

Порождающие грамматики

Лекция №4

Теории Андрея Маркова (младшего)

1. Нормальный алгоритм Маркова
2. Цепи Маркова
 - статистический подход к обработке языков
 - ключевая парадигма в NLP *

* NLP — Natural Language Processing

(обработка естественных языков)

Нормальный алгоритм Маркова

Абстрактная машина:

1. Обрабатывает цепочки символов
2. Имеет набор правил преобразования
 - $AB \rightarrow Ade$
 - $C \rightarrow ff$
3. Алгоритм — это конечный ряд подстановок

Эквивалент: машина Тьюринга.

Порождающие грамматики

- Представлены в 1957 году на примере английского языка
 - Книга “Syntactic Structures”, Avram Noam Chomsky

Порождающие грамматики

- Представлены в 1957 году на примере английского языка
 - Книга “Syntactic Structures”, Avram Noam Chomsky
- Изначально разработаны для естественных языков
 - Генеративная лингвистика (1960—1990-е годы)

Порождающие грамматики

- Представлены в 1957 году на примере английского языка
 - Книга “Syntactic Structures”, Avram Noam Chomsky
- Изначально разработаны для естественных языков
 - Генеративная лингвистика (1960—1990-е годы)
- Стали фундаментом формальных языки

Конструирование и разбор

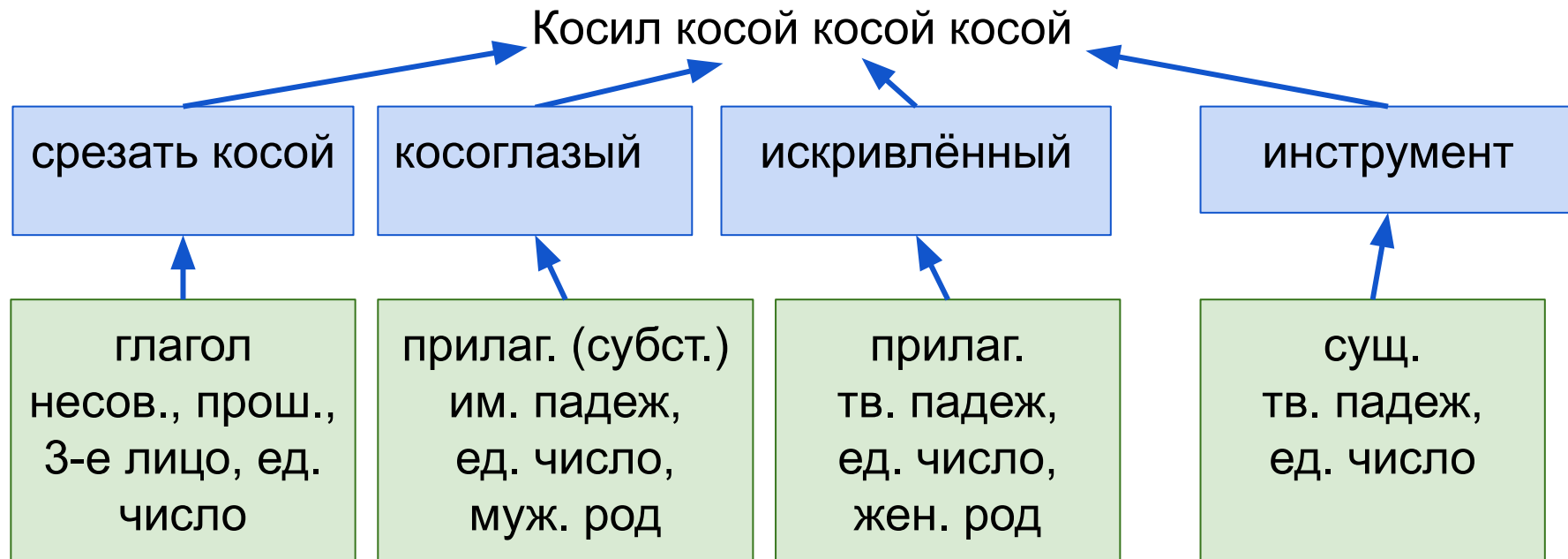
Синтаксический разбор предложения

Косил косой косой косой

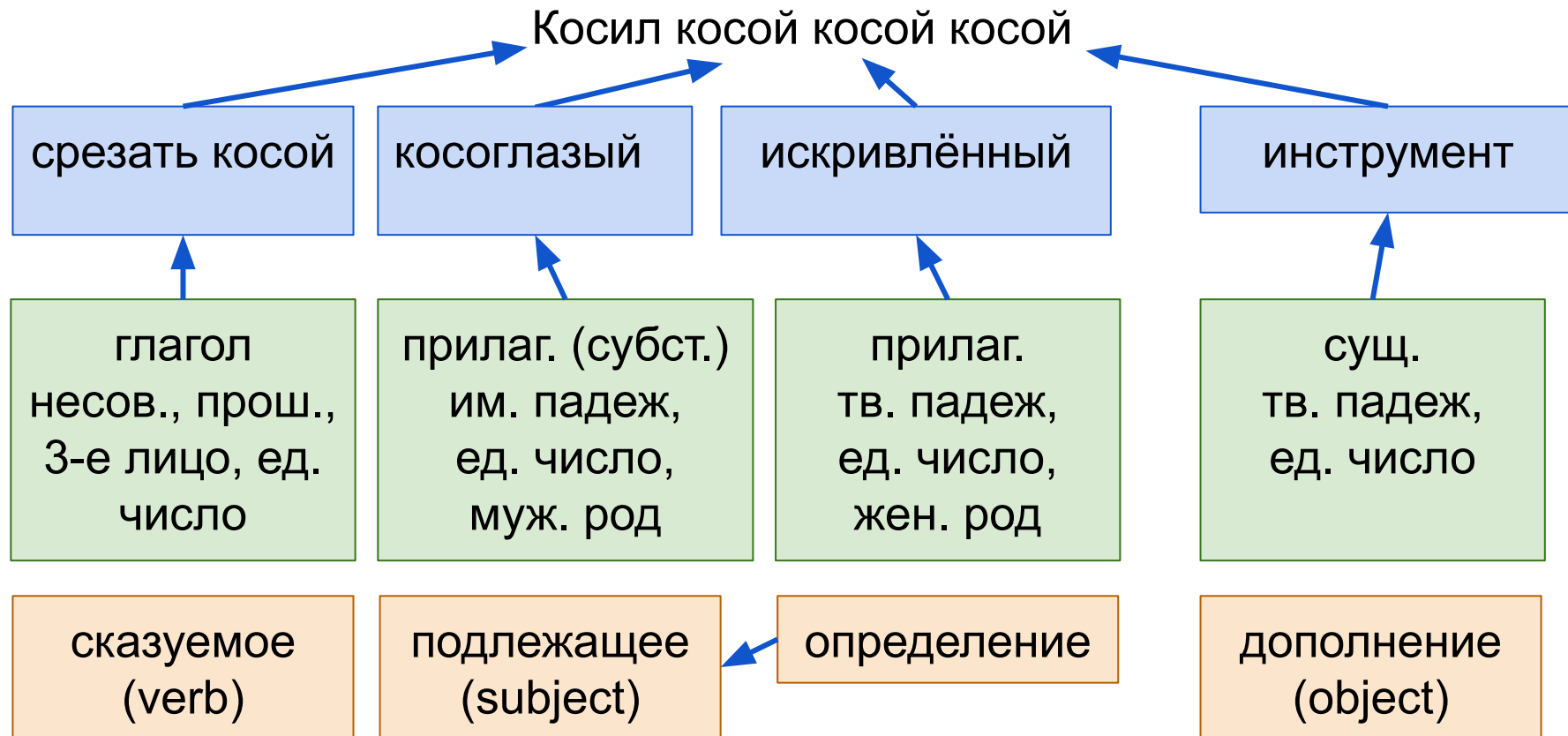
Синтаксический разбор предложения



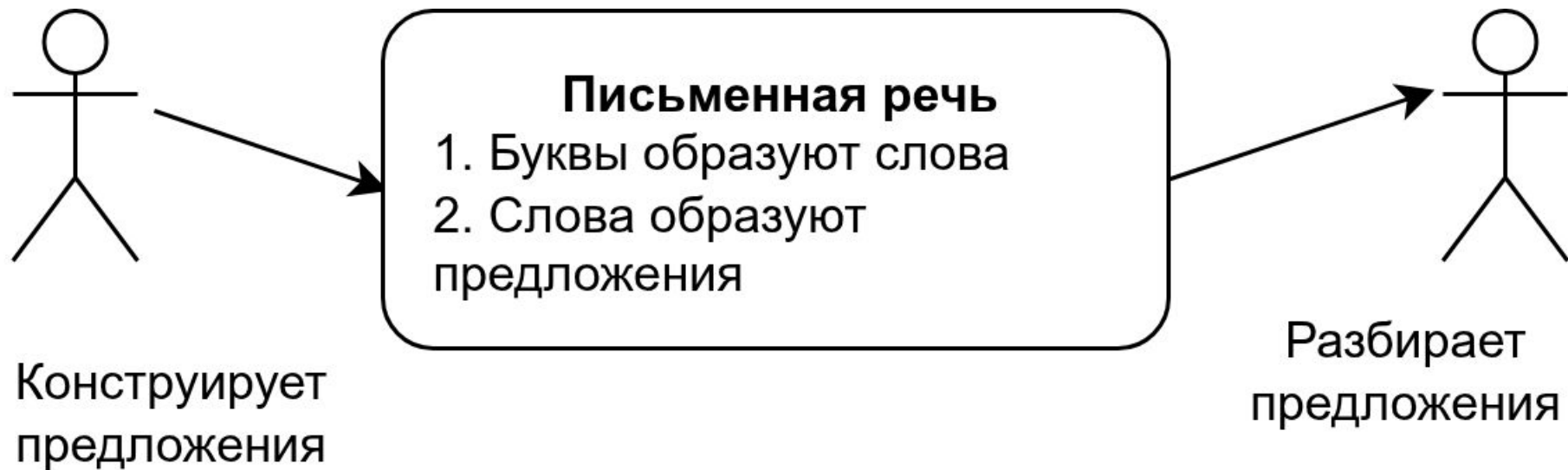
Синтаксический разбор предложения



Синтаксический разбор предложения



Конструирование и разбор



Простая грамматика

предложение = субъект, действие [, объект] ;

Простая грамматика

предложение = субъект, действие [, объект] ;

субъект = СУЩ. | МЕСТ. ;

действие = ГЛАГОЛ ;

объект = ПРИЛ., СУЩ. ;

Терминалы и нетерминалы

предложение = субъект, действие [, объект] ;

субъект = СУЩ. | МЕСТ. ;

действие = ГЛАГОЛ ;

объект = ПРИЛ. , СУЩ. ;

Терминалы (терминальные символы)	СУЩ., МЕСТ., ГЛАГОЛ, ПРИЛ.
Нетерминалы (нетерминальные символы)	предложение, субъект, действие, объект

1-й пример конструирования

предложение = субъект, действие [, объект] ;

субъект = СУЩ. | МЕСТ. ;

действие = ГЛАГОЛ ;

объект = ПРИЛ., СУЩ. ;

предложение

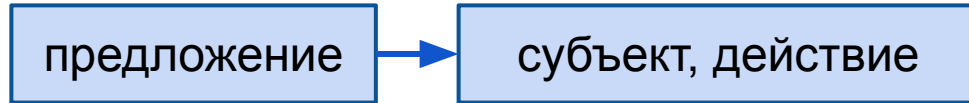
1-й пример конструирования

предложение = субъект, действие [, объект] ;

субъект = СУЩ. | МЕСТ. ;

действие = ГЛАГОЛ ;

объект = ПРИЛ., СУЩ. ;



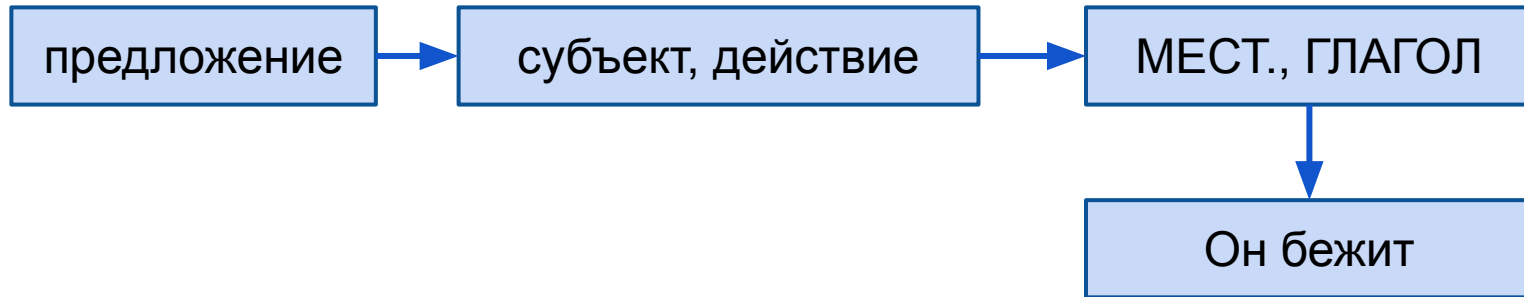
1-й пример конструирования

предложение = субъект, действие [, объект] ;

субъект = СУЩ. | МЕСТ. ;

действие = ГЛАГОЛ ;

объект = ПРИЛ., СУЩ. ;



2-й пример конструирования

предложение = субъект, действие [, объект] ;

субъект = СУЩ. | МЕСТ. ;

действие = ГЛАГОЛ ;

объект = ПРИЛ., СУЩ. ;

предложение

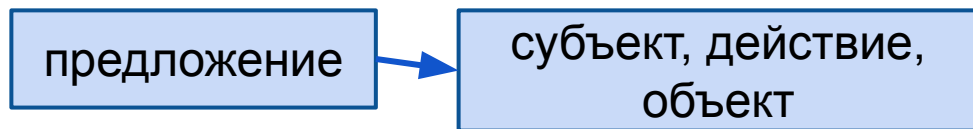
2-й пример конструирования

предложение = субъект, действие [, объект] ;

субъект = СУЩ. | МЕСТ. ;

действие = ГЛАГОЛ ;

объект = ПРИЛ., СУЩ. ;



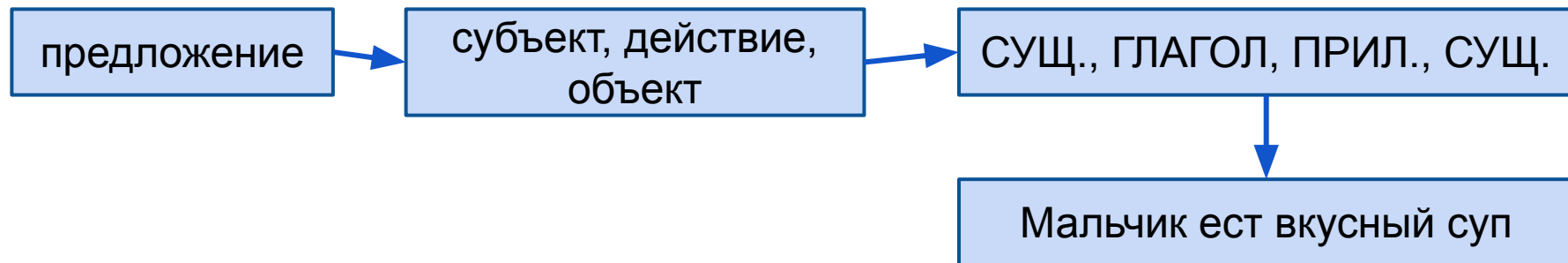
2-й пример конструирования

предложение = субъект, действие [, объект] ;

субъект = СУЩ. | МЕСТ. ;

действие = ГЛАГОЛ ;

объект = ПРИЛ., СУЩ. ;



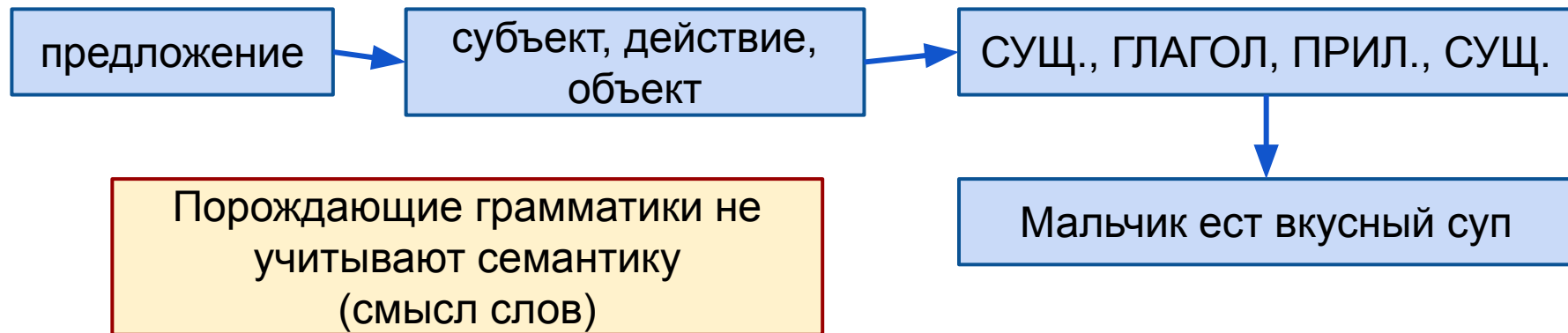
2-й пример конструирования

предложение = субъект, действие [, объект] ;

субъект = СУЩ. | МЕСТ. ;

действие = ГЛАГОЛ ;

объект = ПРИЛ., СУЩ. ;



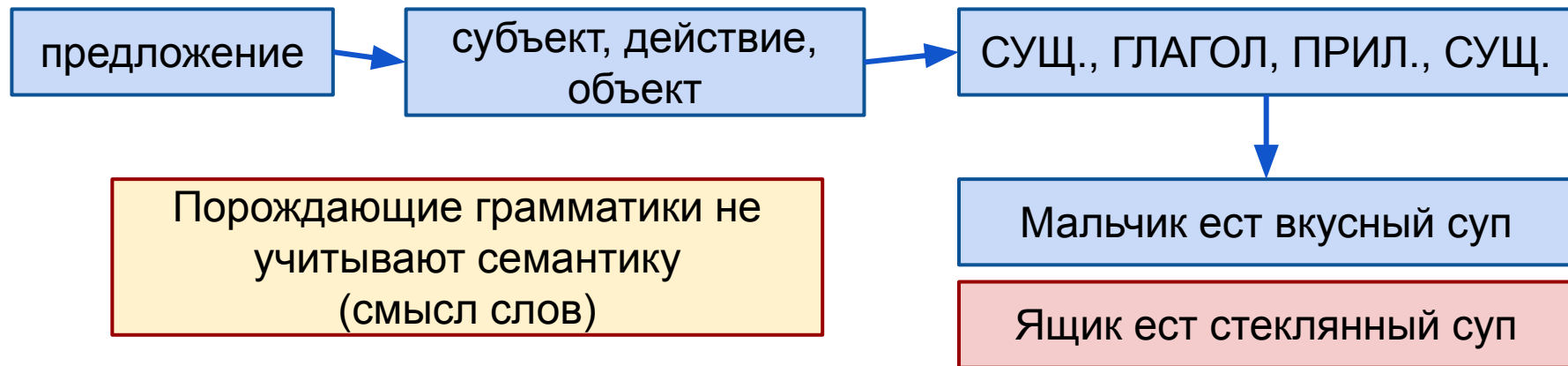
2-й пример конструирования

предложение = субъект, действие [, объект] ;

субъект = СУЩ. | МЕСТ. ;

действие = ГЛАГОЛ ;

объект = ПРИЛ., СУЩ. ;



Переход к абстракциям

предложение = субъект, действие [, объект] ;

субъект = СУЩ. | МЕСТ. ;

действие = ГЛАГОЛ ;

объект = ПРИЛ., СУЩ. ;

Переход к абстракциям

предложение = субъект, действие [, объект] ;

субъект = СУЩ. | МЕСТ. ;

действие = ГЛАГОЛ ;

объект = ПРИЛ., СУЩ. ;

sentence = subject, action, object ;

subject = noun | pronoun ;

action = verb ;

object = adjective, noun | ϵ ;

Переход к абстракциям

предложение = субъект, действие [, объект] ;

субъект = СУЩ. | МЕСТ. ;

действие = ГЛАГОЛ ;

объект = ПРИЛ., СУЩ. ;

sentence = subject, action, object ;

subject = noun | pronoun ;

action = verb ;

object = adjective, noun | ϵ ;

$A \rightarrow BCD$

$B \rightarrow a$ $B \rightarrow b$

$C \rightarrow c$

$D \rightarrow da$ $D \rightarrow \epsilon$

Регулярные грамматики

Виды регулярных грамматик

Левая регулярная грамматика

Допустимы 3 вида правил:

1. $A \rightarrow a$
2. $A \rightarrow aB$
3. $A \rightarrow \varepsilon$

Правая регулярная грамматика

Допустимы 3 вида правил:

1. $A \rightarrow a$
2. $A \rightarrow Ba$
3. $A \rightarrow \varepsilon$

** A и B могут совпадать*

Виды регулярных грамматик

Левая регулярная грамматика

Допустимы 3 вида правил:

1. $A \rightarrow a$
2. $A \rightarrow aB$
3. $A \rightarrow \varepsilon$

** A и B могут совпадать*

Правая регулярная грамматика

Допустимы 3 вида правил:

1. $A \rightarrow a$
2. $A \rightarrow Ba$
3. $A \rightarrow \varepsilon$

Вопрос: как выразить правило
 $A \rightarrow aaaB$
в регулярной грамматике?

Вопрос: это регулярная грамматика?

предложение = субъект, действие [, объект] ;

субъект = СУЩ. | МЕСТ. ;

действие = ГЛАГОЛ ;

объект = ПРИЛ., СУЩ. ;

$A \rightarrow BCD$

$B \rightarrow a$ $B \rightarrow b$

$C \rightarrow c$

$D \rightarrow da$ $D \rightarrow \varepsilon$

Разбор леворегулярной грамматики

Правила:

Входные строки:

Соответствие грамматике:

$A \rightarrow a$

1. ba

1. ???

$A \rightarrow b$

2. ab

2. ???

3. bbba

3. ???

$A \rightarrow bA$

Разбор леворегулярной грамматики

Правила:	Входные строки:	Соответствие грамматике:
$A \rightarrow a$	1. ba	1. Соответствует
$A \rightarrow b$	2. ab	2. ???
$A \rightarrow bA$	3. bbba	3. ???

$A \rightarrow bA \rightarrow ba$

$A \rightarrow bA$

$A \rightarrow a$

Разбор леворегулярной грамматики

Правила:

Входные строки:

Соответствие грамматике:

$A \rightarrow a$

1. ba

1. Соответствует

$A \rightarrow b$

2. ab

2. Не соответствует

3. bbba

3. ???

$A \rightarrow bA$

Разбор леворегулярной грамматики

Правила:

Входные строки:

Соответствие грамматике:

$A \rightarrow a$

1. ba

1. Соответствует

$A \rightarrow b$

2. ab

2. Не соответствует

3. bbba

3. Соответствует

$A \rightarrow bA$

$A \rightarrow bA \rightarrow bbA \rightarrow bbbA \rightarrow bbba$

$A \rightarrow bA$

$A \rightarrow bA$

$A \rightarrow bA$

$A \rightarrow a$

Преимущества регулярных грамматик

Правила:

$$A \rightarrow a$$
$$A \rightarrow b$$
$$A \rightarrow bA$$

Преимущества регулярных грамматик

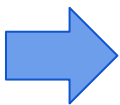
1. Эквивалентны регулярным выражениям

Правила:

$A \rightarrow a$

$A \rightarrow b$

$A \rightarrow bA$



Регулярное

выражение:

$b^*[a|b]$

Преимущества регулярных грамматик

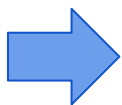
1. Эквивалентны регулярным выражениям
2. Эквивалентны конечному автомату

Правила:

$A \rightarrow a$

$A \rightarrow b$

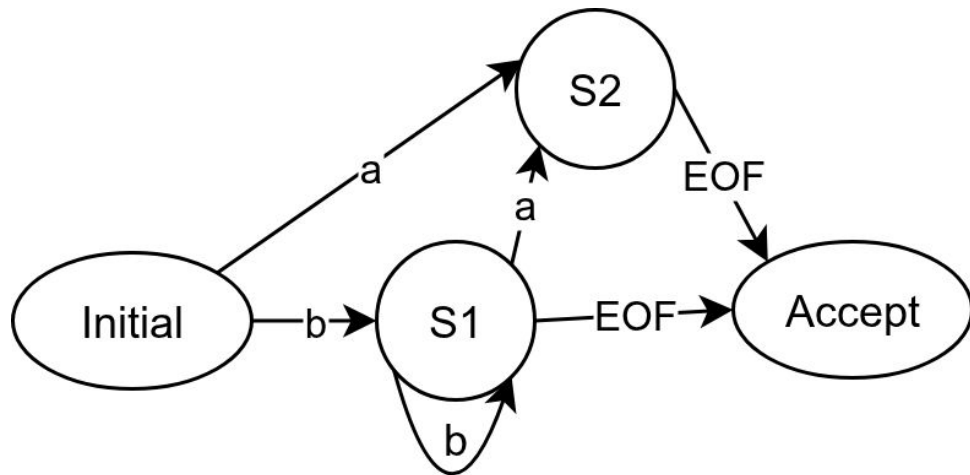
$A \rightarrow bA$



Регулярное

выражение:

$b^*[a|b]$



Преимущества регулярных грамматик

1. Эквивалентны регулярным выражениям
2. Эквивалентны конечному автомату
 - Следовательно, могут быть разобраны за один проход без предпросмотра (ДКА) или с предпросмотром (НКА)
 - Идеальны для лексического анализа

Правила:

$A \rightarrow a$



Регулярное

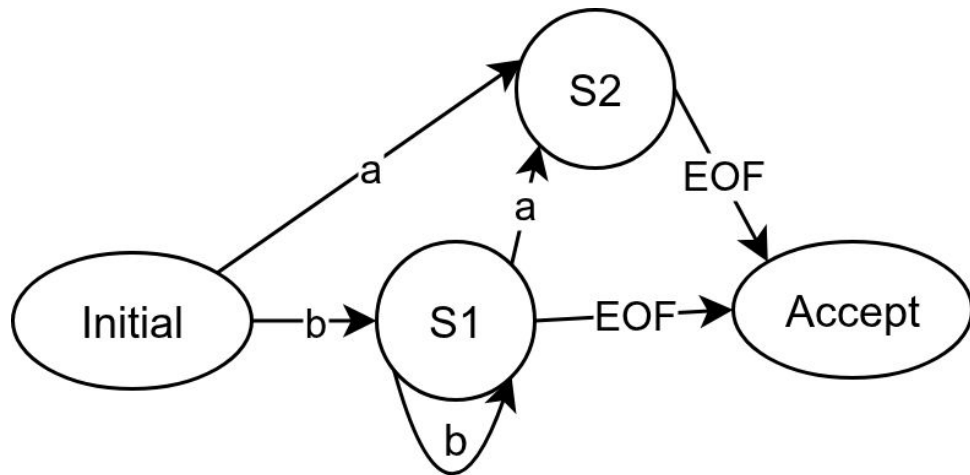
выражение:

$b^*[a|b]$



$A \rightarrow b$

$A \rightarrow bA$



Ограничения регулярных грамматик

Не могут выражать иерархические правила:

1. Сбалансированные скобки: $()$, $(())$, $((()))$, ...
2. Закрывающие/открывающие теги: `<div><span></span></div>`
3. Операторы с приоритетами: $17 + a * b / 2$
4. Выражения со скобками: $(17 + a) * b / 2$

Конечный автомат не «запоминает» пройденную часть иерархии

Контекстно-свободные грамматики (КС-грамматики)

Контекстно-свободная грамматика

Состоит из правил вида

$$A \rightarrow \alpha$$

Где:

- A — нетерминал
- α — конечная цепочка терминалов и нетерминалов
 - может быть пустой

Грамматика сбалансированных скобок

EBNF:

$\text{expr} = "(" , \text{expr} , ")" ;$

$\text{expr} = ;$

Формула:

$A \rightarrow aAb$

$A \rightarrow \varepsilon$

Грамматика арифметических выражений

EBNF с левой рекурсией:

```
expr = expr, [ ("+" | "-"), term];
```

```
term = term, [ ("*" | "/"), factor];
```

```
factor = identifier | number
```

Грамматика арифметических выражений со скобками

EBNF с левой рекурсией:

```
expr = expr, [ ("+" | "-"), term];
```

```
term = term, [ ("*" | "/"), factor];
```

```
factor = "(", expr, ")" | identifier | number
```

Преимущества КС грамматик

1. Могут выражать рекурсивные правила
 - Подходят для разбора формальных языков

Преимущества КС грамматик

1. Могут выражать рекурсивные правила
 - Подходят для разбора формальных языков
2. Могут быть разобраны с помощью рекурсии или стека
 - Автомат с магазинной памятью

Контекстно-зависимые грамматики

Контекстно-зависимая грамматика

Состоит из правил вида

$$\alpha A \beta \rightarrow \alpha \omega \beta$$

Где:

- A — нетерминал
- ω — конечная непустая цепочка терминалов и нетерминалов
- α, β — конечные цепочки терминалов и нетерминалов
 - могут быть пустыми

Язык $a^n b^n c^n$

Допустимые предложения: abc , $aabbcc$, $aaabbbccc$, ...

Грамматика:

$S \rightarrow a B C$

$b B \rightarrow b b$

$S \rightarrow a S B C$

$b C \rightarrow b c$

$C B \rightarrow B C$

$c C \rightarrow c c$

$a B \rightarrow a b$

Иерархия грамматик

Иерархия порождающих грамматик Хомского

- Тип 0 — неограниченные
- Тип 1 — контекстно-зависимые (неукорачивающие)
- Тип 2 — контекстно-свободные
- Тип 3 — регулярные

Иерархия порождающих грамматик Хомского

Тип	Способ вычисления	Применение в компиляторах
Неограниченные	Нормальный алгоритм Маркова (Машина Тьюринга)	—
Контекстно-зависимые	Линейно ограниченный автомат	—
Контекстно-свободные	Автомат со стеком	Синтаксический разбор
Регулярные	Конечный автомат	Лексический разбор

Ссылки

1. [Простым языком о языковых моделях и цепи Маркова \(Markov Chain\)](#)
2. [Лемма о разрастании \(лемма о накачке\)](#)

Конец!
Вопросы?