Underlying Principles and Recurring Ideas of Formal Grammars

after a paper by Alexander Okhotin*

talk by Tobias Denkinger

Freitagsseminar 2018-04-20

* [Okhotin 18]

Overview

1. Grammar types
2. Parse trees
3. Language equations
4. Parsing
5. Characterisation results

Grammar types I

- ordinary grammar (aka context-free grammar) [Chomsky 56]

  ... is a tuple (N, Σ, R, S) of finite sets

  with rules (in R) of the form

  $$A \to u_0 B_1 u_1 \cdots B_k u_k \qquad u_i \in \Sigma^*, B_i \in N$$

Grammar types I

- ordinary grammar (aka context-free grammar) [Chomsky 56]

  ... is a tuple (N, Σ, R, S) of finite sets

  with rules (in R) of the form

  $$A \to u_0 B_1 u_1 \cdots B_k u_k \qquad u_i \in \Sigma^*, B_i \in N$$

characterised by 3 elements:

- constituents (CON) — here: strings
- operations on constituents (C-OP) — here: concatenation $\cdot$
- logical operations (L-OP) — here: disjunction $+$

Grammar types II

- linear grammars: $A \to u B v$ $\quad u, v \in \Sigma^*, A, B \in N$

  C-OP: only concatenation of fixed strings to constituents

Grammar types II

- linear grammars: $A \to uBv$ $\quad u, v \in \Sigma^*, A, B \in N$

  C-OP: only concatenation of fixed strings to constituents

- unambiguous grammars:

  C-OP: $K \cdot L$ only if $\forall w: \exists! (u,v): (u \cdot v = w) \wedge (u \in K) \wedge (v \in L)$

  L-OP: $K + L$ only if $K \cap L = \emptyset$

Grammar types II

- linear grammars: $A \to uBv$ $\quad u,v \in \Sigma^*, A,B \in N$

  C-OP: only concatenation of fixed strings to constituents

- unambiguous grammars:

  C-OP: $K \cdot L$ only if $\forall w: \exists! (u,v): (u \cdot v = w) \wedge (u \in K) \wedge (v \in L)$

  L-OP: $K + L$ only if $K \cap L = \emptyset$

- conjunctive grammar [Okhotin 01, 13]:

  L-OP: + and conjunction &

Grammar types II

- linear grammars: $A \to uBv$ $\quad u,v \in \Sigma^*, A,B \in N$

  C-OP: only concatenation of fixed strings to constituents

- unambiguous grammars:

  C-OP: $K \cdot L$ only if $\forall w: \exists! (u,v): (u \cdot v = w) \wedge (u \in K) \wedge (v \in L)$

  L-OP: $K + L$ only if $K \cap L = \emptyset$

- conjunctive grammar [Okhotin 01, 13]:

  L-OP: + and conjunction &

- Boolean grammar [Okhotin 04, 13]: L-OP: +, &, negation

Grammar types III

- M-component grammars [Vijay-Shanker+87, Seki+91]

  $B_0(\alpha_{1,1}, \ldots, \alpha_{m_0}) \leftarrow B_1(x_{1,1}, \ldots, x_{1,m_1}) \cdots B_k(x_{k,1}, \ldots, x_{k,m_k})$

  CON: ($\leq M$)-tuples of strings $[m_i \leq M]$

  C-OP: "combination" of string tuples

Grammar types III

- M-component grammars [Vijay-Shanker+87, Seki+91]

  $B_0(\alpha_{1,1}, \ldots, \alpha_{m_0}) \leftarrow B_1(x_{1,1}, \ldots, x_{1,m_1}) \cdots B_k(x_{k,1}, \ldots, x_{k,m_k})$

  CON: ($\leq M$)-tuples of strings $[m_i \leq M]$

  C-OP: "combination" of string tuples

- well-nested M-component grammars:

  CON: ($\leq M$)-tuples of strings

  C-OP: "combination" of string tuples, but unrelated variables may not cross each other

Grammar types III

- M-component grammars [Vijay-Shanker+87, Seki+91]

$$B_0(\alpha_{1,1}, \ldots, \alpha_{m_0}) \leftarrow B_1(x_{1,1}, \ldots, x_{1,m_1}) \cdots B_k(x_{k,1}, \ldots, x_{k,m_k})$$

  CON: ($\leq M$)-tuples of strings $\quad [m_i \leq M]$

  C-OP: "combination" of string tuples

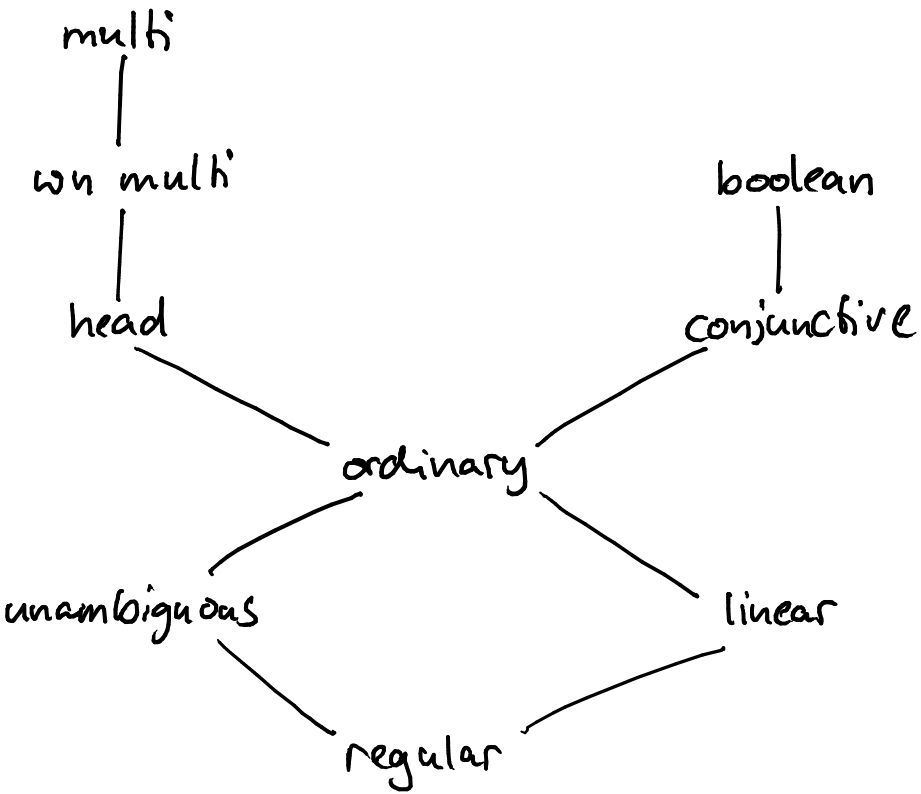
- well-nested M-component grammars:

  CON: ($\leq M$)-tuples of strings

  C-OP: "combination" of string tuples, but unrelated variables may not cross each other

- head grammars = well-nested 2-component grammars

Hierarchy of grammar formalisms (by inclusion)



Parse trees I

- ordinary grammar

$$\frac{B(u) \quad C(v)}{A(uv)}\,(A \to BC)$$

Parse trees I

- ordinary grammar

$$\frac{B(u) \quad C(v)}{A(uv)}\,(A \to BC)$$

example:

$$\dfrac{\dfrac{}{S(c)}\,(S \to c) \quad \dfrac{\dfrac{}{S(\varepsilon)}\,(S \to \varepsilon)}{S(ab)}\,(S \to aSb)}{S(cab)}\,(S \to SS)$$

Parse trees I

- ordinary grammar

$$\frac{B(u) \quad C(v)}{A(uv)}\,(A \to BC)$$

example:

$$\dfrac{\dfrac{}{S(c)}\,(S \to c) \quad \dfrac{\dfrac{}{S(\varepsilon)}\,(S \to \varepsilon)}{S(ab)}\,(S \to aSb)}{S(cab)}\,(S \to SS)$$

- conjunctive grammars

$$\frac{B(u) \quad C(v) \quad D(x) \quad E(y)}{A(uv)}\,(A \to BC \,\&\, DE)$$

Parse trees I

- ordinary grammar

$$\frac{B(u) \quad C(v)}{A(uv)}\,(A \to BC)$$

example:

$$\dfrac{\dfrac{\dfrac{}{S(c)}\,(S \to c) \quad \dfrac{\dfrac{}{S(\varepsilon)}\,(S \to \varepsilon)}{S(ab)}\,(S \to aSb)}{S(cab)}\,(S \to SS)$$

- conjunctive grammars

$$\frac{B(u) \quad C(v) \quad D(x) \quad E(y)}{A(uv)}\,(A \to BC \,\&\, DE)$$

example:

$$\dfrac{\dfrac{\dfrac{}{A(\varepsilon)}}{A(a)} \quad \dfrac{\dfrac{}{B(\varepsilon)}}{B(bc)} \quad \dfrac{\dfrac{}{D(\varepsilon)}}{D(ab)} \quad \dfrac{\dfrac{}{C(\varepsilon)}}{C(c)}}{S(abc)}\,(S \to AB \,\&\, DC)$$

Parse trees II

- multi-component

$$\frac{A(t,u) \qquad B(v,w)}{S(tvuw)} \left(S(x_1 y_1 x_2 y_2) \rightarrow A(x_1, x_2)\ B(y_1, y_2)\right)$$

Parse trees II

- multi-component

$$\frac{A(t,u) \qquad B(v,w)}{S(tvuw)}\ (S(x_1y_1x_2y_2) \rightarrow A(x_1,x_2)\ B(y_1,y_2))$$

example:

$$\dfrac{\dfrac{\dfrac{}{A(\varepsilon,\varepsilon)}}{A(a,c)} \qquad \dfrac{\dfrac{}{B(\varepsilon,\varepsilon)}}{B(b,d)}}{\dfrac{A(aa,cc) \qquad B(b,d)}{S(aabccd)}\ (S(x_1y_1x_2y_2) \rightarrow \ldots)}$$

Language Equations

- ordinary grammars

G: $S \to \varepsilon$ $\quad S \to aSb$ $\quad S \to SS$

Language Equations

- ordinary grammars

G: $S \to \varepsilon$ $\quad S \to aSb$ $\quad S \to SS$

$S = \{\varepsilon\} \cup (\{a\} \cdot S \cdot \{b\}) \cup (S \cdot S)$

$L(G)$ is the least fixed point of the equation (system)

Language Equations

- ordinary grammars

G: $S \to \varepsilon$ $\quad S \to aSb$ $\quad S \to SS$

$$S = \{\varepsilon\} \cup (\{a\} \cdot S \cdot \{b\}) \cup (S \cdot S)$$

$L(G)$ is the least fixed point of the equation (system)

- head/multi-component grammars

→ use sets of tuples of strings as domain

Language Equations

- ordinary grammars

G: $S \to \varepsilon$ $\quad S \to aSb$ $\quad S \to SS$

$S = \{\varepsilon\} \cup (\{a\} \cdot S \cdot \{b\}) \cup (S \cdot S)$

$L(G)$ is the least fixed point of the equation (system)

- head/multi-component grammars

→ use sets of tuples of strings as domain

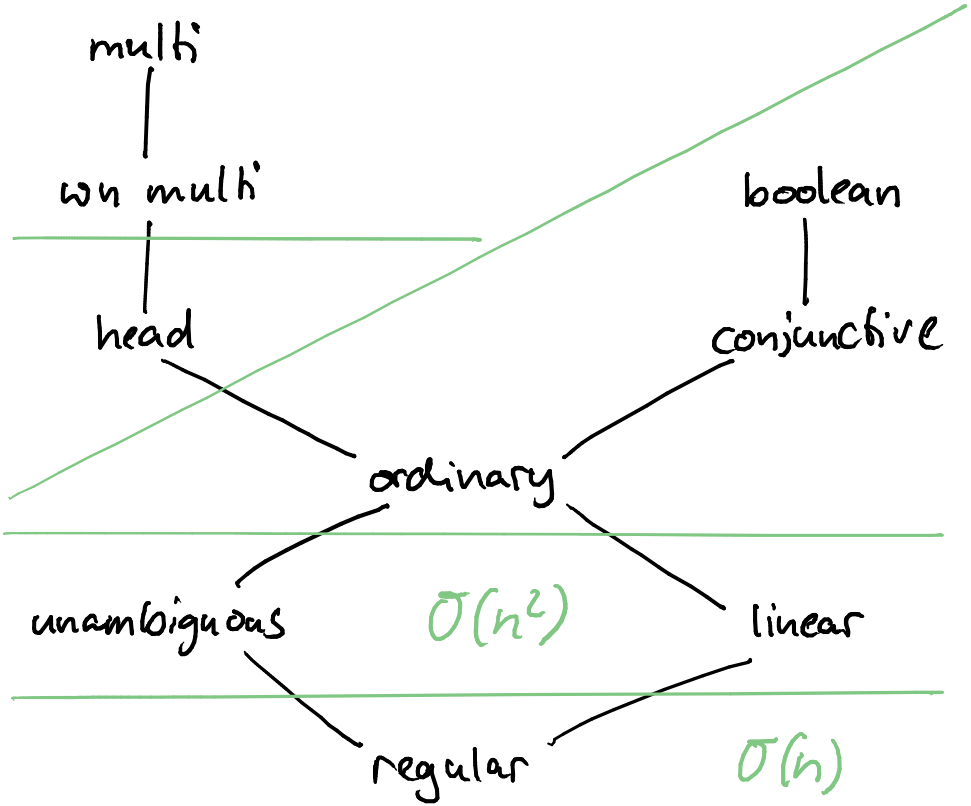
- conjunctive/Boolean grammars

→ & becomes $\cap$, $\neg S$ becomes $\Sigma^* \setminus S$

Parsing

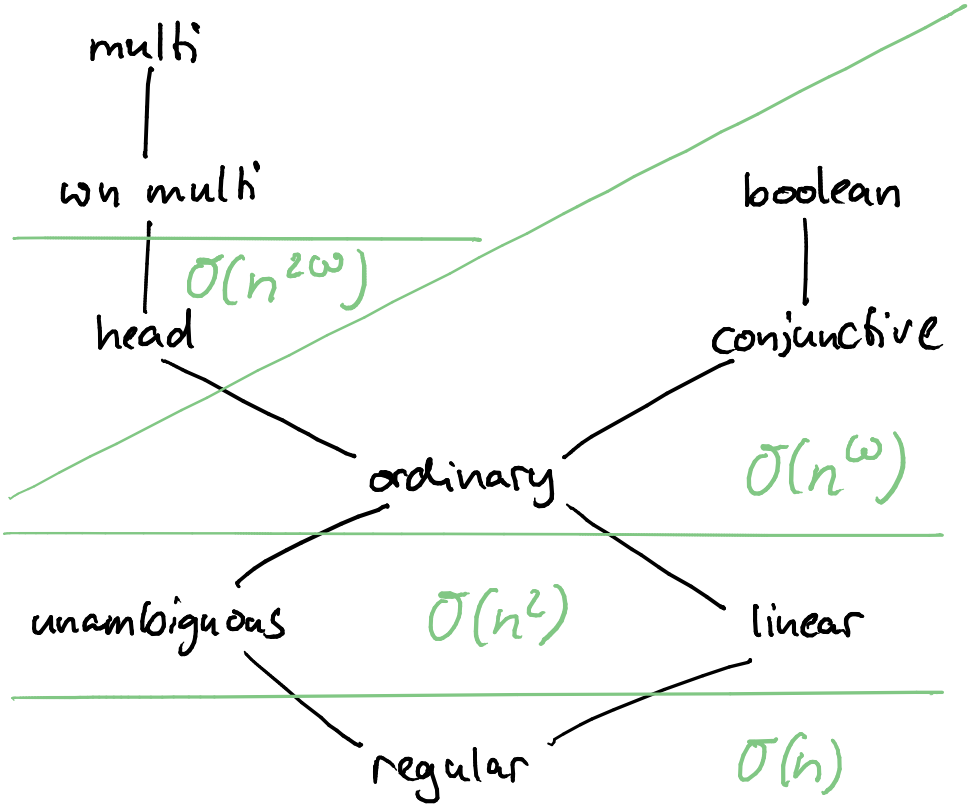
- use dynamic programming algorithm (CKY-like)

Parsing



- use dynamic programming algorithm (CKY-like)

Parsing

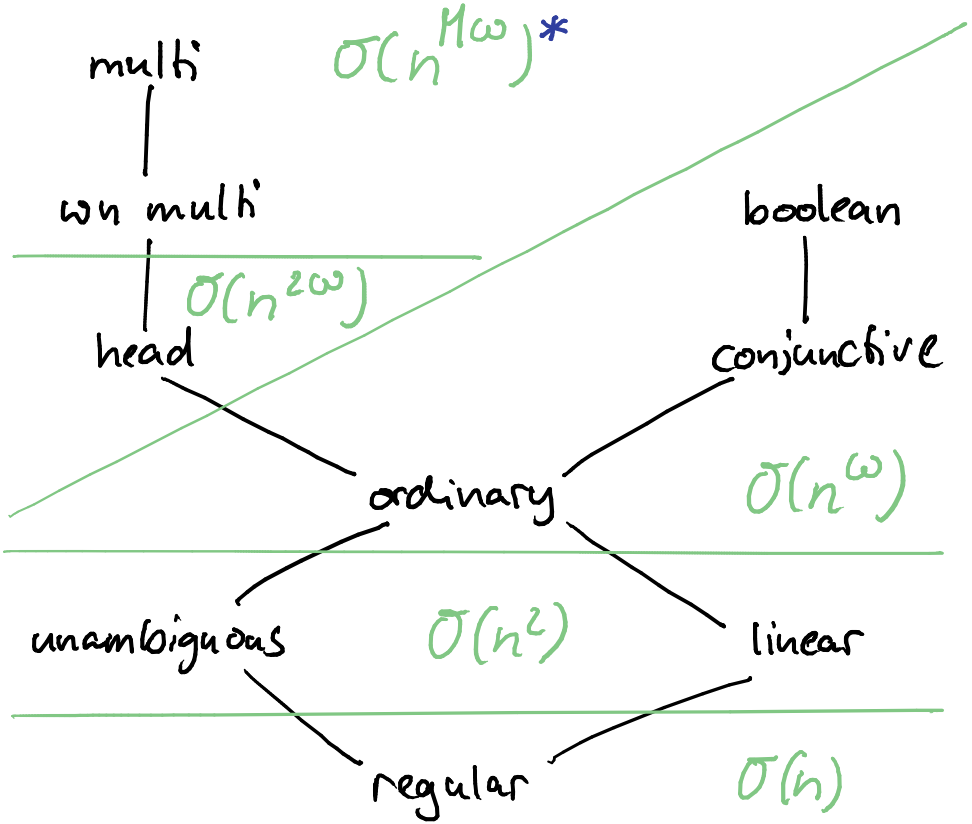


- use dynamic programming algorithm (CKY-like)

- two $n \times n$ matrices can be multiplied in time $O(n^{\omega})$ [Valiant 75]

  $\omega \leq 2.38$

Parsing



- use dynamic programming algorithm (CKY-like)
- two $n \times n$ matrices can be multiplied in time $O(n^{\omega})$ [Valiant 75]
  $\omega \leq 2.38$

* [Cohen+16]

Characterisation results (a selection)

- by logic (Büchi results): find X s.t. $\forall G\ \exists \varphi$ with

  $L(G) = L(\varphi)$ $\quad$ φ is an X-formula

Characterisation results (a selection)

- by logic (Büchi results): find X s.t. $\forall G\ \exists \varphi$ with

  $L(G) = L(\varphi)$ $\quad$ φ is an X-formula

- by automata: find Y s.t. $\forall G\ \exists \mathcal{A}$ with

  $L(G) = L(\mathcal{A})$ $\quad$ $\mathcal{A}$ is an Y-automaton

Characterisation results (a selection)

- by logic (Büchi results): find X s.t. $\forall G\ \exists \varphi$ with

  $L(G) = L(\varphi)$ $\quad$ φ is an X-formula

- by automata: find Y s.t. $\forall G\ \exists \mathcal{M}$ with

  $L(G) = L(\mathcal{M})$ $\quad$ $\mathcal{M}$ is an Y-automaton

- homomorphic characterisation (Chomsky-Schützenberger results)

  find $\mathcal{Z}$ s.t. $\forall G\ \exists h, R, D$ with

  $L(G) = h(R \cap D)$ $\quad$ $R \in REG$, $h \in HOM$,

  D is a $\mathcal{Z}$-Dyck language

References I (in order of appearance)

[Okhotin 18] A. Okhotin: Underlying Principles and Recurring Ideas of Formal Grammars, 2018. DOI.

[Chomsky 56] N. Chomsky: Three models for the description of language, 1956. DOI.

[Okhotin 01] A. Okhotin: Conjunctive Grammars, 2001.

[Okhotin 04] A. Okhotin: Boolean Grammars, 2004. DOI.

[Okhotin 13] A. Okhotin: Conjunctive and Boolean grammars, the true general case of the context-free grammars, 2013. DOI.

[Vijay-Shankes 87] K. Vijay-Shankes, D.J. Weir: Characterizing structural descriptions produced by various grammar formalisms, 1987. DOI.

References II (in order of appearance)

[Seki+91] H. Seki, T. Matsumura, M. Fujii, T. Kasami: On multiple context-free grammars, 1991. DOI.

[Valiant75] L. G. Valiant: General context-free recognition in less than cubic time, 1975. DOI.

[Cohen+16] S. B. Cohen, D. Gildea: Parsing Linear Context-Free Rewriting Systems with Fast Matrix Multiplication, 2016. DOI.