HW06

Для доказательства факта того, что пересечение регулярного языка и КН-грамматики является КН-грамматикой, достаточно привести явный алгоритм построения КН-грамматики для пересечения. Этим мы и займёмся.

Алгоритм

Нам даны автомат A и КН-грамматика G. Сразу будем считать, что G находится в НФ Хомского, потому что на лекциях было подробно разобрано, как любую КН-грамматику можно к ней привести.

Символами новой грамматики будем называть тройки вида $(start, chain, end)$, где $start, end$ – какие-то вершины автомата A, а $chain$ – нетерминал G такой, что ему будет соответствовать цепочка какого-то пути между $start$ и end в автомате. Тогда начальным состояниям будут соответствовать тройки $(A_{start}, G_{initial}, A_terminal)$. Для каждой такой тройки T_i добавим продукцию $S : T_i$.

Действительно, любая порождаемая автоматом цепочка начнётся в начальной вершине и закончится в терминальной, а любая цепочка из порождаемых КН-грамматикой начнётся с начального нетерминала.

Есть ещё один важный случай – грамматики, содержащие пустое слово. В терминах автоматов это означает, что стартовая вершина тоже является терминальной. Добавим такое правило – $(A_{start}, G_{initial}, A_{start}) : \epsilon$.

Научимся же строить такие тройки: все уже достигнутые состояния будем хранить в очереди. Сначала заполним её тройками $(from, N, to)$, где $(from \rightarrow to)$ – какое-то ребро автомата A, а N – нетерминал, у которого есть правило, содержащее терминал. Таким образом, сначала мы рассмотрим все цепочки из одного символа, соответствующие правилам вида $N : t$ в НФ Хомского. Это означает, что в дальнейшем мы с ними уже не встретимся. Теперь остались правила вида $N : PQ$: для их использования нам необходимо наличие троек $(u, P, v), (v, Q, w)$. Значит, мы будем последовательно исследовать нашу очередь и для текущего нетерминала рассматривать все правила, в которых он содержится. После извлечения обоих нетерминалов правило можно будет применить. Но нам нужно это как-то согласовать с автоматом. Для определённости будем считать, что последним мы вытащили Q в виде тройки (v, Q, w). Тогда переберём вершины u и правила P, и если мы уже посетили (u, P, v), то добавим новое достижимое состояние (u, N, w). Для любой пары вершин автомата мы переберём все его вершины и все правила G – сложность выйдет $O(|A|^3|G|)$.

Теперь строго покажем, почемы мы получили именно пересечение: пусть построенная нами G' породила некоторую цепочку x. Так как переходы мы делали только по правилам грамматики G, то $x \in L(G)$. Теперь найдём её в автомате: мы начинали в $(start, \_, terminal)$, а все переходы были вида $(u, \_, w) : (u, \_, v)(v, \_, w)$ – то есть они сохраняли связность пути в автомате. Значит наша цепочка соответствует некоторому пути в A между начальной вершиной и терминальной – $x \in L(A)$.

Теперь наоборот, надо по цепочке x из пересечения найти её в G'. Будем доказывать этот факт индукцией по $|x|$. Случаи с длинами $0, 1$ были рассмотрены на этапе описания алгоритма – база есть. Будем показывать индукционный переход. Так как длина > 1, а цепочка принадлежит пересечению, то данную цепочку порождает некоторое правило $N : PQ$. Из принадлежности цепочки автомату мы можем заключить, что есть некоторый путь $(u, \ldots, w)$, соответствующий x. Найдём в дереве разбора префикс p, соответствующий нетерминалу P из правила. Пусть этому префиксу соответствует вершина v в пути $(u, \ldots, w)$. Теперь вспомним, что мы удалили все ϵ-продукции, что означает, что каждый нетерминал породит непустую цепочку (если он, конечно, не ϵ). Следовательно, оба куска $pq = x$ короче $|x|$, а, значит, по индукционному предположению, соответствующие им вершины $(u, P, v), (v, Q, w)$ достижимы $\Rightarrow$ получаемая из них вершина (u, N, w) тоже достижима. ЧТД.