Introduction to Circuit Complexity: Errata & Addenda
(July 12, 2005)

Page 5

- Line 15: are on) \(\sim\) are on for \(1 \leq i \leq n\)

Page 6

- Line -4: \(3 \sim 4\)

Page 9

- Line 5: \(\alpha(v_1) < \cdots < \alpha(v_k) \sim \alpha(v_1, v) < \cdots < \alpha(v_k, v)\)

Page 12

- Line 17: \(s, d: N \to N \sim s, d: N \to N, s(n) \geq n\)

Page 15

- Line -20: \(B_0 \sim B_1\)

Page 22

- Eq. (1.2): The lower index of the sum should be \(k = 0\).

Page 32

- Line 3: \(f \in FSIZE-DEPTH(s(n^{O(1)}), d(n^{O(1)}) + \log n)\)

- Line 4: \(f \in FUnbSIZE-DEPTH(s(n^{O(1)}), d(n^{O(1)}) \cap \text{UnbDEPTH}(1) \cap \text{UnbSIZE}(1))\)

Page 35

- Lemma 2.2 and Observation 2.3: \(\text{SIZE}(1) \cap \text{DEPTH}(1) \sim \text{UnbSIZE}(1) \cap \text{UnbDEPTH}(1)\)

Page 44

- Line -8: \(\langle x, f(|x|) \rangle \sim \langle x, f(1|x|) \rangle\)

Page 52

- Line -12: root \(k\) \(\sim\) root \(v\)
Lemma 1.31  ⇔  Theorem 1.31

▶ page 55 line –8
\[ f(|x|) \Rightarrow O(f(|x|)) \]  

▶ page 67 line –13
processors \( P_1, \ldots, P_{p(n)} \)  ⇔  processors \( P_1, \ldots, P_{p(n)} \) (where \( n \) is the input length, as defined on the next page)

▶ page 73 line 2
processors \( P_1, \ldots, P_{p(n)}(n) \)  ⇔  processors \( P_1, \ldots, P_{p(n)}(n) \) (where \( n \) is the input length, as defined on the next page)

▶ page 77 line –14
if \( C_f = 1 \) then begin  ⇔  while \( C_f = 1 \) do begin
Additionally, something about synchronization should be mentioned here. It has to be ensured that all processors enter the while loop at the same time and need the same time to complete one execution of the loop. This can be achieved by inserting a necessary number of dummy statements (e.g., \( R_0 \leftarrow R_0 \)) in the different cases of the if-statements.

▶ page 82 line –1
CRCW-PRAM  ⇔  CRCW-PRAM with multiplication and division as unit-time operations

▶ page 90 line –5
Add after first sentence: “By the above, we assume that every \( D_n \) is layered.”

▶ page 91 line 13
Replace second sentence by: “Pick \( n_1 > n_0 \) such that \( 2^{n_1} + r(1 - (n_1 + r)^{-k}) \geq \frac{a}{m} 2^{n_1} \) and \( (\log(n_1 + r))^2 \leq \sqrt{n_1} \).”

▶ page 101 line 24 and 25
Replace both lines by the following:
\[ F_0(x_1, \ldots, x_n) = q_{n+r}(x_1, \ldots, x_n, 0, \ldots, 0) \]
\[ F_i(x_1, \ldots, x_n) = q_{n+r}(x_1, \ldots, x_n, 1, \ldots, 1, 0, \ldots, 0) \] for \( 1 \leq i < r \).

▶ page 119 line –7
\[ \sum_{i=m}^{k} d(\log n_m)^j \Rightarrow \sum_{m=1}^{k} d(\log n_m)^j \]

▶ page 126 line 8
ATIME-ALT(\( \log n, 1 \))  ⇔  ATIME-ALT(\( \log n, O(1) \))

▶ page 133 line 2
\( \{0,1\} \)-programs  ⇔  \( \{0,1\} \)-programs of polynomial size

2
\[
\{(w \in \{0,1\} \times \mathcal{P}(V))^+ \mid w \models \phi\} \rightarrow \{w \in (\{0,1\} \times \mathcal{P}(V))^+ \mid w \models \phi\}
\]

\[
a \in i \rightarrow a \in \{0,1\}
\]

results in a for \(\rightarrow\) results in a circuit for

\[
L \leq (L')^2 \rightarrow t(L) \leq (L')^2
\]

\[
\text{step number } i \rightarrow \text{step number } t
\]

the position in \(\rightarrow\) the symbol in

 Replace every occurrence of \(g_n\) by \(g_m\).

 \[
\text{AC[6]} \rightarrow \text{AC[6]}
\]

It was shown recently in


that \(\text{FO[<, bit]} = \text{FO[bit]}\). This means that the \(<\)-predicate may be omitted in the statement of Lemma 4.72, Theorem 4.73, and Corollary 4.77.

This is just the statement of Corollary 4.54.

Replace first two sentences by: “Such a program computes a function \(f_P: \mathcal{R}^n \rightarrow \mathcal{R}\) as follows: Let \(x = (x_1, \ldots, x_n) \in \mathcal{R}^n\).”

The definition of \(\text{accepting subcircuit}\) of \(C\) on input \(x\) is maybe a bit misleading. To be more explicit, one should claim that: \(H\) contains the output gate of \(C\); for every \(\wedge\) gate \(v\) in \(H\) all \(\text{input wires}\) of \(v\) are in \(H\); for every \(\vee\) gate \(v\) in \(H\) exactly one \(\text{input wire}\) of \(v\) is in \(H\); only wires and gates thus introduced belong to \(H\); and all gates in \(H\) evaluate to 1 on input \(x\).

\[
\Sigma\text{-programs} \rightarrow \Sigma\text{-programs of polynomial size}
\]

\[
v_i \rightarrow v_0
\]
matrix programs ⇝ matrix programs of polynomial size

Add: “Note that, according to Definition 5.31, the definition of \( M_s \) above actually gives two matrices, one for each value of \( x_k \).

See the remark on Question 3, p. 209 below.

See the remarks on Question 1, p. 209 below.

As a partial answer to this question, non-uniformly \( \#NC^1 \subseteq FAC^1 \) (and hence, \( \text{Gap-NC}^1 \subseteq FAC^1 \), leading to an additional arc in Fig. 5.6 on p. 207) is known. The result follows from Theorem 5.39 (p. 198f) and the inclusion \( \text{SIZE-DEPTH}(n^{O(1)}, \log n \log \log n) \subseteq AC^1 \) (Theorem 4.3 in [CSV84]). The proof of the latter inclusion is as follows: Divide the fan-in 2 circuit of depth \( \log n \log \log n \) into \( \log n \) levels of depth \( \log \log n \) each. For every level, the nodes on the output of the level depend on at most \( \log n \) nodes at the input of the level. Thus the value of each such node at the output of the level can be expressed as a polynomial size DNF of the input nodes (see Exercise 1.1, p. 32).

The question of how much this construction can be made uniform depends on the complexity of the required operation of division.

(See next remark.)

The summer of 2001 brought a complete clarification of the complexity of the operation of division. First, it was shown in


that division can be computed in logspace and even in \( U_L-NC^1 \). This is already a major improvement to the result by Beame, Cook, and Hoover (see Exercise 1.19 and [BCH86] in the book), placing division in \( U_P-NC^1 \) (actually, \( U_P-TC^0 \)). But it was even surpassed a little bit later by

W. Hesse. Division is in uniform \( TC^0 \). In *Proceedings 28th International Colloquium on Automata, Languages and Programming*, Lecture Notes in Computer Science 2076, p. 104–114, Springer-Verlag, Berlin, 2001,

showing that division is in \( U_D-TC^0 \).

An immediate consequence already of the result by Chiu et al. is an improvement to the preceding remark on Question 1, p. 209, that I added on 19.4.00: We now know \( \text{Gap-NC}^1 \subseteq FL \).

Hesse’s result, that division is in \( U_D-TC^0 \) (see remark on Question 1, p. 209, above), implies that \( TC^0 = C \land AC^0 = CAC^0 \) (Theorem 5.46) holds even in the dlogtime-setting. This issue is discussed in

\[ a \cdot (b + c) \iff a \times (b + c) \]

For helpful comments, leading to some of the above corrections, I am grateful to Eric Allender (Rutgers, New Brunswick) Ramon Bejar (Lleida), Koen Claessen (Chalmers, Gothenburg), Matthias Galota (Würzburg), Jan Johannsen (München), Michal Koucky (Rutgers, New Brunswick), Sebastian Maneth (Leiden), Pascal Michel (Paris), Holger Petersen (Stuttgart), Bernhard Schwarz (Würzburg), and Joachim Selke (Hannover).