Introduction to Circuit Complexity: Errata & Addenda

(July 12, 2005)

▶**page 5** line 15 ——— 16.5.00

are on) $\rightsquigarrow$ are on) for $1 \leq i \leq n$

▶**page 6** line –4 ——— 16.5.00

3 $\rightsquigarrow$ 4

▶**page 9** line 5 ——— 28.4.00

$\alpha(v_1) < \cdots < \alpha(v_k)$ $\rightsquigarrow$ $\alpha\big((v_1, v)\big) < \cdots < \alpha\big((v_k, v)\big)$

▶**page 12** line 17 ——— 22.5.00

$s, d \colon \mathbb{N} \to \mathbb{N}$ $\rightsquigarrow$ $s, d \colon \mathbb{N} \to \mathbb{N}$, $s(n) \geq n$

▶**page 15** line –20 ——— 29.5.00

$\mathcal{B}_0$ $\rightsquigarrow$ $\mathcal{B}_1$

▶**page 22** line 6 ——— 22.1.03

MAJ $\rightsquigarrow$ MULT

▶**page 22** Eq. (1.2) ——— 22.1.03

The lower index of the sum should be $k = 0$.

▶**page 32** line 3 ——— 18.5.00

$\mathcal{B}_0$ $\rightsquigarrow$ $\mathcal{B}_1$

▶**page 32** line 8 ——— 18.5.00

$\ell^{(k)} = 1$ $\rightsquigarrow$ $\ell^{(k)} \leq 2$

▶**page 32** line 15 ——— 6.7.04

UnbSIZE-DEPTH$(n^2, 1)$ $\rightsquigarrow$ UnbSIZE-DEPTH$(n, 1)$

▶**page 32** Exercise 1.8 ——— 22.1.03

a circuit $\rightsquigarrow$ a circuit whose underlying graph is connected

▶**page 32** line –3 ——— 20.6.00

$f \in$ FSIZE-DEPTH$\big(s^{O(1)}, d\big)$ $\iff$ there are $s' \in s^{O(1)}$ and $d' \in O(d)$ such that all bit functions f_i^n can be computed in size s' and depth d'.

▶**page 33** line 2 ——— 20.6.00

$f \in$ FSIZE-DEPTH$\big(s(n^{O(1)}) \cdot n^{O(1)}, d(n^{O(1)}) + \log n\big)$

▶**page 33** line 4 ——— 20.6.00

$f \in$ FUnbSIZE-DEPTH$\big(s(n^{O(1)}) \cdot n^{O(1)}, d(n^{O(1)})\big)$

▶**page 35** Lemma 2.2 and Observation 2.3 ——— 15.10.01

SIZE(1) $\cap$ DEPTH(1) $\rightsquigarrow$ UnbSIZE(1) $\cap$ UnbDEPTH(1)

▶**page 44** line –8 ——— 12.7.05

$\langle x, f(|x|)\rangle$ $\rightsquigarrow$ $\langle x, f(1^{|x|})\rangle$

▶**page 45** line 6 ——— 20.6.00

root k $\rightsquigarrow$ root v

▶**page 52** line –12 ——— 20.6.00

Lemma 1.31 ⇝ Theorem 1.31

▶**page 55** line –8 ———— 14.1.02

$f(|x|)$ ⇝ $O\big(f(|x|)\big)$

▶**page 67** line –13 ———— 20.6.00

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 ———— 20.6.00

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 77** line –14 ———— 20.6.00

CRCW-PRAM ⇝ CRCW-PRAM with multiplication and division as unit-time operations

▶**page 82** line –1 ———— 12.7.05

$\beta(\alpha(x_1,x_2),x_2))$ ⇝ $\beta(\alpha(x_1,x_2),x_1)$

▶**page 90** line –5 ———— 22.1.03

Add after first sentence: "By the above, we assume that every D_n is layered."

▶**page 91** line 13 ———— 22.1.03

t_g ⇝ t

▶**page 101** line 18 ———— 22.1.03

words $x \in \{0,1\}^n$ we have $q_n(x) = 1$ ⇝ words $x = x_1 \ldots x_n \in \{0,1\}^n$ we have $q_n(x_1,\ldots,x_n) = 1$

▶**page 101** line 21 ———— 22.1.03

Delete "and k".

▶**page 101** line 21 ———— 22.1.03

Replace second sentence by: "Pick $n_1 > n_0$ such that $2^{n_1+r}\big(1-(n_1+r)^{-k}\big) \geq \frac{9}{10}2^{n_1}$ and $\big(\log(n_1+r)\big)^c \leq \sqrt{n_1}$."

▶**page 101** line 24 and 25 ———— 22.1.03

Replace both lines by the following:

$$F_0(x_1,\ldots,x_n) = q_{n+r}(x_1,\ldots,x_n,\underbrace{0,\ldots,0}_{r})$$

$$F_i(x_1,\ldots,x_n) = q_{n+r}(x_1,\ldots,x_n,\underbrace{1,\ldots,1}_{r-i},\underbrace{0,\ldots,0}_{i}) \text{ for } 1 \leq i < r.$$

▶**page 119** line –7 ———— 9.8.99

$\displaystyle\sum_{i=m}^{k} d(\log n_m)^j$ ⇝ $\displaystyle\sum_{m=1}^{k} d(\log n_m)^j$

▶**page 126** line 8 ———— 14.1.02

ATIME-ALT$(\log n, 1)$ ⇝ ATIME-ALT$\big(\log n, O(1)\big)$

▶**page 133** line 2 ———— 22.1.03

$\{0,1\}$-programs ⇝ $\{0,1\}$-programs of polynomial size

▶**page 138** line –7 26.5.99

$\{ (\boldsymbol{w} \in \{0,1\} \times \mathcal{P}(V))^+ \mid \boldsymbol{w} \models_I \phi \} \quad \rightsquigarrow \quad \{ \boldsymbol{w} \in (\{0,1\} \times \mathcal{P}(V))^+ \mid \boldsymbol{w} \models_I \phi \}$

▶**page 140** line 18 22.8.03

$a \in i \quad \rightsquigarrow \quad a \in \{0,1\}$

▶**page 140** line 25 22.8.03

results in a for $\rightsquigarrow$ results in a circuit for

▶**page 143** line 11 6.7.04

$L \leq (L')^2 \quad \rightsquigarrow \quad \ell(L) \leq (L')^2$

▶**page 144** line -18 6.7.04

step number i $\rightsquigarrow$ step number t

▶**page 145** line 9 6.7.04

the position in $\rightsquigarrow$ the symbol in

▶**page 156** lines 12–24 25.7.00

Replace every occurence of g_n by g_m.

▶**page 158** line –11 12.7.05

closure of X $\rightsquigarrow$ closure of S

▶**page 163** line 8 11.10.99

AC[6] $\rightsquigarrow$ $\mathrm{AC}^0[6]$

▶**page 164** after line 2 16.9.99

It was shown recently in

> A. Dawar, K. Doests, S. Lindell, and S. Weinstein. Elementary properties of the finite ranks. *Mathematical Logic Quarterly*, 44:349-353, 1998

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

▶**page 167** Exercise 4.25 14.3.02

This is just the statement of Corollary 4.54.

▶**page 179** lines 4ff 22.1.03

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

▶**page 179** lines 23 22.1.03

$f \colon \{0,1\}^n \to \{0,1\} \quad \rightsquigarrow \quad f \colon \mathcal{R}^n \to \mathcal{R}$

▶**page 183** Remark 5.21 21.4.99

The definition of *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 *input wires* of v are in H; for every $\vee$ gate v in H exactly one *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.

▶**page 191** line 17 22.1.03

Σ-programs $\rightsquigarrow$ Σ-programs of polynomial size

▶**page 192** line 8 5.7.99

$v_i \quad \rightsquigarrow \quad v_0$

▶**page 192** lines 11 and 13 ______ 22.1.03

matrix programs ⇝ matrix programs of polynomial size

▶**page 197** line 11 ______ 22.1.03

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.

▶**page 206** Theorem 5.46 ______ 28.9.01

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

▶**page 207** Fig. 5.6 ______ 19.4.00

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

▶**page 209** Question 1 ______ 19.4.00

As a partial answer to this question, non-uniformly $\#\mathrm{NC}^1 \subseteq \mathrm{FAC}^1$ (and hence, $\text{Gap-NC}^1 \subseteq \mathrm{FAC}^1$, 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 $\text{SIZE-DEPTH}(n^{O(1)}, \log n \log\log n) \subseteq \mathrm{AC}^1$ (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.)

▶**page 209** Question 1 ______ 28.9.01

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

> A. Chiu, G. Davida, and B. Litow. Division in logspace-uniform NC^1. *Theoretical Informatics and Applications*, 2001, to appear

that division can be computed in logspace and even in $\mathrm{U_L}$-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 $\mathrm{U_P}$-NC^1 (actually, $\mathrm{U_P}$-TC^0). But it was even surpassed a little bit later by

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

showing that division is in $\mathrm{U_D}$-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 $\text{Gap-NC}^1 \subseteq \mathrm{FL}$.

▶**page 209** Question 3 ______ 28.9.01

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

> Eric Allender. The division breakthroughs. The Computational Complexity Column, ed. by Lance Fornow. *EATCS Bulletin*, 74, 2001.

▶**page 213** line –11 ______ 16.4.99

TC^1 ⇝ FTC^1

▶**page 235** line –1 ______ 10.6.99

class yields ⇝ class with "F" yields

▶**page 239** line 4 ______ 22.1.03

$a \cdot (b + c) \quad \rightsquigarrow \quad a \times (b + c)$

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).