Языки программирования и методы трансляции

Лекция 8.

Распознающие автоматы для контекстно-свободных грамматик

Распознающие автоматы

Линейная последовательность символов из Σ. За один такт считывается один символ.

Множество состояний и отображение, описывающее переход к следующему состоянию в зависимости от текущего входного символа и данных из вспомогательной памяти

Поддерживает функции доступа к памяти и преобразования памяти (например, замена символа в вершине стека на цепочку).

Распознаватель допускает входную цепочку w, если, начиная с начальной конфигурации, в которой w записана во входной очереди, распознаватель может выполнить последовательность тактов, завершающуюся конечной конфигурацией.

Конечный автомат

Входная очередь

↓

Устройство управления с конечной памятью

Недетерминированный конечный автомат – пятерка объектов: $K=(Q, \Sigma, \delta, q_0, F)$

Q – множество состояний УУ,
Σ – алфавит входных символов,
$\delta: Q\times \Sigma\rightarrow P(Q)$ – функция переходов,
q_0 – начальное состояние,
F – множество заключительных состояний

Конечные преобразователи

Входная очередь

↓

Устройство управления с конечной памятью

↓

Выходная очередь

Недетерминированный конечный преобразователь – шестерка объектов:
$K=(Q, \Sigma, \Delta, \delta, q_0, F)$
Q – множество состояний УУ,
Σ – алфавит входных символов,
Δ – алфавит выходных символов
$\delta: Q\times \Sigma \rightarrow P(Q) \times \Delta$ – функция переходов,
q_0 – начальное состояние,
F – множество заключительных состояний

Конфигурация конечного преобразователя - тройка (q, ax, y), где q – состояние, x – необработанная часть входной цепочки, y – построенная часть выходной цепочки.

Такт работы конечного преобразователя – переход от конфигурации (q, ax, y) к конфигурации (r, x, yz), если δ(q,a) содержит (r, z).

Регулярный перевод

Формализация такта КП как отношения $\vdash$ на множестве конфигураций:

Для всех $q \in Q$, $a \in \Sigma$, $x \in \Sigma$ и $y \in \Delta$, таких, что $\delta(q, a)$ содержит (r, z), конфигурации $(q, ax, y) \vdash (r, x, yz)$.

Транзитивное замыкание отношения $\vdash$ обозначим $\vdash^+$

Рефлексивно-транзитивное замыкание отношения $\vdash$ обозначим $\vdash^*$

Цепочка $y \in \Delta$ называется выходом для цепочки $x \in \Sigma$, если $(q0, x, \varepsilon) \vdash^* (q, \varepsilon, y)$

Переводом, определяемым конечным преобразователем, называется множество пар цепочек:

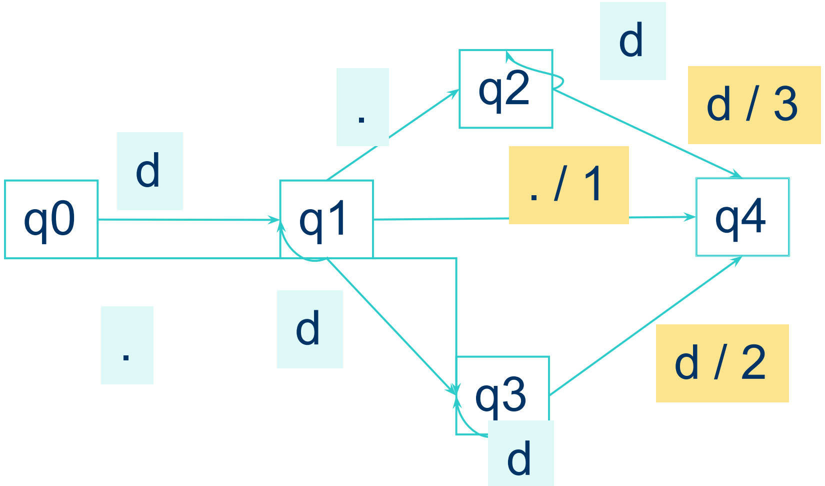
$$\tau(M) = \{ (x, y) \mid (q0, x, \varepsilon) \vdash^* (q, \varepsilon, y) \text{ для некоторого } q \in F\}$$

(регулярный, или конечный, перевод)

Пример конечного преобразователя: распознавание действительных чисел

M={(q0, q1, q2, q3, q4), {d, .}, {1, 2, 3}, δ, q0, {q4} }
δ(q0, d) = {(q1, ε)}
δ(q0, .) = {(q3, ε)}
δ(q1, d) = {(q1, ε)}
δ(q1, .) = {(q2, ε), (q4, 1)}
δ(q2, d) = {(q2, ε), (q4, 3)}
δ(q3, d) = {(q3, ε), (q4, 2)}

Выход:
1 – dd…d.
2 - .dd…d
3 – dd…d.dd…d



Пример: входная цепочка 234.17

(q0, 234.17, ε) ├ (q1, 34.17, ε) ├
├ (q1, 4.17, ε) ├ (q1, .17, ε) ├ (q2, 17, ε) ├
├ (q2, 7, ε) ├ (q4, ε, 3)

Автомат с магазинной памятью

Входная очередь

Устройство управления с конечной памятью

магазин (стек)

Z
α
Z_0

МП-автоматом называется семерка
$P = (Q, \Sigma, \Gamma, \delta, q0, Z0, F)$, где
Q – множество состояний УУ
Σ - алфавит входных символов
Γ – алфавит выходных символов
δ- функция переходов: $Q\times(\Sigma\cup\{\varepsilon\})\times\Gamma\rightarrow Q\times\Gamma^*$
q0 – начальное состояние УУ
Z0 – начальный символ (дно) магазина
F – множество заключительных состояний

Конфигурация МП-автомата
(q, x, Z):
$q\in Q$ – состояние УУ
$x \in \Sigma^*$ - необработанная часть входной цепочки
$Z \in \Gamma^*$ - содержимое магазина (левый символ – вершина стека)

Такт работы МП-автомата:
переход от конфигурации (q, aw, Z) к конфигурации $(r, w, \gamma\alpha)$, если $\delta(q, a, Z)$ содержит (r, γ).

$$(q, aw, Z) \vdash (r, w, \gamma\alpha)$$

Пример: МП-автомат для распознавания симметричных цепочек

P = ({q0, q1, q2}, {a, b}, {Z, a, b}, δ, q0, Z, {q2})

Функции переходов:

1) δ(q0, a, Z)={(q0, aZ)}
2) δ(q0, b, Z)={(q0, bZ)}
3) δ(q0, a, a)={(q0, aa), (q1, ε)}
4) δ(q0, a, b)={(q0, ab)}
5) δ(q0, b, a)={(q0, ba)}
6) δ(q0, b, b)={(q0, bb), (q1, ε)}
7) δ(q1, a, a)={(q1, ε)}
8) δ(q1, b, b)={(q1, ε)}
9) δ(q1, ε, Z)={(q2, ε)}

Примеры:

(q0, baab, Z) ├ (q0, aab, bZ) ├
├ (q0, ab, abZ) ├ (q0, b, aabZ) ├
├ (q0, ε, baabZ) – ошибка

(q0, baab, Z) ├ (q0, aab, bZ) ├
├ (q0, ab, abZ) ├ (q1, b, bZ) ├
├ (q1, ε, Z) ├ (q2, ε)

Расширенный МП-автомат

Лемма:

Если $(q, w, A) \vdash^n (q', \varepsilon, \varepsilon)$, то $(q, w, A\alpha) \vdash^n (q', \varepsilon, \alpha)$ для всех $A \in \Gamma$ и $\alpha \in \Gamma^*$.

Расширенным МП-автоматом называется МП-автомат, который за один такт может заменять цепочку конечной длины в вершине магазина другой цепочкой конечной длины. Функция переходов δ: $Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma^* \to Q \times \Gamma^*$

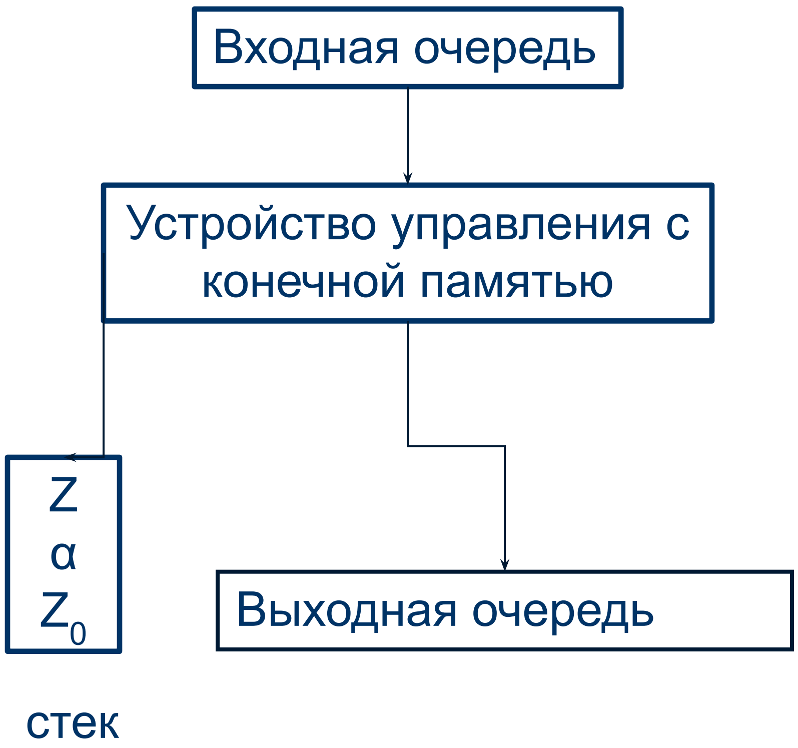
Пример расширенного МП-автомата для языка ww^R

$P = (\{q0, q1\}, \{a, b\}, \{S, Z, a, b\}, \delta, q0, Z, \{q1\})$

Функции переходов:

1) $\delta(q0, a, \varepsilon)=\{(q0, a)\}$
2) $\delta(q0, b, \varepsilon)=\{(q0, b)\}$
3) $\delta(q0, \varepsilon, \varepsilon)=\{(q0, S)\}$
4) $\delta(q0, \varepsilon, aSa)=\{(q0, S)\}$
5) $\delta(q0, \varepsilon, bSb)=\{(q0, S)\}$
6) $\delta(q0, \varepsilon, SZ)=\{(q1, \varepsilon)\}$

Преобразователи с магазинной памятью



Размеченный вывод цепочек в КС-грамматике

Предложение: $i+i*i$

Грамматика:

(1) $E \to E+T$

(2) $E \to T$

(3) $T \to T*P$

(4) $T \to P$

(5) $P \to i$

(6) $P \to (E)$

Левый вывод:

E=>(1) $E+T$=>(2) $T+T$=>(4) $P+T$=>

=>(5) $i+T$=>(3) $i+T*P$=>

=>(4) $i+P*P$=>(5) $i+i*P$=>(5) $i+i*i$

Правый вывод:

E=>(1) $E+T$=>(3) $E+T*P$=>(5) $E+T*i$=>

=>(4) $E+P*i$=>(5) $E+i*i$=>(2) $T+i*i$=>

=>(4) $P+i*i$=>(5) $i+i*i$

Дерево вывода

Упорядоченное дерево, каждый узел которого помечен символом из $\Sigma \cup N \cup \{\varepsilon\}$, называется деревом вывода в КС-грамматике G=(N,Σ, P, S), если:

1. Корень дерева помечен основным символом грамматики S.
2. Каждый прямой потомок узла, помеченного символом A, помечен таким символом X, что грамматика содержит правило вывода A→X.
3. Если узел D_i помечен символом X_i из N, то поддерево D_i должно быть деревом вывода в грамматике G_i=(N,Σ, P, X_i).
4. Если X_i - символ из Σ, то поддерево D_i состоит из одной вершины, помеченной X_i
5. Если корень дерева состоит из единственного потомка, помеченного ε, то этот потомок образует дерево, состоящее из единственной вершины, и в множестве правил грамматики содержится правило S→ε.

Определение разбора

Цепочка для КС-грамматики *разобрана*, если известно её дерево вывода.

Пусть заданы КС-грамматика G, правила которой перенумерованы целыми числами 1, 2, …, p, и цепочка терминальных символов α. Тогда:
левым разбором цепочки α называется последовательность правил, примененных при её левом выводе из S;
правым разбором цепочки α называется обращение последовательности правил, примененных при правом выводе цепочки из S.

Эквивалентность КС-грамматик и МП-автоматов

Лемма о нисходящем разборе

Пусть $G=(N,\Sigma, P, S)$ – КС-грамматика,
$P =(\{q\}, \Sigma, \Sigma\cup N, \delta, q, S, \{q\})$ – МП-автомат с правилами перехода δ :

1. Если $A\rightarrow\alpha$ - правило вывода грамматики G, то $\delta(q, \varepsilon, A)$ содержит (q, α)
2. $\delta(q, a, a) = \{(q, \varepsilon)\}$ для всех a из Σ.

Тогда $A \Rightarrow^m w$ тогда и только тогда, когда $(q, w, A) \vdash^n (q, \varepsilon, \varepsilon)$ для некоторых m, n.

Пример: МП-автомат для скобочных выражений

Грамматика:

$G=(\{E, T, M\}, \{i, +, *, (,)\}, \{E\to E+T, E\to T, T\to T*M, T\to M, M\to (E), M\to i\}, E)$

МП-автомат:

$P=(\{q\}, \{i, +, *, (,), E, T, M\}, \delta, q, E, \{q\})$

с правилами перехода:

1) $\delta(q, \varepsilon, E)=\{(q, E+T), \{(q,T)\}\}$
2) $\delta(q, \varepsilon, T)=\{(q, T*M), \{(q,M)\}\}$
3) $\delta(q, \varepsilon, M)=\{(q, (E)), \{(q,i)\}\}$
4) $\delta(q, a, a)=\{(q, \varepsilon)\}$ для всех a из $\{i, +, *, (,)\}$

Пример: разбор цепочки i*(i+i)

(q, i*(i+i), E) ├ (q, i*(i+i), T) ├ (q, i*(i+i), T*M)
├ (q, i*(i+i), i*M) ├ (q, *(i+i), *M) ├ (q, *(i+i), *M)
├ (q, (i+i), M) ├ (q, (i+i), (E)) ├ (q, i+i), E)) ├ (q, i+i), E+T))
├(q, i+i), T+T)) ├(q, i+i), M+T)) ├(q, i+i), i+T))
├ (q, +i), +T)) ├ (q, i), T)) ├(q, i), M)) ├ (q, i), i))
├ (q,),)) ├ (q, ε, ε)

Последовательность тактов МП-автомата соответствует левому выводу цепочки i*(i+i) в грамматике G.

Восходящий синтаксический анализ (свертка слева)

Правый вывод цепочки i+i*i:

E=>(1) $E+T$=>(3) $E+T*P$=>(5) $E+T*i$ =>(4) $E+P*i$=>(5) $E+i*i$=>(2) => $T+i*i$=>(4) $P+i*i$=>(5) $i+i*i$

Обращение правого вывода цепочки i+i*i:

$i+i*i <= P+i*i <= T+i*i <= E+i*i <= E+P*i <= E+T*i <= E+T*P <= E+T <= E$

Пусть G – КС-грамматика, S=>αAw=>αβw=>xw – правый вывод в ней. Правовыводимую цепочку αβw можно свернуть слева к правовыводимой цепочке αAw с помощью правила вывода A→β. Это вхождение цепочки β в цепочку αβw называется основой цепочки αβw.

Пример: правила вывода {S→Ac, S →Bd, A →aAb, A →ab, B →aBbb, B →abb}. Цепочка aabbbbd – правовыводимая, abb – основа этой цепочки (т.к. abb – правая часть правила B →abb и aBbbd – правовыводимая цепочка. Цепочка ab – не основа, т.к. это правая часть правила A →ab, но aAbbbd – не правовыводимая цепочка.

Лемма о восходящем разборе

Пусть G=(N,Σ, P, S) – КС-грамматика. Тогда можно построить такой расширенный МП-автомат, что L(G)=L(P).

P=({q, r}, Σ, $\Sigma \cup N \cup \{\perp\}$, δ, q, ⊥, {r})

Правила перехода:

1. δ(q, a, ε) = {(q, a)} для всех терминальных a, т.е. символы со входной ленты переносятся в стек.
2. Если A→α – правило вывода грамматики G, то δ(q, ε, α) содержит (q, A).
3. δ(q, ε, ⊥S) = {(r, ε)}.

Автомат P строит правовыводимые цепочки грамматики G, начиная с терминальной цепочки на входной ленте и заканчивая цепочкой S.