

Рекурсивно-определенные предикаты.

Пример: множество натуральных чисел (с нулем).

A: $N(0)$ // 0 – натуральное.

B: $\forall x(N(s(x)) \leftarrow N(x))$ // $s(x)$ – натуральное, если x – натуральное.

Проверим, что $s(s(0))$ – натуральное:

$N(s(s(0)))$? B: $x=s(0): N(s(0))$? B: $x=0: N(0)$? A: \square .

Результат:

Рекурсивный алгоритм распознавания правильности построения формулы:

Быть натуральным числом = быть правильно построенной формулой в алфавите $\{ 0, s, (,) \}$.

Рекурсивное определение для двухместной операции сложения (обозначенной формулах символом “+”).

$$A: \forall x (x+0=x)$$

$$B: \forall x \forall y (x+s(y)=s(x+y))$$

Здесь = - бинарное отношение (двухместный предикат).

Вообще можно сказать, что мы задаем только отношения:

Е - равенство (эквивалентность):

$$\forall x E(x,x)$$

$$\forall x \forall y (E(x,y) \rightarrow E(y,x))$$

$$\forall x \forall y \forall z ((E(x,y) \& E(y,z)) \rightarrow E(x,z))$$

S – “Следовать за” (инъективная функция, не покрывающая 0):

$$\forall x \forall y \forall u \forall v ((S(x,y) \& S(u,v) \& E(y,v)) \rightarrow E(x,u))$$

$$\neg \exists x (S(x,0))$$

A – быть суммой (трехместное отношение):

$$\forall x \forall z (A(x,0,z) \rightarrow E(z,x))$$

$$\forall x \forall y \forall z \forall r ((S(y,z) \& A(x,z,r) \& A(x,y,u) \& S(u,v)) \rightarrow E(r,v))$$

Применение правил для вычисления результата рекурсивно определенной операции +

A: $\forall x (x+0=x)$

B: $\forall x \forall y (x+s(y)=s(x+y))$

“1+2=3”:

$$s(0)+s(s(0)) = s(s(0)+s(0)) = s(s(s(0)+0)) = s(s(s(0)))$$

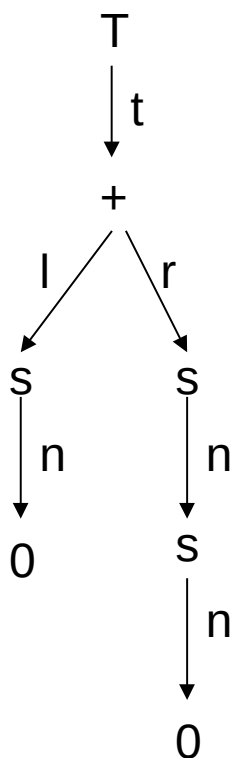
B: $x=s(0), y=s(0)$

B: $x=s(0), y=0$

A: $x=s(0)$

Представление формул ориентированными размеченными графами (ориентированными деревьями)

$s(0)+s(s(0))$



Отношения:

l – быть левым операндом

r – быть правым операндом

n – быть единственным операндом

t – быть корнем

Разметка вершин:

символы алфавита формулы

(плюс символ корня T)

Разметка ребер:

символы отношений позиций операндов

Фрагмент

T



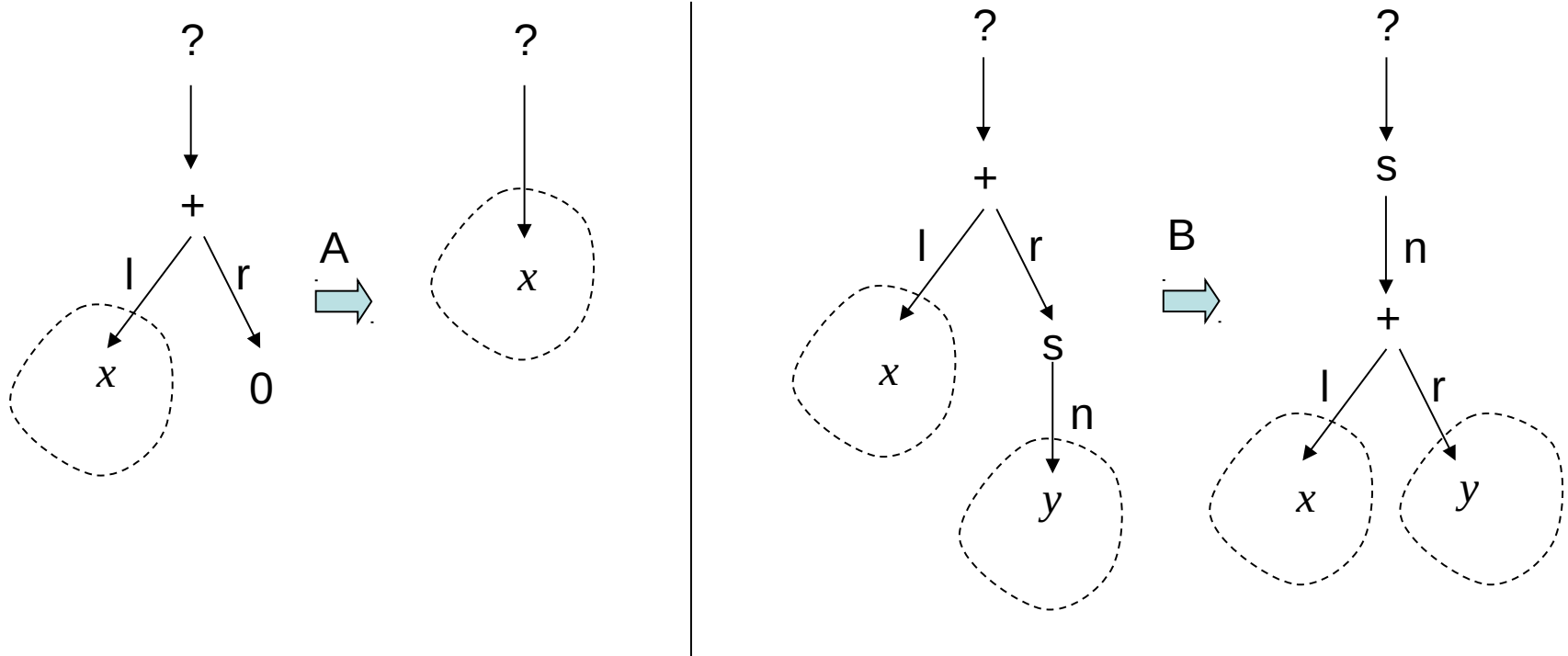
будем
опускать

Правила преобразования формул – правила трансформации фрагментов диаграмм

(и построения новых графов-деревьев формул)

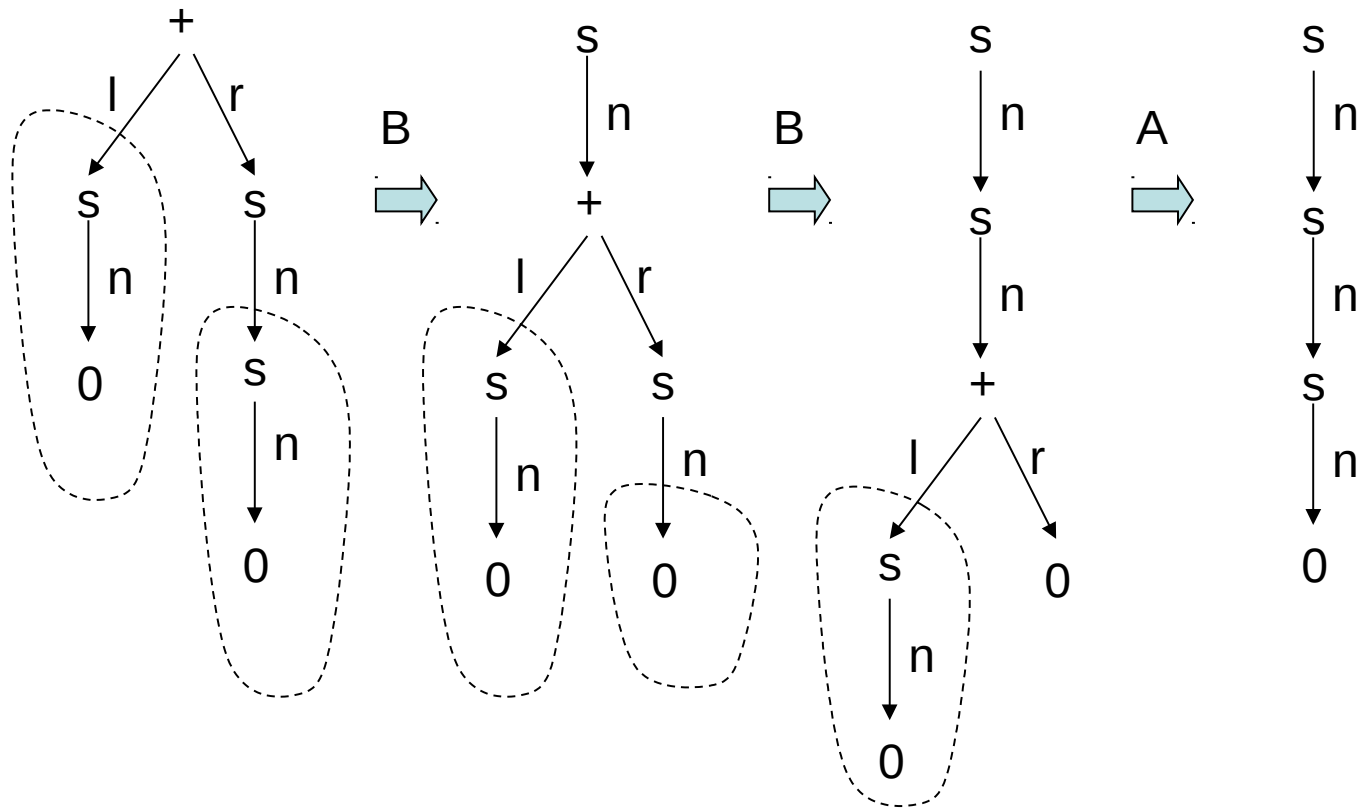
$$A: \forall x (x+0=x)$$

$$B: \forall x \forall y (x+s(y)=s(x+y))$$



Пример: вычисление $1+2=3$ графически

$$s(0)+s(s(0)) = s(s(0)+s(0)) = s(s(s(0)+0)) = s(s(s(0)))$$



Описание правил трансформации диаграмм (деревьев)

(возможный синтаксис языка описания трансформаций)

$\langle \text{Условие} \rangle \rightarrow \langle \text{Что менять} \rangle / \langle \text{На что менять} \rangle$

T = вся диаграмма, а также ее корневой символ в условии

T_l = левая поддиаграмма, если из корня идет стрелка l .

T_{rn} = поддиаграмма, достижимая из корня по маршруту стрелок r, n

Конструктор новых диаграмм (от корня по месту замены):

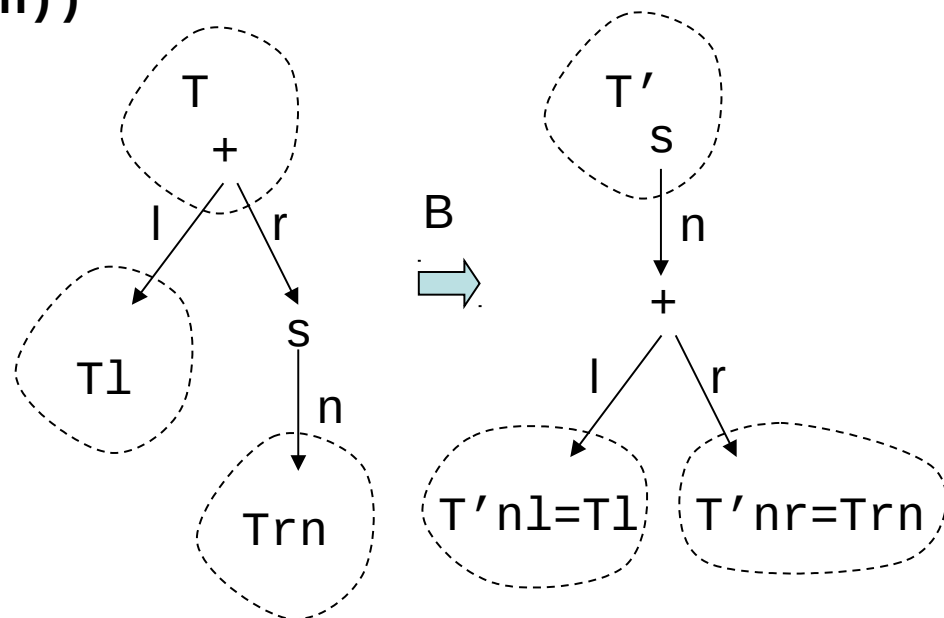
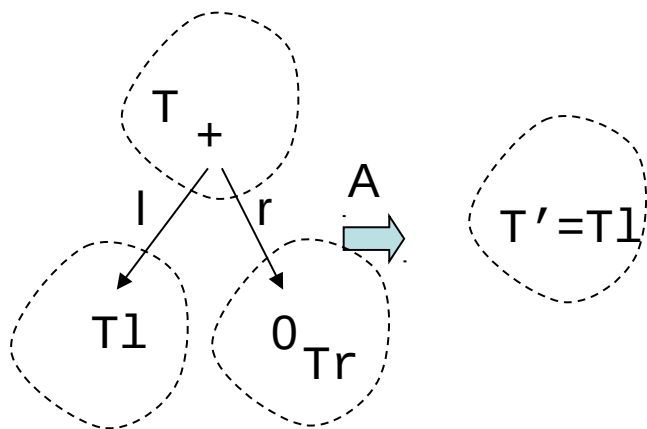
$s(n: \langle \text{Конструктор} \rangle)$ = вершина s , из которой идет стрелка n , и далее как указано

$+(l: \langle \text{конструктор1} \rangle, r: \langle \text{конструктор2} \rangle)$ = вершина $+$, со стрелками l и r ...

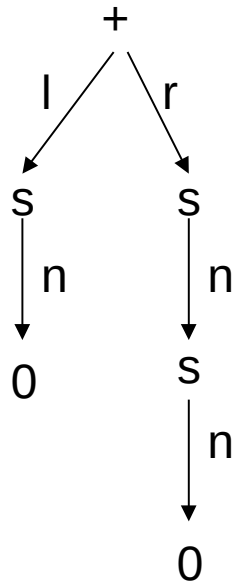
$(T=+) \& (Tr=0) \rightarrow T/T_l$

Нет подстановочных символов!!!

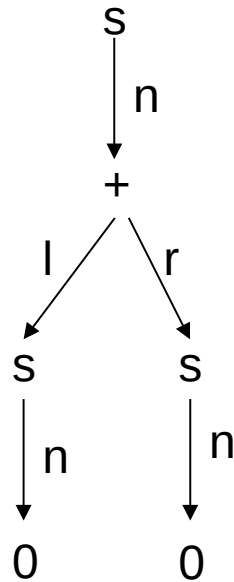
$(T=+) \& (Tr=s) \rightarrow T/s(n: +(l: T_l, r: T_{rn}))$



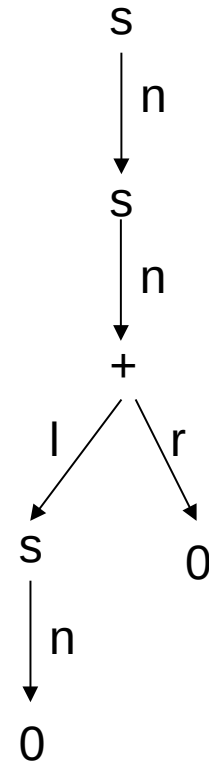
Представление деревьев строками в алфавите {0,s,+}
(обратная польская бесскобочная запись)



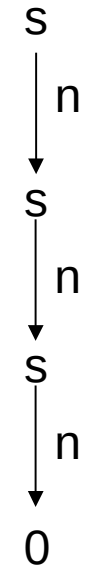
0s0ss+



0s0s+s



0s0+ss



0sss

Алгоритм перевода дерева в бесскобочную формулу-строку

Для каждого алфавита и валентности символов указать рекурсивную схему

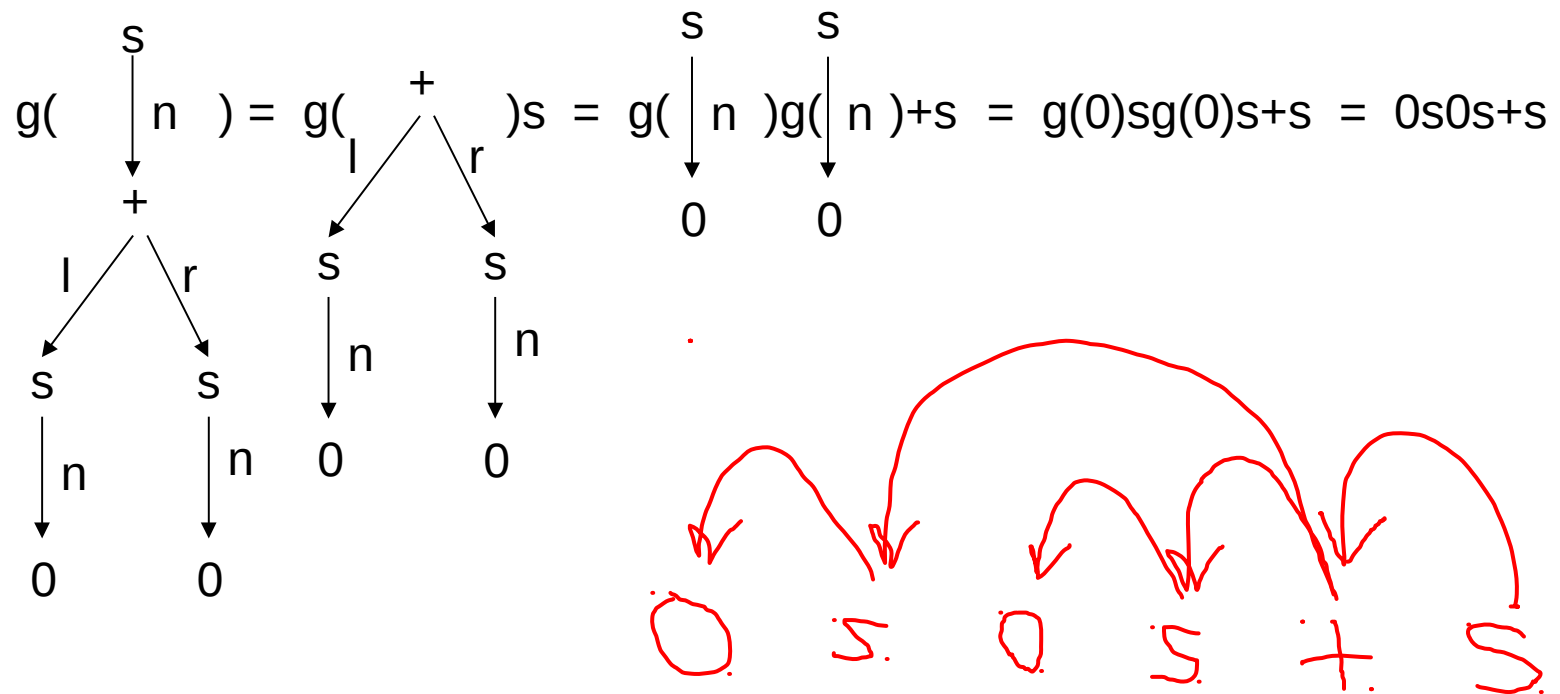
g – функция построения бесскобочной записи для своего аргумента-диаграммы

0-местный символ (терминальная вершина): $g(0)=0$

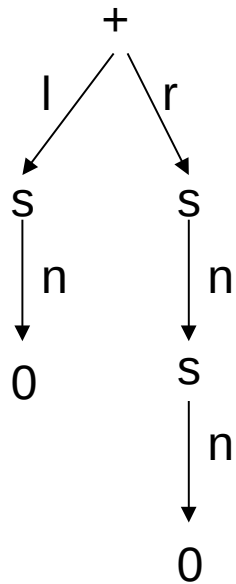
1-местный символ: $g\left(\begin{array}{c} s \\ \downarrow \\ x \end{array}\right) = g(x)s$

2-местный символ: $g\left(\begin{array}{cc} & + \\ & \swarrow \searrow \\ l & r \\ \swarrow \searrow & \swarrow \searrow \\ x & y \end{array}\right) = g(x)g(y) +$

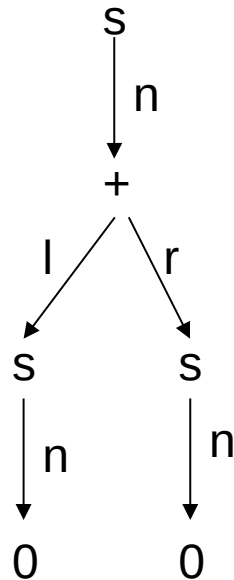
Пример перевода в бесскобочную запись



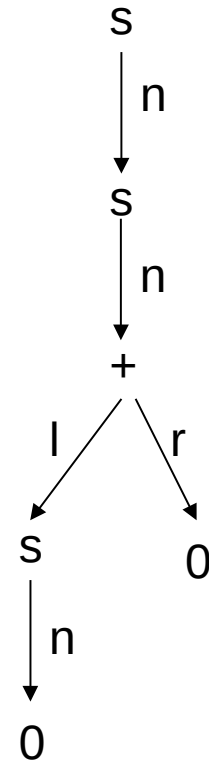
Алгоритмы на строках символов алфавита $\{0,s,+\}$



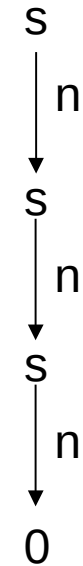
0s0ss+



0s0s+s



0s0+ss



0ssss

Каковы правила трансформации строк?

$s+ \rightarrow +s$

$0+ \rightarrow \varepsilon$

Здесь ε – пустая строка

Нормальный алгоритм Маркова

Правила замен подстрок (стратегия применения правил замен):

-Упорядочены сверху вниз

-Применяется к самой левой подстроке первое подходящее правило

-Если подстрок более нет, получена строка-результат

-Можно отметить правило как заключительное

Схема:

(1) $s+ \rightarrow +s$

(2) $0+ \rightarrow \varepsilon$

Применение:

