Теория формальных языков
Рубежный контроль №2

Вариант №22

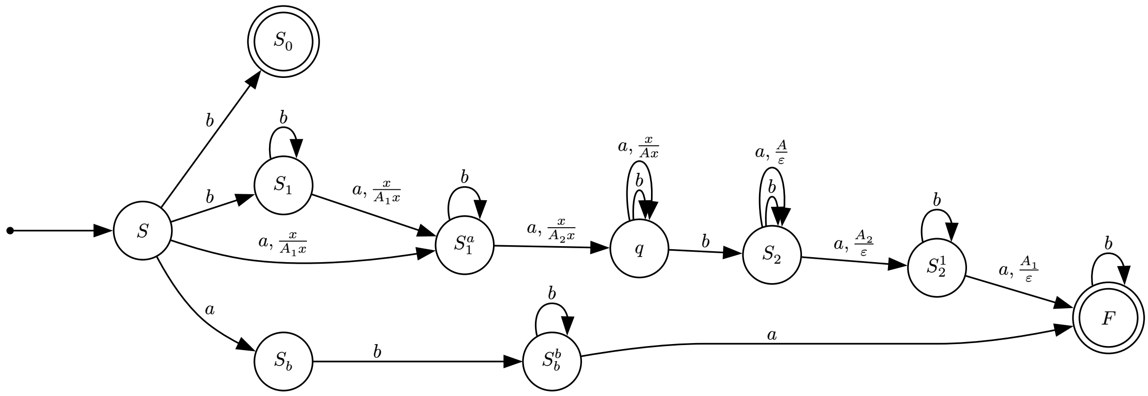
Никитин Сергей ИУ9-51Б

Задание 1

Язык SRS $a \to ab, a^2 \to ba^2$ над множеством базисных слов $a^n ba^n$

Решение

Для доказательства контекстной свободы построим недетерминированный PDA:



Теперь проверим язык на регулярность. Рассмотрим регулярный язык a^*ba^*. Пересечение нашего языка с этим состоит из слов, имеющих ровно одну b. В нашем языке такие слова можно получить только если вообще не применять правил, т.к. каждое правило увеличивает количество букв b. Из этого следует, что пересечением этих 2-х языков является $\{a^n ba^n \mid n \geq 0\}$, однако данный язык не является регулярным. Следовательно и исходный язык из условия задачи также не является регулярным.

Задание 2

Язык $\{\omega_1 aa\omega_2 \mid \omega_1 = \omega_3 b\omega_4 \quad \& \quad \omega_2 = \omega_5 b\omega_6 \quad \& \quad |\omega_1| = |\omega_2|\}$. Алфавит $\{a, b\}$.

Решение

Предположим, что наш язык является DCFL. Воспользуемся замкнутостью DCFL относительно операции пересечения с регулярным языком. Рассмотрим регулярный язык $a^*ba^*aaa^*ba^*$. Его пересечение с нашим языком будет иметь вид $\{a^n ba^m aaa^l ba^k \mid m + n = l + k\}$. По предположению, пересечение должно быть DCFL. Проверим так ли это. В таком случае должен существовать DPDA, который распознаёт этот язык. Рассмотрим префикс $u_n = a^n ba^m$. После прочтения этого префикса автомат будет читать первый символ из центральной строки aa, однако первая a в этом блоке может быть, например, всё ещё частью префикса. В этом случае автомат должен:

- в первом случае — перейти к проверке правой половины слова (начать доставать символы из стека)
- во втором случае — продолжать накапливать информацию о его длине (класть символы в стек)

Эти действия различны, несмотря на то, что входной символ один и тот же и состояния и вершины стека одинаковы. Следовательно, переход не может быть детерминированным и пересечение исходного языка с регулярным не является DCFL, а это значит, что наше исходное предположение является ложным.

Задание 3

Язык атрибутной грамматики:

$S \to aSbS \quad ; \quad S_0.a \coloneqq S_1.a + S_2.a,\ S_1.a == S_2.a$

$S \to T \quad ; \quad S.a \coloneqq T.a$

$T \to aT \quad ; \quad T_0.a \coloneqq T_1.a + 1$

$T \to bT \quad ; \quad T_0.a \coloneqq [T_1.a/2]$

$T \to c \quad ; \quad T.a \coloneqq 0$

Решение

Все атрибуты в грамматике являются синтезируемыми, но в первом правиле есть условие $S_1.a == S_2.a$, которое не вычисляет значение, а проверяет его. Данное условие вводит контекстную зависимость, так как требует равенства атрибутов двух разных поддеревьев. Это условие не может быть выражено в рамках контекстно-свободной грамматики по нескольким причинам:

- Атрибут $.a$ вычисляется как целое число на основе строки, порожденной из T или S.
- Вычисление атрибута зависит от всей подстроки и его значение может быть сколь угодно большим.
- Проверка равенства двух таких атрибутов требует сравнения результатов двух независимых вычислений, что эквивалентно проверке глобального свойства.

Рассмотрим подмножество языка, порождаемого грамматикой:

$$L' = \{a\omega_1 cb\omega_2 c \mid f(\omega_1) = f(\omega_2)\},$$

где $f(\omega)$ — значение атрибута, вычисляемое для строки w по правилам для T. Если бы L' был контекстно-свободным, то и пересечение с регулярным языком, например, с a^*cb^*c, тоже давало бы КС язык. Проблема здесь в том, что условие $f(\omega_1) = f(\omega_2)$ требует равенства двух чисел, представленных в виде последовательностей операций $+1$ и $/2$. Более того, функция $f(w)$ может принимать произвольные натуральные значения, и проверка её равенства для двух подстрок потребует неограниченной памяти.

Таким образом данный язык атрибутной грамматики не является КС, хотя, если бы условие отсутствовало, то грамматика была бы чисто синтезируемой (S-атрибутной) и порождала бы КС язык.