Студент: Никита Сазанович
Группа: CS

Домашнее задание #6

Задача 1.

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

Доказательство.

Пусть у нас есть G – контестно-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: $R_{N_I} = \{(v_s, v_t) \mid v_s, v_t \in V \wedge \exists p = v_s \to v_1 \to \dots \to v_k \to v_t \wedge w(p) \in L(G)\}$, 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 это 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 ≥ 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 состояниям будут следующие: $[automatonStartNode, initialNode, automatonTerminalNode]$, где $automatonStartNode$ – стратовая вершина в автомате, $initialNode$ – стартовый нетерминал в G, а $automatonTerminalNode$ – терминальная вершина в A. Мы добавим по продукции из стратового 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, это те, которые начинаются в стартовой вершине, проходят какой-то путь, и заканчивают в терминальной вершине. А все цепочки, которые порождаются грамматикой, это те, которые могут быть образованы по правилам из начального нетерминала. То есть в итоге, мы получим ровно пересечение.

Отдельно стоит обратить внимание на грамматики, в которых есть ϵ. Тогда, ϵ будет лежать в пересечении $\iff$ в автомате существует вершина, которая является и начальной, и терминальной. Т.к. мы добавили все правила $S : [automatonStartNode, initialNode, automatonTerminalNode]$, все что нам остается сделать для корректности – это добавить правила получения ϵ 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 ($N : a$ или $N : t$). 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 было достижимо и A, и B. Тогда, каждый раз вынимая состояние из очереди, мы будем рассматривать все правила, в которых участвует текущий нетерминал. Когда мы вынимем 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 $[nodeS, N, complementingNode]$.

Таким образом, мы рассмотрим все правила грамматики G, и при этом воспользуемся только теми правилами, которые согласуются с автоматом, т.е. построим пересечение.

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.

$\square$