# 46. Определите синтаксически управляемый перевод и входную и выходную цепочки СУ-схемы

Рассмотренная схема перевода относится к классу схем, называемых *схемами*

*синтаксически управляемого перевода*.

**Определение.** Схемой синтаксически управляемого перевода называется

пятёрка Тb = (V, Σ, Δ, P, S), где:

V — конечное множество нетерминальных символов;

Σ — конечный входной алфавит;

Δ — конечный выходной алфавит;

P — конечное множество правил вида A→ α, β, где α ∈ (V ⋃ Σ)\*, β ∈ (V ⋃

Δ)\* и вхождения не терминалов в цепочку β образуют перестановку вхождений

не терминалов в цепочку α;

S — начальный символ (S ∈ V).

По определению, каждому вхождению определённого не терминала в

цепочку α соответствует некоторое вхождение этого не терминала в цепочку β.

Если некоторый не терминал входит в цепочку α более одного раза, то может

возникнуть неоднозначность. Для исключения этого будем использовать

верхний целочисленный индекс, например в правиле A → B1CB2, B2B1C

первой, второй и третьей позиции цепочки В1СВ2 соответствуют вторая, третья

и первая позиции цепочки В2В1С.

**Примечание:** выводимая пара цепочек Тb СУ-схемы определяется

рекурсивно следующим образом:

1. (S, S) — выводимая пара, в которой символы S соответствуют друг другу.

2. Если (αAβ, α’Aβ’) — выводимая пара, в которой два выделенных

вхождения не терминала А соответствуют друг другу, и A → α, β — правило из

P, то (α𝛾β, α’𝛾’ β’) — выводимая пара.

Вхождения не терминалов в цепочки 𝛾 и 𝛾’ соответствуют друг другу точно

так же, как они соответствовали в правиле СУ-схемы. Вхождения не

терминалов в цепочки α и β соответствуют вхождениям не терминалов в

цепочки α' и β' в новой выводимой паре точно так же, как они соответствовали в

158

старой выводимой паре. При необходимости это соответствие будет

указываться верхними индексами.

Если между парами цепочек (αAβ, α’Aβ’) и (α𝛾β, α’𝛾’β’) установлена

описанная ранее связь, то будем писать (αAβ, α’Aβ’) ⇒Tb (α𝛾β, α’𝛾’ β’).

Транзитивное замыкание, рефлексивно-транзитивное замыкание и k-ю

степень отношения ⇒Tb будем обозначать как ⇒+

Tb, ⇒\*Tb и ⇒kTb

соответственно. В очевидных случаях можно опускать индекс Тb.

**47. Определите L-атрибутную и S-атрибутную транслирующие грамматики**

**Определение.** АТ-грамматика называется L-атрибутной тогда и только тогда,

когда выполняются следующие условия:

1. Аргументами правила вычисления значения унаследованного атрибута

символа из правой части правила вывода могут быть только унаследованные

атрибуты символа из левой части и произвольные атрибуты символов из

правой части, расположенные левее рассматриваемого символа.

2. Аргументами правила вычисления значения синтезированного атрибута

символа из левой части правила вывода являются унаследованные атрибуты

этого символа или произвольные атрибуты символов из правой части.

3. Аргументами правила вычисления значения синтезированного атрибута

операционного символа могут быть только унаследованные атрибуты этого

символа.

Является ли произвольная АТ-грамматика L-атрибутной, можно проверить,

независимо исследуя каждое правило вывода и каждое правило вычисления

значения атрибута. Примером L-атрибутной транслирующей грамматики

является грамматика 𝐺 .

**Определение:** АТ-грамматика называется S-атрибутной тогда и только тогда,

когда она является L-атрибутной и все атрибуты нетерминалов

синтезированные.

Ограничения, накладываемые на L-атрибутную транслирующую грамматику,

позволяют вычислять значения атрибутов в процессе нисходящего анализа

входной цепочки. Нисходящий детерминированный анализатор для

LL(1)-грамматик требует, чтобы L-атрибутная транслирующая грамматика,

описывающая перевод, имела форму простого присваивания.

**48. Определите  копирующее правило вычисления атрибутов**

**Определение. П**равило вычисления атрибутов называется **копирующим**

**правилом** тогда и только тогда, когда левая часть правила - это атрибут или

список атрибутов, а правая часть – константа или атрибут. Правая часть

называется источником копирующего правила, а каждый атрибут из левой части

– приемником копирующего правила.

Если источники нескольких копирующих правил совпадают, то их приемники

можно объединить в одну левую часть. Например, правила z, w← a и х, у←z

можно записать в виде: х, у, z, w← a , т. к. источнику второго правила z,

согласно первому правилу, присваивается значение а. Аналогично х←у и a, b ←

y можно записать как а, b, х ←у.

**49. В каком случае АТ-грамматика имеет форму простого присваивания?**

**Определение.** Атрибутная транслирующая грамматика имеет форму

простого присваивания тогда и только тогда, когда:

1. Некопирующими являются только правила вычисления синтезируемых

атрибутов операционных символов.

2. Для каждого правила вывода грамматики соответствующее множество

копирующих правил независимо.

**50. Расширить восходящий анализатор действиями по выполнению перевода.**

1. Каждой функции ƒ(х1 , х2, ... ,х ), входящей 𝑥 в правило вычисления 1 𝑥2 𝑥𝑛

атрибутов, связанное с некоторым правилом вывода грамматики, создать

соответствующий ей операционный символ {ƒ}, который определяется

следующим образом:

{ƒ} 𝑥1 ,х2 ... ,х , р

унаследованные 𝑥1, 𝑥2 , ... , 𝑥𝑛

синтезированный р

p ← ƒ(𝑥1, 𝑥2 , ... , 𝑥𝑛 )

2. Для каждого некопирующего правила 𝑧𝑖, 𝑧2,…, 𝑧n ← ƒ(𝑦1, 𝑦2 , ... , 𝑦𝑛)

связанного с некоторым правилом вывода грамматики, найти символы а1 ... , а„,

которые не содержатся в этом правиле вывода, и вставить в его правую часть

символ{ƒ}𝑎n ← 𝑎2,…, 𝑎𝑛. Заменить некопирующее правило на следующие (n + 𝑛

1) копирующих правил:

𝑎 𝑖 ← 𝑦𝑛 для каждого аргумента 𝑦𝑖 ,

𝑧1 , 𝑧2 ,…, 𝑧𝑚 ←r

При включении в правило вывода операционного символа необходимо

соблюдать следующие условия:

● операционный символ должен располагаться правее всех символов

правой части правила вывода, атрибутами которых являются

аргументы y1, …, yn ;

● операционный символ должен располагаться левее всех символов

правой части правила вывода, атрибутами которых служат z1 ,…, zn ;

● с учётом предыдущих ограничений операционный символ может

быть вставлен в любое место правой части правила вывода, но

предпочтение следует отдать самой левой из возможных позиций,

т.к. это позволяет упростить реализацию синтаксического

анализатора.

2. Два копирующих правила, соответствующие одному и тому же правилу

вывода, необходимо объединить, если источник одного из них входит в

другое. Это достигается удалением правила с лишним источником и

объединением его приемников с приемниками оставшегося правила.

Если в качестве источника копирующего правила используется константная

функция, являющаяся процедурой-функцией без параметров (например,

функция GETNEW из АТ-грамматики), то такие атрибутные правила

объединять нельзя, т. к. два разных вызова функции без параметров могут

давать разные значения.

**51. Представление в магазине и вычисление унаследованных и синтезированных атрибутов символов грамматики в S-Атрибутном ДМП-процессоре.**

В S-атрибутном ДМП-процессоре каждый магазинный символ - объект

для представления атрибутов.

!!!заменить на объект

Аналогично L-атрибутному ДМП-процессору, примем, что магазинный

символ с n атрибутами представляется в магазине (п + 1)-ой ячейками, верхняя

из которых содержит имя символа, а остальные - поля для атрибутов. Поля

магазинного символа, предназначенные для атрибутов, заполняются значениями

атрибутов в момент вталкивания символа в магазин и не изменяются до

момента выталкивания его из магазина.

В S-атрибутном ДМП-процессоре **операция переноса** расширяется таким

образом, что значения атрибутов переносимого входного символа помещаются в

соответствующие поля вталкиваемого при переносе магазинного символа.

При выполнении **операции свертки** для правила с номером i верхние

символы магазина представляют собой правую часть i-го правила вывода

входной грамматики, а поля магазинных символов содержат значения атрибутов

соответствующих символов грамматики.

Расширенная операция свертки использует эти значения для вычисления

значений всех атрибутов операционных символов, связанных с правилом

вывода транслирующей грамматики, и значений всех атрибутов нетерминала из

левой части правила. Значения атрибутов операционных символов

используются для выдачи результатов в выходную ленту или выполнения

других действий, определяемых этими символами. Атрибуты нетерминала из

левой части правила вывода записываются в соответствующие поля

магазинного символа, который соответствует этому нетерминалу и вталкивается

в магазин во время свертки.

**52. Случай АТ-грамматика с формой простого присваивания.**

**Пример.** Пусть дана L-атрибутная транслирующая грамматика, порождающая

префиксные арифметические выражения над константами, в форме простого

присваивания. Атрибутные правила вывода этой грамматики имеют следующий

вид: Ep - синтезированный р , {ответ}r - унаследованный r:

G = ({S, Ep}, {cr, \*, +}, {{ответ}r , P, S}), где

P = {

1. S → Ep{ответ}r

r ← p

2. Ep → +Eq Er {сложить}A,B,R

p ← q+ r

3. Ep → \*Eq Er

p ← q\* r

4. Ep → cr

p ← r

}

**53. Приведите процедуру преобразования произвольной АТ-грамматики в форму простого присваивания.**

**Алгоритм преобразования** произвольной L-атрибутной транслирующей

грамматики в эквивалентную L-атрибутную грамматику в форме простого

присваивания.

Вход: KС грамматика G = (T, V, P, S0)

Выход: эквивалентная КС грамматика G' = (T, V, P', S0)

foreach (Ap → Xa1,..., Xan, {OP1,..., OPak }, p <- F{f1(a1,...,an, ak),...,fn(a1,...,an, ak)})

foreach (f F)

a = ∅ пустое множество атрибутов

if (f ≠ fo ) не копирующее, fo - копирующая функция

foreach (aj {a1,...,an, ak})

if(aj ∉ T)

F = F ∪ f0(aj', aj)

a = a ∪ aj'

end if

end foreach

F = F ∪ f0(a1, a1'\_ans) - копирующая функция

a = a ∪ a1'\_ans

F = F – f //удалить из правил правило f

OP1 ,..., OPak = OP1 ,..., OPak ∪ Opa добавить новый операционный символ

end if

end foreach

end foreach

Входными символами грамматики являются: лексема с, представляющая

собой целочисленную константу, и знаки арифметических операций: «+» и «\*».

Входной символ с имеет один атрибут, значением которого является значение

константы. Нетерминальный символ Е и операционный символ {ОТВЕТ} также

имеют по одному атрибуту. Значением синтезированного атрибута символа Е

является значение подвыражения, порождаемого этим символом, а значением

унаследованного атрибута символа {ОТВЕТ} — значение всего выражения,

порождаемого грамматикой.

Исходная грамматика содержит два некопирующих правила: р ← q + r и р ←

q\* r, правые части которых представляют собой функции сложения и

умножения соответственно. Для преобразования заданной грамматики в форму

простого присваивания введем операционные символы {СЛОЖИТЬ}A, B, R и

{УМНОЖИТЬ}A, B, R каждый из которых имеет по два унаследованных

атрибута А и В и один синтезированный атрибут R. Для операционного символа

{СЛОЖИТЬ}A, B, R атрибутное правило имеет вид: R ← A + В, а для

операционного символа {УМНОЖИТЬ}A, B, R — R ←А X В.

Для того чтобы преобразованная грамматика также была L-атрибутной,

символ {СЛОЖИТЬ} необходимо поместить правее всех символов правой части

правила вывода (2), т. к. одним из аргументов сложения является атрибут самого

правого символа Е. Атрибут, получающий в качестве своего значения результат

сложения, в определении места расположения символа {СЛОЖИТЬ} не

участвует, т. к. он не приписан ни к одному из символов правой части.

Аналогичные рассуждения относительно операционного символа

{УМНОЖИТЬ} определяют крайнюю правую позицию правой части правила

(3), как единственно возможное место расположения этого символа.

Полученная в результате преобразования L-атрибутная грамматика в форме

простого присваивания имеет вид:

Ер синтезированный р

{ОТВЕТ}r, унаследованный r

{СЛОЖИТЬ}A, B, R унаследованный А, В

R ← А + В синтезированный R

{УМНОЖИТЬ}A, B, R унаследованный А, В

R ←А\* В синтезированный R

S →Е {ОТВЕТ}r 𝑝

r ← р

Е → + {СЛОЖИТЬ}A, B, R 𝑝 Е𝑞 Е𝑝

А← q

В ←r

Р ←R

Е → \* {УМНОЖИТЬ}A, B, R 𝑝 Е𝑞 Е𝑝

А←q

В← r

p ←R

Е → 𝑝 𝑐𝑟

р←r

Эта грамматика порождает те же входные цепочки и значение

синтезированного атрибута нетерминала Е, что и исходная грамматика. Однако

формально она не определяет того же самого перевода, т. к. преобразованные

правила (2) и (3) удлиняют активную цепочку, включив в нее действия,

обеспечивающие вычисление функций сложения и умножения соответственно.

Для того чтобы преобразованная грамматика определяла тот же перевод, что

и исходная, введенные в процессе преобразования операционные символы не

следует выдавать в выходную цепочку.

**54. Преобразование деревьев под управлением СУ-схем**

**Алгоритм 6.1** Преобразование деревьев при помощи СУ-схемы

**Вход:** СУ-схема Тb = (V, Σ, Δ, Pi, S), входная грамматика Gi= (V, Σ, Рi, S),

выходная грамматика Gj = (V, Δ, Рj, S) и дерево вывода D в Gi, с кроной,

принадлежавшей Σ\*.

**Выход:** Дерево вывода D' в G0, такое, что если х и у - кроны деревьев D и D'

соответственно, то (х, у) ∈ τ(Tb).

Шаг 1. Применяется к внутренней вершине n дерева D, имеющей k прямых

потомков n1,...,nk.

1.1 Устранить из множества вершин n1,...,nk листья, помеченные

терминальными символами или ε.

1.2 Пусть A→α - правило входной грамматики Gi, соответствующее

вершине n и ее прямым потомкам, т. е. А - метка вершины n, и α образуется

конкатенацией меток вершин n1,..., nk. Выбрать из Pi некоторое правило вида А→ α, β (если таких правил несколько, то выбор произволен).

Переставить оставшиеся прямые потомки вершины n (если они есть)

согласно соответствию между вхождениями не терминалов в α и β.

(Поддеревья, корнями которых служат эти потомки, переставляются вместе с

ними).

1.3 Добавить в качестве прямых потомков вершины и листья с метками так,

чтобы метки всех ее прямых потомков образовали цепочку β.

1.4 Применить шаг 1 к прямым потомкам вершины n, не являющимися

листьями, в порядке слева направо.

2. Повторять шаг 1 рекурсивно, начиная с корня дерева D.

Результат дерево D'.

**55.Приведите алгоритм построения управляющей таблицы для транслирующей грамматики цепочного перевода, входной грамматикой которого является LL(1)-грамматика.**

**Алгоритм** построения управляющей таблицы для транслирующей

грамматики цепочечного перевода, входной грамматикой которой является

LL(1)-грамматика.

Вход: Транслирующая грамматика цепочечного перевода GTb = (T, Σi, Σa, Р, S),

входная грамматика которой является LL(1)- грамматикa.

Выход : Корректная управляющая таблица М для грамматики GTb.

Описание алгоритма:

Управляющая таблица М определяется на множестве (N ∪ Σi ∪ Σa ∪{⊥}) X

Σi ∪ {ε}) по следующим правилам:

Если А→yβ - правило вывода грамматики GT, где у ∈ Σa\* , а у — либо ε,

либо цепочка, начинающаяся с терминала или нетерминала, то М{А, а) = (β, у)

для всех а≠ ε, принадлежащих множеству FIRST(β).

Если ε ∈ FIRST(β), то М(А, b) = (β, у) для всех b ∈ FOLLOW (A).

Заметим, что при вычислении FIRST(β) операционные символы, входящие в

цепочку β, вычеркиваются.

М(Х, а) = ВЫДАЧА(Х) для всех x ∈ Σa И a ∈ Σi ∪ {ε}.

М(а, а) = ВЫБРОС для всех а ∈ Σi .

M (⊥ , ε) = допуск

В остальных случаях М(Х, а) = ОШИБКА для X ∈ (N ∪ Σi ∪ Σa ∪ {⊥}) и а ∈

Σi ∪{ ε }.

Построим управляющую таблицу для транслирующей грамматики,

описывающей перевод инфиксных арифметических выражений в ПОЛИЗ. Эта

транслирующая грамматика получена из входной LL(1)-грамматики G1 путем

включения в нее операционных символов {i}, {+} и {\*}:

E →E ′ (5)T′ →\* P{\*} T′

Е ′ →+ T{+} Е′6)T′ →ε

Е ′ →ε

Р →i {i}

Т →РT′

202

(8)P →(Е)

Управляющая таблица должна содержать 14 строк, помеченных символами из

множества N∪Σi ∪ Σa ∪ {⊥}, и 6 столбцов, помеченных символами из

множества Σi ∪{ ε }.

Построение управляющей таблицы для строк, отмеченных символами из

множества N∪Σi ∪ {⊥}, ничем не отличается от построения таблицы для

соответствующей входной LL(1)-грамматики (табл. 1.1), а строки управляющей

таблицы, отмеченные операционными символами, содержат значения

ВЫДАЧА{{Х}), где X ∈{{i}, {+},{\*}}. Заметим, что элементы таблицы,

соответствующие строкам, помеченным операционными символами, для всех

столбцов одинаковые, т. к. действия, выполняемые ДМП-процессором в случае,

когда верхним символом магазина является операционный символ, не зависят

от входного символа.

Таблица 1.1. Управляющая таблица М для транслирующей грамматики,

описывающей перевод инфиксных арифметических выражений в ПОЛИЗ

i

( ) + \* ε

E ТЕ’

,ε

ТЕ’,ε

E’ ε, ε \*P{\*} T’, ε ε, ε

T PЕ’,

ε

PЕ’,ε

T’ ε, ε ε, ε \*P{\*} T’,

ε

ε, ε

P i{i},

ε

(E),ε

i ВЫБ

РОС

( ВЫБРОС

) ВЫБРОС

+ ВЫБРОС

\* ВЫБРОС

⊥ ДОПУСК

{i} ВЫДАЧА({i})

{+} ВЫДАЧА({+})

{\*} ВЫДАЧА({\*})

Начальное содержимое магазина - Е⊥

Начальное содержимое магазина — E⊥

Для входной цепочки i +i \* i ДМП-процессор проделает следующую

последовательность тактов:

(ct203)

|-(i +i \* i , E⊥, ε) (i +i \* i , TE'⊥, ε)

|- (i +i \* i , PT'E'⊥, ε)

|- (i +i \* i , {I} T'E'⊥, ε

|- (+ i \* i , {i}T'E'⊥, ε)

|- (+ i \* i , {i}T'E'⊥, {i})

|- (+i \* i , {i}E'⊥, {i})

|- (+i \* i , +T {+} E'⊥,{i})

|- (i \* i , T {+} E'⊥,{i})

|- (i \* i , PT '{+} E'⊥, {i})

|- (i \* i , i {i} T '{+} E'⊥, {i})

|- (\*i , {i} T '{+} E'⊥, {i})

|- (\* i , T '{+} E'⊥, {i} {i})

|- (\* i , \*P{\*}T'{+}E'⊥, {i} {i})

|- (i , P{\*}T'{+}E'⊥, {i} {i})

|- ( i , i {i} {\*}T'{+}E'⊥, {i} {i})

|- (ε , {i}{\*}T'{+}E'⊥, {i} {i})

|- (ε , {i}{\*}T'{+}E'⊥, {i} {i})

|- (ε , T'{+}E'⊥, {i} {i}{i}{\*})

|- (ε , {+}E'⊥, {i} {i}{i}{\*})

|- (ε , E'⊥, {i} {i}{i}{\*}{+})

|- (ε , ⊥, {i} {i}{i}{\*}{+})

ДМП-процессор можно использовать в качестве базового процессора для

других видов синтаксически управляемого перевода, если операцию выдачи

операционного символа в выходную ленту заменить операциями вызова

соответствующих семантических процедур.

**56. Определите транслирующую грамматику.**

**Определение.** Транслирующей грамматикой называется пятерка объектов

GT **=** (V, Σi, Σa, Р, S)**,** где Σi - словарь входных символов, Σa - словарь

операционных символов, V - нетерминальный словарь, S ϵ V - начальный

символ транслирующей грамматики, P - конечное множество правил вывода

вида A → α, в которых A принадлежит V, а α принадлежит ( Σi Σa V )\*.

Транслирующая грамматика - это КС-грамматика, множество терминалов

которой разбито на два множества: множество входных и множество

операционных символов.

**57. Опишите, каким образом представляются в магазине и вычисляются унаследованные и синтезированные атрибуты символов грамматики в L-атрибутном ДМП-процессоре.**

Начальная конфигурация. В магазине находится маркер дна и начальный

символ грамматики. Поля начального символа грамматики, соответствующие

унаследованным атрибутам, заполняются начальными значениями атрибутов,

которые задаются L-атрибутной транслирующей грамматикой, а в поля,

соответствующие синтезированным атрибутам, заносятся пустые указатели,

которые служат маркерами конца списков.

М(Х, а) = ВЫБРОС (символ в верхушке магазина совпадает с текущим

входным символом). В этом случае каждое поле верхнего магазинного символа

содержит указатель на список полей магазина, в которых требуется поместить

значение атрибута текущего входного символа. Операция ВЫБРОС

расширяется таким образом, что каждый атрибут текущего входного символа

копируется во все поля списка на который указывает соответствующее поле

верхнего магазинного символа.

М(Х, а) = ВЫДАЧА(Х) (в верхушке магазина находится операционный

символ). Операция ВЫДАЧА(Х) ДМП-процессора расширяется следующим

образом:

унаследованных атрибутов извлекаются из соответствующих полей верхнего

магазинного символа и используются затем при выдаче символа в выходную

цепочку. При этом следует помнить, что если операционный символ появился в

результате преобразования исходной атрибутной транслирующей грамматики в

форму простого присваивания, то такой символ в выходную цепочку не

выдается;

значение синтезированных атрибутов вычисляются по правилам вычисления

атрибутов, связанным с данным операционным символом, после чего значение

каждого синтезированного атрибута помещается во все поля списка, на который

указывает соответствующее поле символа из верхушки магазина.

М(Х, а) = (β, у) (в верхушке магазина находится нетерминал). В этом случае

L-атрибутный ДМП-процессор вталкивает в магазин цепочку символов β из

правой части распознаваемого правила B DSLFTN WTGJXRE операционных

символов у. При этом вычисляются атрибуты операционных символов, которые

не вталкиваются в магазин, и заполняются поля атрибутов магазинных

символов и символов цепочки β, вталкиваемых в магазин.

Источниками атрибутных правил, связанных с правилами вывода L-

атрибутной транслирующей грамматики в форме простого присваивания, могут

быть только константы, унаследованные атрибуты нетерминалов из левой части

правил вывода, атрибуты входных и операционных символов, синтезированные

205

атрибуты нетерминалов из правой части правил вывода. В табл. 1.2 приведены

значения источников копирующих правил в момент времени, когда верхним

символом магазина является нетерминалы унаследованные атрибуты символов

из правой части правил вывода и унаследованные атрибуты символов из правой

части правил вывода. В табл. 1.3 приведены поля магазина, соответствующие

приемникам атрибутных правил, которые необходимо заполнить во время

перехода L-атрибутного ДМП-процессора при М{Х, а) = (β, у).

**58. Какие СУ-схемы называются простыми.**

**Определение.** СУ-схема Тb = (V, Σ, Δ, P, S) называется простой, если для

каждого правила A → α, β принадлежит P соответствующие друг другу вхождения не терминалов встречаются в α и β в одном и том же порядке. Перевод, определяемый *простой* СУ-схемой, называется *простым* синтаксически управляемым переводом. Соответствие нетерминалов в выводимой паре простой СУ-схемы определяется порядком, в котором эти нетерминалы появляются в цепочках, поэтому для нее легко построить транслятор, представляющий собой преобразователь с магазинной памятью

**59. Приведите процедуру преобразования ДМП-процессора в L-атрибутный ДМП-процессор.**

Процедура преобразования ДМП-процессора в атрибутный ДМП-процессор, которая описывается в данном пособии, требует, чтобы **АТ-грамматика, определяющая перевод, имела форму простого присваивания.**

Рассмотрим построение L-атрибутного ДМП-процессора, реализующего перевод, определяемый L-атрибутной транслирующей грамматикой в форме простого присваивания, входной грамматикой которого является LL(1)- грамматика.

**КР** L-атрибутный ДМП-процессор  **:**  Сначала построим ДМП-процессор, реализующий цепочечный перевод, описываемый транслирующей грамматикой цепочечного перевода, которая получается из заданной L-атрибутной транслирующей грамматики после удаления из нее всех атрибутов. Затем расширим полученный таким образом ДМП- процессор, включив для каждого магазинного символа множество полей для представления атрибутов символа и дополнив управляющую таблицу действиями по вычислению атрибутов и записи их в соответствующие поля

**60. Определите активную цепочку, ее входную и операционную части.**

**Определение.** Входную часть активной цепочки называют **входной цепочкой**, операционную часть - **выходной цепочкой**, а транслирующую грамматику G0T - грамматикой цепочечного перевода если при реализации появление входного символа в активной цепочке интерпретируется как операция чтения этого символа считывающим устройством, а вхождению операционного символа в правило вывода сопоставляется операция выдачи символа, заключённого в фигурные скобки.

Грамматика G0T является примером грамматики цепочечного перевода:

1. E → E + Т {+}

2. E → Т

4. Т → Р

3. Т → Т \* Р {\*}

5. Р → i {i}

6. Р → (E)

Операционные символы интерпретируются именами семантических процедур, вызываемых при обработке входной цепочки. Активная цепочка задает последовательность наступления событий вызовов этих процедур по отношению к событиям чтения входных символов.

**61. Как программируются процедуры в методе рекурсивного спуска, реализующего перевод, описываемый L-атрибутной транслирующей грамматикой.**

**Пример 7.1.** Рассмотрим L-атрибутную транслирующую грамматику в качестве примера реализации атрибутного перевода с использованием метода рекурсивного спуска.

Для повышения наглядности переименуем атрибуты символов грамматики таким образом, чтобы всем вхождениям символов в правые части разных правил вывода соответствовали разные имена атрибутов, а также выберем одинаковые имена для атрибутов нетерминала R из левой части правил вывода с номерами (3), (4) и (5). Пусть р — унаследованный атрибут нетерминала R, a t — его синтезированный атрибут. После преобразования получим следующую грамматику:

Et синтезированный t

Rp, t унаследованный р синтезированный t

Атрибуты операционных символов унаследованные.

(0) → S ⊥

(1) S → := {:=}a, b

(2) →

(3) → {+}p, , r1

r1 ← GETNEW

(4) → X {\*}p, , , r1

r2 ← GETNEW

(5) → ε

t←p

**62. Определите входную и выходную грамматики транслирующей грамматики**

**Определение.** G0 называется входной грамматикой для транслирующей

грамматики G0

T, если она получена из транслирующей грамматики путем

вычеркивания всех операционных символов. L(G0) называется входным языком.

Про выходную грамматику не нашла

**Определение.** Входную часть активной цепочки называют **входной**

**цепочкой**, операционную часть - **выходной цепочкой**, а транслирующую

грамматику G0

T - грамматикой цепочечного перевода если при реализации

появление входного символа в активной цепочке интерпретируется как

операция чтения этого символа считывающим устройством, а вхождению

операционного символа в правило вывода сопоставляется операция выдачи

символа, заключённого в фигурные скобки.

**63. Как программируются процедуры в методе рекурсивного спуска, реализующие перевод, описываемый транслирующей грамматикой.**

Сначала рассмотрим, каким образом можно изменить процедуры для распознавания цепочек, порождаемых нетерминальными символами грамматики, для реализации перевода, описываемого транслирующей грамматикой цепочечного перевода.

В этом случае правила составления процедур дополняются следующим правилом: если текущим символом правой части правила вывода грамматики является операционный символ Y, ему соответствует вызов процедуры записи операционного символа в выходную цепочку output (Y).

В качестве примера реализации перевода с использованием метода рекурсивного спуска рассмотрим транслирующую грамматику, описывающую перевод инфиксных арифметических выражений в польскую инверсную запись. Эта грамматика построена на основе входной грамматики G1 и имеет следующие правила:

S → E⊥

E → TE'

E' →+T{+} E'│ε

T → PT '

T'→\*P{\*} T'│ ε

P → (E)│i {i}

**64. Как граф зависимостей используется при вычислении атрибутов**

По определению, атрибут можно вычислить, если известны значения всех атрибутов, от которых зависит его значение. Число деревьев вывода так же, как и входных цепочек, бесконечно, поэтому важно по самой грамматике уметь определять, является ли множество её правил вычисления атрибутов корректным.

Алгоритмы проверки корректности АТ-Грамматики приведены в [2, 25]. Они основаны на построении графа зависимостей и еuо анализе.

Узлами графа зависимостей служат атрибуты, которые нужно вычислить, а дугам ставятся в соответствие зависимости, определяющие, какие атрибуты вычисляются раньше, а какие позже.

Граф зависимостей R(D) представляет собой ориентированный Граф, кото- рый строится для некоторого дерева вывода D следvующим образом:

* узлами R(D) являются пары (Х, а), где Х - узел дерева D, а *а* - атрибут символа, служащего меткой узла Х;
* дуга из узла (, ) в узел (, ) проводится, если семантическое правило, вычисляющее значение атрибута , непосредственно использует значение атрибута .

Вычисление значения двоичного числа по его символьному представлению можно задать при помощи следующей атрибутной грамматики [24]:

→ . →

← + ←

← 0 ←

← - ←1

→ → 0

← ← 0

← 0

→ → 1

← + ← 1

←

← + 1

← + 1

В этой грамматике атрибуты имеют следующую семантику: v - значение (рациональное число), s - масштаб (целое число) и - длина числа (целое число).

Используя грамматику, построим дерево D вывода цепочки 101.01

Для построения узлов графа зависимостей необходимо последовательно рассмотреть вершины дерева D и для каждой из них построить столько узлов, сколько атрибутов имеет символ, которым помечена эта вершина.

Например, для корня дерева, обозначенного символом N с одним атрибуттом, нужно в граф включить один узел, поместив его парой (N, ).

Для определения дуг графа необходимо использовать семантические правила. Например, рассмотрим узел Дерева зависимостей (N, ). При построении дерева D непосредственные потомки корня N определялись правилом грамматики → . . Атрибут не терминала N зависит, в соответствии с семантическим правилом ← + , от атрибута левого сына, помеченного символом L, и атрибута правого сына. Поэтому в граф зависимостей необходимо включить дуги ((L, ), (N, )) и ((L, ), (N, )). Поступая аналогичным образом, мы можем построить граф зависимостей.

**65. С какой целью осуществляется переименование имен атрбутов нетерминальных символов из левых частей правил вывода L-атрибутной транслирующей грамматики при реализации вывода L-атрибутного перевода методом рекурсивного спуска.**

**66. Как связаны вывод цепочки и дерево вывода активной цепочки-**

(из инета) Вывод – это процесс порождения предложений языка на основе правил грамматики.

Цепочка β наз-ся непосредственно выводимой из цепочки α, грамматики G(VT,VN,P,S), где V=VT U VN, γ € V\*

β=δ1 γ δ2

α= δ1 ω δ2

δ1,δ2€ V\*, ω € V+

если во мн-ве пр-я грамматики существует правило ω→γ∈Р

ω→ γ € P, α => β (непосредственная выводимость)

Цепочка вывода – это последовательность непосредственно выводимых цепочек. Каждый переход от одной непосредственно выводимой цепочки к следующей называется шагом вывода.

Деревом вывода – граф, которое соответствует некоторой цепочке вывода и удовлетворяет следующим условиям:

- каждая вершина дерева обозначается символом грамматики;

- корнем дерева всегда является целевой символ грамматики;

- листьями дерева всегда являются терминальные символы грамматики;

- если некоторый узел обозначен нетерминальным символом A, а связанные с ним узлы – символами b1,b2,…,bn, то в грамматике G существует правило A-> b1,b2,…,bn.

Дерево вывода в указанном виде всегда можно построить только для КС-грамматик и регулярных грамматик.

**67. С какой целью осуществляется переименование имен атрбутов нетерминальных символов из левых частей правил вывода L-атрибутной транслирующей грамматики при реализации вывода L-атрибутного перевода методом рекурсивного спуска**

(смотреть вопрос 65)

**68. Определите атрибутную транслирующую грамматику.**

Атрибутная транслирующая грамматика - это транслирующая грамматика,

обладающая следующими дополнительными свойствами:

1. Каждый символ грамматики (входной, нетерминальный и операционной)

имеет конечное множество атрибутов, и каждый атрибут имеет (возможно,

бесконечное) множество допустимых значений.

2. Все атрибуты нетерминальных и операционных символов делятся на

синтезированные и унаследованные.

3.Унаследованные атрибуты подчиняются следующим правилам:

• каждому вхождению унаследованного атрибута в правую часть правела

вывода сопоставляется правило, позволяющее вычислить значение этого

атрибута как функцию некоторых других атрибутов символов, входящих в

левую или правую часть данного правила;

• для каждого унаследованного атрибута начального символа грамматики

задаётся начальное значение.

4.Правила вычисления значений синтезированных атрибутов определяются

следующим образом:

• каждому вхождению синтезированного атрибута нетерминального символа

в левую Часть правила вывода сопоставляется правило, позволяющее

вычислить значение этого атрибута как функцию некоторых других атрибутов

символов, входящих в левую или правую часть данного правила;

• каждому синтезированному атрибуту операционного символа

сопоставляется правило, позволяющее вычислить значение этого атрибута как

функцию некоторых других атрибутов данного символа.

• каждому синтезированному атрибуту операционного символа

сопоставляется правило, позволяющее вычислить значение этого атрибута как

функцию некоторых других атрибутов данного символа.

Атрибуты записываются в виде индексов соответствующих символов АТ-

грамматики, при этом для каждого атрибута указывается его тип. Например:

Xa,b синтезированный a, унаследованный b

Правила вычисления атрибутов записываются в виде операторов

присваивания, левая часть которых - атрибут или список атрибутов, а правая

часть - функция.

( Это не то но также есть в методичке

Транслирующая грамматика описывает перевод только той части символа,

который задает его тип. Расширением понятия грамматики цепочечного

перевода, использующей при переводе оба компонента входного символа

является **атриибитно транслирующая грамматики (АТ-грамматики)**.

АТ-грамматики были предложены Дональдом Кнутом

В АТ-грамматике символы снабжаются **атрибутами**, которые могут

принимать значения из некоторого множества допустимых значений и

**интерпретируются** как **семантическая** информация, связанная с конкретным

вхождением символа в правило вывода грамматики.

При таком подходе значения атрибутов связываются с вершинами дерева

вывода в АТ-грамматике, а правила вычисления значений атрибутов

сопоставляются правилам вывода грамматики.

**Атрибуты нетерминальных и операционных символов делятся на**

**синтезированные и унаследованные атрибуты.**

**)**

**69. Опишите, каким образом представляются в магазине и вычисляются унаследованные и синтезированные атрибуты символов грамматики в L-атрибутном ДМП-процессоре**

(смотреть вопрос 57)

**70. Какие атрибуты называются унаследованными, синтезированными.**

Все атрибуты нетерминальных и операционных символов делятся на синтезированные и унаследованные

Унаследованные атрибуты подчиняются следующим правилам:

• каждому вхождению унаследованного атрибута в правую часть правела вывода сопоставляется правило, позволяющее вычислить значение этого атрибута как функцию некоторых других атрибутов символов, входящих в левую или правую часть данного правила;

• для каждого унаследованного атрибута начального символа грамматики задаётся начальное значение.

Правила вычисления значений синтезированных атрибутов определяются следующим образом:

• каждому вхождению синтезированного атрибута нетерминального символа в левую Часть правила вывода сопоставляется правило, позволяющее вычислить значение этого атрибута как функцию некоторых других атрибутов символов, входящих в левую или правую часть данного правила;

• каждому синтезированному атрибуту операционного символа сопоставляется правило, позволяющее вычислить значение этого атрибута как функцию некоторых других атрибутов данного символа.

• каждому синтезированному атрибуту операционного символа сопоставляется правило, позволяющее вычислить значение этого атрибута как функцию некоторых других атрибутов данного символа.

Атрибуты записываются в виде индексов соответствующих символов АТ- грамматики, при этом для каждого атрибута указывается его тип. Например:

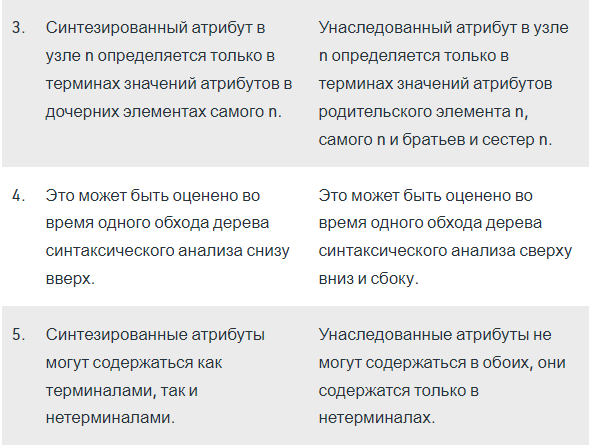
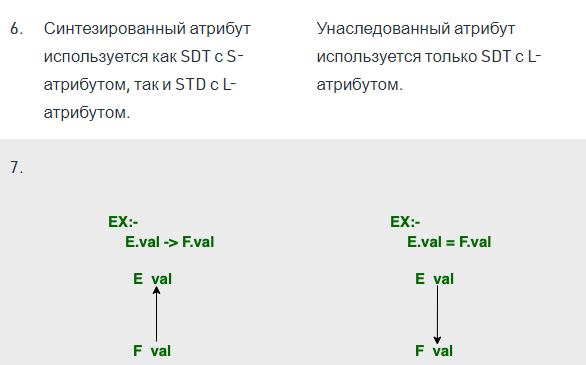
Xa,b синтезированный a, унаследованный b

**71. С какой целью осуществляется переименование имен атрбутов нетерминальных символов из левых частей правил вывода L-атрибутной транслирующей грамматики при реализации вывода L-атрибутного перевода методом рекурсивного спуска.**

(смотреть вопрос 67)

**72. Определите различия между унаследованными и синтезированными атрибутами**

(из инета) 

**73.Определите представление в магазине и вычисление унаследованных и синтезированных атрибутов символов грамматики в S-Атрибутном ДМП-процессоре.**

(смотреть вопрос 51)

**74. Определите правило вычисления атрибутов.**

**Определение. П**равило вычисления атрибутов называется **копирующим правилом** тогда и только тогда, когда левая часть правила - это атрибут или список атрибутов, а правая часть – константа или атрибут. Правая часть называется источником копирующего правила, а каждый атрибут из левой части – приемником копирующего правила.

Если источники нескольких копирующих правил совпадают, то их приемники можно объединить в одну левую часть. Например, правила z, w← a и х, у←z можно записать в виде: х, у, z, w← a , т. к. источнику второго правила z, согласно первому правилу, присваивается значение а. Аналогично х←у и a, b ← y можно записать как а, b, х ←у.

**75. Приведите алгоритм построения управляющей таблицы для транслирующей грамматики цепочного перевода, входной грамматикой которого является LL(1)-грамматика**

(смотреть вопрос 55)

**76. Как строится граф зависимостей.**

Граф зависимостей R(D) представляет собой ориентированный Граф, кото- рый строится для некоторого дерева вывода D следvующим образом:

* узлами R(D) являются пары (Х, а), где Х - узел дерева D, а *а* - атрибут символа, служащего меткой узла Х;
* дуга из узла (, ) в узел (, ) проводится, если семантическое правило, вычисляющее значение атрибута , непосредственно использует значение атрибута .

**77. Случай АТ-грамматика с формой простого присваивания.**

(смотреть вопрос 52)

**78. Типы ошибок при тестировании АТ-грамматики.**

**(из инета)Тестирование АТ-грамматики.**Тестирование АТ- грамматики заключается в моделировании работы построенного по ней процессора, выполняющего перевод. При большом числе тестов и/или большой их длине эта работа может быть выполнена только при использовании соответствующих средств автоматизации. Для простых языковых конструкций, когда длина теста и их число невелики, а тестирование выполняется отдельно для каждой конструкции, начиная с простейших, работа эта не только необходима, но и реально выполнима.

Тестирование АТ-грамматики позволяет определять два вида ошибок:

* ошибки в структуре выходной цепочки (неверный порядок следования символов в выходной цепочке);
* ошибки в значениях символов выходной цепочки, что связано с ошибками в передаче значений атрибутов.

При тестировании реального описания перевода желательно использовать комплексные тесты, определяющие оба вида ошибок.

**79. Приведите процедуру преобразования ДМП-процессора в L-атрибутный ДМП-процессор.**

L-атрибутный ДМП-процессор  **:**  Сначала построим ДМП-процессор, реализующий цепочечный перевод, описываемый транслирующей грамматикой цепочечного перевода, которая получается из заданной L-атрибутной транслирующей грамматики после удаления из нее всех атрибутов. Затем расширим полученный таким образом ДМП- процессор, включив для каждого магазинного символа множество полей для представления атрибутов символа и дополнив управляющую таблицу действиями по вычислению атрибутов и записи их в соответствующие поля.

**80. Этапы построения транслирующей грамматики**

**Алгоритм** трансформации КС-грамматики G0 в транслирующую грамматику.

Вход : КС-грамматиа G0

Выход : G0

T - транслирующая грамматика

1. Входной язык L описывается входной КС-грамматикой G0.

2. В правила вывода G0 грамматики вставляются операционные символы для

описания семантических действий, связанных с соответствующими правилами.

В восходящих методах обработки языков широко применяются постфиксные

транслирующие грамматики.

**81. Определите представление в магазине и вычисление унаследованных и синтезированных атрибутов символов грамматики в S-Атрибутном ДМП-процессоре.**

(см вопрос 51)

**82. Виды объектов включаемые в выходную цепочку при переводе.**

Для перевода могут быть применены два метода:

1. схема трансляции — описание грамматики с встроенными

семантическими операциями A → B + C {+};

2. семантические правила с атрибутами: грамматика может быть применена

для выполнения семантических правил над атрибутами символов.

правило продукций семантические правила

A→ B + C A.val → B.val + C.val

Атрибуты могут включать в себя тип переменной, значение выражения, и т.п.

Для символа A с атрибутом val обращение к атрибуту выглядит как A.val.

2. применение автомата, распознающего входную цепочку и выдающего на

выходе перевод этой цепочки.

Перейдём к рассмотрению простейших способов описания перевода.

**Определение.** Пусть Σ — входной алфавит и Δ — выходной алфавит.

Переводом с языка L1 ⊆ Σ\* на язык L2 ⊆ Δ\* называется отношение τ из Σ\* в Δ\* ,

для которого L1 — область определения, а L2 — множество значений.

Если (х, у) ϵ τ, то цепочка у называется выходом для цепочки х. В общем

случае в переводе τ для одной входной цепочки может быть задано более одной выходной цепочки. Как программа то выход один.

**83. Определите L-атрибутную и S-атрибутную транслирующие грамматики.**

**(Смотреть 46)**

L-атрибутные и S-атрибутные транслирующие грамматики - два подкласса корректных АТ-грамматик, часто используемых при проектировании языковых процессоров

**Определение.** АТ-грамматика называется L-атрибутной тогда и только тогда, когда выполняются следующие условия:

1. Аргументами правила вычисления значения унаследованного атрибута символа из правой части правила вывода могут быть только унаследованные атрибуты символа из левой части и произвольные атрибуты символов из правой части, расположенные левее рассматриваемого символа.
2. Аргументами правила вычисления значения синтезированного атрибута символа из левой части правила вывода являются унаследованные атрибуты этого символа или произвольные атрибуты символов из правой части.
3. Аргументами правила вычисления значения синтезированного атрибута операционного символа могут быть только унаследованные атрибуты этого символа.

**Определение:** АТ-грамматика называется S-атрибутной тогда и только тогда, когда она является L-атрибутной и все атрибуты нетерминалов синтезированные.

Ограничения, накладываемые на L-атрибутную транслирующую грамматику, позволяют вычислять значения атрибутов в процессе нисходящего анализа входной цепочки. Нисходящий детерминированный анализатор для LL(1)-грамматик требует, чтобы L-атрибутная транслирующая грамматика, описывающая перевод, имела форму простого присваивания.