i on the left-hand side; and a state in the second set expresses that the jth rule with nonterminal X_i on the left-hand side has been read up to the kth character.

The stack alphabet is given by

$$\Gamma = \Sigma \cup \{\bot\} \cup \{B_{i,j} \,|\, 1 \leq i \leq n, 1 \leq j \leq n_i\},$$

in which the symbol $B_{i,j}$ represents the jth rule of X_i.

Now define the transition function δ as follows: for every rule in P_i satisfying case (1), i. e. for every $1 \leq j \leq n_i$ with $X_i \to \alpha_{i,j,0} \cdots \alpha_{i,j,n_{i,j}} \in P_i$ and $\alpha_{i,j,0}, \cdots, \alpha_{i,j,n_{i,j}} \in \Sigma \cup V \setminus \{X_i\}$, define

1. $\delta(q_{X_i}, \varepsilon, A) \ni (q_{i,j,0}, A)$,
   for all $A \in \Gamma$;
2. $\delta(q_{i,j,k}, \alpha_{i,j,k+1}, A) \ni (q_{i,j,k+1}, A)$,
   for all $A \in \Gamma, \alpha_{i,j,k+1} \in \Sigma, 0 \leq k < n_{i,j}$;
3. $\delta(q_{i,j,k}, \varepsilon, A) \ni (q_{X_{i'}}, A)$,
   for all $A \in \Gamma, \alpha_{i,j,k+1} = X_{i'} \in V, 0 \leq k < n_{i,j}$;
4. $\delta(q_{i,j,n_{i,j}}, \varepsilon, A) \ni (q_{i',j',k'}, A)$,
   for all $A \in \Gamma, X_{i'} \to \alpha_{i',j',0} \cdots \alpha_{i',j',n_{i',j'}} \in P$ with $\alpha_{i',j',k'} = X_i$.

For every rule in P_i satisfying case (2), i. e. for every $1 \leq j \leq n_i$ with $X_i \to \alpha_{i,j,0} \cdots \alpha_{i,j,n_{i,j}} \in P_i$ and $\alpha_{i,j,0}, \cdots, \alpha_{i,j,n_{i,j}} \in \Sigma \cup \{X_i\}$, define

1. $\delta(q_{X_i}, \varepsilon, A) \ni (q_{i,j,0}, A)$,
   for all $A \in \Gamma$;

2. $\delta(q_{i,j,k}, \alpha_{i,j,k+1}, A) \ni (q_{i,j,k+1}, A)$,
   for all $A \in \Gamma, \alpha_{i,j,k+1} \in \Sigma, 0 \leq k < n_{i,j}$;

3. $\delta(q_{i,j,k}, \varepsilon, A) \ni (q_{X_i}, B_{i,j}A)$,
   for all $A \in \Gamma, \alpha_{i,j,k+1} = X_i, 0 \leq k < n_{i,j}$;

4. $\delta(q_{i,j,n_{i,j}}, \varepsilon, B_{i,j}) \ni (q_{i',j',k'}, \varepsilon)$,
   for all $X_{i'} \to \alpha_{i',j',0} \cdots \alpha_{i',j',n_{i',j'}} \in P$ with $\alpha_{i',j',k'} = X_i$.

Now the set of final states can be specified as the set of states that P resides in after completely reading the last letter of a derivation from the starting symbol X_0. That is

$$F = \{q_{0,j,n_{0,j}} \mid 1 \leq j \leq n_0\}.$$

It remains to show that $L(G) = L(P)$. If $X_{i_0} \Rightarrow_G w \in \Sigma^\star$ is a one-step derivation of some nonterminal X_{i_0} in G, then there is a production $X_{i_0} \to w \in P$ and, according to the above definitions, $(q_{X_{i_0}}, wx, \gamma) \longmapsto_P^\star \delta(q_{i_0,j,n_{i_0,j}}, x, \gamma)$, for all $x \in \Sigma^\star, \gamma \in \Gamma^\star$.

Otherwise, if $X_{i_0} \Rightarrow_G v_0 X_{i_1} v_1 \cdots v_{s-1} X_{i_s} v_s$ is a one-step derivation using the jth rule in P_{i_0}, then for all $w = v_0 w_0 v_1 \cdots v_{s-1} w_s v_s \in \Sigma^\star$ with $X_{i_l} \Rightarrow_G^\star w_l$ for $1 \leq l \leq s$ there is a sequence of configurations

$$\begin{aligned}
(q_{X_{i_0}}, w, \gamma) &\longmapsto_P (q_{i_0,j,0}, wx, \gamma) \\
&\longmapsto_P^\star (q_{i_0,j,|v_0 w_0 \cdots v_l|}, w_l v_{l+1} \cdots v_{s-1} w_s v_s x, \gamma) \\
&\longmapsto_P (q_{X_{i_l}}, w_l v_{l+1} \cdots v_{s-1} w_s v_s x, \gamma) \\
&\longmapsto_P^\star (q_{i_0,j,|v_0 w_0 \cdots v_l w_l|}, v_{l+1} \cdots v_{s-1} w_s v_s x, \gamma) \\
&\longmapsto_P^\star (q_{i_0,j,n_{i_0},j}, x, \gamma)
\end{aligned}$$

for all $x \in \Sigma^\star$, $\gamma \in \Gamma^\star$, $0 \leq l \leq s$. Hence, by induction on the length of a derivation and the choice $i_0 = 0$, $x = \varepsilon$ and $\gamma = \bot$, $(q_{X_0}, w, \bot) \longmapsto_P^\star$

$(q_X, \varepsilon, \bot)$ is obtained. On the other hand, if such a sequence of configurations exists, one can analogously construct a derivation of G. Thus $L(G) = L(P)$. □

Unfortunately this superclass of linear recursive grammars does not capture the context-free languages expressible in FO[+], what strengthens conjecture 3.27.

Theorem 3.30. FO[+] ∩ CFL ⊈ finite-turn CFL.

Proof. Consider the context-free language

$$L = \big\{ w \in \{\mathrm{a}, \mathrm{b}\}^\star \,\big|\, w = w_1 \cdots w_n, w_i = \mathrm{a}^{m_i}\mathrm{b}^{m_i}, m_i > 0, 1 \le i \le n \big\}.$$

Clearly $L \in$ CFL. L is definable in FO[+] by a formula expressing that

1. the word begins with an a,
2. the word ends with a b, and
3. all strings between adjacent positions of the substring ba (including the first a and last b) match $\mathrm{a}^n\mathrm{b}^n$ for some $n > 0$.

The following formula $\varphi \in$ FO[+] defines L:

$$\begin{aligned}\varphi \equiv\ & \underbrace{\exists x\big(\forall y(x \le y) \wedge Q_\mathrm{a}(x)\big)}_{1.} \wedge \underbrace{\exists x\big(\forall y(y \le z) \wedge Q_\mathrm{b}(x)\big)\wedge}_{2.} \\ & \underbrace{\begin{aligned}\forall x \forall y \Big(& x < y \wedge Q_\mathrm{a}(x) \wedge \big(Q_\mathrm{b}(x-1) \vee \forall z(x \le z)\big) \wedge \\ & Q_\mathrm{b}(y) \wedge \big(Q_\mathrm{a}(y+1) \vee \forall z(z \le y)\big) \wedge \\ & \forall z\big(z < x \vee y > z \vee \neg(Q_\mathrm{b}(z) \wedge Q_\mathrm{a}(z+1))\big) \\ & \to \exists z\big(Q_\mathrm{a}(z) \wedge Q_\mathrm{b}(z+1) \wedge \exists d(x + d = z \wedge z + d = y)\big)\Big).\end{aligned}}_{3.}\end{aligned}$$

Next assume L is recognizable by a finite-turn PDA. Let $k \in \mathbb{N}$ be the minimal number of stack-operation turns necessary to recognize L and

choose any word $w \in L$ for which this number of turns is reached. w is of the form $w_1 \cdots w_n, n \in \mathbb{N}$, with $w_i = \mathrm{a}^{m_i}\mathrm{b}^{m_i}$ for all $1 \leq i \leq n$. In order to test the number of a's and the number of b$'s$ in some w_i for equality, at least one stack operation turn is necessary:

Assume there is a PDA P that could test this equality without any stack operation turn. Let Q be the set of states of P and let Γ be its stack alphabet, then there are only $|Q| \cdot |\Gamma|$ possible configurations while reading the block of a's. Thus, if $w_i = \mathrm{a}^{m_i}\mathrm{b}^{m_i}$, with $m_i > |Q| \cdot |\Gamma|$, P has to enter some configuration c twice. Let l be the number of letters read by P between two successive occurrences of c. Then P accepts $w' = w_1 \cdots w_{i-1} w_i' w_{i+1} \cdots w_n, w_i' = a^{2l+d}b^{l+d}$ for all $d \in \mathbb{N}$, because it resides in the same configuration after reading $w_1 \cdots w_{i-1}a^{l+d}$ and $w_1 \cdots w_{i-1}a^{2l+d}$; this contradicts the assumption of the existence of P.

Hence, in order to recognize the word w, at least $n \leq k$ stack operations turns are necessary. Now consider the word $w' = w_1' \cdots w_{k-n+1}'$ with $w_i' = \mathrm{a}^{m_i'}\mathrm{b}^{m_i'}, 1 \leq i \leq k-n+1$ and the concatenation $ww' = w_1 \cdots w_n w_1' \cdots w_{k-n+1}' = \tilde{w}_1 \cdots \tilde{w}_{k+1} \in L$ for suitable $\tilde{w}_i \in \Sigma^\star, 1 \leq i \leq k+1$. It is compelling that ww' is only recognizable with at least $k+1 > k$ stack operation turns. This contradicts the existence of a finite-turn PDA recognizing L. □

From an intuitive point of view, theorem 3.30 states that FO[+] is capable of defining context-free languages not being recognizable with finite-turn PDAs. Hence first-order logic contains formalisms to finitely describe patterns that require a PDA to perform arbitrary many stack operation turns.

Note that the converse of theorem 3.30 does not hold either. The language of words with an even number of occurrences of the symbol a is regular and can therefore be recognized by a finite-turn PDA (which in fact does not use its stack at all). Along with corollary 3.10, it follows:

Corollary 3.31. *1.* Finite-turn CFL $\not\subseteq$ FO[+].

2. Finite-turn CFL *and* FO[+] *are incomparable.*

The resulting inclusion diagram is depicted in figure 3.3.

Besides, $L \in \mathfrak{B}$(linear recursive CFL) holds again, as its complement is generated by a linear recursive grammar with the rules

$$\begin{array}{llll} S \to \mathrm{a}S\mathrm{a}, & T \to \mathrm{a}T\mathrm{b}, & A \to \mathrm{a}A, & B \to \mathrm{b}B, \\ S \to \mathrm{a}S\mathrm{b}, & T \to A, & A \to \mathrm{a}, & B \to \mathrm{b}, \\ S \to \mathrm{b}S\mathrm{a}, & T \to B, & & \\ S \to \mathrm{b}S\mathrm{b}, & & & \\ S \to \mathrm{b}T\mathrm{a} & & & \end{array}$$

and starting symbol S.

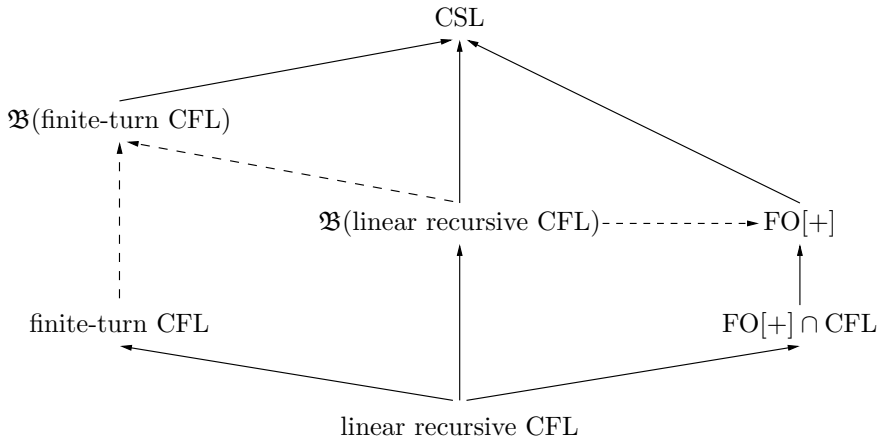

Figure 3.3: Dashed lines are inclusions, solid lines are proper inclusions.

3.5 Context-Free Numerical Predicates

Theorem 3.32. *Let $\mathcal{N}$ denote the set of all numerical predicates, then* $\mathrm{FO}[\mathcal{N}] \cap \mathrm{REG} = \mathrm{FO}[<, \equiv]$, *where $\equiv$ denotes equality modulo n for all $n \in \mathbb{N} \setminus \{0\}$.*

Proof sketch. The inclusion from right to left is trivial regarding section 3.1. The opposite inclusion is proven using the result $\mathrm{FO}[\mathcal{N}] = \mathrm{AC}^0$ (see e. g. [Str74]). Hence it suffices to prove $\mathrm{AC}^0 \cap \mathrm{REG} \subseteq \mathrm{FO}[<, \equiv]$. This is accomplished by showing that the syntactic monoid of any language $L \in \mathrm{AC}^0 \cap \mathrm{REG}$ contains no nontrivial group: otherwise there existed a cyclic group of cardinality $q > 0$ permitting the definition of $\{a_1 \cdots a_n \in \{0,1\}^\star \mid \sum_{i=1}^{n} a_i \equiv 0 \pmod q\} \in \mathrm{AC}^0$, which contradicts $\mathrm{AC}^0 \subsetneq \mathrm{TC}^0$. A detailed version of this proof can be found in [Str74, section IX.3]. □

Theorem 3.32 implies that all regular languages definable in first-order logic with arbitrary numerical predicates are already definable in $\mathrm{FO}[<, \equiv]$. First of all, this shows that $\mathrm{FO}[+]$ does not exceed the definability of $\mathrm{FO}[<, \equiv]$ on the level of the regular languages:

Corollary 3.33. $\mathrm{REG} \cap \mathrm{FO}[+] \setminus \mathrm{FO}[<, \equiv] = \emptyset$.

Subsequently this also raises the question whether there is a set of numerical predicates that captures all context-free languages definable in first-order logic. Due to section 3.1, such a set must be a superset of the addition.

Corollary 3.34. $\mathrm{FO}[+] \cap \mathrm{CFL} \subseteq \mathrm{FO}[\mathcal{N}] \cap \mathrm{CFL}$.

Since $\mathrm{FO}[\mathcal{N}] \cap \mathrm{CFL}$ is not closed under intersection, these two classes can not coincide. Hence the boolean closure of CFL is considered instead. But is the inclusion strict, or are there still other predicates grasping essential parts of context-freeness?

Conjecture 3.35. $\mathrm{FO}[+] \cap \mathfrak{B}(\mathrm{CFL}) = \mathrm{FO}[\mathcal{N}] \cap \mathfrak{B}(\mathrm{CFL})$.

4 Conclusions and Further Work

The central subject considered in this thesis is the relationship of first-order logic to subsets of the context-free languages.

Therefore, a brief summary of the elementary results concerning FO[$<$] and REG has been given initially. Then investigation turned towards FO[+], an extension of ordered first-order logic, by which its definability is extended beyond the star-free regular languages. Examples for languages that are definable and languages that are indefinable in FO[+] have been given and a general upper bound for the definability of FO[+] has been derived.

It has been shown that the boolean closure of linear recursive context-free languages is captured by FO[+]. Further on, both separation and matching of these classes failed, yielding conjecture 3.27:

Is $\mathfrak{B}$(linear recursive CFL) = FO[+]?

In aspiration to gain the opposite of the former inclusion, i. e. the inclusion of FO[+] in a superclass of linear recursive CFL, the relationship of FO[+] to finite-turn CFL has been examined. It has been shown that this superclass does not capture all languages in FO[+] $\cap$ CFL, making the latter both incomparable.

Finally section 3.5 focused on which set of numerical predicates suffices to define all the context-free languages definable in first-order logic (with arbitrary numerical predicates) at all. As no new results were obtained, the situation has been briefly surveyed and the question was raised whether the inclusion FO[$\mathcal{N}$] $\cap$ $\mathfrak{B}$(CFL) $\supseteq$ FO[+] is strict? And if, for which set of numerical predicates does equality hold?

Which set of relations $\mathcal{R}$ satisfies FO[$\mathcal{N}$] $\cap$ $\mathfrak{B}$(CFL) = FO[$\mathcal{R}$]?

One possible approach to this problem offers [BLM93], in which CFL has been proven to coincide with the set of FO[<]-formulas with a single unary Lindström quantifier, $Q^{\text{un}}_{\text{Grp}}\text{FO}[<]$. Hence $\text{FO}[\mathcal{N}] \cap \mathfrak{B}(\text{CFL}) = \text{FO}[\mathcal{N}] \cap \text{FO}[Q^{\text{un}}_{\text{Grp}}]$. Likewise the power of FO[+] on the level of CFL can be expressed as $\text{FO}[+] \cap \text{CFL} = \text{FO}[+] \cap Q^{\text{un}}_{\text{Grp}}\text{FO}[<]$. But no viable application of this reformulation has been found yet, since it simply relocates the context-freeness into an "oracle" question to the Lindström quantifier.

Bibliography

[ABB97] Jean-Michel Autebert, Jean Berstel, and Luc Boasson, *Context-free languages and pushdown automata*, Handbook of Formal Languages, Vol. 1: Word, Language, Grammar (1997), 111–174.

[BIL+01] David A. Mix Barrington, Neil Immerman, Clemens Lautemann, Nicole Schweikardt, and Denis Thérien, *The crane beach conjecture*, Logic in Computer Science, 2001, pp. 187–196.

[BIS90] David A. Mix Barrington, Neil Immerman, and Howard Straubing, *On uniformity within NC*, Journal of Computer and System Sciences **41** (1990), no. 3, 274–306.

[BLM93] F. Bédard, F. Lemieux, and P. McKenzie, *Extensions to Barrington's M-program model*, Theoretical Computer Science **107** (1993), 31–61.

[GS66] S. Ginsburg and E. H. Spanier, *Semigroups, presburger formulas and languages*, Pacific Journal of Mathematics **16** (1966), 285–296.

[Har78] Michael A. Harrison, *Introduction to formal language theory*, Adddion-Wesley, Reading, Massachusetts, 1978.

[Lib04] Leonid Libkin, *Elements of finite model theory*, Springer-Verlag, Berlin, Germany, 2004.

[LMSV01] Clemens Lautemann, Pierre McKenzie, Thomas Schwentick, and Heribert Vollmer, *The descriptive complexity approach to LOGCFL*, Journal of Computer and System Sciences **62** (2001), no. 4, 629–652.

[LS01] Clemens Lautemann and Nicole Schweikardt, *An Ehrenfeucht-Fraïssé approach to collapse results for first-order queries over embedded databases*, STACS '01: Proceedings of the 18th Annual Symposium on Theoretical Aspects of Computer Science (London, UK), Springer-Verlag, 2001, pp. 455–466.

[LST95] Clemens Lautemann, Thomas Schwentick, and Denis Thérien, *Logics for context-free languages*, CSL '94: Selected Papers from the 8th International Workshop on Computer Science Logic (London, UK), Springer-Verlag, 1995, pp. 205–216.

[Lyn82] James F. Lynch, *On sets of relations definable by addition*, Journal of Symbolic Logic **47** (1982), no. 3, 659–668.

[Ruh99] Matthias Ruhl, *Counting and addition cannot express deterministic transitive closure*, Logic in Computer Science, 1999, pp. 326–334.

[Sch01] Nicole Schweikardt, *On the expressive power of first-order logic with built-in predicates*, Ph.D. thesis, Fachbereich Mathematik und Informatik, Johannes Gutenberg-Universität Mainz, Germany, December 2001.

[Str74] Howard Straubing, *Finite automata, formal logic, and circuit complexity*, Birkhäuser Verlag, Basel, Switzerland, 1974.