Грамматики и синтаксический анализ

Евгений Борисов

Синтаксический анализ

Уровни сложности при автоматической обработке текстов

Прагматика (Дискурс) - смысловые контексты
Семантика - смыслы последовательностей слов
Синтаксис - последовательности слов
Лексика - отдельные слова и устойчивые словосочетания

Синтаксический анализ

Задача автоматического синтаксического разбора

Применение:

- машинный перевод

- извлечение информации

- диалоговые системы

Синтаксический анализ

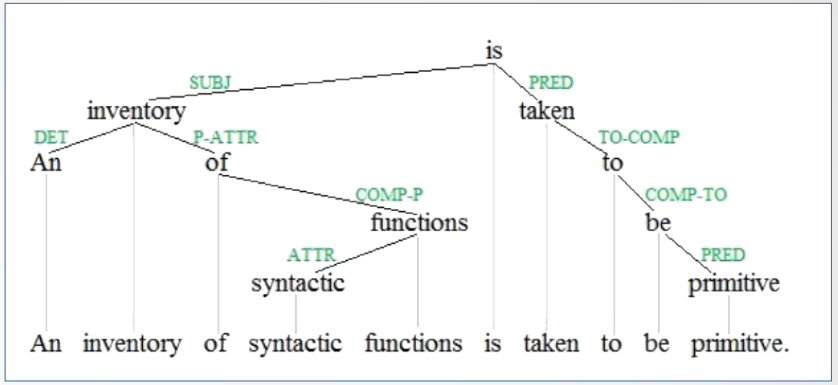
Грамматика — способ описания языка

- грамматика зависимостей

- грамматика составляющих

Синтаксический анализ

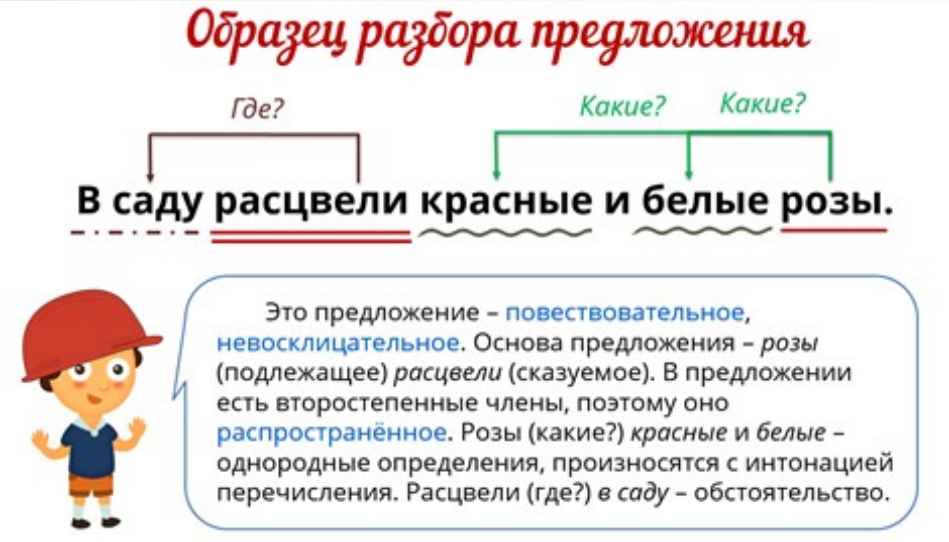
Грамматика зависимостей (Dependency grammar)



Главные члены предложения

Подлежащее — предмет. кто? что?

Сказуемое — что делать? что сделать? каков?



Второстепенные члены предложения

Определение — признак предмета.
какой? чей? который?

Обстоятельство — время, место, способ действия.
где? когда? куда? откуда? почему? зачем? как?

Дополнение — предмет. кого? чего? кому? чему? кого? что? кем? чем? о ком? о чём?

Синтаксический анализ

Грамматика составляющих (Constituency grammar)

предложение (П; sentence, S).

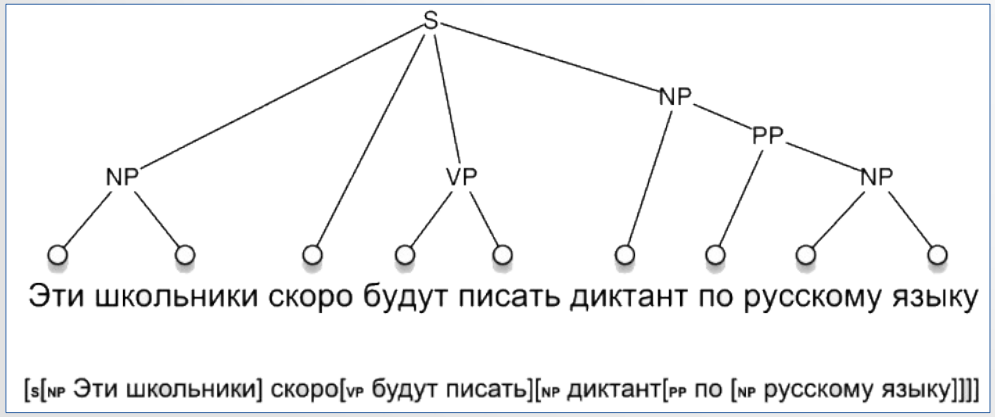
именная группа (группа существительного, ИГ; noun phrase, NP) — возглавляется существительным;

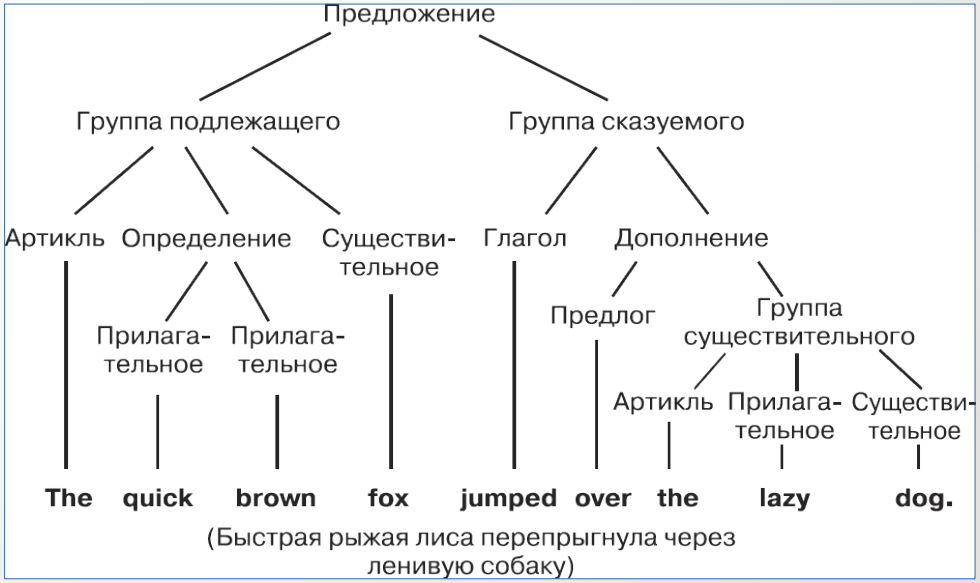
группа прилагательного (ГПрил; adjectival phrase, AP) — возглавляется прилагательным;

наречная группа (НарГ; adverbial phrase, AdvP) — возглавляется наречием;

предложная группа (ПрГ; prepositional phrase, PP) — возглавляется предлогом;

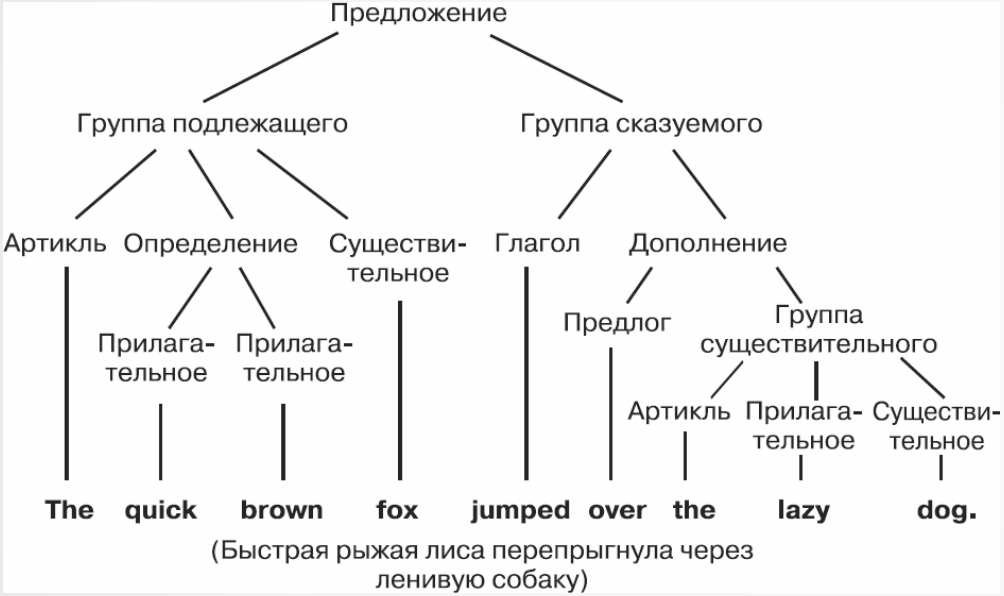
глагольная группа (ГГ; verb phrase, VP) — возглавляется глаголом;



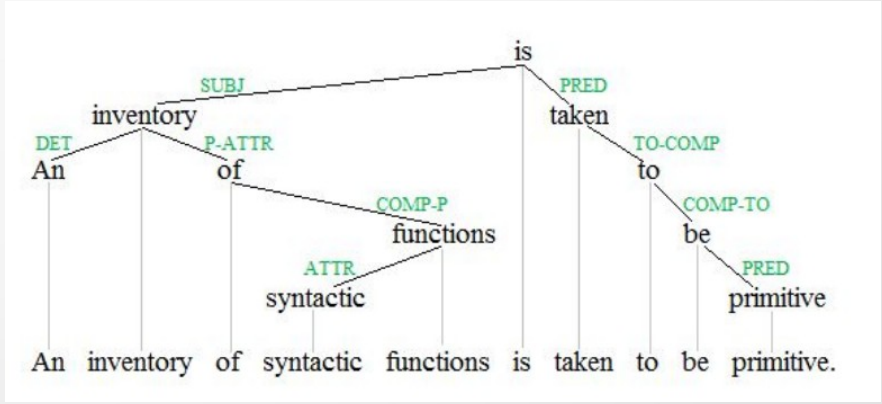


Синтаксический анализ

грамматика составляющих



грамматика зависимостей



Грамматика составляющих

Синтаксический анализ

грамматика составляющих (constituency grammar)
— разметка (вложенных) групп

именная группа (группа существительного, ИГ; англ. noun phrase, NP) — возглавляется существительным;

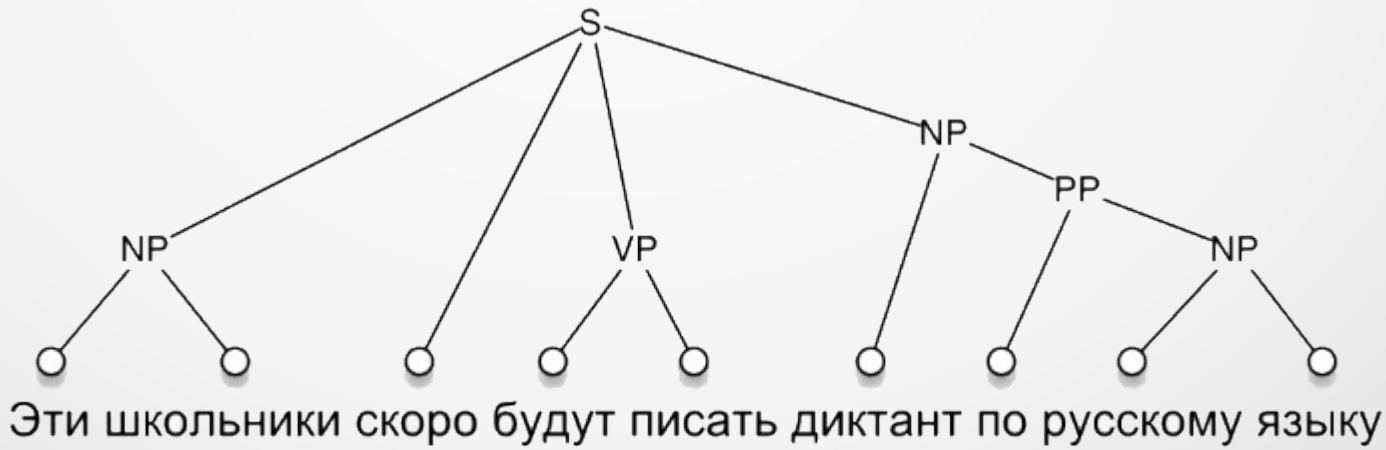
группа прилагательного (ГПрил; adjectival phrase, AP) — возглавляется прилагательным;

наречная группа (НарГ; adverbial phrase, AdvP) — возглавляется наречием;

предложная группа (ПрГ; prepositional phrase, PP) — возглавляется предлогом;

глагольная группа (ГГ; verb phrase, VP) — возглавляется глаголом;

предложение (П; sentence, S).



[S[NP Эти школьники] скоро[VP будут писать][NP диктант[PP по [NP русскому языку]]]]

Синтаксический анализ

Формальная грамматика - способ описания языка

$$G=(N,\Sigma,R,s);V=N\cup\Sigma$$

N — множество (алфавит) нетерминальных символов (синтаксические переменные или понятия)

Σ - множество (алфавит) терминальных символов (не пересекается с N)

V - словарь грамматики G

s - начальный нетерминал (принадлежит алфавиту нетерминалов N)

R - конечное множество правил вывода (продукции),
вида A → b, где A, b — последовательности символов из алфавита V грамматики G

Нетерминальные символы
- объекты, обозначающие какую-либо сущность языка (предложение, формула и т.д.).

Терминальные символы
- объекты непосредственно присутствующие в языке.

Синтаксический анализ

Форма Бэкуса-Наура (БНФ) - способ представления КС-грамматик

Lex/Yacc калькулятор:

```
%token INTEGER

expr: INTEGER
    | "-"  expr
    | "(" expr ")"
    | expr "-" expr
    | expr "+" expr
    | expr "*" expr
    | expr "/" expr
```

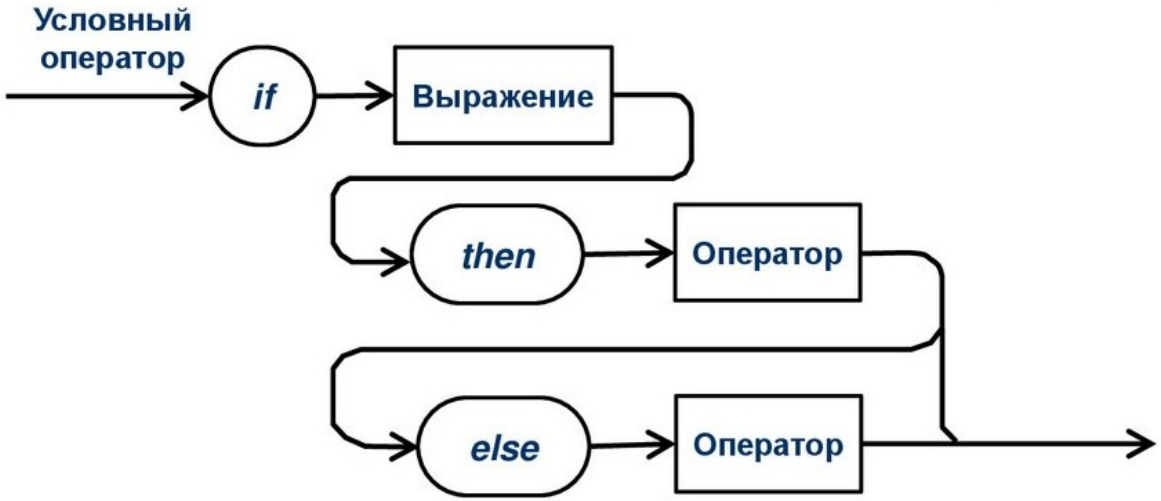
expr — нетерминальный символ (объект, обозначающий сущность языка)

- + ( ) * / число — терминальные символы (объекты непосредственно присутствующие в языке)

(пример описывает только INT)

Синтаксический анализ

БНФ и диаграммы Вирта



<Условный оператор> ::=
if <Выражение> *then* <Оператор>
[*else* <Оператор>]

Синтаксический анализ

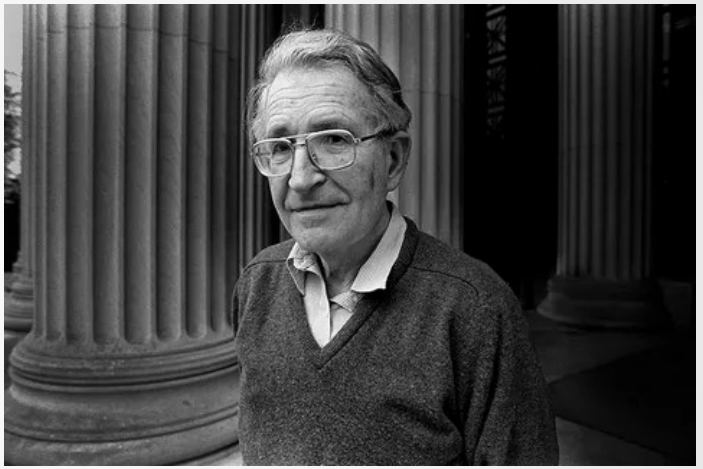
классификация формальных грамматик по Хомскому

неограниченные

R : B → β

B — любая непустая последовательность из V содержащая нетерминалы
β — любая (в т.ч. пустая) последовательность из V

$$G=(N,\Sigma,R,s);V=N\cup\Sigma$$



Avram Noam Chomsky

Синтаксический анализ

классификация формальных грамматик по Хомскому

неограниченные

R : B → β

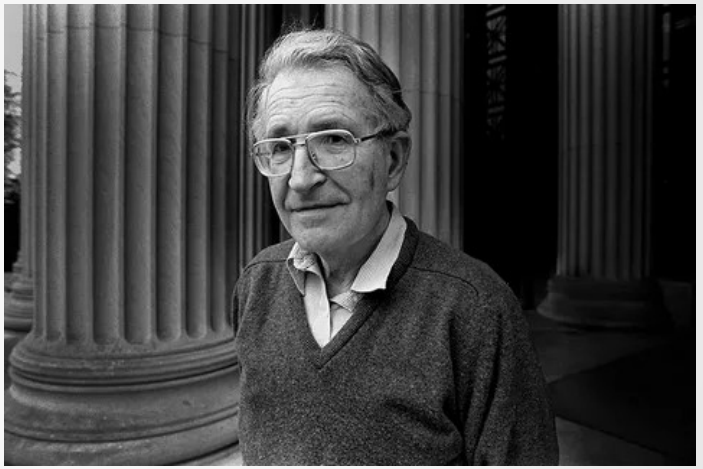
B — любая непустая последовательность из V содержащая нетерминалы
β — любая (в т.ч. пустая) последовательность из V

контекстно-зависимые

R : αAβ → αγβ

A — нетерминал из N
α,β — любые (в т.ч. пустые) последовательности из V
γ — любая непустая последовательность из V

$$G=(N,\Sigma,R,s);V=N\cup\Sigma$$



Avram Noam Chomsky

Синтаксический анализ

классификация формальных грамматик по Хомскому

неограниченные

R : B → β

B — любая непустая последовательность из V содержащая нетерминалы
β — любая (в т.ч. пустая) последовательность из V

контекстно-зависимые

R : αAβ → αγβ

A — нетерминал из N
α,β — любые (в т.ч. пустые) последовательности из V
γ — любая непустая последовательность из V

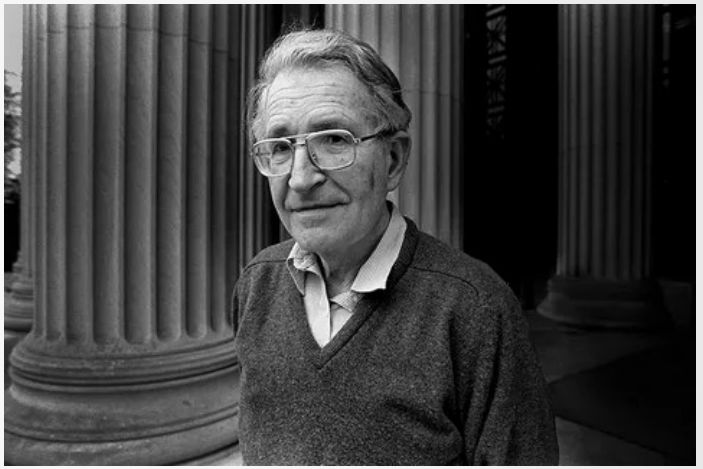
контекстно-свободные

R : A → β

A — нетерминал из N
β — любая (в т.ч. пустая) последовательность из V

применяются для описания компьютерных языков

$$G=(N,\Sigma,R,s);V=N\cup\Sigma$$



Avram Noam Chomsky

Синтаксический анализ

классификация формальных грамматик по Хомскому

неограниченные

R : B → β

B — любая непустая последовательность из V содержащая нетерминалы
β — любая (в т.ч. пустая) последовательность из V

контекстно-зависимые

R : αAβ → αγβ

A — нетерминал из N
α,β — любые (в т.ч. пустые) последовательности из V
γ — любая непустая последовательность из V

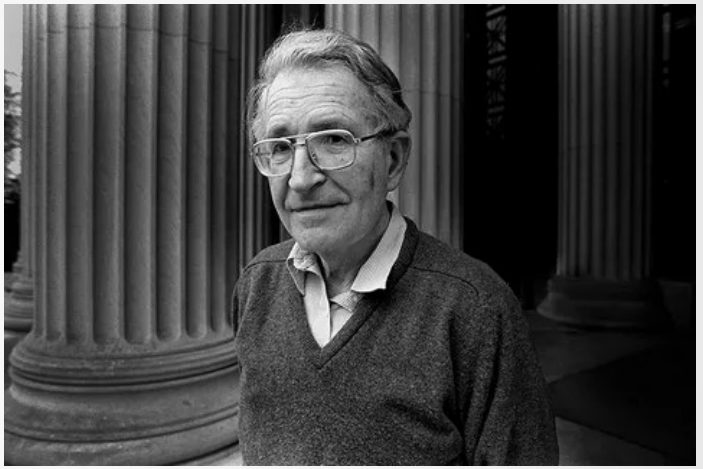
контекстно-свободные

R : A → β

A — нетерминал из N
β — любая (в т.ч. пустая) последовательность из V

применяются для описания компьютерных языков

$$G=(N,\Sigma,R,s);V=N\cup\Sigma$$



Avram Noam Chomsky

регулярные

R : A → Aβ , A → β (леворекурсивные)

R : A → βA , A → β (праворекурсивные)

A — нетерминал из N
β — последовательность (в т.ч. пустая) терминалов из Σ

применяются для описания простых конструкций

Синтаксический анализ

Нормальная форма грамматики по Хомскому (CNF)

продукции имеют вид: $A \to BC$, $A \to \alpha$, $s \to \varepsilon$,

где
A, B, C – нетерминалы (B и C не могут являться начальными символами),
α – терминальный символ,
s – начальный символ,
ε – пустая строка (грамматика может порождать пустую строку)

Эквивалентность грамматик

- сильная (совпадает язык и дерево разбора)

- слабая (совпадает язык, деревья разбора могут отличаться)

Теорема:
любая КС-грамматика может быть преобразована в эквивалентную CNF-грамматику.

Синтаксический анализ

Методы синтаксического разбора

- рекурсивный спуск (top-down parsing)

- восходящий анализ (bottom-up parsing)

- алгоритм Кока-Янгера-Касами (CKY parsing)

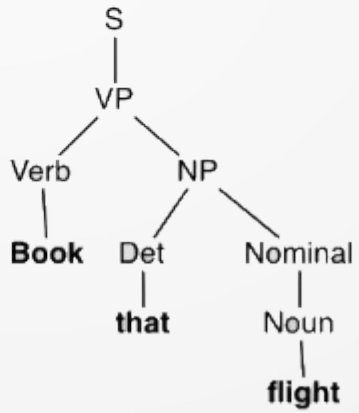
- алгоритм Эрли (Earley parser)

...

Синтаксический анализ

S → NP VP
S → Aux NP VP
S → VP
NP → Pronoun
NP → Proper-Noun
NP → Det Nominal
Nominal → Noun
Nominal → Nominal Noun
Nominal → Nominal PP
VP → Verb
VP → Verb NP
VP → Verb NP PP
VP → Verb PP
VP → VP PP
PP → Preposition NP

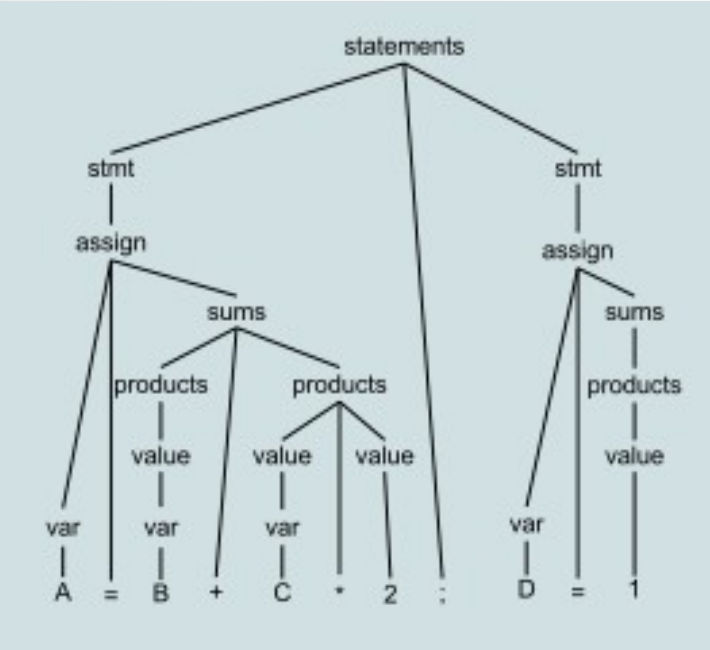
Det → that | this | a
Noun → book | flight | meal | money
Verb → book | include | prefer
Pronoun → I | she | me
Proper-Noun → Houston | TWA
Aux → does
Preposition → from | to | on | near | through

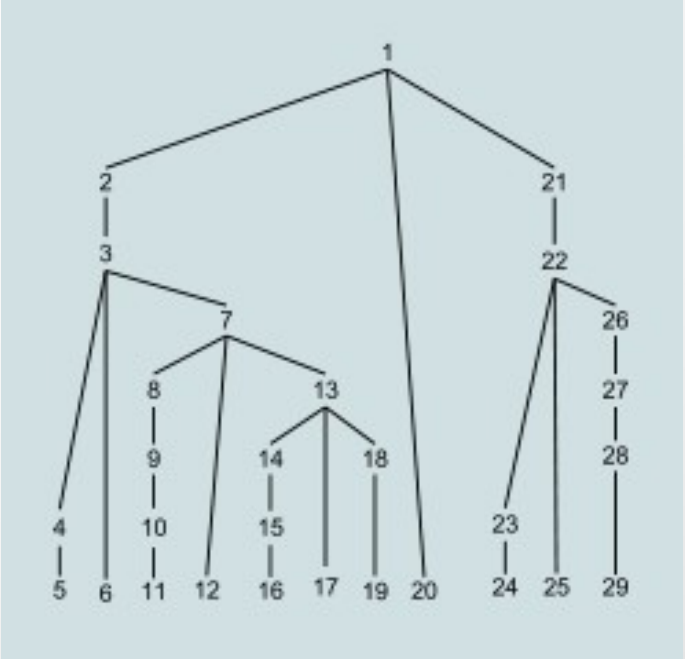


Синтаксический анализ

Анализ рекурсивным спуском (top-down parsing)

правила формальной грамматики раскрываются, начиная со стартового символа, до получения требуемой последовательности токенов.



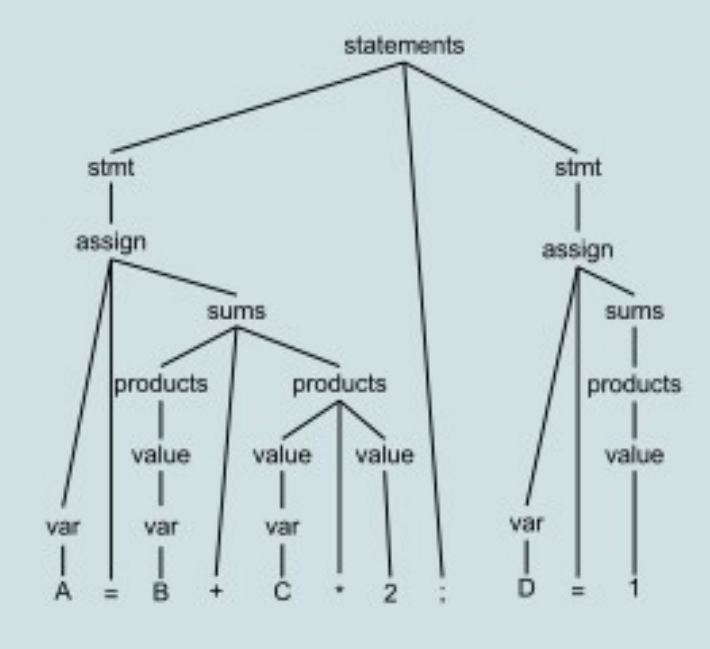


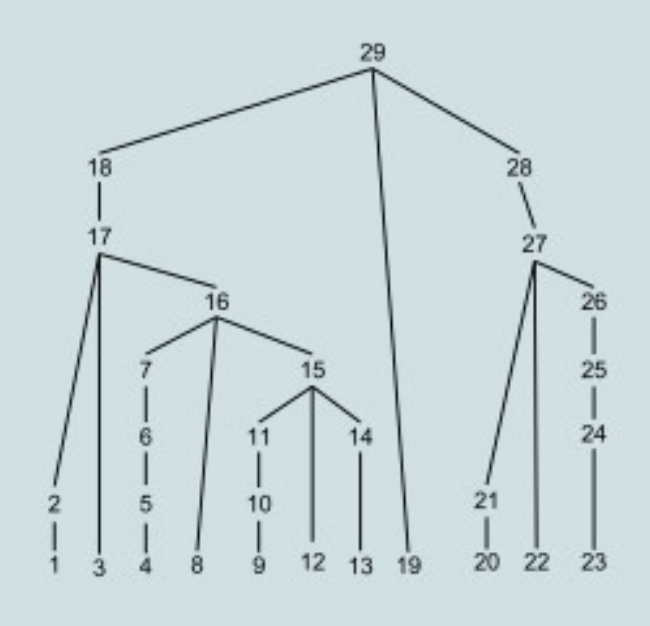
https://wikijaa.ru/wiki/Bottom-up_parsing

Синтаксический анализ

Восходящий анализ (bottom-up parsing)

сначала распознает мелкие детали самого низкого уровня текста, а затем его структуры среднего уровня, и оставляет общую структуру самого высокого уровня на потом.





Синтаксический анализ

Частичный разбор (группировка, применение группы правил)

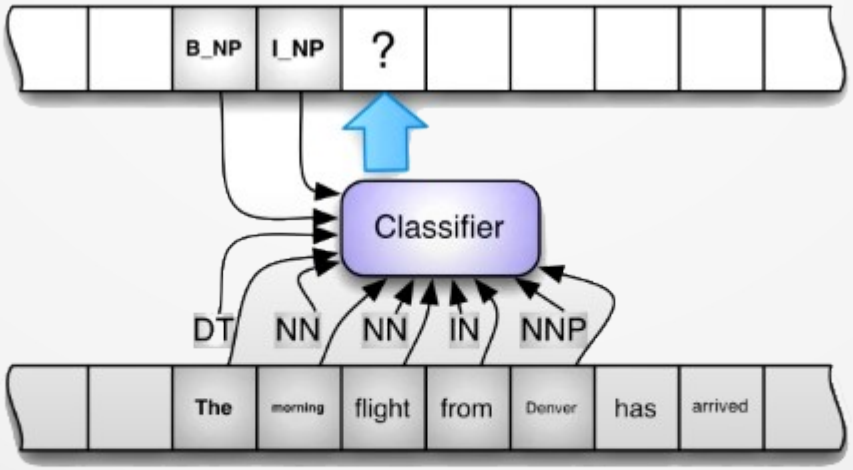
Применяется для извлечения именованных сущностей

- Partial parsing, Shallow parsing
- Chunking, фрагментирование
  - –[NP The morning flight][PP from][NP Denver][VP has arrived]
  - –[NP The morning flight] from [NP Denver] has arrived

Синтаксический анализ

Группировка на основе машинного обучения

- Классы BIO (begin, inside, outside)
- Тренировочное множество - Treebank



Признаки: *The, DT, B_NP, morning, NN, I_NP, flight, NN, from, IN, Denver, NNP*

Синтаксический анализ

Статистические КС-грамматики

$$G=(N,\Sigma,R,s);V=N\cup\Sigma$$

N — множество (алфавит) нетерминальных символов (синтаксические переменные или понятия)

Σ - множество (алфавит) терминальных символов (не пересекается с N)

V - словарь грамматики G

s - начальный нетерминал (принадлежит алфавиту нетерминалов N)

R - конечное множество правил вывода (продукции),
вида **A → β[p]**
где
A — нетерминал из N
β — последовательности символов из алфавита V грамматики G
p — вероятность правила P(β|A) (сумма вероятностей всех правил вида A → * равна 1)

Нетерминальные символы
- объекты, обозначающие какую-либо сущность языка (предложение, формула и т.д.).

Терминальные символы
- объекты непосредственно присутствующие в языке.

Синтаксический анализ

Статистические КС-грамматики

Грамматика	Вероятность		
S → NP VP	0.8		
S → Aux NP VP	0.1	+	1.0
S → VP	0.1		
NP → Pronoun	0.2		
NP → Proper-Noun	0.2	+	1.0
NP → Det Nominal	0.6		
Nominal → Noun	0.3		
Nominal → Nominal Noun	0.2	+	1.0
Nominal → Nominal PP	0.5		
VP → Verb	0.2		
VP → Verb NP	0.5	+	1.0
VP → VP PP	0.3		
PP → Prep NP	1.0		

Лексикон

Det → the | a | that | this
0.6 0.2 0.1 0.1

Noun → book | flight | meal | money
0.1 0.5 0.2 0.2

Verb → book | include | prefer
0.5 0.2 0.3

Pronoun → I | he | she | me
0.5 0.1 0.1 0.3

Proper-Noun → Houston | NWA
0.8 0.2

Aux → does
1.0

Prep → from | to | on | near | through
0.25 0.25 0.1 0.2 0.2

https://www.youtube.com/watch?v=Cq9E4Fa2Jsk

Синтаксический анализ

Статистические КС-грамматики

Разрешение многозначности

- Вероятность разбора

$$P(T,S) = \prod_{i=1}^{n} P(RHS_i|LHS_i)$$

- Вероятность $P(T,S) = P(T)P(S|T) = P(T)$
- Выбор наиболее вероятного дерева разбора $\hat{T}(S) = \arg\max_T P(T|S)$

$$\hat{T}(S) = \arg\max_T \frac{P(T,S)}{P(S)}$$

$$\hat{T}(S) = \arg\max_T P(T,S)$$

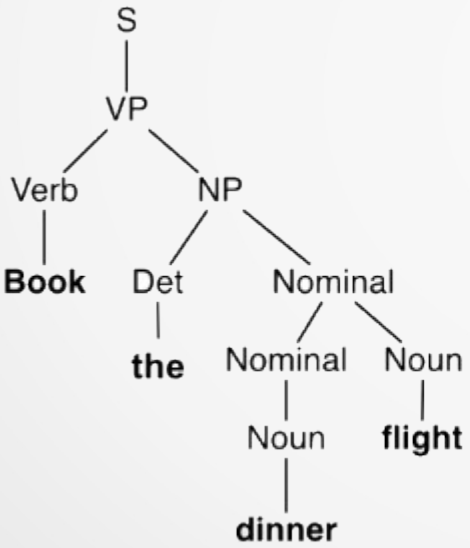
$$\hat{T}(S) = \arg\max_T P(T)$$

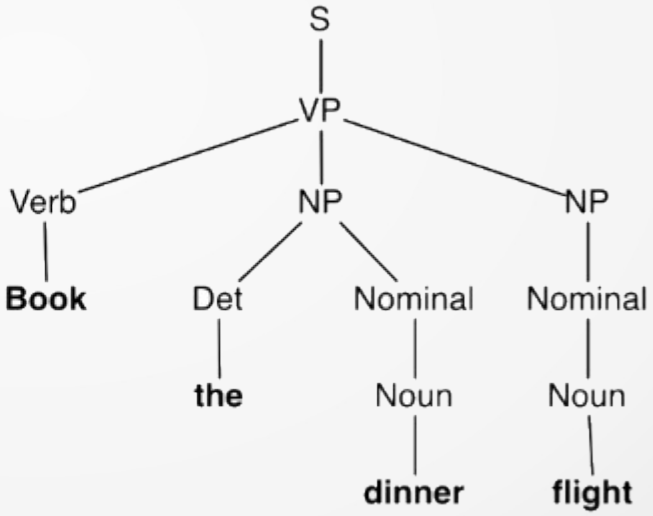
https://www.youtube.com/watch?v=Cq9E4Fa2Jsk

Синтаксический анализ

Статистические КС-грамматики

Разрешение многозначности





P(T-left) = .05*.20*.20*.20*75*.30*.60*.10*.40=**2.2*10 $^{-6}$**
P(T-right) = .05*.10*.20*.15*.75*.75*.30*.60*.10*.40=**6.1*10 $^{-7}$**

https://www.youtube.com/watch?v=Cq9E4Fa2Jsk

Синтаксический анализ

Статистические КС-грамматики

Обучение СКС

- Вычисление вероятности на основе банка деревьев

$$P(\alpha \rightarrow \beta|\alpha) = \frac{Count(\alpha \rightarrow \beta)}{\sum_{\gamma} Count(\alpha \rightarrow \beta)} = \frac{Count(\alpha \rightarrow \beta)}{Count(\alpha)}$$

- Вывод без тренировочного множества (EM)
  - На основе множества предложений построить множество наиболее вероятных синтаксических разборов
  - Обновить значения вероятностей на основе полученных данных
  - (Manning and Schutze 1999)

https://www.youtube.com/watch?v=Cq9E4Fa2Jsk

Синтаксический анализ

Оценка качества алгоритма

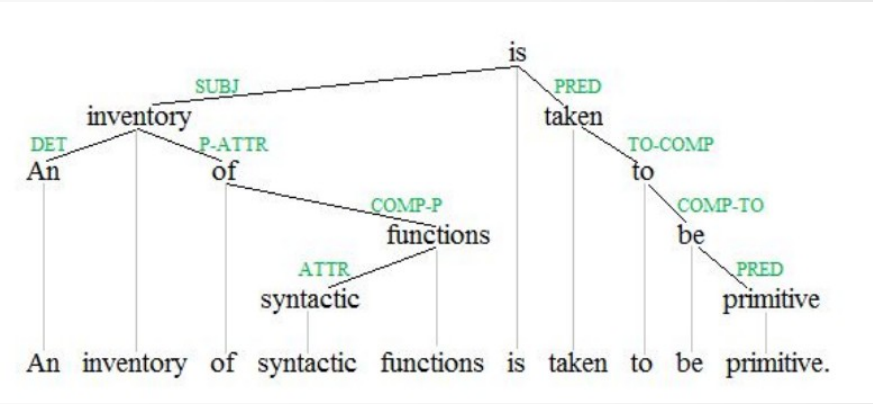
- Метрика PARSEVAL: пусть P - дерево разбора, созданное алгоритмом, T - дерево разбора, созданное экспертами
  - Точность = (# правильных компонент в P) / (# компонент в T)
  - Полнота = (# правильных компонент в P) / (# компонент в P)
  - F-мера = 2PR / (P + R)
- Современные алгоритмы показывают точность и полноту более 90%

https://www.youtube.com/watch?v=Cq9E4Fa2Jsk

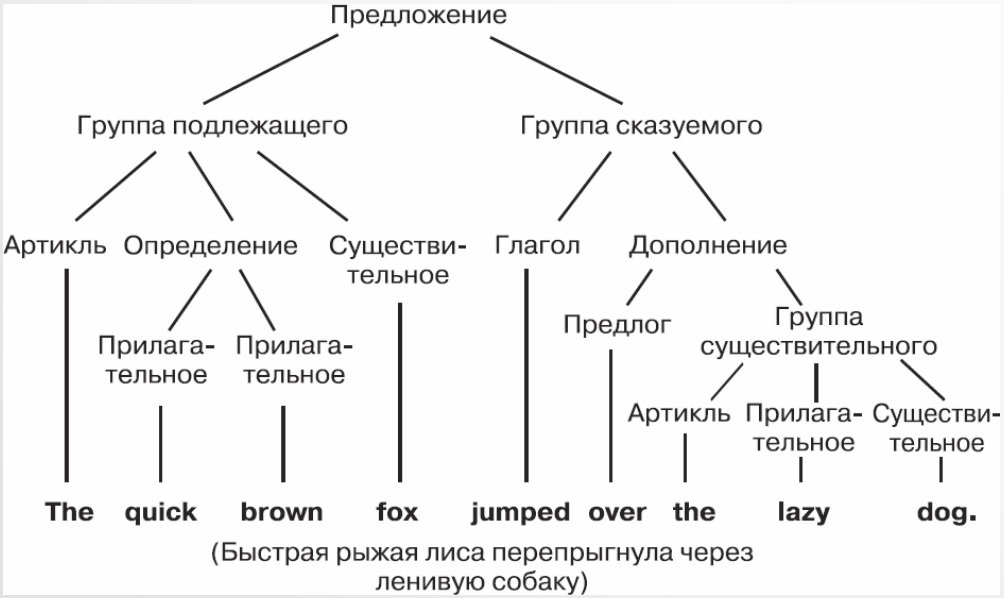
Грамматика зависимостей

Синтаксический анализ

грамматика зависимостей

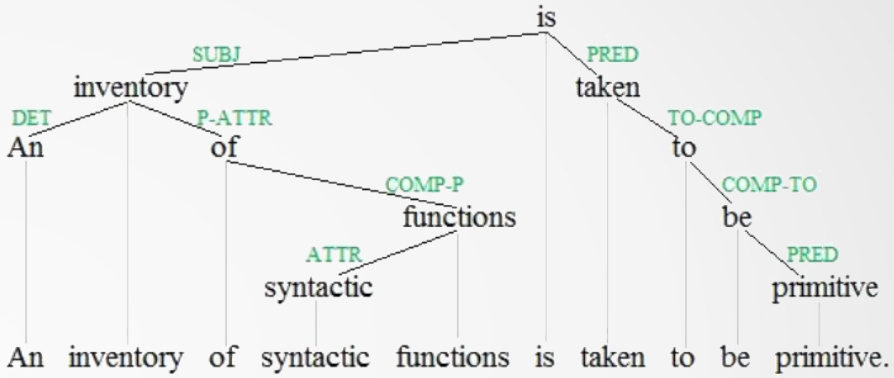


грамматика составляющих



Синтаксический анализ

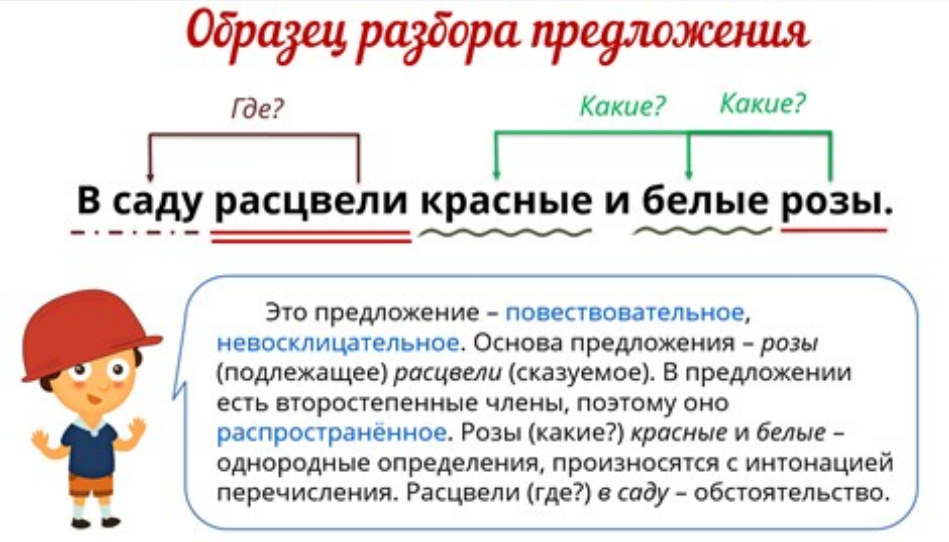
Грамматика зависимостей (Dependency grammar)



Главные члены предложения

Подлежащее — предмет. кто? что?

Сказуемое — что делать? что сделать? каков?



Второстепенные члены предложения

Определение — признак предмета.
какой? чей? который?

Обстоятельство — время, место, способ действия.
где? когда? куда? откуда? почему? зачем? как?

Дополнение — предмет. кого? чего? кому? чему? кого? что? кем? чем? о ком? о чём?

Синтаксический анализ

Разбор в грамматику зависимостей

- Dependency parser
  - –Malt parser (2006)
  - –Stanford Neural Network Dependency Parser (2014)

строим ориентированный граф зависимостей
на упорядоченном множестве слов

https://www.youtube.com/watch?v=Cq9E4Fa2Jsk

Синтаксический анализ

Разбор в грамматику зависимостей

- Итеративный алгоритм разбора предложения $w_1, w_2, \ldots, w_n$
- Состояние парсера $c = (s, b, A)$:
  - стек $s = [Root]$
  - буфер $b = [w_1, w_2, \ldots, w_n]$
  - множество дуг зависимостей $A = \emptyset$

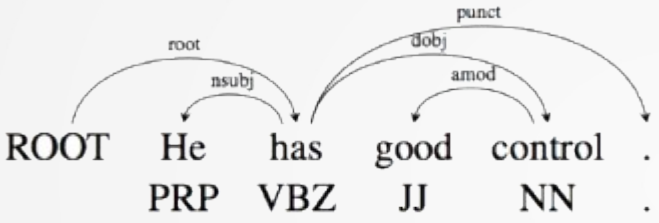
- Остановка
  - стек содержит один узел *ROOT*
  - и буфер пуст
- На каждой итерации происходит выбор одного из трех правил (для выбора используется классификатор)
  - *LEFT-ARC(label): добавление дуги* $s_1 \rightarrow s_2$ *с меткой label и удаление* s_2 *из стека. Предусловие:* $|s| \geq 2$
  - *RIGHT-ARC(label): добавление дуги* $s_2 \rightarrow s_1$ *с меткой label и удаление* s_1 *из стека. Предусловие:* $|s| \geq 2$
  - *SHIFT: перенос* b_1 *из буфера в стек.* $|b| \geq 1$

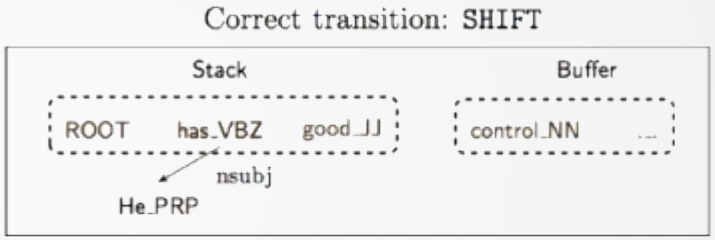
используем ML классификатор для выбора правила на каждом шаге

https://www.youtube.com/watch?v=Cq9E4Fa2Jsk

Синтаксический анализ

Пример





Transition	Stack	Buffer	A
	[ROOT]	[He has good control .]	$\emptyset$
SHIFT	[ROOT He]	[has good control .]	
SHIFT	[ROOT He has]	[good control .]	
LEFT-ARC(nsubj)	[ROOT has]	[good control .]	$A \cup$ nsubj(has,He)
SHIFT	[ROOT has good]	[control .]	
SHIFT	[ROOT has good control]	[.]	
LEFT-ARC(amod)	[ROOT has control]	[.]	$A \cup$amod(control,good)
RIGHT-ARC(dobj)	[ROOT has]	[.]	$A \cup$ dobj(has,control)
...	...	...	...
RIGHT-ARC(root)	[ROOT]	[]	$A \cup$ root(ROOT,has)

https://www.youtube.com/watch?v=Cq9E4Fa2Jsk

Синтаксический анализ

Литература

Борисов Е.С. Методы машинного обучения. 2024
https://github.com/mechanoid5/ml_lectorium_2024_I

Борисов Е.С. Методы обработки текстов на естественном языке. 2024
https://github.com/mechanoid5/ml_nlp_2024_I

Турдаков Д.Ю. Основы обработки текстов. Лекция 7.
Формальные грамматики и синтаксический анализ. ИСП РАН, 2017
https://www.youtube.com/watch?v=TkMtUm-D6aE

Steven Bird, Ewan Klein, and Edward Loper Analyzing Text with the Natural Language Toolkit
https://www.nltk.org/book/

D.Jurafsky,J.H.Martin Speech and Language Processing. third edition, 2020

А. Ахо, Дж. Ульман. Теория синтаксического анализа, перевода и компиляции. М.: Мир, 1978.

Д.Кук, Г.Бейз Компьютерная математика - Москва: Наука, 1990

В.С.Проценко, П.Й.Чаленко Элементы компиляции - Киев: УМК ВО, 1988

Е.С.Борисов Методы и средства построения грамматических анализаторов.
http://mechanoid.su/programming-grammar-analysis.html