16. A computation that uses only $\oplus$ and complementation produces nothing but affine functions (see exercise 7.1.1–132). Suppose $f(x) = f(x_1, \ldots, x_n)$ is a non-affine function computable in minimum memory. Then $f(x)$ has the form $g(Ax + c)$ where $g(y_1, y_2, \ldots, y_n) = g(y_1 \wedge y_2, y_2, \ldots, y_n)$, for some nonsingular $n \times n$ matrix A of 0s and 1s, where x and c are column vectors and the vector operations are performed modulo 2; in this formula the matrix A and vector c account for all operations $x_i \leftarrow x_i \oplus x_j$ and/or permutations and complementations of coordinates that occur after the most recent non-affine operation that was performed. We will exploit the fact that $g(0, 0, y_3, \ldots, y_n) = g(1, 0, y_3, \ldots, y_n)$.

Let α and β be the first two rows of A; also let a and b be the first two elements of c. Then if $Ax + c \equiv y$ (modulo 2) we have $y_1 = y_2 = 0$ if and only if $\alpha \cdot x \equiv a$ and $\beta \cdot x \equiv b$. Exactly 2^{n-2} vectors x satisfy this condition, and for all such vectors we have $f(x) = f(x \oplus w)$, where $Aw \equiv (1, 0, \ldots, 0)^T$.

Given α, β, a, b, and w, with $\alpha \neq (0, \ldots, 0)$, $\beta \neq (0, \ldots, 0)$, $\alpha \neq \beta$, and $\alpha \cdot w \equiv 1$ (modulo 2), there are $2^{2^n - 2^{n-2}}$ functions f with the property that $f(x) = f(x \oplus w)$ whenever $\alpha \cdot x \bmod 2 = a$ and $\beta \cdot x \bmod 2 = b$. Therefore the total number of functions computable in minimum memory is at most 2^{n+1} (for affine functions) plus

$$(2^n - 1)(2^n - 2)2^2(2^{n-1})(2^{2^n - 2^{n-2}}) < 2^{2^n - 2^{n-2} + 3n + 1}.$$

17. Let $f(x_1, \ldots, x_n) = g(x_1, \ldots, x_{n-1}) \oplus (h(x_1, \ldots, x_{n-1}) \wedge x_n)$ as in 7.1.1–(16). Representing h in CNF, form the clauses one by one in x_0 and AND them into x_n, obtaining $h \wedge x_n$. Representing g as a sum (mod 2) of conjunctions, form the successive conjunctions in x_0 and XOR them into x_n when ready.

(It appears to be impossible to evaluate all functions inside of $n+1$ registers if we disallow the non-canalizing operators $\oplus$ and $\equiv$. But $n+2$ registers clearly do suffice, even if we restrict ourselves to the single operator $\overline{\wedge}$.)

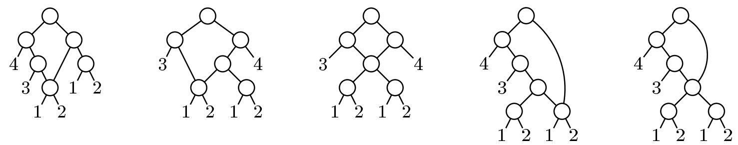
18. As mentioned in answer 14, we should extend the text's definition of minimum-memory computation to allow also steps like $x_{j(i)} \leftarrow x_{k(i)} \circ_i x_{l(i)}$, with $k(i) \neq j(i)$ and $l(i) \neq j(i)$, because that will give better results for certain functions that depend on only four of the five variables. Then we find $C_m(f) = (0, 1, \ldots, 13, 14)$ for respectively (2, 2, 5, 20, 93, 389, 1960, 10459, 47604, 135990, 198092, 123590, 21540, 472, 0) classes of functions . . . leaving 75,908 classes (and 575,963,136 functions) for which $C_m(f) = \infty$ because they cannot be evaluated *at all* in minimum memory. The most interesting function of that kind is probably

$$(x_1 \wedge x_2) \vee (x_2 \wedge x_3) \vee (x_3 \wedge x_4) \vee (x_4 \wedge x_5) \vee (x_5 \wedge x_1),$$

which has $C(f) = 7$ but $C_m(f) = \infty$. Another interesting case is $\big(((x_1 \vee x_2) \oplus x_3) \vee ((x_2 \vee \bar{x}_4) \wedge x_5)\big) \wedge \big((x_1 \equiv x_2) \vee x_3 \vee x_4\big)$, for which $C(f) = 8$ and $C_m(f) = 13$. One way to evaluate that function in eight steps is $x_6 = x_1 \vee x_2$, $x_7 = x_1 \vee x_4$, $x_8 = x_2 \oplus x_7$, $x_9 = x_3 \oplus x_6$, $x_{10} = x_4 \oplus x_9$, $x_{11} = x_5 \vee x_9$, $x_{12} = x_8 \wedge x_{10}$, $x_{13} = x_{11} \wedge \bar{x}_{12}$.

19. If not, the left and right subtrees of the root must overlap, since case (i) fails. Each variable must occur at least once as a leaf, by hypothesis. At least two variables must occur at least twice as leaves, since case (ii) fails. But we can't have $n+2$ leaves with $r \leq n+1$ internal nodes, unless the subtrees fail to overlap.

20. Now Algorithm L (with '$f = g \oplus h$' omitted in step L5) shows that some formulas must have length 15; and even the footprint method of exercise 11 does no better than 14. To get truly minimum chains, the 25 special chains for $r = 6$ in the text must be supplemented by five others that can no longer be ruled out, namely



and when $r = (7, 8, 9)$ we must also consider respectively $(653, 12387, 225660)$ additional potential chains that are not special cases of the top-down and bottom-up constructions. Here are the resulting statistics, for comparison with Table 1:

$C_c(f)$	Classes	Functions	$U_c(f)$	Classes	Functions	$L_c(f)$	Classes	Functions	$D_c(f)$	Classes	Functions
0	2	10	0	2	10	0	2	10	0	2	10
1	1	48	1	1	48	1	1	48	1	1	48
2	2	256	2	2	256	2	2	256	2	7	684
3	7	940	3	7	940	3	7	940	3	59	17064
4	9	2336	4	9	2336	4	7	2048	4	151	47634
5	24	6464	5	21	6112	5	20	5248	5	2	96
6	30	10616	6	28	9664	6	23	8672	6	0	0
7	61	18984	7	45	15128	7	37	11768	7	0	0
8	45	17680	8	40	14296	8	27	10592	8	0	0
9	37	7882	9	23	8568	9	33	11536	9	0	0
10	4	320	10	28	5920	10	16	5472	10	0	0
11	0	0	11	6	1504	11	30	6304	11	0	0
12	0	0	12	5	576	12	3	960	12	0	0
13	0	0	13	3	144	13	8	1472	13	0	0
14	0	0	14	2	34	14	2	96	14	0	0
15	0	0	15	0	0	15	4	114	15	0	0

The two function classes of depth 5 are represented by $S_{2,4}(x_1, x_2, x_3, x_4)$ and $x_1 \oplus S_2(x_2, x_3, x_4)$; and those two functions, together with $S_2(x_1, x_2, x_3, x_4)$ and the parity function $S_{1,3}(x_1, x_2, x_3, x_4) = x_1 \oplus x_2 \oplus x_3 \oplus x_4$, have length 15. Also $U_c(S_{2,4}) = U_c(S_{1,3}) = 14$. The four classes of cost 10 are represented by $S_{1,4}(x_1, x_2, x_3, x_4)$, $S_{2,4}(x_1, x_2, x_3, x_4)$, $(x_4?\ x_1 \oplus x_2 \oplus x_3\colon \langle x_1 x_2 x_3 \rangle)$, and $[(x_1 x_2 x_3 x_4)_2 \in \{0, 1, 4, 7, 10, 13\}]$. (The third of these, incidentally, is equivalent to (20), "Harvard's hardest case.")

21. (The authors stated that their table entries "should be regarded only as the most economical operators known to the present writers.") The minimum cost of their hardest function (20) is still unknown, but David Stevenson has shown that $V(f) \le 17$:

$$g = \mathrm{AND}\big(\mathrm{NAND}(w, x), \mathrm{NAND}(\bar{w}, \bar{x})\big);$$
$$f = \mathrm{OR}\big(\mathrm{AND}\big(\mathrm{NOT}(g), \mathrm{NAND}(w, \bar{z}), \mathrm{NAND}(y, z)\big), \mathrm{AND}\big(\mathrm{NOT}(\mathrm{NOT}(g)), \mathrm{NAND}(y, \bar{z}), \mathrm{NAND}(\bar{y}, z)\big)\big).$$

Although they failed to find this particular construction, the Harvard researchers did remarkably well, in some cases beating the footprint heuristic by as many as 6 grids!

22. $\nu(x_1x_2x_3x_4x_5) = 3$ if and only if $\nu(x_1x_2x_3x_4) \in \{2, 3\}$ and $\nu(x_1x_2x_3x_4x_5)$ is odd. Similarly, $S_2(x_1, x_2, x_3, x_4, x_5) = S_3(\bar{x}_1, \bar{x}_2, \bar{x}_3, \bar{x}_4, \bar{x}_5)$ incorporates $S_{1,2}(x_1, x_2, x_3, x_4)$: