Рябых Владислав, Б05-905

Алгоритмы и модели вычислений. Домашнее задание № 7

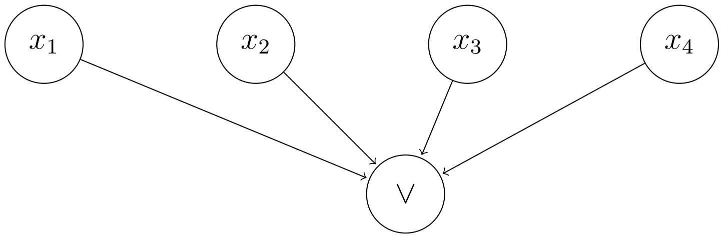
***Задача** 1.* 1. Докажите, что $NC^d \subseteq AC^d \subseteq NC^{d+1}$.

2. Докажите, что $\bigcup\limits_{d=1}^{\infty} AC^d = NC$.

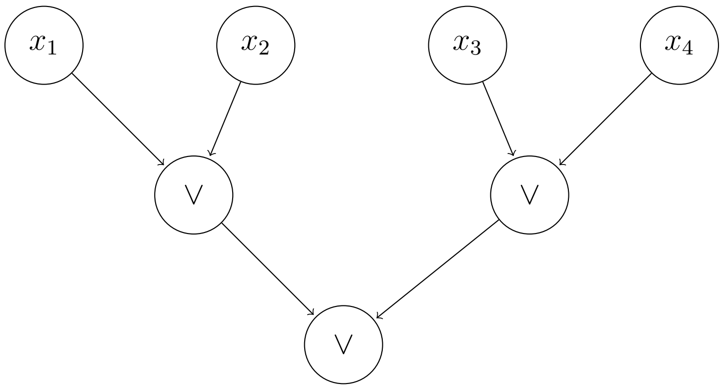
Решение. 1. Вложение $NC^d \subseteq AC^d$ очевидно: если некоторый язык распознаётся схемами с двумя входными литералами с глубиной $O(\log^d n)$, то он распознаётся схемами с произвольным количеством входных литералов с глубиной $O(\log^d n)$ (можно просто взять ту же последовательность схем)

Докажем теперь вложение $AC^d \subseteq NC^{d+1}$: для этого сначала покажем, как мы будем разворачивать вершины $\vee$ и $\wedge$ произвольных схем из AC^d:

Изначально имеем следующую схему:



Разворачиваем её следующим образом:



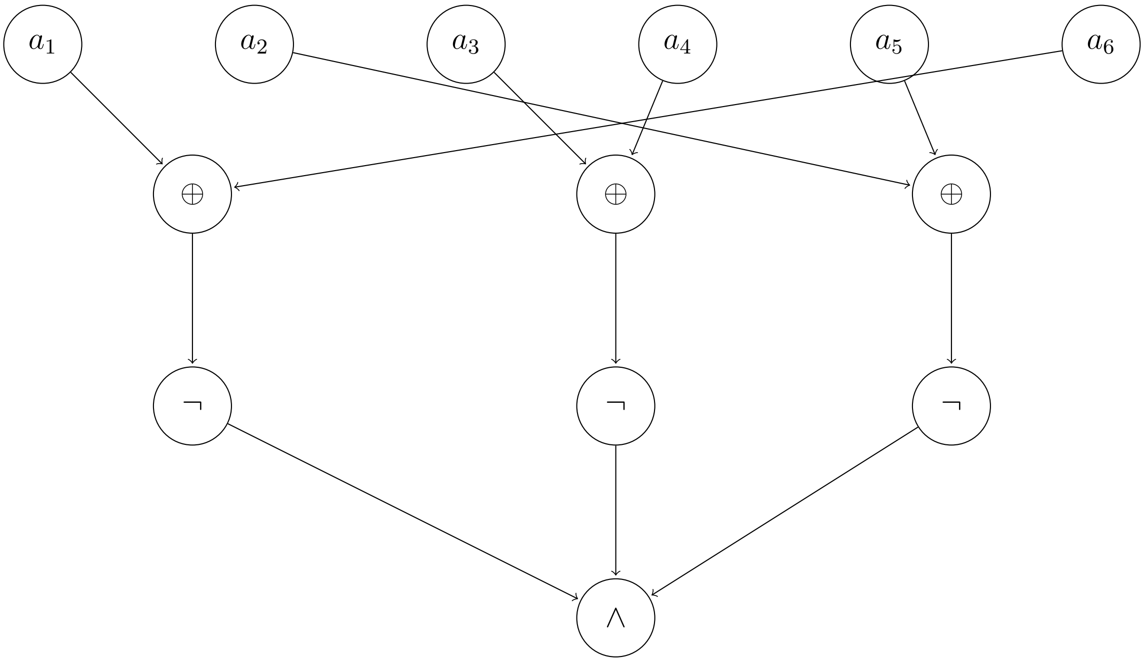
Докажем, что если мы в произвольной схеме глубины $O(\log^d n)$ развернём все вершины $\vee$ и $\wedge$ входящей степени > 2, то мы получим схему глубиной $O(\log^{d+1} n)$. Пусть некоторая вершина исходной схемы имеет входную степень k. Тогда заметим, что так как в нашей развёртке количество входов с каждым уравнем сокращается вдвое, то эта вершина преобразуется в дерево глубины $\log_2 k$. Таким образом глубина исходной схемы $O(\log^d n)$, а глубина каждого из её уровней после преобразования есть $O(\log n)$, то есть итоговая глубина преобразованной схемы будет $O(\log^d n) \cdot O(\log n) = O(\log^{d+1} n)$, таким образом $AC^d \subseteq NC^{d+1}$, что и требовалось.

2. По доказанному в прошлом пункте $NC^d \subseteq AC^d \subseteq NC^{d+1}$, тогда $NC = \bigcup\limits_{d=1}^{\infty} NC^d \subseteq \bigcup\limits_{d=1}^{\infty} AC^d \subseteq \bigcup\limits_{d=1}^{\infty} NC^{d+1} \subseteq \bigcup\limits_{d=1}^{\infty} NC^d = NC$

   То есть $NC \subseteq \bigcup\limits_{d=1}^{\infty} AC^d \subseteq NC$, то есть $\bigcup\limits_{d=1}^{\infty} AC^d = NC$, что и требовалось.

Задача 2. Докажите, что язык $PAL = \{a \mid a = a^R\}$, где a^R — слово a, записанное в обратном порядке, лежит в AC^0

Решение. AC^0 — язык слов, распознаваемых схемами типа AC и имеющих при этом константную глубину. Зафиксируем базис $\neg, \vee, \wedge$. Так как $a \oplus b = (a \wedge \neg b) \vee (\neg a \wedge b)$, то добавление в базис символа $\oplus$ увеличивает размер схемы в константу раз (будем использовать $\oplus$, так как предикат «a_i совпадает с a_j» можно записать в виде $\neg(a_i \oplus a_j)$)
Таким образом $a = a_1 a_2 \ldots a_n \in PAL \Leftrightarrow \neg(a_1 \oplus a_n) \wedge \neg(a_2 \oplus a_{n-1}) \wedge \ldots = 1$, и, собственно, мы можем вычислить эту функцию на схеме типа AC константной глубины:



Мы получили схему константной длины, которая вычисляет язык $PAL \Rightarrow PAL \in AC^0$, что и требовалось.