



Автоматы и формальные языки

Карпов Юрий Глебович
профессор, д.т.н., зав.кафедрой
“Распределенные вычисления и компьютерные сети”
Санкт-Петербургского Политехнического университета
karpov@dcn.infos.ru

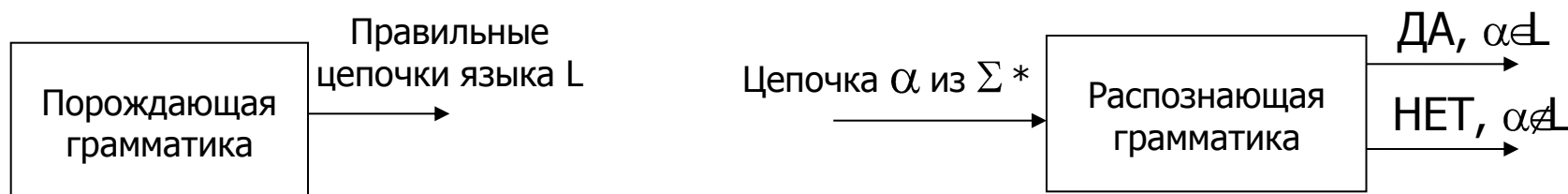


Структура курса

- Конечные автоматы-распознаватели – 4 л
- Порождающие грамматики Хомского – 3 л
 - Лекция 5. Синтаксически-ориентированная трансляция и грамматики Хомского
 - Лекция 6. Иерархия грамматик Хомского
 - Лекция 7. Абстрактные распознающие автоматы
- Атрибутные трансляции и двусмысленные КС-грамматики – 2 л
- Распознаватели КС-языков и трансляция – 6 л
- Дополнительные лекции - 2 л

Порождающая и распознающая грамматики

Грамматика – конечное формальное описание языка



- Существует два типа грамматик: порождающие и распознающие
 - *Порождающая грамматика* языка L - это конечный набор правил, позволяющих строить все "*правильные*" предложения языка L , и применение которых не дает ни одного "*неправильного*" предложения, не принадлежащего L
 - *Распознающая грамматика* задает критерий принадлежности произвольной цепочки данному языку. Это, фактически, алгоритм, принимающий в качестве входа символ за символом произвольную цепочку над словарем V и дающий на выходе один из двух возможных ответов: "*данная цепочка принадлежит языку L* " либо "*данная цепочка **не** принадлежит языку L* "

Конечные автоматы недостаточны

- Конечные автоматы – удобный формализм распознавания языков, но **только очень узкий класс языков являются автоматными**
- Примеры неавтоматных языков:
 - $\Sigma = \{a, b, c\}$ $L = \{a^n b c^n \mid n \geq 0\}$
 $aabcc \in L$ $cbaa \notin L$
 - $\Sigma_5 = \{a, b, c\}$ $L_5 = \{w c w^R \mid w \in \Sigma^*\}$ палиндромы (лёшанаполкЕклопанашёл)
 $abaabcbaaba \in L_5$ $cbaa \notin L_5$
 - $\Sigma_6 = \{a, b, c\}$ $L_6 = \{\alpha \in \Sigma_6^* \mid \text{в } \alpha \text{ количества вхождений } a, b \text{ и } c \text{ равны}\}$ $cscbaba \in L_6$,
 $cscb \notin L_6$
 - $\Sigma_9 = \{ '(', ') '\}$ $L_9 = \text{множество правильных скобочных выражений}$
 $(())() \in L_9$ $))(() \notin L_9$
 - $\Sigma_{12} = \{a\}$ $L_{12} = \text{цепочки из } a \text{ длиной } 1, 4, 8, 16, \dots, n^2$
 $aaaa \in L_{12}$ $a \in L_{12}$, $aaa \notin L_{12}$

Для распознавания произвольных цепочек этих языков конечной фиксированной памяти конечного автомата недостаточно!!

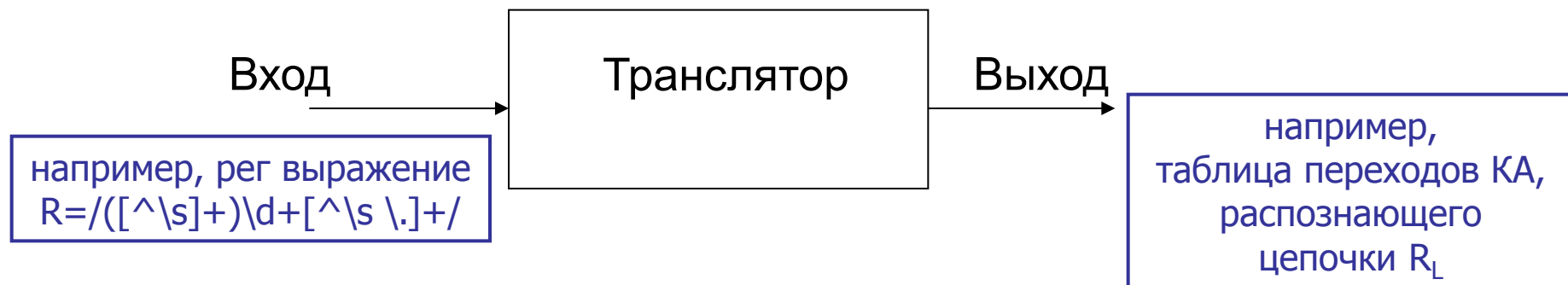


Нужна другая теория

Цель курса: КАК ПОСТРОИТЬ ТРАНСЛЯТОР?

НАША ЦЕЛЬ: Изучение

теоретических основ,
алгоритмов,
практических методов построения трансляторов



- Описание языка грамматикой – только первый шаг.
Более важным является проблема “понимания” “**смысла**” входной цепочки языка, чтобы осуществить перевод этой цепочки в некоторый нужный выход, выражающий этот смысл

Для выявления “смысла” входной цепочки языка необходимы новые теории, формальные модели и методы



Как выявить “смысл” входной цепочки?



- Словарь языка конечен, смыслы слов фиксированы
 - Как выявить смысл цепочки языка, имея смыслы всех слов в цепочке?
 - Нельзя по набору слов предложения определить его смысл. **Смысл предложения определяется не набором слов, а именно их порядком!**
 - Предложений бесконечное число, нельзя для всех предложений языка заранее определить смыслы (например, таблицей)
 - Значение, “смысл” каждого предложения **нужно вычислять**. КАК?
-
- Может ли для всех (или для большей части практических) языков существовать **единый систематический метод** преобразования входных цепочек символов в нечто, отражающее СМЫСЛЫ этих цепочек?
-
- Такой метод существует, он называется **“Метод синтаксически-ориентированной трансляции”**



Как вычислять “смысл” предложений языка?

- Ноам Хомский предложил :
рассматривать ДВУСМЫСЛЕННЫЕ предложения, чтобы понять, как человек приписывает смысл таким предложениям

Порядок сменит хаос

Смысл зависит от того, какое слово считаем ПОДЛЕЖАЩИМ

“Водитель “Ауди” выжила после того, как ее автомобиль раздавил КАМАЗ”

Смысл зависит от того, какую РОЛЬ играет каждое слово

- **Идея Хомского следующая:**
 - Человек понимает смысл ДВУСМЫСЛЕННОГО предложения, приписывая ту или иную **структуру** предложению
 - Понимание смысла однозначному предложению, в соответствии с идеей Хомского, производится также на основе **структуры** предложения



Как работает переводчик естественного языка

- Переводчик **не переводит пословно**. Перевод осуществляется в две стадии:
 - Анализ предложения входного языка:
 - выявляется структура фразы, т.е. грамматические категории (группа подлежащего, группа сказуемого, дополнение, ...) и связи между грамматическими категориями
 - Синтез (генерация выхода)
 - определяется значение каждого слова (слов КОНЕЧНОЕ ЧИСЛО)
 - по структуре каждой грамматической категории и значениям слов, его составляющих, устанавливается смысл этой грамматической категории
 - по структуре предложения и смыслам грамматических конструкций устанавливается мысль, "смысл", "значение" всего предложения
 - "значение" предложения формируется в виде внутренней структуры (в голове переводчика)
 - на основании внутреннего представления "значений" слов и грамматических категорий строится РЕЗУЛЬТАТ - предложение выходного языка
- Для построения алгоритмов трансляции ВСЕ эти понятия нужно формализовать (язык, структура, грамматические категории, семантика , ...)

Значение предложения не есть сумма значений составляющих его слов

Казнить нельзя помиловать – КАК ПОНИМАТЬ?

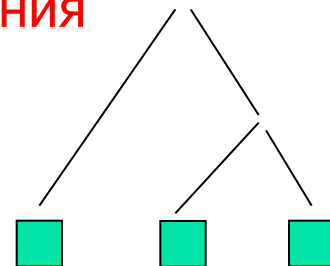
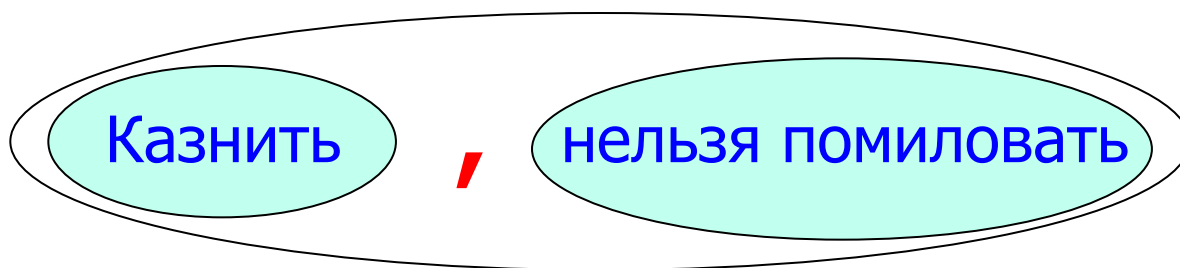
Казнить / нельзя помиловать
ИЛИ

(нужно казнить)

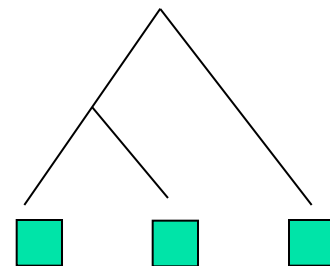
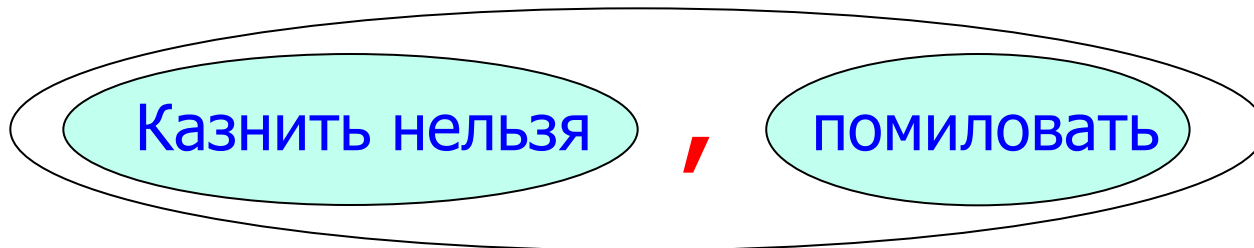
Казнить нельзя / помиловать

(нужно помиловать)

Однозначный смысл фразе придает запятая: два предложения



Более тесную связь удобно показать деревом связей



Двусмысленные предложения

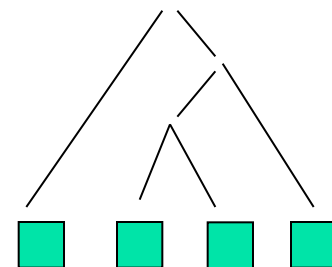
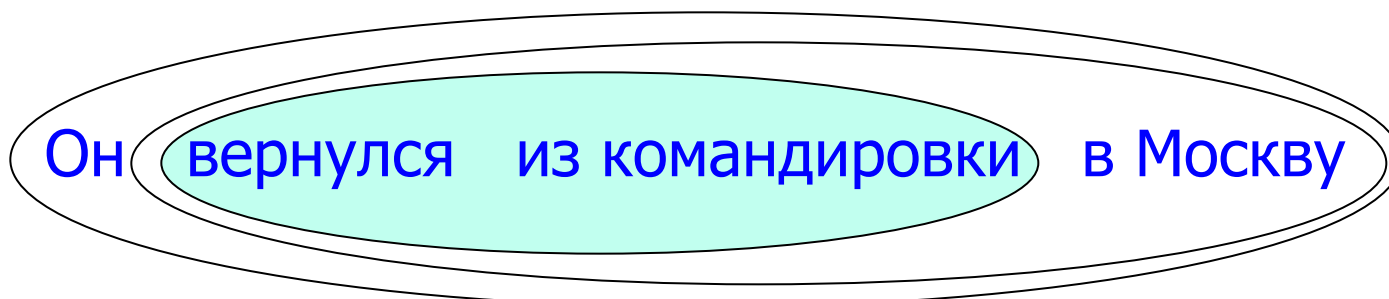
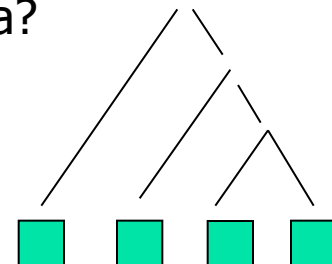
Он вернулся из командировки в Москву – **КАК ПОНИМАТЬ?**

Два смысла: *Он был в командировке в Москве* (и вернулся из Москвы)

или

Он вернулся из командировки (и приехал в Москву)

Как связать смысл с этой фразой? Наш мозг переключается с одной интерпретации на другую, и обе правильные. Почему два смысла?

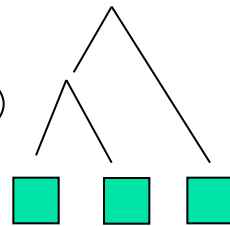


Теория Н.Хомского (Noam Chomsky) – на основе анализа двусмысленных предложений

Человек понимает смысл предложения естественного языка на основе анализа структуры текста и роли структурных конструкций в предложении

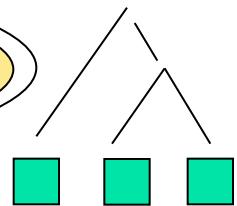
Куплю коляску для новорожденного синего цвета

коляска синяя



Куплю коляску для новорожденного синего цвета

новорожденный синий



Два смысла потому, что две структуры



Примеры двусмысленных предложений

- Петерстар поглотил Мегафон
 - кто кого поглотил?
- Ножницы для стрижки волос 20 см
 - другие волосы стричь нельзя?
- Облицовка кафелем заказчика
 - бедный заказчик!
- Дети до пяти лет проходят в цирк на руках
 - дети-акробаты
- Бытие определяет сознание
 - Что определяется чем? основной постулат марксизма – двусмысленен!
- Я встретил ее на поляне с цветами
 - кто был с цветами? Я, она **или** поляна?
- И вскрикнул внезапно ужаленный князь
 - *вскрикнул внезапно, **или** был ужален внезапно?*
- I made her duck
 - *я приготовил для нее утку, **или** я приготовил утку, принадлежащую ей **или** я превратил ее в утку*
- I have drawn butter
 - *я намазан маслом **или** у меня есть намазанное масло?*
- Formal property verification
 - *verification of formal properties **or** formal methods for property verification*



Лента новостей только за один день

Двусмысленные предложения – везде!!

- “Путин рассказал о приоритетах России на саммите G20”
 - Рассказал о приоритетах, или рассказал на саммите? За полгода до саммита!
- “Митинг в поддержку фигурантов “Болотного дела” в Петербурге собрал 200 человек”
 - Болотное дело – не в Петербурге!
- “... буква закона позволяет рабочим из соседних республик платить меньше”
 - Рабочие будут платить меньше (например, взятки), или они будут получать меньше (меньшую зарплату)?
- “Избиратели Айовы предпочитают Джошуа Маккаби Хиллари Клинтон”
 - Кого кому предпочитают?
- “Автоинспектора премируют за отказ от взятки в сумме 1 млн рублей”
 - Взятка в 1 млн рублей или премия в 1 млн рублей
- “Фигурант «болотного дела» пожаловался на избиение в суде”
 - 01.10.2013 – lenta.ru
- “В Руанде бхуту убивают тутси”
 - Кто кого убивает?

Примеры двусмысленных сообщений СМИ

- Обама пообещал давать по \$100 млрд развивающимся странам в год на борьбу с выбросами по \$100 млрд каждой стране, или по \$100 млрд в год?
- Плющенко будет готовиться к олимпиаде в Сочи
будет готовиться в Сочи или будет готовиться к Олимпиаде, которая пройдет в Сочи?
- Патаркацишвили отравил лично Саакашвили, подложив в пищу яд
кто кого отравил? Поскольку Саакашвили живой, то это он подложил?
- А.Проханов: "В сталинские годы консерватизм заменил собой авангард"
что было заменено чем?
- На Московском шоссе фура упала на иномарку, погиб ее водитель погиб
водитель фуры или водитель иномарки?
- Дорожные фонды наполнят акцизы на бензин кто кого наполнит?
- "Я мечтаю опять поехать в Париж" мечтаю опять, или опять поехать?
- 01.02.2014. Из новостей: "*Первый вертолетоносец типа «Мистраль» будет оснащать вооружением Кронштадтский морской завод*".
Кто кого будет оснащать?
- "*Привет освободителям Харькова от немецко-фашистских захватчиков*"
Кому привет от кого??

КАК ПОНЯТЬ СМЫСЛ ПРЕДЛОЖЕНИЙ? Их два (или больше)

Как человек понимает предложение языка?

- Хомский сосредоточился на центральном факте естественных языков:
любой человек может понимать и воспроизводить потенциально бесконечное множество предложений на родном языке

Его выводы:

1. знание о синтаксисе родного языка должно состоять из **конечного описания бесконечного множества предложений**, т.е. существует какая-то бессознательная грамматика в нашей голове. Эта грамматика состоит из правил, которые **порождают** все предложения языка
2. Предложение должно иметь **структуру**, которая позволяет “вычислить” смысл предложения по смыслам слов и конструкций составляющих предложение
(**не только для двусмысленных, но и для однозначных предложений**).

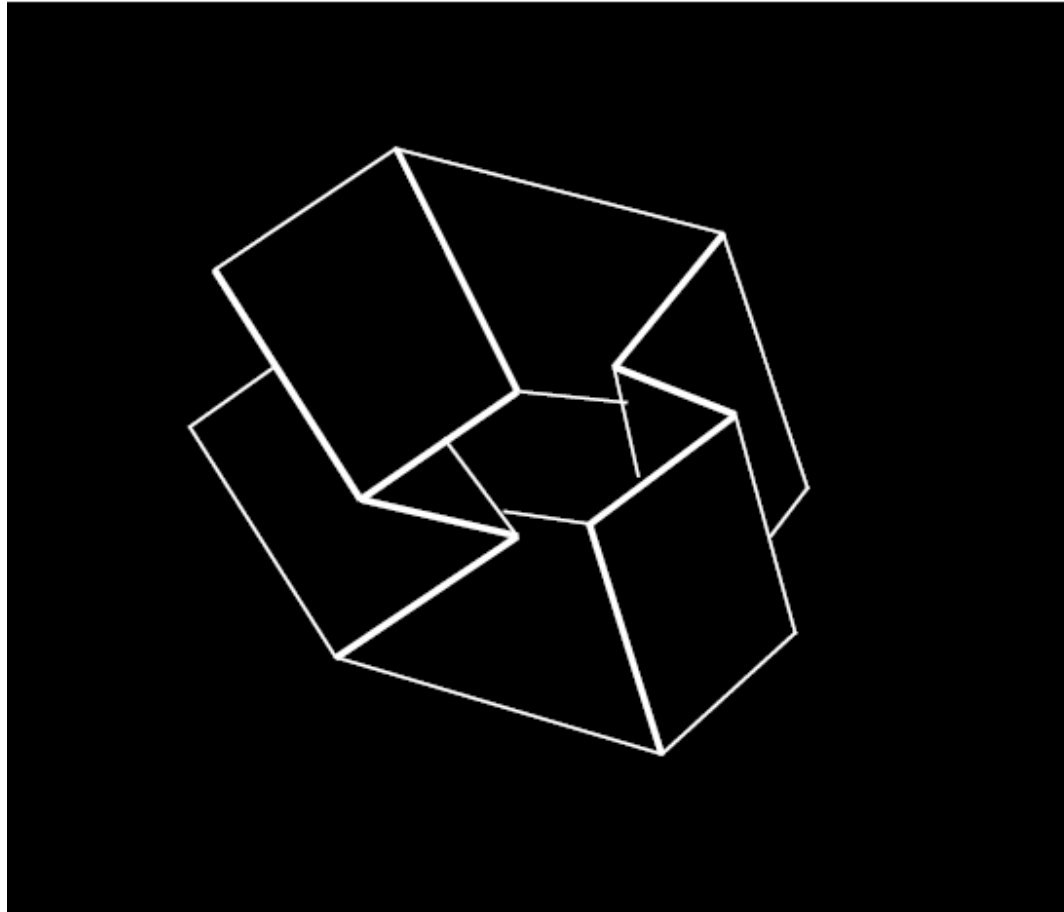


Примеры неоднозначных образов



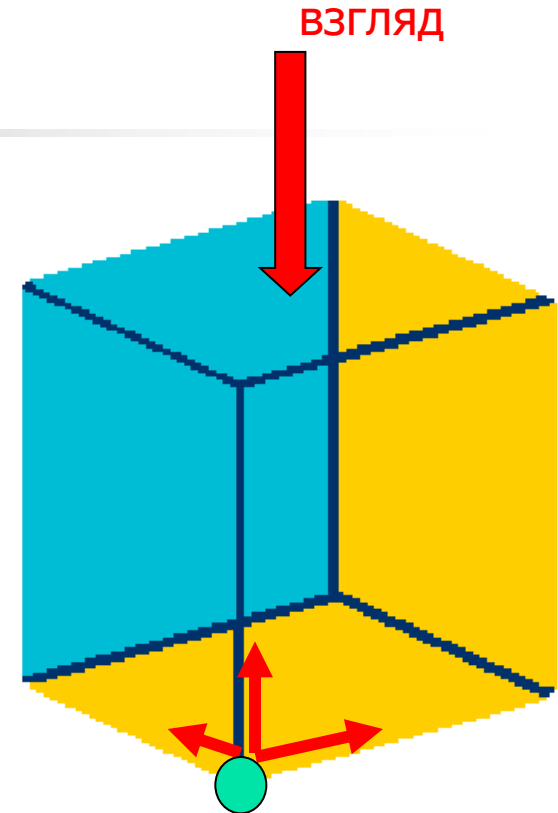
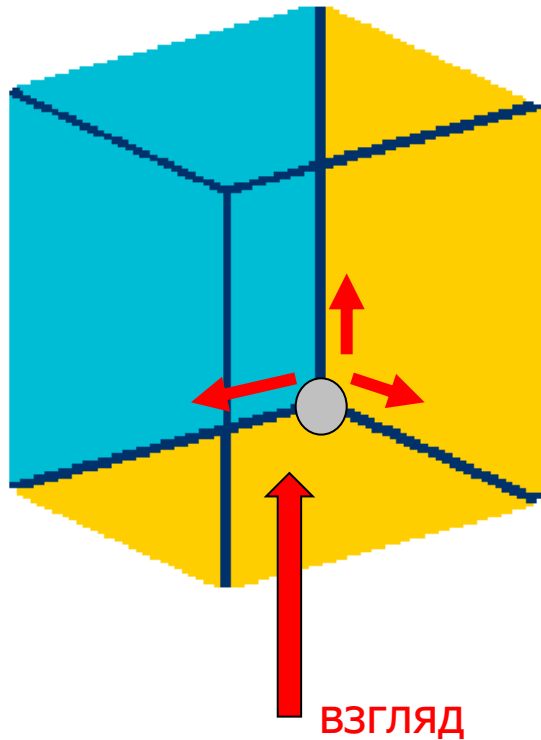
Рыбы или птицы? Гравюра Мориса Эшера

Примеры неоднозначных образов



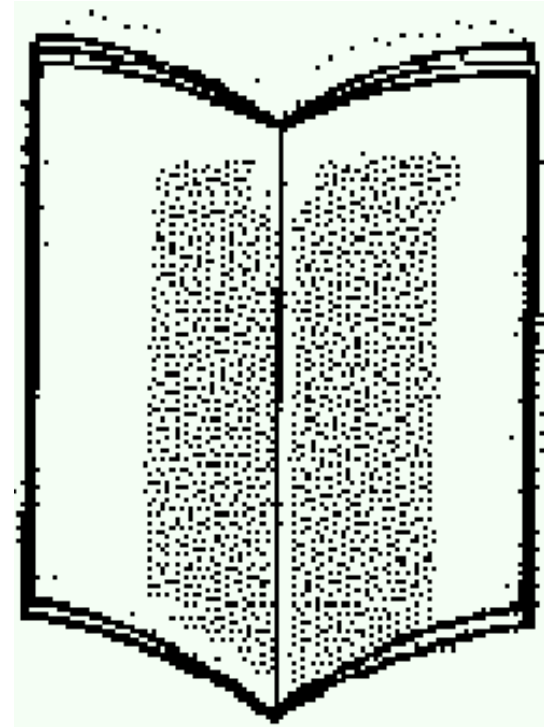
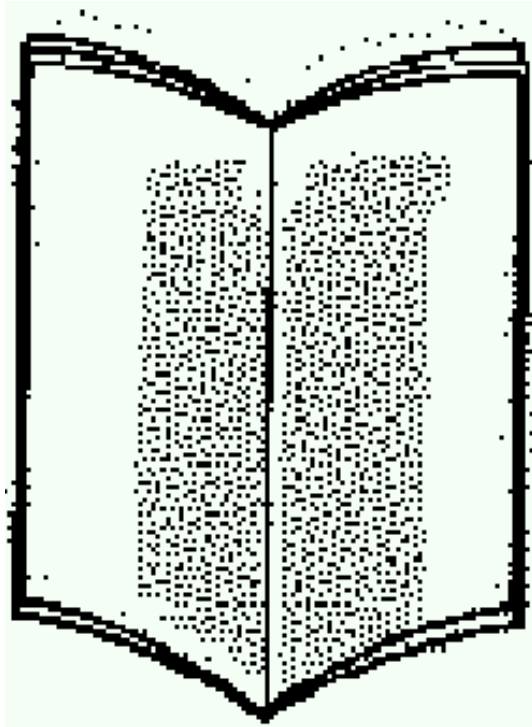
Морис Эшер: [фигура с несколькими интерпретациями](#)

Примеры неоднозначных образов



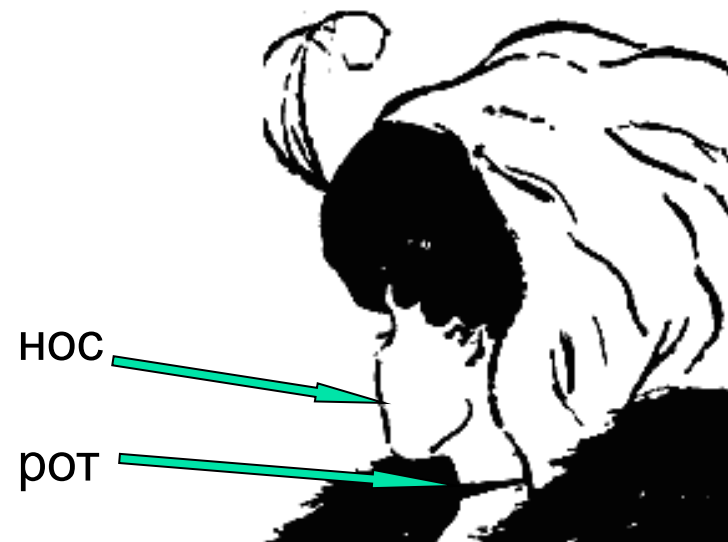
- Куб Неккера. При наблюдении фигура спонтанно «переворачивается»: одна объемная проекция сменяется другой. Два равноправных **“правильных”** решения перцептивной проблемы: что есть данный объект? Мозг «пробует» каждую из этих гипотез поочередно, не останавливаясь окончательно ни на одной из них

Примеры неоднозначных образов



- Фигура Маха похожа на корешок книги, обращенной к нам то страницами, то обложкой

Примеры неоднозначных образов



- Картина американского психолога Э. Дж. Боринга «Неоднозначная теща». Воспринимается то как портрет прелестной молодой девушки, то как лицо ужасной старухи, причем когда воспринимается один объект, совершенно «исчезает» другой.

Не имея данных, на основе которых можно сделать однозначный выбор смыслов, мозг "перескакивает" от одной модели к другой!

Для фиксации конкретного смысла всего образа нужно связать фрагменты рисунка с абстрактными "конструкциями": нос, подбородок, ...

Какое отношение это все имеет к нам?

- Существует два условных оператора:

- полный

if <условие> then <оператор> else <оператор>

- сокращенный

if <условие> then <оператор>

- Пусть $x=y=0$. Чему будет равно y после выполнения оператора

if $x>0$ then if $y>0$ then $x:=1$ else $y:=1$?

- Если весь оператор сокращенный (а полный – внутри): $x=y=0$

if $x>0$ then [if $y>0$ then $x:=1$ else $y:=1$], y остается 0

- Если этот оператор полный (а сокращенный – внутри):

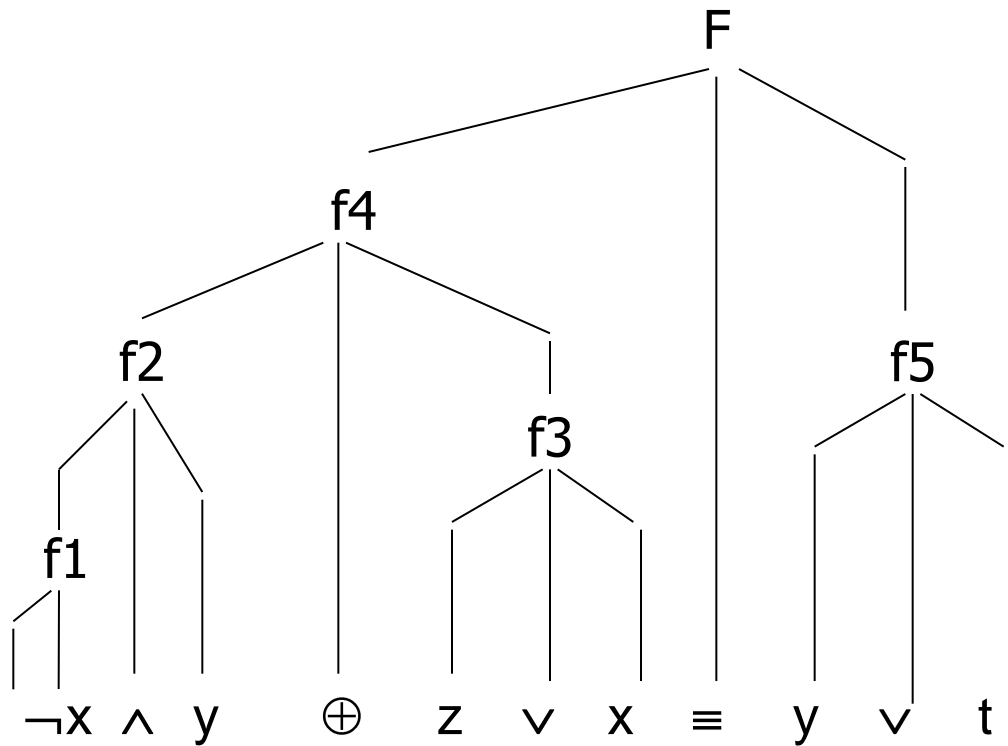
if $x>0$ then [if $y>0$ then $x:=1$] else $y:=1$, y станет 1

Если структура цепочки построена, то с ней можно связать смысл

Разные структуры □ разные смыслы

Порядок вычисления функции определяется структурой формулы (смысл связан со структурой)

$\neg x \wedge y \oplus z \vee x \equiv y \vee t$ - Как понимать?



$f1 = \neg x$
 $f2 = f1 \wedge y$
 $f3 = z \vee x$
 $f4 = f2 \oplus f3$
 $f5 = y \vee t$
 $F = f4 \equiv f5$

Приоритеты:

\neg НЕ
 \wedge И
 \vee ИЛИ
 $\Rightarrow \oplus \equiv, \dots$ Все остальные

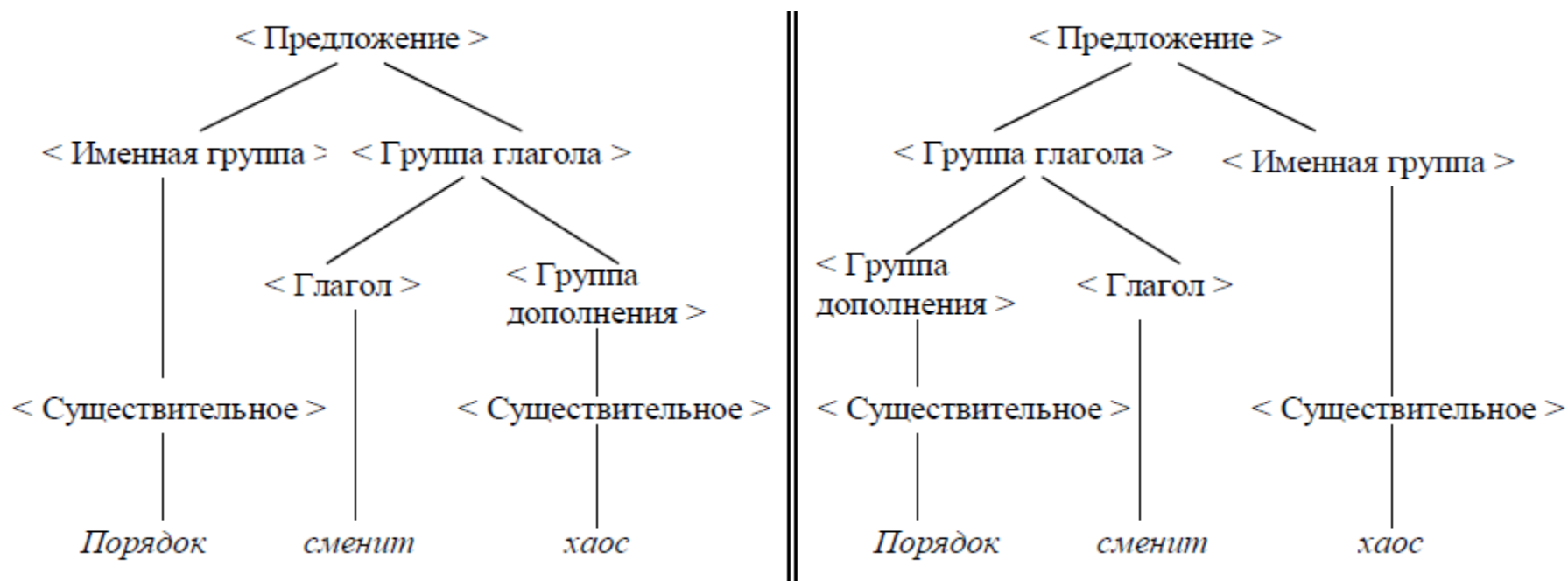
$$F = (((\neg x) \wedge y) \oplus (z \vee x)) \equiv (y \vee t)$$

скобки определяют структуру явно

Всегда вычисляем последовательно по структуре, по дереву

Различные структуры двусмысленного предложения

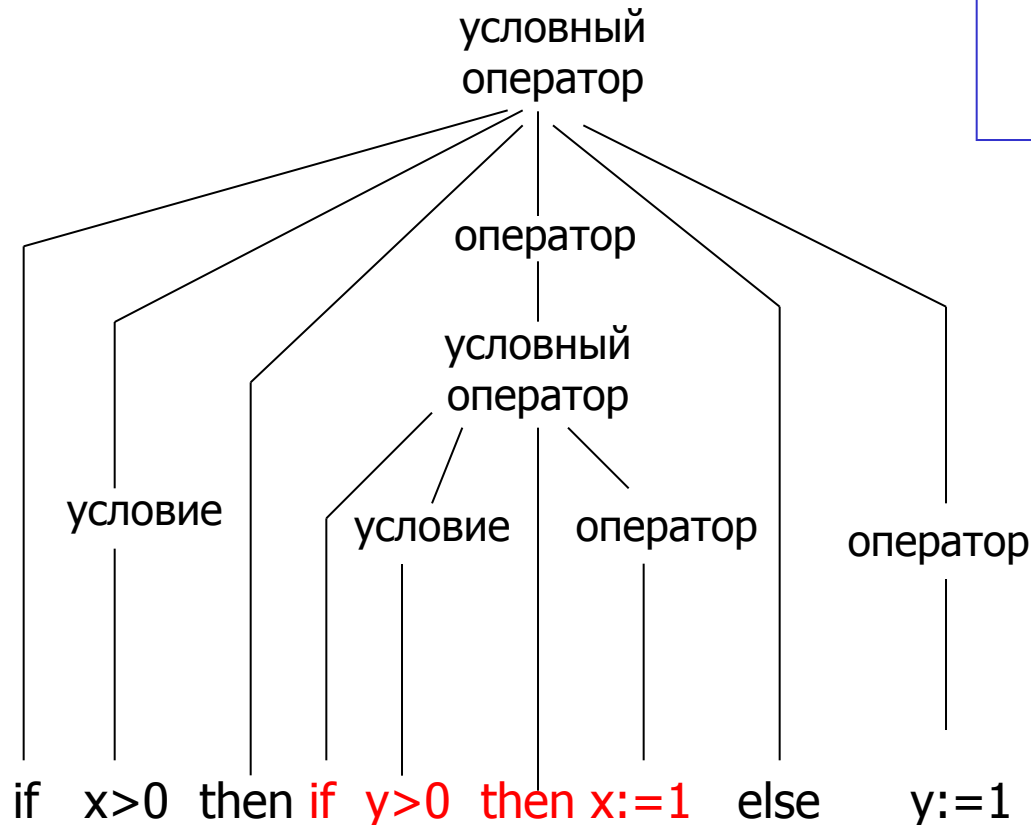
В русском языке возможно, чтобы сначала шла группа глагола, а потом группа подлежащего ("*ехал вдоль реки казак молодой*")



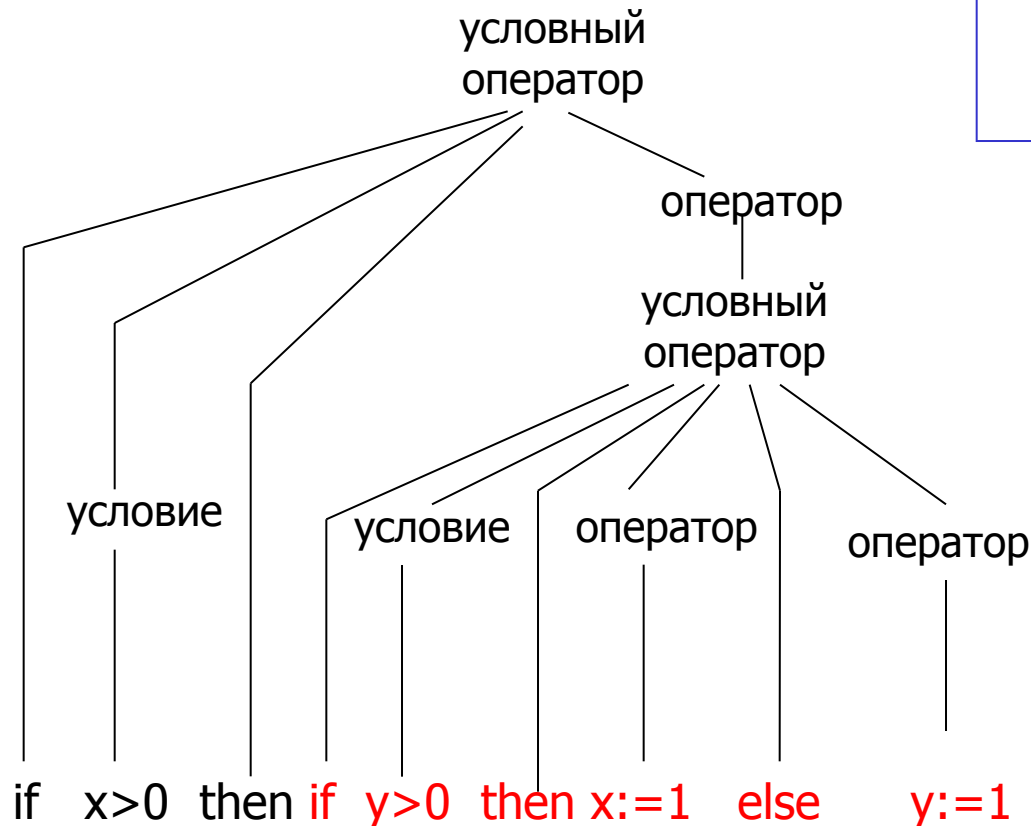
Идея Хомского: основой для понимания смысла предложения является его грамматическая структура, определяющая роль каждого слова и отдельных групп слов в предложении. Рисунок показывает, что **слова *порядок* и *хаос* могут играть в предложении различную роль** – либо роль подлежащего (активного объекта), либо дополнения (объекта, на которое направлено действие).

У предложения ДВЕ структуры, поэтому оно может пониматься двояко

при $x=0$ и $y=0$
у становится 1



Как проанализировать условный оператор



при $x=0$ и $y=0$
 y остается 0

- Проблема возникает, когда компилятор понимает одним образом, а программист - другим

Синтаксически-ориентированный подход

- Ноам Хомский: "Как ребенок понимает предложения, которые он раньше никогда не слышал?"
- *"Звук и значение в слове никак не связаны между собой"* – Л.С.Выготский. Смыслы слов нужно заучить. Смыслы предложений определяются структурой
- Структура используется для понимания ЛЮБЫХ предложений
- Мозг имеет врожденную способность вычленять структуру в предложении

По Хомскому, человеческий мозг разбивает процесс понимания предложения на два шага. На первом шаге производится построение структуры входного предложения; на втором эта структура используется для "вычисления" смысла



На основе разработанной Н.Хомским структурной теории языков была построена теория формальных языков и грамматик и разработаны современные алгоритмы синтаксического анализа и трансляции языков программирования



Идея синтаксически-ориентированной трансляции

- В соответствии с идеей Хомского, характеристикой предложения, однозначно связывающей смысл с цепочкой слов предложения, является структура предложения, т.е. объединения слов в подструктуры и роли, которые приписываются этим подструктурам
- Каждая пара – (*предложение, структура*) однозначно определяет смысл предложения. Если структур две или больше, то обычно и смыслов у одного и того же предложения два или больше
- Хомский предположил, что структура предложения естественного языка важна не только в том случае, когда человек пытается понять двусмысленное предложение, структура важна и при понимании смысла однозначного предложения
- Time flies like arrow – Время летит, как стрела (flies – глагол)
- Time flies like arrow – Временные мухи любят стрелу (flies– существительное)

ОСНОВНАЯ структура транслятора

Структура одна не определяет смысл предложения: предложения

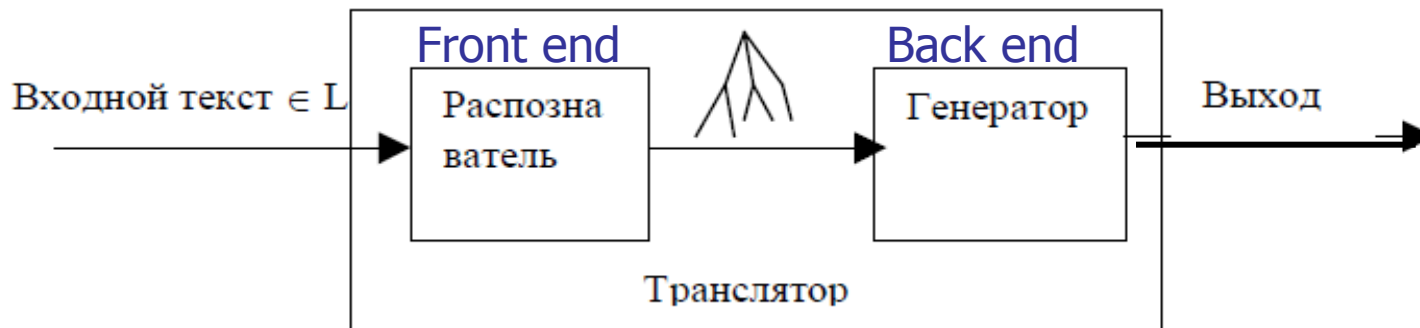
"Маша велела Ивану уйти" и

"Маша обещала Ивану уйти" –

не различаются по поверхностной структуре. Но в первом - Иван должен уйти, а во втором Маша должна уйти. Поэтому нужен и семантический анализ на основе структуры



- В соответствии с идеей синтаксически-ориентированной трансляции, процесс трансляции в информатике связывается с двумя основными этапами:
 - На первом этапе блок, который можно назвать распознавателем, строит структуру входной цепочки
 - На втором этапе построенная структура используется для семантического анализа и генерации выхода, выражающего смысл входной цепочки



Во многих трансляторах процессы распознавания и генерации разделены не так явно. Но во всех случаях метод синтаксически-ориентированной трансляции основан на том, что целиком или по частям строится структура входной цепочки



Как связать структуру с предложением?

Грамматики Хомского

Модель Хомского: порождающие грамматики

- В 1956 г. Ноам Хомский предложил модель порождающей грамматики, которая оказалась весьма удобной для задания искусственных языков. Одно из удобств этой модели в том, что каждой порождаемой цепочке языка эта модель позволяет сопоставить ее структуру
- **Определение.** Порождающая грамматика Хомского: $G=(T,N,S,R)$, где:
 - T – конечное множество символов (терминальный словарь)
 - N – конечное множество символов (нетерминальный словарь)
 - $S \in N$ – начальный нетерминал
 - R – конечное множество правил вида $\alpha \rightarrow \beta$, где α и β – цепочки над словарем $T \cup N$.

Пример: $G_0=(\{a, b, c\}, \{S, A, B\}, S, R)$, где:

$R = \left\{ \begin{array}{ll} S & \rightarrow aSbA, \\ aS & \rightarrow bA, \\ bA & \rightarrow B, \\ SbA & \rightarrow \varepsilon, \\ B & \rightarrow b \end{array} \right\}$

Левая часть правила должна содержать по крайней мере один нетерминал:

$\alpha \in (T \cup N)^* N (T \cup N)^*, \beta \in (T \cup N)^*$

Если нетерминалы – большими буквами, терминалы – маленькими, то при известном начальном нетерминале грамматика – это просто множество правил R

$G_0 =$

1. $S \rightarrow aSbA$
2. $aS \rightarrow bA$
3. $bA \rightarrow B$
4. $SbA \rightarrow \varepsilon$
5. $B \rightarrow b$


“Принцип действия” грамматики Хомского

G_0 :

1. $S \rightarrow aSbA$
2. $aS \rightarrow bA$
3. $bA \rightarrow B$
4. $SbA \rightarrow \varepsilon$
5. $B \rightarrow b$

- Как грамматика порождает цепочки языка?
- **Определение.** Из цепочки α *непосредственно выводима* цепочка β в грамматике G (обозначается $\alpha \Rightarrow \beta$), если:
 - цепочку α можно представить $\alpha = \mu\phi\gamma$ (некоторые из цепочек μ , ϕ , γ м. б. пусты);
 - цепочку β также можно представить как конкатенацию $\beta = \mu\psi\gamma$;
 - в G есть продукция $\phi \rightarrow \psi$, разрешающая подстановку ψ вместо ϕ

$\alpha \Rightarrow \beta$

μ	ϕ	γ
$\alpha =$	XXXXXX yy ZZZZZZZ	
		
$\beta =$	XXXXXX WWWW ZZZZZZZ	
μ	ψ	γ

$yy \rightarrow WWWW$

в грамматике

$bAcaS \Rightarrow BcaS$, если по правилу 3 подцепочку bA заменим на цепочку B (в G_0)

$bAcaS \Rightarrow bAc bA$, если по правилу 2 подцепочку aS заменим на bA

$bAcaS \Rightarrow bAca aSbA$, если по правилу 1 заменим S на $aSbA$

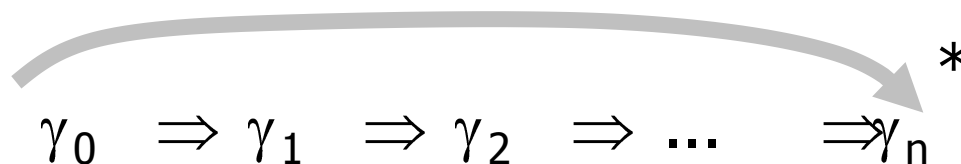
Условие выводимости цепочки

G_0 : 1 $S \rightarrow aSbA$
2 $aS \rightarrow bA$
3 $bA \rightarrow B$
4 $SbA \rightarrow \varepsilon$
5 $B \rightarrow b$

$$\alpha \Rightarrow^* \beta$$

- **Определение.** Из цепочки α в грамматике G выводима цепочка β (обозначается $\alpha \Rightarrow^* \beta$), если $\alpha = \gamma_0 \Rightarrow \gamma_1 \Rightarrow \gamma_2 \Rightarrow \dots \Rightarrow \gamma_n = \beta$.

Иными словами, цепочка β выводима из цепочки α в грамматике G , если β можно получить по правилам этой грамматики из α за конечное число шагов непосредственной выводимости. Символ " \Rightarrow^* " означает рефлексивное транзитивное замыкание отношения " \Rightarrow " (* - операция Клини)

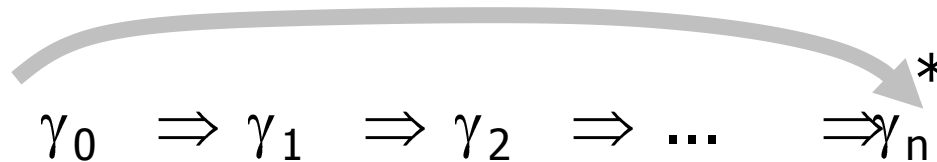


$$\alpha \Rightarrow^* \beta$$

Условие выводимости цепочки

G_0 :

- 1 $S \rightarrow aSbA$
- 2 $aS \rightarrow bA$
- 3 $bA \rightarrow B$
- 4 $SbA \rightarrow \varepsilon$
- 5 $B \rightarrow b$



- Из цепочки $bAcaS$ в G_0 можно вывести несколько цепочек. Например:
 - $bAcaS \Rightarrow^* bcb$, поскольку $\mathbf{bAcaS} \Rightarrow \mathbf{BcaS} \Rightarrow \mathbf{bcaS} \Rightarrow \mathbf{bcbA} \Rightarrow \mathbf{bcB} \Rightarrow \mathbf{bcb}$;
 - $bAcaS \Rightarrow^* Bcaaa$, поскольку $\mathbf{bAcaS} \Rightarrow \mathbf{bAcaaaSbA} \Rightarrow \mathbf{bAcaaa} \Rightarrow \mathbf{Bcaaa}$.
- Грамматика Хомского - это механизм порождения одних символьных цепочек из других символьных цепочек
- Из одной и той же цепочки можно породить несколько различных цепочек, иногда бесконечное их количество
- Например, в G_0 цепочка $bAcaS$ после подстановки вместо S цепочки $aSbA$ по первому правилу превращается в цепочку, также содержащую S . Из нее по тому же правилу опять можно породить цепочку, содержащую S , и т.д.



Как грамматика порождает язык?

G_0 : 1 $S \rightarrow aSbA$
2 $aS \rightarrow bA$
3 $bA \rightarrow B$
4 $SbA \rightarrow \varepsilon$
5 $B \rightarrow b$

Определение.

Языком, порождаемым грамматикой G , называется множество терминальных цепочек, выводимых из начального символа грамматики

$$L(G) = \{\alpha \in T^* \mid S \Rightarrow_G^* \alpha\}$$

Любой вывод цепочек языка начинается только с начального нетерминала. Если после произвольного конечного числа подстановок подцепочек в соответствии с правилами грамматики полученная в результате цепочка состоит из **терминалов**, то это – цепочка порождаемого грамматикой языка

Например, цепочка bb принадлежит языку, порождаемому грамматикой G_0 .
Действительно: $S \Rightarrow aSbA \Rightarrow bAbA \Rightarrow BbA \Rightarrow BB \Rightarrow bB \Rightarrow bb$

Отсюда ясны названия *терминальные-нетерминальные* символы

Только *терминальные*, 'окончательные' символы могут встретиться в цепочках языка, заканчивающих вывод

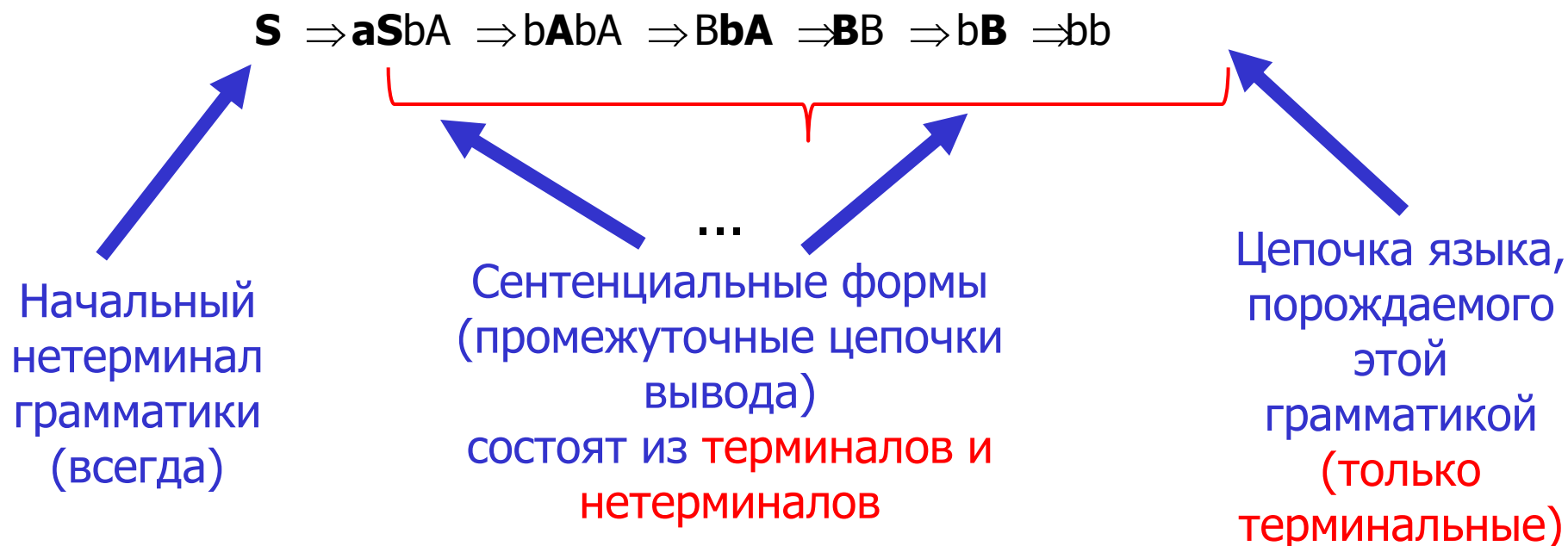
Нетерминальные символы – это вспомогательные символы, они никогда не встречаются в цепочках языка

Сентенциальные формы

Sentence – предложение

G_0 :
1 $S \rightarrow aSbA$
2 $aS \rightarrow bA$
3 $bA \rightarrow B$
4 $SbA \rightarrow \varepsilon$
5 $B \rightarrow b$

- Вывод цепочки языка из начального символа грамматики



Каждая сентенциальная форма получена из предыдущей по одному из правил грамматики

Какой язык порождает грамматика?

G_0 :
1 $S \rightarrow aSbA$
2 $aS \rightarrow bA$
3 $bA \rightarrow B$
4 $SbA \rightarrow \varepsilon$
5 $B \rightarrow b$

- Для замены S в G_0 есть только одно правило: $S \rightarrow aSbA$.
Заменять S по первому правилу можно многократно, следовательно, мы в общем случае можем получить промежуточные цепочки вывода в этой грамматике: $S \Rightarrow^* a^n S (bA)^n$
- Далее вывод может пойти двумя путями: либо мы используем правило $aS \rightarrow bA$ для того, чтобы заменить подцепочку aS на цепочку, не включающую S , либо по правилу 4 подцепочка SbA будет заменена на пустую цепочку
 - В первом случае имеем:
 $S \Rightarrow^* a^n S (bA)^n = a^{n-1} aS (bA)^n \Rightarrow a^{n-1} bA (bA)^n \Rightarrow^* a^{n-1} B^{n+1} \Rightarrow^* a^{n-1} b^{n+1}$
 - Во втором:
 $S \Rightarrow^* a^n S (bA)^n = a^n SbA (bA)^{n-1} \Rightarrow a^n (bA)^{n-1} \Rightarrow^* a^n B^{n-1} \Rightarrow^* a^n b^{n-1}$

Окончательно: $L(G_0) = \{ a^{n-1} b^{n+1} \mid n > 0 \} \cup \{ a^n b^{n-1} \mid n > 0 \}$

Примеры цепочек, порождаемых грамматикой G_0 :

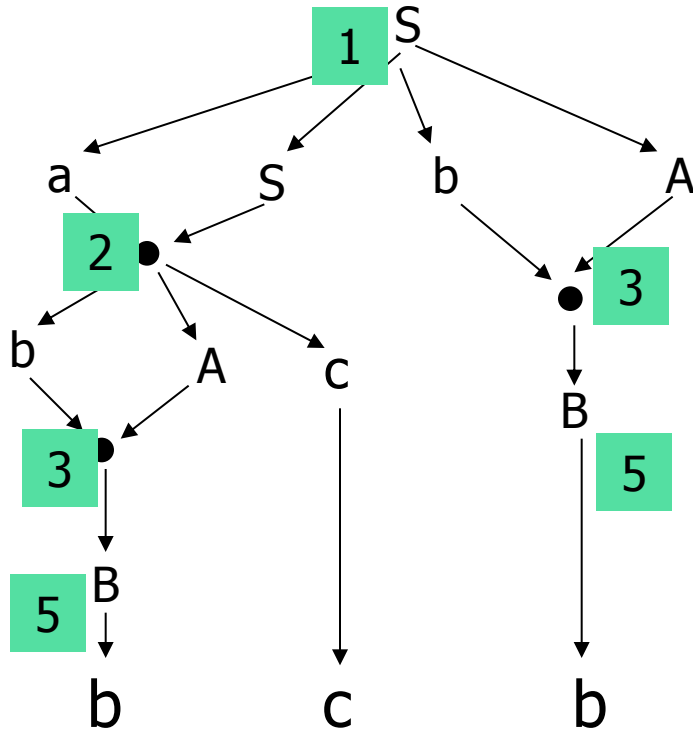
$L(G_0) = \{ bb, abbb, aabbbb, aaabbbbb, \dots, a, aab, aaabb, aaaabbb, \dots \}$

G1: 1 S \rightarrow aSbA
2 aS \rightarrow bAc
3 bA \rightarrow B
4 SbA \rightarrow ϵ
5 B \rightarrow b

- Порождающая грамматика Хомского - это "рецепт",
в соответствии с которым можно породить любое предложение языка. Но
грамматика не говорит, КАК породить конкретную цепочку языка

Например, цепочка `bcb` принадлежит языку, порождаемому грамматикой `G1`.

Действительно: $\mathbf{S} \Rightarrow \mathbf{aSbA} \Rightarrow \mathbf{bAcbA} \Rightarrow \mathbf{VcbA} \Rightarrow \mathbf{VcB} \Rightarrow \mathbf{bcB} \Rightarrow \mathbf{bcb}$



Структура выведенной цепочки

Насколько мощны порождающие грамматики Хомского?

- Порождающая грамматика Хомского для конечного языка

$\Sigma_1 = \{a, b, c\}$ $L_1 = \{abc, cc\}$ – **конечный язык**

$cc \in L_1$ $cbc \notin L_1$

$G_1::$ $S \rightarrow abc$
 $S \rightarrow cc$

Порождающая грамматика Хомского для конечных языков строится тривиально

- $\Sigma_2 = \{a, b, c\}$ $L_2 = \emptyset$ – **пустой язык**, $cc \notin L_2$

$G_2::$ $S \rightarrow S$

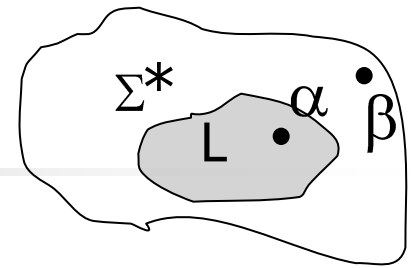
В G_2 не выводится ни одной терминальной цепочки

- $\Sigma_3 = \{a, b, c\}$ $L_3 = \Sigma^*$ – **все возможные цепочки из a, b и c**

$G_3::$ $S \rightarrow aS$
 $S \rightarrow bS$
 $S \rightarrow cS$
 $S \rightarrow \epsilon$

$S \Rightarrow bS \Rightarrow baS \Rightarrow babS \Rightarrow babbS \Rightarrow babb$

Примеры порождающих грамматик

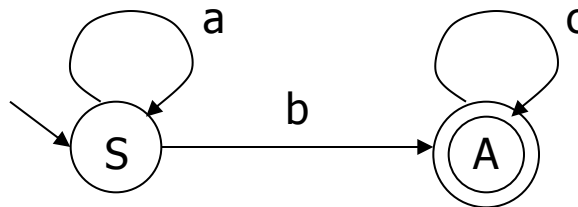


- Порождающая грамматика Хомского для автоматного языка

$\Sigma_4 = \{a, b, c\}$ $L_4 = \{a^n b c^m \mid n, m \geq 0\}$ $aaaabcc \in L_4$ $cbaa \notin L_4$

$G_4::$ $S \rightarrow aS$
 $S \rightarrow bA$
 $A \rightarrow cA$
 $A \rightarrow \epsilon$

$S \xRightarrow{*} a^n S \Rightarrow a^n b A \xRightarrow{*} a^n b c^m A \Rightarrow a^n b c^m$



Грамматика Хомского с легкостью порождает автоматные языки.
Между конечными автоматами и грамматиками Хомского - тесная связь:
автоматные грамматики – подкласс грамматик Хомского

Грамматика Хомского для неавтоматных языков

Многие языки представляются порождающими грамматиками Хомского

□ $\Sigma_5 = \{a, b, c\}$ $L_5 = \{a^n b c^n \mid n \geq 0\}$ $aabcc \in L_5$ $cbaa \notin L_5$

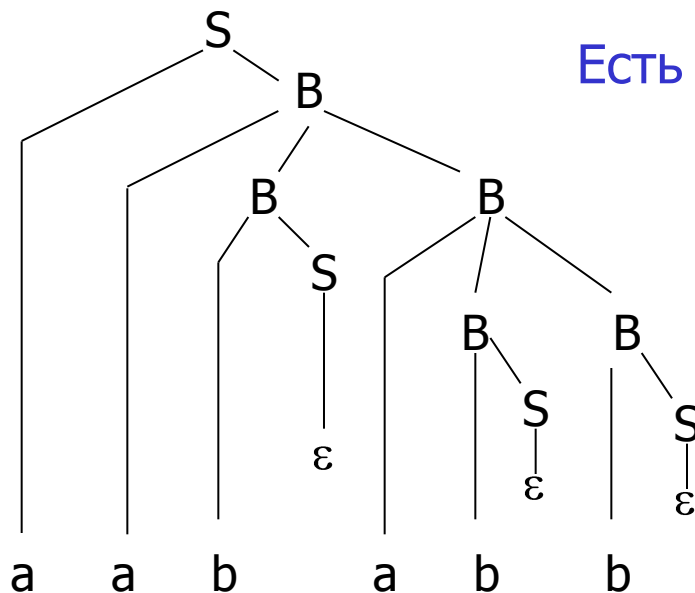
$G_5::$ $S \rightarrow aSc$
 $S \rightarrow b$

$S \Rightarrow aSc \Rightarrow aaScc \Rightarrow aaaSccc \Rightarrow aaa b ccc$

□ $\Sigma_6 = \{a, b\}$ $L_6 = \{\alpha \in \Sigma_6^* \mid \text{в } \alpha \text{ количества вхождений } a \text{ и } b \text{ равны}\}$

$G_6::$ $S \rightarrow aB$
 $S \rightarrow bA$
 $B \rightarrow aBB$
 $B \rightarrow bS$
 $A \rightarrow bAA$
 $A \rightarrow aS$
 $S \rightarrow \epsilon$

Есть и другое дерево вывода





Пример грамматики Хомского: язык согласованных скобок

- Язык согласованных скобок строится над алфавитом $\Sigma = \{ (,) \}$.
- Как конечным образом строго определить все возможные цепочки согласованных скобок, которых бесконечное число?
 - Правильные “скобочные скелеты” записей функций (но что значит “правильные”?)
 - Можно определить рекурсивно:
 - пустая цепочка является правильным скобочным выражением
 - если цепочки x и y являются правильными скобочными выражениями, то (x) и xy – тоже правильные скобочные выражения
 - других правильных скобочных выражений нет
 - Грамматика Хомского позволяет задать язык правильных скобочных выражений полностью в соответствии с этим рекурсивным определением. Эта грамматика содержит только три правила:

G::	$S \rightarrow \epsilon$	// пустая цепочка – правильное скобочное выражение
	$S \rightarrow (S)$	// правильное скобочное выражение, взятое в скобки, тоже является правильным скобочным выражением
	$S \rightarrow SS$	// два рядом стоящих правильных скобочных выражения тоже является правильным скобочным выражением

Грамматики Хомского для различных языков

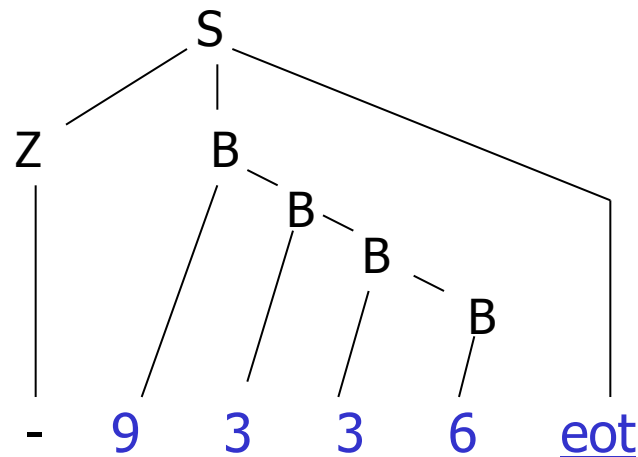
- $\Sigma_7 = \{0,1\}$ L_7 = множество четных двоичных чисел $100 \in L_7$ $01101 \notin L_7$

$G_7::$ $S \rightarrow 0 S$
 $S \rightarrow 1 S$
 $S \rightarrow 0$

- $\Sigma_8 = \{ '+', '-', '0', \dots, '9' \}$; L_8 = множество целых констант $-67 \in L_8$ $2+3-4 \notin L_8$

$G_8::$ $S \rightarrow '+' B$
 $S \rightarrow '-' B$
 $S \rightarrow \epsilon C$
 $B \rightarrow \epsilon C$
 $C \rightarrow \epsilon C$
 $C \rightarrow \underline{eot} D$
 $D \rightarrow \epsilon$

$G'_8::$ $S \rightarrow Z B \underline{eot}$
 $Z \rightarrow '-'$
 $Z \rightarrow '+'$
 $Z \rightarrow \epsilon$
 $B \rightarrow \epsilon B$
 $C \rightarrow \epsilon$



Для одного и того же языка можно построить бесконечное множество порождающих этот язык грамматик



Эквивалентных грамматик много (бесконечное число)

- Для любого языка существует множество грамматик Хомского, порождающих этот язык, но проблема эквивалентности грамматик Хомского в общем случае алгоритмически неразрешима

Арифметические выражения

- $V = \{ (,), +, -, *, /, i \}$; L = арифметические выражения

Пример цепочки в языке: $(i + i) * i$

$$E \Rightarrow E * E \Rightarrow (E) * E \Rightarrow (E + E) * E \Rightarrow (i + E) * E \Rightarrow (i + i) * E \Rightarrow (i + i) * i$$

G: $E \rightarrow i$

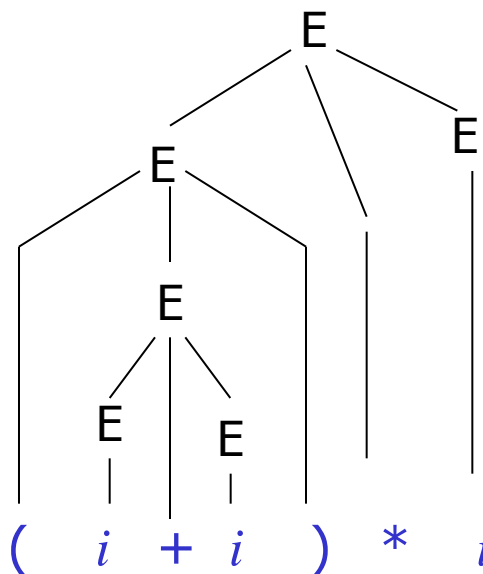
$E \rightarrow (E)$

$E \rightarrow E + E$

$E \rightarrow E - E$

$E \rightarrow E / E$

$E \rightarrow E * E$

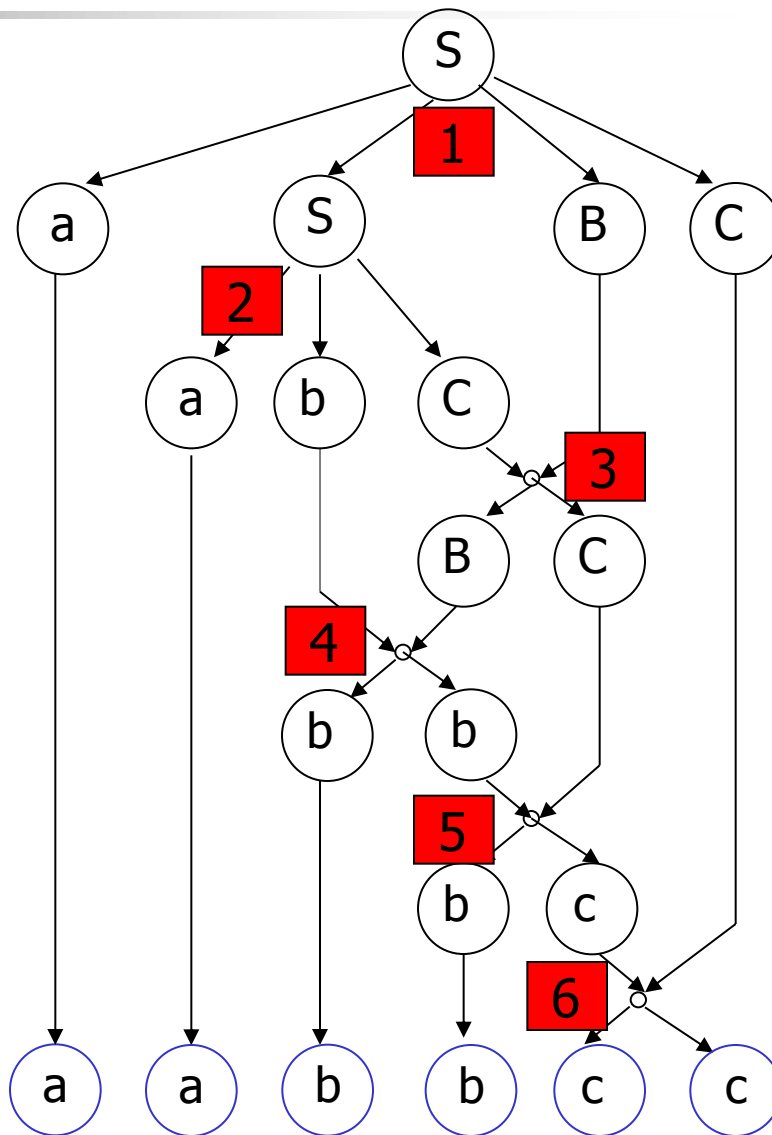


Графическое представление вывода цепочки

- $\Sigma = \{a, b, c, \}$; $L = \{a^n b^n c^n \mid n > 0\}$
 $aaabbbccc \in L$

1. $S \rightarrow aSBC$
2. $S \rightarrow abC$
3. $CB \rightarrow BC$
4. $bB \rightarrow bb$
5. $bC \rightarrow bc$
6. $cC \rightarrow cc$

$S \Rightarrow_1 aSBC \Rightarrow_2 aabCBC \Rightarrow_3 aabBCC \Rightarrow_4$
 $aabbCC \Rightarrow_5 aabbcC \Rightarrow_6 aabbcc$



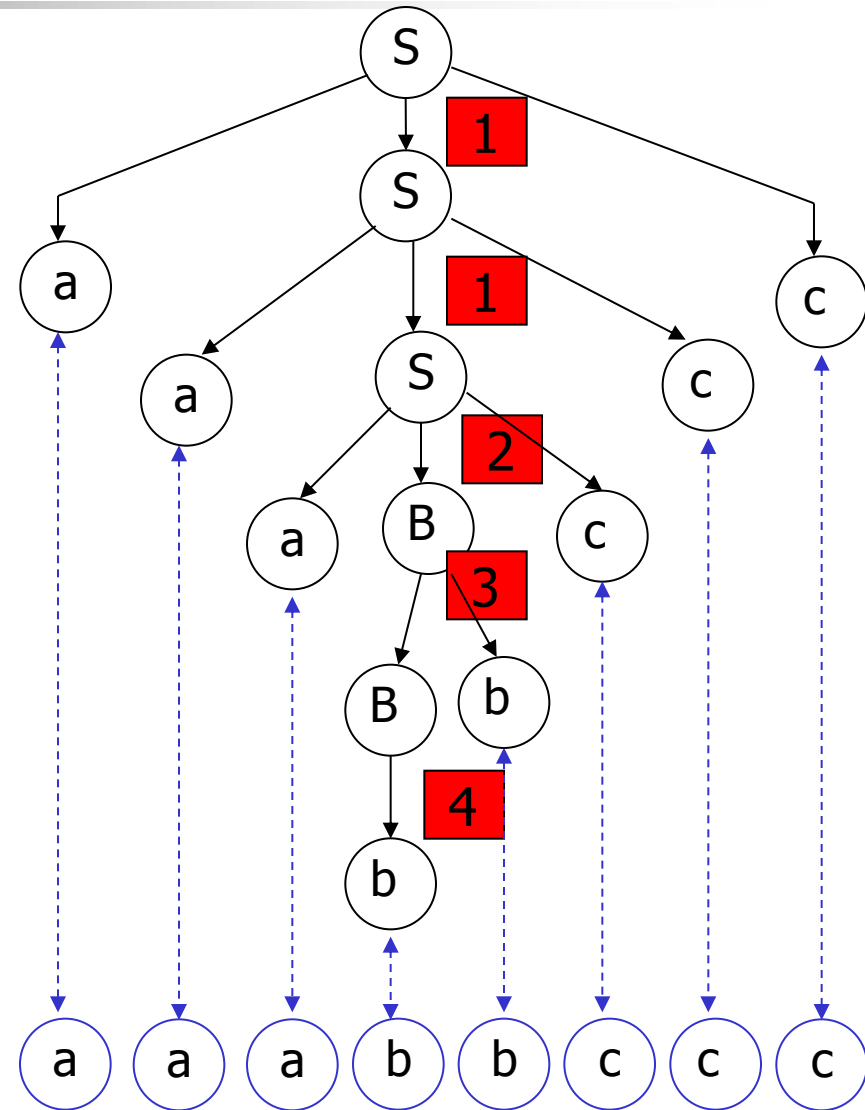
Графическое представление вывода цепочки (2)

- $\Sigma = \{a, b, c, \}$; $L = \{a^n b^m c^n \mid n > 0\}$
 $aaabbccsc \in L$

1. $S \rightarrow aSc$
2. $S \rightarrow aBc$
3. $B \rightarrow Bb$
4. $B \rightarrow b$

$S \Rightarrow_1 aSc \Rightarrow_1 aaSc \Rightarrow_2 aaaBccc \Rightarrow_3$
 $aaaBbccc \Rightarrow_4 aaabbccsc$

Так называемые КС-грамматики
(контекстно-свободные),
с правилами вида $A \rightarrow \alpha$
имеют **дерево** вывода



Пример КС-грамматики для фрагмента русского языка

- $\Sigma_{13} = \{\text{словоформы русского языка}\}$ $L_{13} = \text{русский язык}$

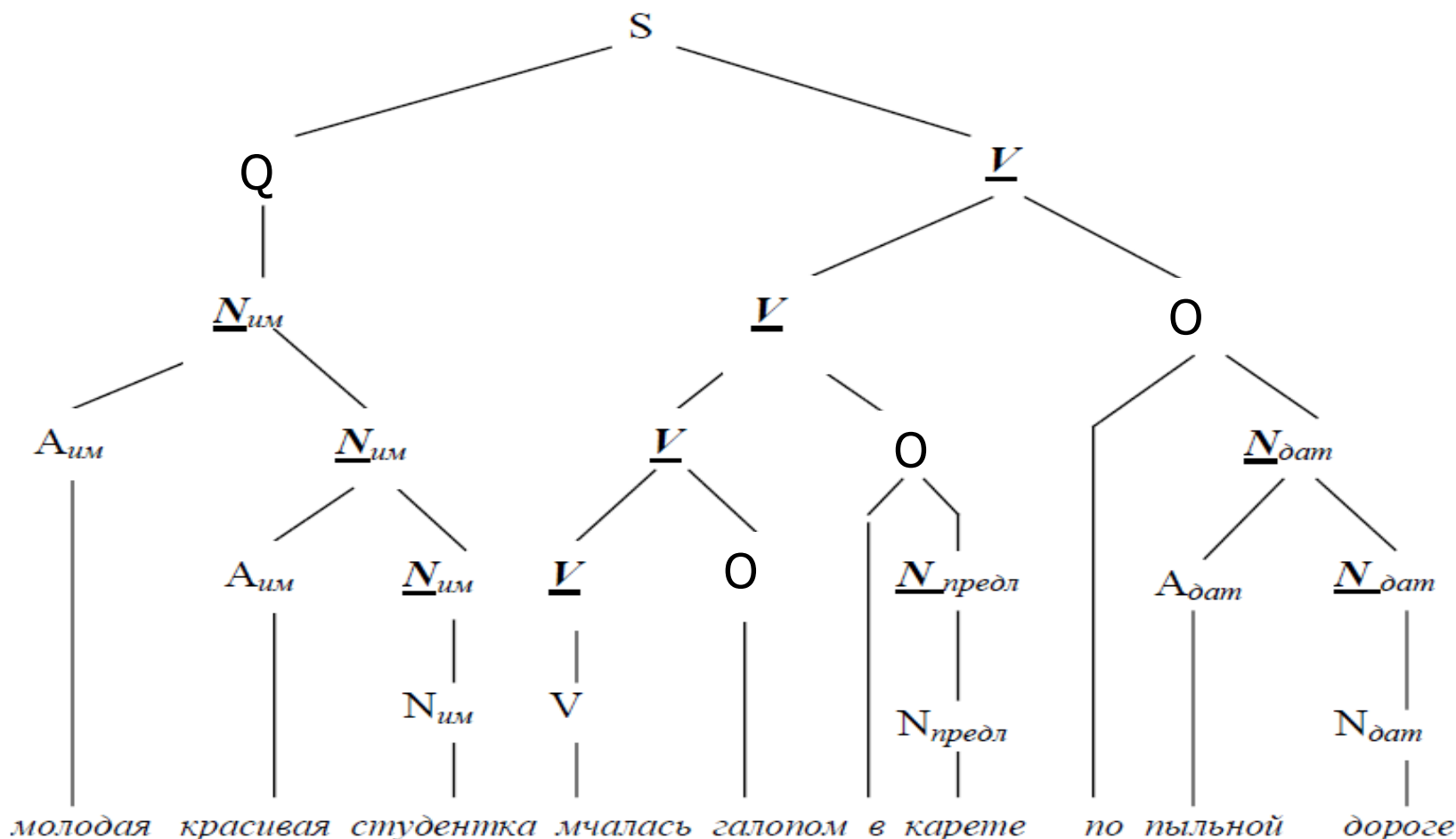
молодая красивая студентка мчалась галопом в карете по пыльной дороге $\in L_{13}$

Граматики Хомского для порождения естественных языков не используются. Иногда их можно использовать для порождения подмножеств естественных языков

S - предложение;
Q – группа подлежащего;
V – глагол;
V – группа сказуемого;
 N_z – суц в падеже z;
 \underline{N}_z - гр суц в падеже z;
 A_z – прилагат в падеже z;
O - обстоятельство

G_{13} : $S \rightarrow Q V \mid V Q$
 $Q \rightarrow \underline{N}_{им}$
 $\underline{V} \rightarrow \underline{V} O \mid O \underline{V} \mid V$
 $\underline{N}_z \rightarrow A_z \underline{N}_z \mid N_z$
 $V \rightarrow \text{шла} \mid \text{бежала} \mid \text{летела} \mid \text{скакала} \mid \dots$
 $N_z \rightarrow \text{деревня}_z \mid \text{дорога}_z \mid \text{студентка}_z \mid$
 $\text{ведьма}_z \mid \text{метла}_z \dots$
 $A_z \rightarrow \text{красивая}_z \mid \text{молодая}_z \mid \text{широкая}_z \mid$
 $\text{пыльная}_z \mid \text{ночная}_z \dots$
 $O \rightarrow \text{пешком} \mid \text{верхом} \mid \text{галопом} \mid$
 $\text{на } \underline{N}_{предл} \mid \text{над } \underline{N}_{твор} \mid \text{в } \underline{N}_{предл} \mid$
 $\text{по } \underline{N}_{дат} \dots$

Пример. Дерево вывода фразы естественного языка



Не каждое предложение, правильное с точки зрения этой грамматики, имеет смысл. Но это относится к любым грамматикам естественных языков. Например каковы смыслы:

“Ехала деревня мимо мужика, вдруг из-под собаки лают ворота”, “Чем пахнет надежда?”

Пример: простое подмножество естественного языка

- Формулы темпоральной логики LTL не очень ясны разработчикам программных систем при задании требований к программам. Например: $G(\text{req} \Rightarrow F \text{ack})$ – *во всех состояниях если послан запрос req, то когда-то в будущем обязательно придет подтверждение ack*
- Существует несколько систем, которые позволяют разработчикам использовать “шаблоны” при записи таких требований:

$G(\text{req} \Rightarrow F \text{ack})$ “assert *always (if req then eventually ack)*”

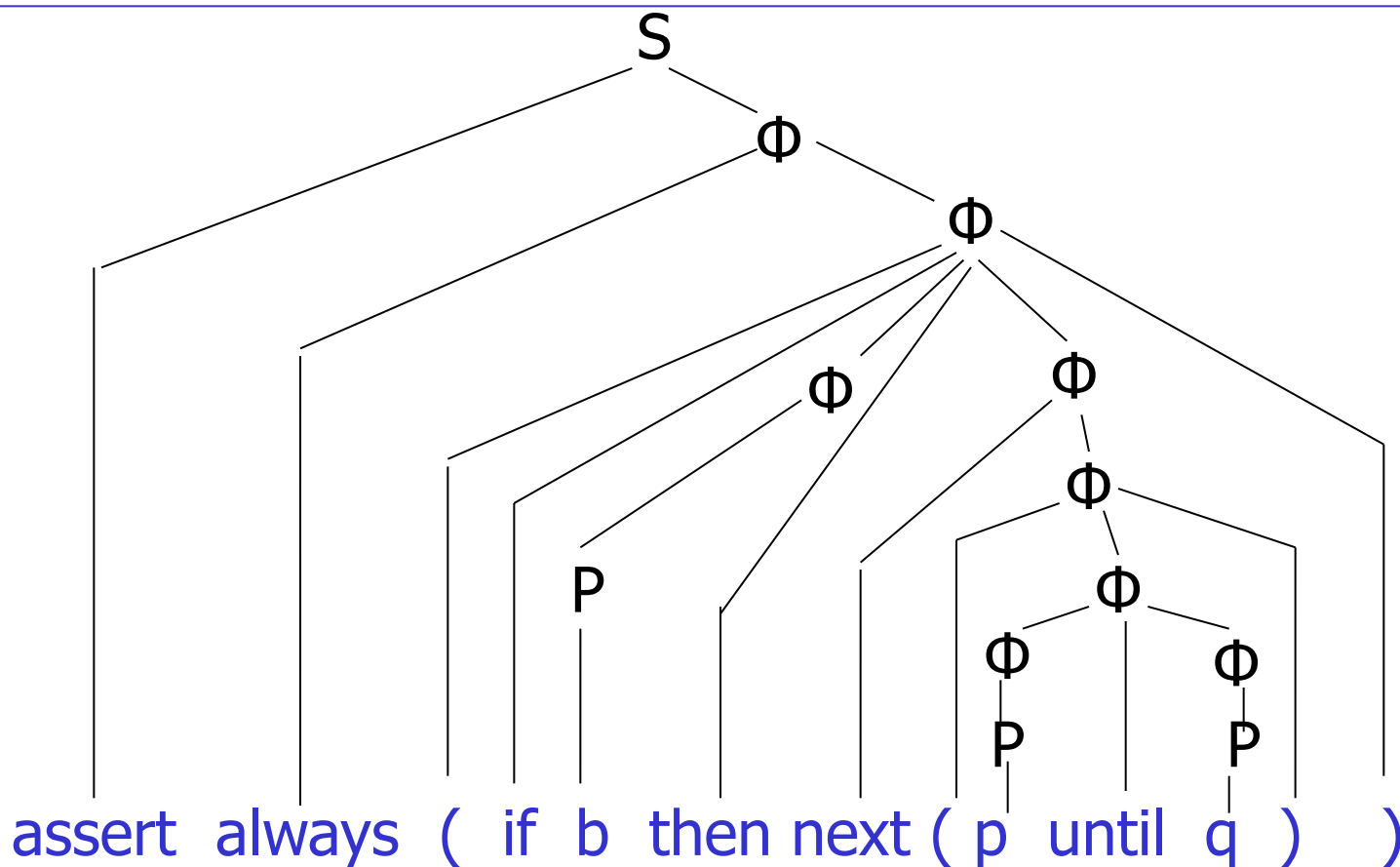
$G(b \Rightarrow X (p \cup q))$ “assert *always (if b then next (p until q))*”

G: $S \rightarrow \text{assert } \Phi$
 $\Phi \rightarrow P \mid (\Phi) \mid \text{not } \Phi \mid (\Phi \text{ or } \Phi) \mid (\Phi \text{ and } \Phi) \mid (\text{if } \Phi \text{ then } \Phi) \mid$
 $\text{always } \Phi \mid \text{next } \Phi \mid \text{eventually } \Phi \mid \Phi \text{ until } \Phi$
 $P \rightarrow \text{req} \mid \text{ack} \mid a \mid b \mid c \mid \dots \mid x \mid y \mid z$

Трансляция цепочек этого конкретного специализированного языка – простая замена названий темпоральных операторов на X, F, G, U

КС-грамматика подмножества естественного языка

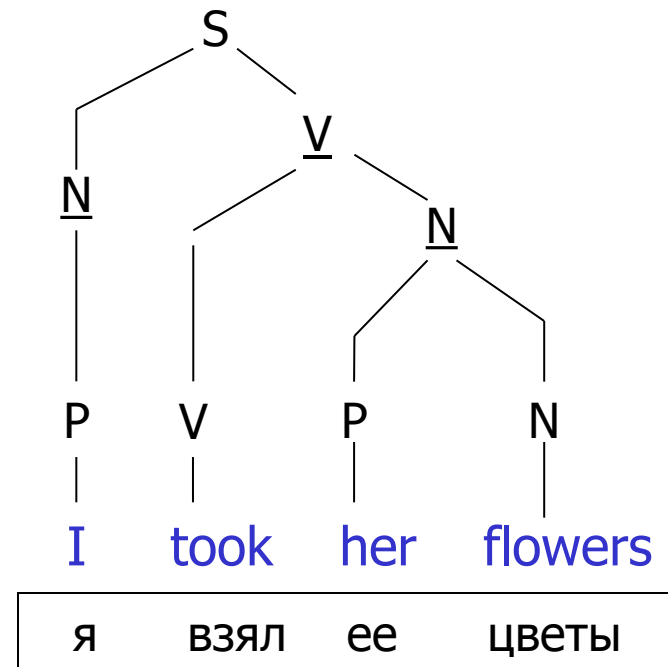
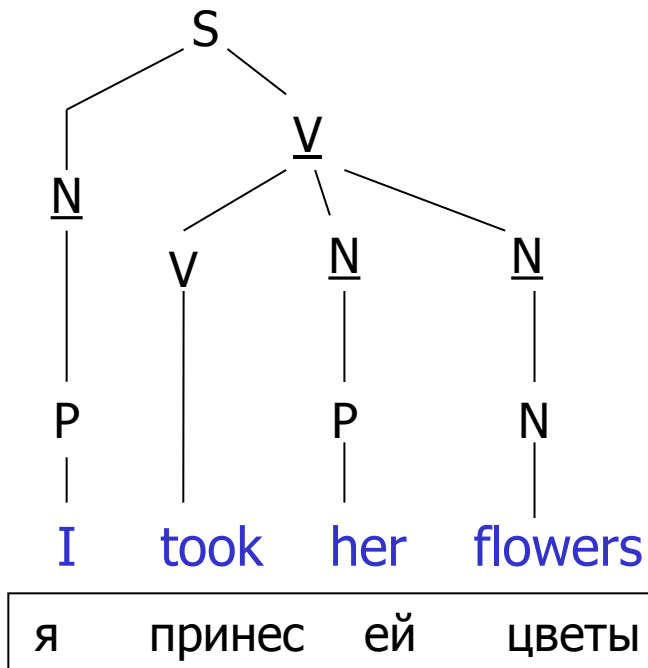
G: $S \rightarrow \text{assert } \Phi$
 $\Phi \rightarrow P \mid (\Phi) \mid \text{not } \Phi \mid (\Phi \text{ or } \Phi) \mid (\Phi \text{ and } \Phi) \mid (\text{if } \Phi \text{ then } \Phi) \mid$
 $\text{always } \Phi \mid \text{next } \Phi \mid \text{eventually } \Phi \mid \Phi \text{ until } \Phi$
 $P \rightarrow \text{req} \mid \text{ack} \mid a \mid b \mid c \mid \dots \mid x \mid y \mid z$



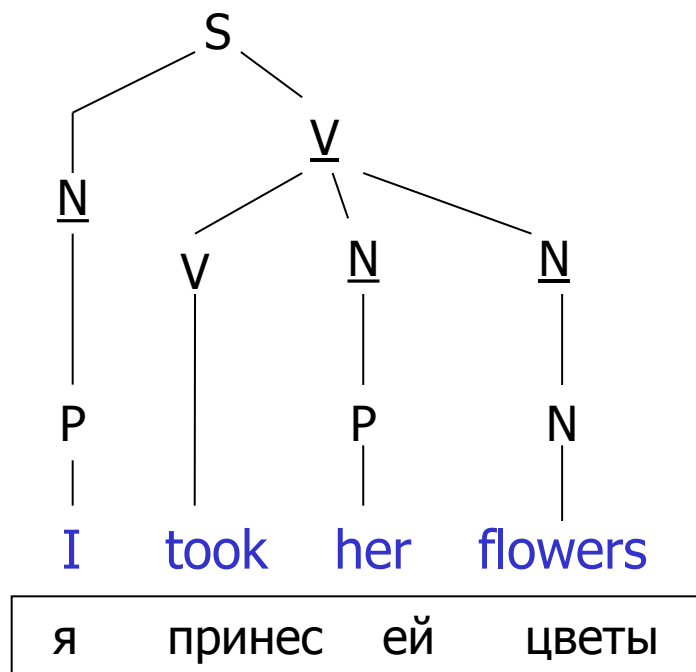
Грамматика подмножества английского языка

S - предложение;
N – группа существительного;
V – группа сказуемого;
V – глагол;
N – существительное;
P – местоимение.

G'_{13} : $S \rightarrow \underline{N} \underline{V}$
 $\underline{N} \rightarrow N \mid P \mid P N$
 $\underline{V} \rightarrow V \underline{N} \mid V \underline{N} \underline{N}$
 $N \rightarrow \textit{flowers} \mid \textit{cup} \mid \textit{chair} \mid \dots$
 $V \rightarrow \textit{was} \mid \textit{took} \mid \textit{bought} \mid \dots$
 $P \rightarrow I \mid \textit{they} \mid \textit{her} \mid \dots$



Смысл терминальных и нетерминальных символов



- Терминальные символы грамматики – это символы, которые появятся в конце, после вывода цепочки из начального нетерминального символа
- Нетерминальные символы – названия конструкций языка, фактически, начальные символы подязыков, задающих все возможные цепочки, которыми можно заменить подконструкции
- Правила порождающей грамматики – правила представления одних фрагментов языка другими фрагментами языка
- Неоднозначность цепочек языка (двусмысленность их трактовки) является следствием того, что мы можем по-разному группировать элементы предложения в соответствии с грамматикой, и приписывать им различные роли



КС-грамматика, задающая КС-грамматики

□ Грамматика ::= 'ALPHABET' (ОписаниеНетерминала) (Правило)

ОписаниеНетерминала ::= ИмяНетерминала

Правило ::= 'RULE' Синтаксис '.'

Синтаксис ::= ИмяНетерминала '::=' ПраваяЧасть

ПраваяЧасть ::= ПраваяЧасть [ЭлементПравойЧасти] | ε

ЭлементПравойЧасти ::= Терминал | Нетерминал



Нормальные формы КС-грамматик

- Нормальная форма Хомского:
правила – в одной из следующих форм:

$$A \rightarrow BC$$

$$A \rightarrow a$$

$$S \rightarrow \epsilon$$

- Нормальная форма Грейбах
(Sheila Greibach, проф. UCLA)

$$A \rightarrow \epsilon$$

$$A \rightarrow a$$

$$A \rightarrow aB$$

$$A \rightarrow aBC$$

Пример КС-грамматики в
нормальной форме Грейбах:

$$S \rightarrow \epsilon \mid AB$$

$$A \rightarrow CD \mid a$$

$$C \rightarrow CD \mid b$$

$$B \rightarrow CA$$

Пример КС-грамматики в
нормальной форме Грейбах:

$$S \rightarrow aST \mid aT$$

$$T \rightarrow bS \mid b$$

Теорема. Любая КС-грамматика может быть представлена в нормальной форма Хомского и в нормальной форме Грейбах



Идея порождающих грамматик Хомского

- *Терминальные символы* – это символы, из которых строятся цепочки языка, порождаемого грамматикой
- *Нетерминалы* – это вспомогательные символы, обозначающие конструкции, категории, понятия языка. Эти символы необходимы, когда мы рассуждаем о языке, но в цепочках языка эти символы не встречаются
- *Правила грамматики Хомского* – это выражение более абстрактных конструкций более конкретными конструкциями
- Например, нетерминал \underline{U} в грамматике русского языка обозначает конструкцию <группа сказуемого>, он нужен для задания языка, но не встречается ни в одной цепочке языка (кроме как у лингвистов). Из начального нетерминала, представляющего самую общую конструкцию грамматики, порождаются все цепочки языка
- Для языка определяется несколько абстрактных конструкций (нетерминалов), и правила порождения определяют, как каждая конструкция или группа конструкций и символов языка может быть построена из других конструкций языка и его символов.



Можно ли построить порождающую грамматику (и вообще конечное описание) для ЛЮБОГО возможного языка?

□ Ответ: **НЕТ!**

Как доказать? Число всех возможных слов (цепочек) над конечным словарем счетно. Любой язык является некоторым подмножеством всех возможных слов над конечным словарем, поэтому множество всех языков, как множество подмножеств счетного множества, имеет мощность континуум.

Для доказательства утверждения необходимо показать, что множество всех конечных описаний не более, чем счетно, т.е. оно равномощно множеству натуральных чисел, и поэтому оно меньше мощности всех возможных языков.

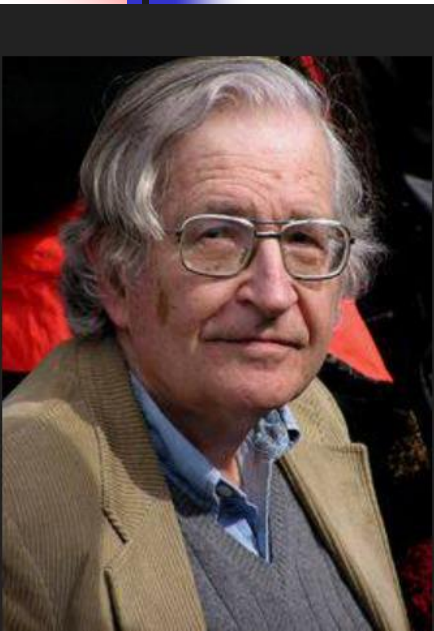
Доказательство. Любое конечное описание строится как строка в конечном алфавите. Например, порождающая грамматика Хомского для языка над словарем $\{a, b\}$ строится в алфавите $\{a, b, A, B, \dots, Z, \rightarrow, ;\}$. Все такие строки можно упорядочить по длине, а строки одной длины – в лексикографическом порядке. Таким образом, любое конечное описание (в частном случае, любая грамматика Хомского) будет иметь свой уникальный номер. Число таких номеров не больше числа натуральных чисел, т.е. не более, чем счетно



Заключение

- Ноам Хомский в 50-х годах XX века разрабатывал модели естественных языков. Его идея состояла в том, что для понимания предложения языка необходимо выявить его структуру. Это очевидно для двусмысленных предложений. Если структура зафиксирована, смысл становится однозначным
- Хомский предположил, что и в понимании недвусмысленных предложений, как и в понимании образов, структура играет определяющую роль
- Хомский предложил понятие порождающей грамматики, которая, фактически, является множеством правил подстановки цепочек. Структура цепочки языка определяется при ее выводе из начального символа
- Грамматики Хомского порождают любой конечный язык, любой автоматный язык, и многие неавтоматные языки.
- Для формальных языков, в частности, языков программирования, грамматики Хомского сегодня являются общепринятым формализмом, позволяющим конечным набором правил задавать бесконечные языки
- Грамматиками Хомского, и вообще конечными описаниями, можно задать лишь часть возможных формальных языков
- Для естественных языков грамматики Хомского не совсем адекватны. Существуют более тонкие модели, позволяющие осуществлять понимание и трансляцию естественных языков

Персоналии: Ноам Хомский



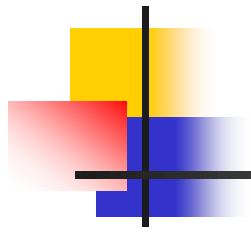
Ноам Хомский (Noam Chomsky, род. 1928) – выдающийся американский лингвист, философ, историк, политический деятель. Почетный профессор MIT (работает там > 50 лет). **Автор более 100 книг и более 1000 статей.**

В 2005 был определен как "**world's top public intellectual**".

Признан "**отцом современной лингвистики**", главной фигурой современной аналитической философии. Его работы - основополагающие в лингвистике, информатике, математике и философии. По оценкам разных индексов, Хомский - наиболее цитируемый автор современности.

Хомский — фамилия славянская. Его родители эмигрировали в США из России в 1912 г.

- Хомский – один из наиболее яростных критиков американской политики, государственного капитализма, американской экономики и *mainstream news media*. В этом качестве он сегодня известен больше, чем выдающийся лингвист. Его считают социалистом и анархистом, определяют как самого выдающегося аналитика в области внешней политики и экономики
- "*Judged in terms of the power, range, novelty and influence of his thought, Noam Chomsky is arguably the most important intellectual alive*" – The New York Times
- "*Obama is in many cases worse than George Bush and Tony Blair -- on Afghanistan, Pakistan, Israel, Egypt -- and would be indicted for war crimes if the Nuremburg principles were applied*"



Спасибо за внимание