

## Домашнее задание #6

### Задача 1.

Опишите формально и докажите конструктивный алгоритм построения контекстно-свободной грамматики, которая порождает все цепочки, лежащие в пересечении заданных контекстно-свободной грамматики и регулярного автомата.

*Доказательство.*

Пусть у нас есть  $G$  – контекстно-свободная грамматика.  $N_G$  и  $T_G$ , соответственно, нетерминалы и терминалы грамматики.  $A$  – регулярный автомат.  $V_A$  и  $E_A$  – вершины и ребра автомата. Далее индексы могут опускаться, ибо понятно, какие сущности имеются в виду.

Доказательство основано на вычислении отношения:  $R_{N_i} = \{(v_s, v_t) \mid v_s, v_t \in V \wedge \exists p = v_s \rightarrow v_1 \rightarrow \dots \rightarrow v_k \rightarrow v_t \wedge w(p) \in L(G)\}$ , где  $N_i$  – это какой-то нетерминал  $G$ ,  $p$  – это путь в автомате по ребрам  $E$ , который начинается в  $v_s$  и заканчивается в  $v_t$ , и цепочка, которая образуется как конкатенация букв на ребрах, порождается грамматикой.

В таком отношении лежат все пары вершин автомата, которые позволяют начать с первой вершины, потом пройти по какому-нибудь пути и закончить во второй вершине, при этом образовав цепочку, которая соответствует классу нетерминалов  $N_i$ . Мы построим такое отношение для каждого класса нетерминалов в  $G$ . Теперь покажем как нужно его строить, что брать за начальный нетерминал и почему порожденной новой грамматикой  $G'$  цепочки, будут совпадать с цепочками пересечения.

Сперва отметим, что  $G$  – грамматика в нормальной форме Хомского. Если это не так, то мы приведем эту грамматику к ней. Шаги к приведению (удаление длинных правых частей, удаление *eps*-продукций, удаление цепных продукций и удаление терминалов в правых частях с длиной  $\geq 2$ ) были представлены на лекции и доказаны нами. Далее полагаем, что  $G$  в НФХ.

В новой грамматике нашими символами будут состояния отношения:  $[nodeS, symbol, nodeT]$ , где  $nodeS$  – начальная вершина автомата,  $nodeT$  – конечная вершина автомата,  $symbol$  – нетерминал  $G$ , которому будет соответствовать цепочка какого-то пути.

Пусть нам удастся построить такие отношения. Тогда начальными состояниями будут следующие:  $[automatonStartNode, initialNode, automatonTerminalNode]$ , где  $automatonStartNode$  – стартовая вершина в автомате,  $initialNode$  – стартовый нетерминал в  $G$ , а  $automatonTerminalNode$  – терминальная вершина в  $A$ . Мы добавим по продукции из стартового нетерминала  $G'$   $S$  в каждое такое состояние для терминальной вершины автомата (ведь терминальное состояние не обязано быть единственным), т.е. правила вида:  $S : [automatonStartNode, initialNode, automatonTerminalNode]$ .

Действительно, все цепочки, которые порождаются автоматом, начинаются в стартовой вершине, проходят какой-то путь, и заканчиваются в терминальной вершине. А все цепочки, которые порождаются грамматикой, должны быть образованы по правилам из начального нетерминала. То есть в итоге, мы получим ровно пересечение.

Отдельно стоит обратить внимание на грамматики, в которых есть  $\epsilon$ . Тогда,  $\epsilon$  будет лежать в пересечении  $\iff$  в автомате существует вершина, которая является и начальной, и терминальной. Т.к. мы добавили все правила  $S : [automatonStartNode, initialNode, automatonTerminalNode]$ , то все что нам остается сделать для корректности – это добавить правила получения  $\epsilon$  для вышеописанных состояний, у которых начальная и терминальная вершины совпадают, т.е.  $[startTermNode, initialNode, startTermNode] : \epsilon$ .

Теперь опишем, как построить такие отношения. Будем использовать очередь для всех уже достигнутых состояний. Сначала положим в очередь все состояния вида  $[edgeStart, nonterminalWithTerminalRule, edgeEnd]$ . Мы рассмотрим все ребра автомата и нетерминалы, у которых есть правила второго типа нормальной формы Хомского ( $N : t$ ). Так мы сможем забыть о всех правилах такого вида, так как они полезны только для цепочек из одного символа, а мы их все рассмотрели.

Остались только правила вида:  $N : AB$ . Заметим, что чтобы таким можно было воспользоваться, нам нужно, чтобы были достижимы состояния  $[v_1, A, v_2]$  и  $[v_2, B, v_3]$ . Тогда, каждый раз вынимая состояние из очереди, мы будем рассматривать все правила, в которых участвует текущий нетерминал. Когда мы извлечем и рассмотрим второй нетерминал, мы сможем применить правило. Для определенности положим, что мы вынули  $A$  последним в виде состояния:  $[nodeS, A, nodeT]$ . Чтобы согласовать это с автоматом, нам нужна вершина *complementingNode* автомата, такая, что есть цепочка класса  $B$ , которую можно получить начав с *nodeT* и закончив путь в *complementingNode*. Переберем все такие вершины и правила, и если мы уже посетили состояние  $[nodeT, B, complementingNode]$ , то добавим новое достижимое состояние  $[nodeS, N, complementingNode]$ .

Сложность алгоритма:  $O(|V|^3|G|)$ , т.к. для любой пары вершин автомата мы переберем третью вершину и для всех таких троек просмотрим все правила в грамматике.

Покажем, почему в итоге цепочки порождающиеся грамматикой  $G'$  совпадают с цепочками, находящимися в пересечении.

Пусть мы смогли породить цепочку  $w$  в грамматике  $G'$ . Покажем, что тогда такая цепочка есть отдельно и в автомате, и в грамматике, и, следовательно, в их пересечении. Т.к. мы начали с *initialNode* грамматики и переходили в новые состояния только по правилам грамматики, то цепочка  $w \in L(G)$ . Оно лежит в автомате, потому что по построению мы начали с состояния  $[startNode, \_, terminalNode]$  и каждый наш переход поддерживал связность пути (т.е. у соседних состояний в дереве разбора общая вершина  $[v_1, \_, v_3] : [v_1, \_, v_2][v_2, \_, v_3]$ ). В итоге мы получили последовательность таких состояний, которые превратились в буквы, а это только ребра. Пройдя этот путь в автомате, мы получим ту же цепочку, а значит  $w \in L(A)$ .

Обратно: пусть есть цепочка  $w$ , которая принадлежит  $L(G) \cap L(A)$ . Покажем, что  $w \in L(G')$ . Будем доказывать по индукции от длины цепочки следующее утверждение: если цепочка лежит в каком-то нетерминале  $N_i$  и при этом может быть порождена в автомате, то достижимо будет и состояние  $[v_s, N, v_t]$ , из которого возможно породить  $w$ .

Про длины 0 и 1 это было показано ранее.

Сейчас же докажем индукционный переход  $k \rightarrow (k+1)$ . Так как  $w \in L(G)$ , то знаем, что есть правило  $N : AB$ , по которому может быть порождена  $w$ . Важное свойство НФХ заключается в том, что каждый нетерминал, кроме *eps*, займет хотя бы 1 символ (ведь мы удалили все *eps*-продукции). Это позволит нам сослаться на индукционное предположение. Из того, что  $w \in L(A)$  имеем, что есть две вершины  $v_s$  и  $v_t$  между которыми есть путь с надписью  $w$ . Теперь из дерева разбора найдем префикс, которому соответствует раскрытие нетерминала  $A$  из правила. Пусть вершина пути, соответствующая этому префиксу – это  $v_m$ . Тогда по предположению имеем, что состояния  $[v_s, A, v_m]$  и  $[v_m, B, v_t]$  достижимы. Тогда и состояние  $[v_s, N, v_t]$ , из которого порождается  $w$ , достижимо.

Ну а все слова в пересечении тогда будут порождены, потому что мы задали как изначальные только состояния с символом начального нетерминала грамматики и парами из стартовой и терминальной вершин автомата.

□