AccessionIndex: TCD-SCSS-V.20191101.004
Accession Date: 1-Nov-2019
Accession By: Dr.Brian Coghlan
Object name: Handbook of Mathematics
Vintage: 1998
Synopsis: Handbook for use in examinations of the Dept.Computer Science, Trinity College Dublin.

Description:
This booklet, prepared and edited by Dr.Michaél Mac an Airchinnigh, Dr.Andrew Butterfield and Arthur Hughes, was the first edition of a mathematics reference handbook for use by students in the examinations of the Dept.Computer Science, Trinity College Dublin.

This item was donated by Brian Coghlan.

The homepage for this catalog is at: https://www.scss.tcd.ie/SCSSTreasuresCatalog/
Click *'Accession Index'* (1st column listed) for related folder, or *'About'* for further guidance.
Some of the items below are more properly part of the other categories of this catalog,
but are listed here for convenience.

Accession Index	Object with Identification
TCD-SCSS-V.20191101.004	Handbook of Mathematics. Handbook for use in examinations of the Dept.Computer Science, Trinity College Dublin. 1998.

References:

1. Trinity College Dublin, *School of Computer Science and Statistics*, see: https://www.scss.tcd.ie/
   Last browsed to on 1-Nov-2019.

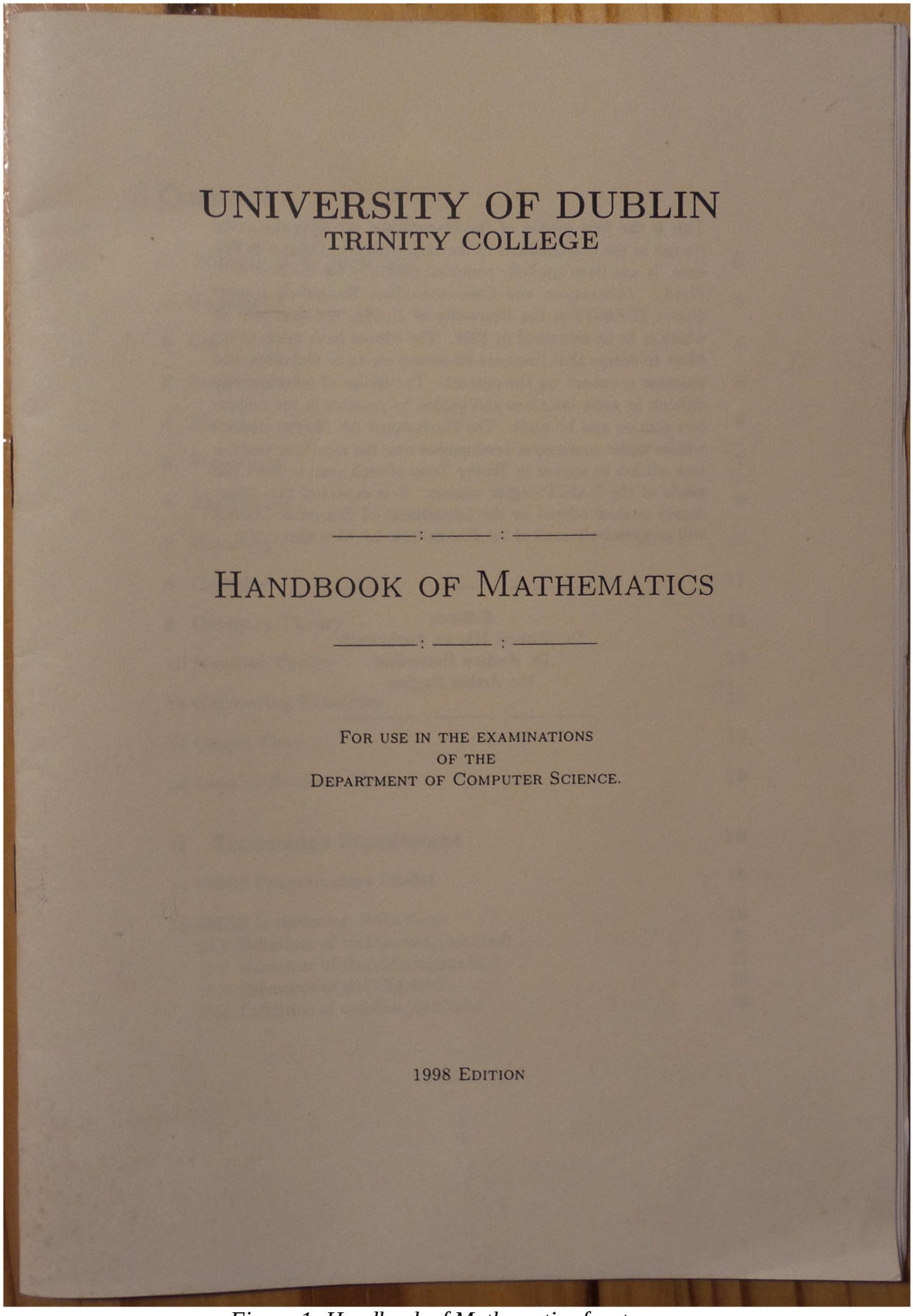


Figure 1: Handbook of Mathematics front cover

This is the *first edition* of the HANDBOOK OF MATHEMATICS for use in the examinations of the Department of Computer Science. It was been specially compiled primarily for the new *B.A. (Mod.) Information and Communications Technology Honors Degree* (BAICT) of the University of Dublin, the first year of which is to be examined in 1998. The editors have made every effort to ensure that there are no serious errors or omissions and welcome comment on the content. The choice of notation was difficult in some instances and guided by practice in the respective courses and by topic. The HANDBOOK OF MATHEMATICS will be under continuous development over the next four years, a new edition to appear in Trinity Term of each year to meet the needs of the BAICT degree courses. It is expected that other degree courses offered by the Department of Computer Science will progressively avail of the HANDBOOK OF MATHEMATICS.

———: ——— : ———

Editors
Dr. Mícheál Mac an Airchinnigh
Dr. Andrew Butterfield
Mr. Arthur Hughes

———: ——— : ———

Figure 2: Handbook of Mathematics fronticepiece

Contents

I Notation & Formulæ 5

1 Numbers 5

2 Sets 5

3 Sequences 6

4 Maps 6

5 Structures 7

6 Algorithms 9

7 Counting 10

8 Calculus 11

9 Category Theory 12

10 Number Theory 13

11 Generating Functions 15

12 Graph Theory 17

13 Laws of Boolean Algebra 18

II Technology Supplement 19

14 68000 Programmers Model 19

15 68000 Instruction Behaviour 19

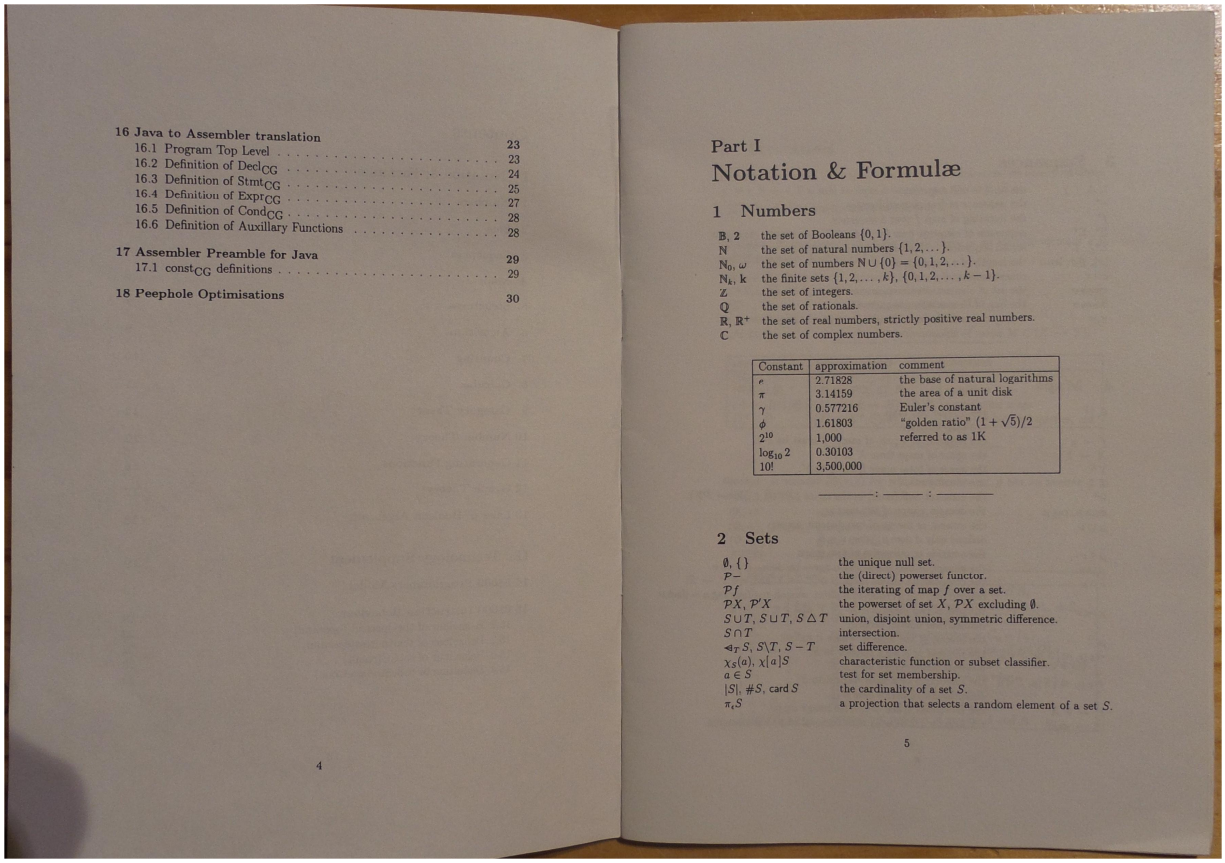
15.1 Behaviour of theOperand$_s$(*operand*) 22

15.2 Behaviour of theAddress(*operand*) 22

15.3 Behaviour of setCC(result) 23

15.4 Definition of codeSize(sizeCode) 23

Figure 3: Handbook of Mathematics table of contents page 1



16 Java to Assembler translation 23
16.1 Program Top Level 23
16.2 Definition of Decl_{CG} 24
16.3 Definition of Stmt_{CG} 25
16.4 Definition of Expr_{CG} 27
16.5 Definition of Cond_{CG} 28
16.6 Definition of Auxillary Functions 28

17 Assembler Preamble for Java 29
17.1 const_{CG} definitions 29

18 Peephole Optimisations 30

Part I

Notation & Formulæ

1 Numbers

$\mathbb{B}$, 2	the set of Booleans $\{0,1\}$.
$\mathbb{N}$	the set of natural numbers $\{1,2,\dots\}$.
$\mathbb{N}_0$, ω	the set of numbers $\mathbb{N}\cup\{0\} = \{0,1,2,\dots\}$.
$\mathbb{N}_k$, k	the finite sets $\{1,2,\dots,k\}$, $\{0,1,2,\dots,k-1\}$.
$\mathbb{Z}$	the set of integers.
$\mathbb{Q}$	the set of rationals.
$\mathbb{R}$, $\mathbb{R}^+$	the set of real numbers, strictly positive real numbers.
$\mathbb{C}$	the set of complex numbers.

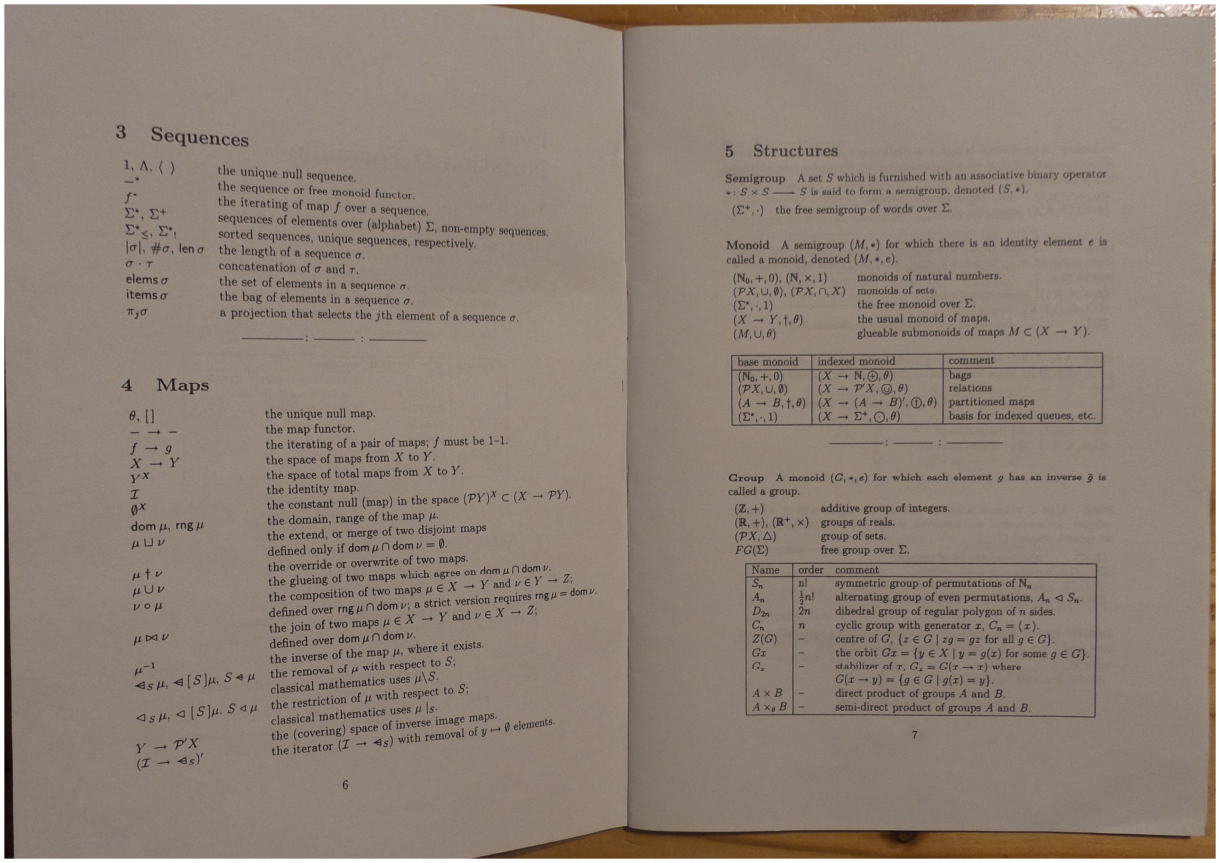
Constant	approximation	comment
e	2.71828	the base of natural logarithms
π	3.14159	the area of a unit disk
γ	0.577216	Euler's constant
ϕ	1.61803	"golden ratio" $(1+\sqrt{5})/2$
2^{10}	1,000	referred to as 1K
$\log_{10} 2$	0.30103	
$10!$	3,500,000	

———: ——— : ———

2 Sets

$\emptyset$, $\{\,\}$	the unique null set.
$\mathcal{P}-$	the (direct) powerset functor.
$\mathcal{P}f$	the iterating of map f over a set.
$\mathcal{P}X$, $\mathcal{P}'X$	the powerset of set X, $\mathcal{P}X$ excluding $\emptyset$.
$S\cup T$, $S\cup T$, $S\,\Delta\,T$	union, disjoint union, symmetric difference.
$S\cap T$	intersection.
$\lhd_T S$, $S\backslash T$, $S-T$	set difference.
$\chi_S(a)$, $\chi[a]S$	characteristic function or subset classifier.
$a\in S$	test for set membership.
$\lvert S\rvert$, $\#S$, $\mathrm{card}\,S$	the cardinality of a set S.
$\pi_r S$	a projection that selects a random element of a set S.

Figure 4: Handbook of Mathematics table of contents page 2 (left) and page 5 (right)



3 Sequences

1, Λ, $\langle\,\rangle$	the unique null sequence.
$-^*$	the sequence or free monoid functor.
f^*	the iterating of map f over a sequence.
Σ^*, Σ^+	sequences of elements over (alphabet) Σ, non-empty sequences.
$\Sigma^*_{\leq}$, $\Sigma^*_{\uparrow}$	sorted sequences, unique sequences, respectively.
$\lvert\sigma\rvert$, $\#\sigma$, $\mathsf{len}\,\sigma$	the length of a sequence σ.
$\sigma\cdot\tau$	concatenation of σ and τ.
$\mathsf{elems}\,\sigma$	the set of elements in a sequence σ.
$\mathsf{items}\,\sigma$	the bag of elements in a sequence σ.
$\pi_j\sigma$	a projection that selects the jth element of a sequence σ.

———: ——— : ———

4 Maps

θ, $[\,]$	the unique null map.
$-\to-$	the map functor.
$f\to g$	the iterating of a pair of maps; f must be 1–1.
$X\to Y$	the space of maps from X to Y.
Y^X	the space of total maps from X to Y.
$\mathcal{I}$	the identity map.
$\emptyset^X$	the constant null (map) in the space $(\mathcal{P}Y)^X\subset(X\to\mathcal{P}Y)$.
$\mathsf{dom}\,\mu$, $\mathsf{rng}\,\mu$	the domain, range of the map μ.
$\mu\sqcup\nu$	the extend, or merge of two disjoint maps defined only if $\mathsf{dom}\,\mu\cap\mathsf{dom}\,\nu=\emptyset$.
$\mu\dagger\nu$	the override or overwrite of two maps.
$\mu\cup\nu$	the glueing of two maps which agree on $\mathsf{dom}\,\mu\cap\mathsf{dom}\,\nu$.
$\nu\circ\mu$	the composition of two maps $\mu\in X\to Y$ and $\nu\in Y\to Z$; defined over $\mathsf{rng}\,\mu\cap\mathsf{dom}\,\nu$; a strict version requires $\mathsf{rng}\,\mu=\mathsf{dom}\,\nu$.
$\mu\bowtie\nu$	the join of two maps $\mu\in X\to Y$ and $\nu\in X\to Z$; defined over $\mathsf{dom}\,\mu\cap\mathsf{dom}\,\nu$.
μ^{-1}	the inverse of the map μ, where it exists.
$\lhd_S\mu$, $\lhd[S]\mu$, $S\lhd\mu$	the removal of μ with respect to S; classical mathematics uses $\mu\backslash S$.
$\lhd_S\mu$, $\lhd[S]\mu$, $S\lhd\mu$	the restriction of μ with respect to S; classical mathematics uses $\mu\,\lvert_S$.
$Y\to\mathcal{P}'X$	the (covering) space of inverse image maps.
$(\mathcal{I}\to\lhd_S)'$	the iterator $(\mathcal{I}\to\lhd_S)$ with removal of $y\mapsto\emptyset$ elements.

5 Structures

Semigroup A set S which is furnished with an associative binary operator $*\colon S\times S\longrightarrow S$ is said to form a semigroup, denoted $(S,*)$.

$(\Sigma^+,\cdot)$ the free semigroup of words over Σ.

Monoid A semigroup $(M,*)$ for which there is an identity element e is called a monoid, denoted $(M,*,e)$.

$(\mathbb{N}_0,+,0)$, $(\mathbb{N},\times,1)$	monoids of natural numbers.
$(\mathcal{P}X,\cup,\emptyset)$, $(\mathcal{P}X,\cap,X)$	monoids of sets.
$(\Sigma^*,\cdot,1)$	the free monoid over Σ.
$(X\to Y,\dagger,\theta)$	the usual monoid of maps.
$(M,\cup,\theta)$	glueable submonoids of maps $M\subset(X\to Y)$.

base monoid	indexed monoid	comment
$(\mathbb{N}_0,+,0)$	$(X\to\mathbb{N},\oplus,\theta)$	bags
$(\mathcal{P}X,\cup,\emptyset)$	$(X\to\mathcal{P}'X,\uplus,\theta)$	relations
$(A\to B,\dagger,\theta)$	$(X\to(A\to B)',①,\theta)$	partitioned maps
$(\Sigma^*,\cdot,1)$	$(X\to\Sigma^+,\bigcirc,\theta)$	basis for indexed queues, etc.

———: ——— : ———

Group A monoid $(G,*,e)$ for which each element g has an inverse $\bar{g}$ is called a group.

$(\mathbb{Z},+)$	additive group of integers.
$(\mathbb{R},+)$, $(\mathbb{R}^+,\times)$	groups of reals.
$(\mathcal{P}X,\Delta)$	group of sets.
$FG(\Sigma)$	free group over Σ.

Name	order	comment
S_n	$n!$	symmetric group of permutations of $\mathbb{N}_n$
A_n	$\frac{1}{2}n!$	alternating group of even permutations, $A_n\lhd S_n$.
D_{2n}	$2n$	dihedral group of regular polygon of n sides.
C_n	n	cyclic group with generator x, $C_n=\langle x\rangle$.
$Z(G)$	–	centre of G, $\{z\in G\mid zg=gz \text{ for all } g\in G\}$.
Gx	–	the orbit $Gx=\{y\in X\mid y=g(x) \text{ for some } g\in G\}$.
G_x	–	stabilizer of τ, $G_x=G(x\to x)$ where $G(x\to y)=\{g\in G\mid g(x)=y\}$.
$A\times B$	–	direct product of groups A and B.
$A\times_\theta B$	–	semi-direct product of groups A and B.

Figure 5: Handbook of Mathematics page 6 (left) and page 7 (right)

Semiring A set S which is both a multiplicative monoid $(S, \otimes, 1)$ and an additive monoid $(S, \oplus, 0)$ for which multiplication distributes over addition is called a semi-ring.

$(\mathbf{N}_0, +, \times, 0, 1)$ semi-ring of natural numbers.
$(\mathcal{P}X, \cup, \cap, \emptyset, X)$ semi-ring of sets.

Ring A set S which is both a multiplicative monoid $(S, \otimes, 1)$ and an additive group $(S, \oplus, 0)$ for which multiplication distributes over addition is called a ring.

$(\mathbf{Z}, +, \times, 0, 1)$ the ring of integers.
$(\mathcal{P}X, \Delta, \cap, \emptyset, X)$ ring of sets.
$\mathbf{Z}[x]$ the ring of polynomials with coefficients in $\mathbf{Z}$.

Field A set S which is both a multiplicative group $(S, \otimes, 1)$ and an additive group $(S, \oplus, 0)$ for which multiplication distributes over addition is called a field.

$\mathbf{Z}_p$ the finite field of integers modulo p, p a prime.
$(\mathbf{Q}, +, \times, 0, 1)$ the rationals.
$(\mathbf{R}, +, \times, 0, 1)$ the real numbers.
$(\mathbf{C}, +, \times, 0, 1)$ the complex numbers.

$$\mathbf{N} \longrightarrow \mathbf{Z} \longrightarrow \mathbf{Q} \longrightarrow \mathbf{R} \longrightarrow \mathbf{C}$$

Poset A set S which is furnished with an ordering relation $\leq$ which is reflexive, antisymmetric, and transitive, is said to form a partially ordered set (poset), denoted $(S, \preceq)$.

$(\mathbf{Z}, \leq)$ totally ordered set of integers.
$(\Sigma^*, \preceq)$ words with prefix ordering.
$(\mathcal{P}X, \subseteq)$ powerset of X with inclusion ordering.
$(\mathbf{div}(n), |)$ divisors of n with divides relation.

Lattice A poset S in which any two elements s and t have both a meet, $s \wedge t$, and a join, $s \vee t$, is called a lattice, denoted $(S, \wedge, \vee)$.

$(\mathcal{P}X, \cap, \cup)$ powerset of X.
$(\mathbf{div}(n), \gcd, \mathrm{lcm})$ divisors of n.

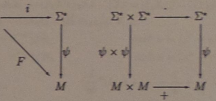
6 Algorithms

The Big O notation Let f be a function from $\mathbf{N}$ to $\mathbf{N}$. Then $f(n)$ is $O(g(n))$ if there is a positive constant k such that $f(n) \leq kg(n)$ for all n in $\mathbf{N}$ (with possibly a finite number of exceptions).

Assuming a machine can execute 10^6 operations per second:

$f(n)$	$n = 20$	$n = 40$	$n = 60$
n	0.00002 sec	0.00004 sec	0.00006 sec
n^2	0.0004 sec	0.0016 sec	0.0036 sec
n^3	0.008 sec	0.064 sec	0.216 sec
2^n	1.0 sec	12.7 days	366 centuries

Σ^***-morphisms** $(\Sigma^*, \cdot, 1) \xrightarrow{\psi} (M, +, e)$



$i(a) := \langle a \rangle$
$\psi \circ i = F$

naïve recursive form

$\psi(aw) = F(a) + \psi(w)$
$\psi(1) = e$

tail-recursive form

$\psi_w(m) = \psi_w(m + F(a))$
$\psi_1(m) = m$
$\psi(w) = \psi_w(e)$

naïve closed form

$\psi(w) = {}^+/F^*w$
$\psi(1) = e$

tail-recursive closed form

$\psi_w(m) = m + {}^+/F^*w$
$\psi_1(m) = m$

Figure 6: Handbook of Mathematics page 8 (left) and page 9 (right)

7 Counting

Pigeon-hole principle: if m objects are distributed into n boxes and $m > n$, then at least one box contains at least 2 objects.

Generalized pigeon-hole principle: if m objects are distributed into n boxes and $m > nr$, then at least one box contains at least $r + 1$ objects.

Sieve principle (Principle of inclusion/exclusion): If $A_1, A_2, \ldots, A_n$ are finite sets and α_i is the sum of the cardinalities of the intersections of the sets taken i at a time $(1 \leq i \leq n)$ then

$$|A_1 \cup A_2 \cup \cdots \cup A_n| := \alpha_1 - \alpha_2 + \alpha_3 - \cdots + (-1)^{n-1}\alpha_n$$

If $A_1, A_2, \ldots, A_n$ are subsets of a given set X, with $|X| = N$, then

$$|X \setminus \{A_1 \cup A_2 \cup \cdots \cup A_n\}| = |X| - |A_1 \cup A_2 \cup \cdots \cup A_n| = N - \alpha_1 + \alpha_2 - \alpha_3 + \cdots + (-1)^n \alpha_n$$

Derangements

$$d_n = n!(1 - \frac{1}{1!} + \frac{1}{2!} - \cdots + (-1)^n \frac{1}{n!})$$

Partitions If there are α_i parts of size i, then the partition of n is written

$$[1^{\alpha_1} 2^{\alpha_2} \cdots i^{\alpha_i} \cdots n^{\alpha_n}]$$

The number of partitions of n into k parts is given by

$$p_k(n) = p_k(n-k) + p_{k-1}(n-k) + \cdots + p_1(n-k)$$

Permutations The type of a permutation $\pi \in S_n$ in cycle notation is the partition of n:

$$[1^{\alpha_1} 2^{\alpha_2} \cdots n^{\alpha_n}]$$

The number of permutations of type $[1^{\alpha_1} 2^{\alpha_2} \cdots n^{\alpha_n}]$ is

$$\frac{n!}{1^{\alpha_1} 2^{\alpha_2} \cdots n^{\alpha_n} \alpha_1! \alpha_2! \cdots \alpha_n!}$$

8 Calculus

fundamental theorem

$$F(x) := \int_a^x f(u)du \quad \Rightarrow \quad F'(x) = f(x)$$

natural logarithm $\log x$, $\ln x$, $\log_e x$

$$\log x := \int_1^x \frac{1}{u} du, \qquad \log(1+x) = \sum_{n=1}^{\infty} (-1)^{n-1} \frac{x^n}{n!}$$

exponential function $\exp x$, e^x

$$x = \log y \Rightarrow y = e^x$$

$$e^x := \sum_{n=0}^{\infty} \frac{x^n}{n!}$$

$$e^{ix} = \cos x + i \sin x, \quad (i^2 = -1)$$
$$\cos x = (e^{ix} + e^{-ix})/2, \qquad \cosh x = (e^x + e^{-x})/2$$
$$\sin x = (e^{ix} - e^{-ix})/2i, \qquad \sinh x = (e^x - e^{-x})/2$$

Gamma function

$$\Gamma(z) := \int_0^\infty e^{-t} t^{z-1} dt, \quad (Re(z) > 0)$$

$$\Gamma(z+1) = z\Gamma(z)$$
$$\Gamma(1) = 1$$

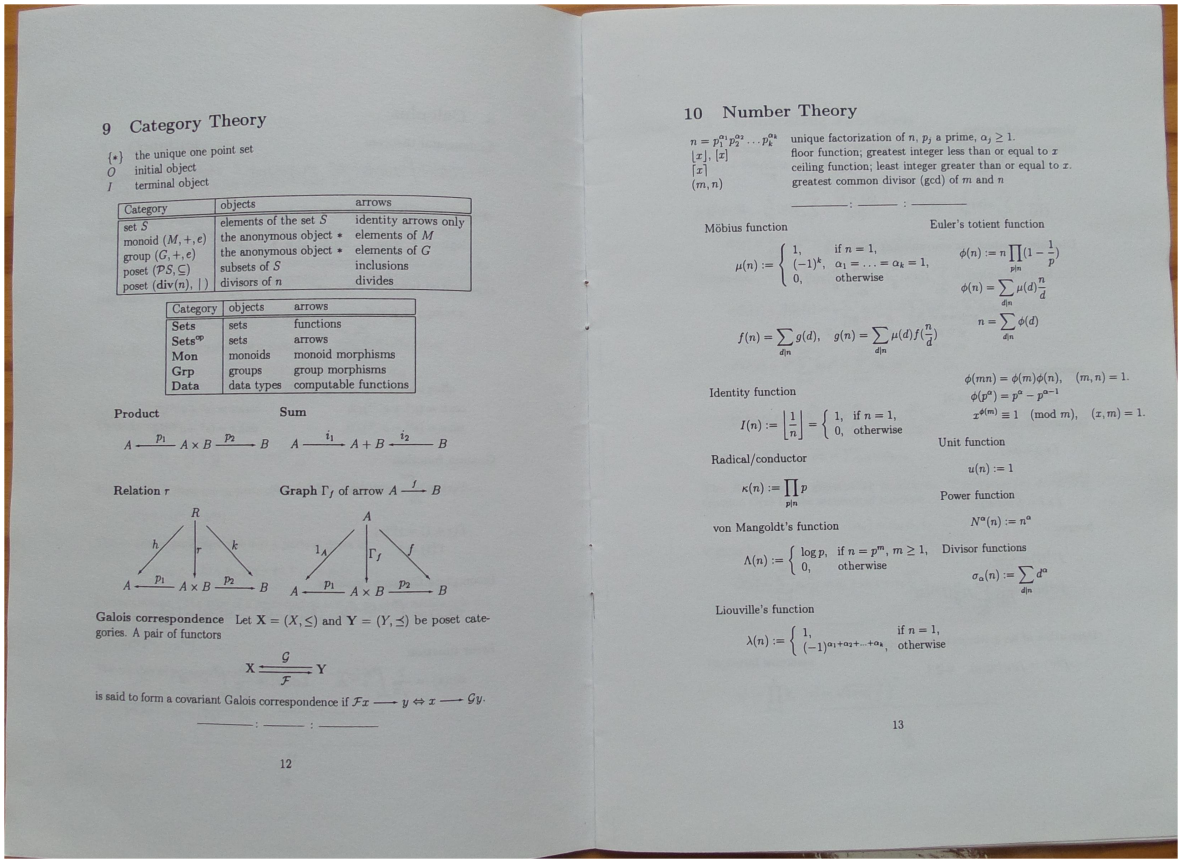
Incomplete Gamma function

$$\gamma(z, x) := \int_0^x e^{-t} t^{z-1} dt, \quad (Re(z) > 0), \qquad \gamma(z, \infty) = \Gamma(z)$$

Error function

$$\mathrm{Erf}(x) := \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt, \qquad \mathrm{Erfc}(x) := \frac{2}{\sqrt{\pi}} \int_x^\infty e^{-t^2} dt$$

Figure 7: Handbook of Mathematics page 10 (left) and page 11 (right)



9 Category Theory

$\{*\}$ the unique one point set
O initial object
I terminal object

Category	objects	arrows
set S	elements of the set S	identity arrows only
monoid $(M,+,e)$	the anonymous object $*$	elements of M
group $(G,+,e)$	the anonymous object $*$	elements of G
poset $(\mathcal{P}S,\subseteq)$	subsets of S	inclusions
poset $(\mathbf{div}(n),\mid)$	divisors of n	divides

Category	objects	arrows
Sets	sets	functions
$\mathbf{Sets}^{\mathbf{op}}$	sets	arrows
Mon	monoids	monoid morphisms
Grp	groups	group morphisms
Data	data types	computable functions

Product

$$A \xleftarrow{p_1} A \times B \xrightarrow{p_2} B$$

Sum

$$A \xrightarrow{i_1} A + B \xleftarrow{i_2} B$$

Relation r

Graph Γ_f of arrow $A \xrightarrow{f} B$

Galois correspondence Let $\mathbf{X} = (X, \leq)$ and $\mathbf{Y} = (Y, \preceq)$ be poset categories. A pair of functors

$$\mathbf{X} \underset{\mathcal{F}}{\overset{\mathcal{G}}{\rightleftarrows}} \mathbf{Y}$$

is said to form a covariant Galois correspondence if $\mathcal{F}x \longrightarrow y \Leftrightarrow x \longrightarrow \mathcal{G}y$.

———— : ———— : ————

10 Number Theory

$n = p_1^{\alpha_1} p_2^{\alpha_2} \dots p_k^{\alpha_k}$ unique factorization of n, p_j a prime, $\alpha_j \geq 1$.
$\lfloor x \rfloor$, $[x]$ floor function; greatest integer less than or equal to x
$\lceil x \rceil$ ceiling function; least integer greater than or equal to x.
(m,n) greatest common divisor (gcd) of m and n

———— : ———— : ————

Möbius function

$$\mu(n) := \begin{cases} 1, & \text{if } n = 1, \\ (-1)^k, & \alpha_1 = \dots = \alpha_k = 1, \\ 0, & \text{otherwise} \end{cases}$$

$$f(n) = \sum_{d|n} g(d), \quad g(n) = \sum_{d|n} \mu(d) f(\tfrac{n}{d})$$

Euler's totient function

$$\phi(n) := n \prod_{p|n} (1 - \frac{1}{p})$$

$$\phi(n) = \sum_{d|n} \mu(d) \frac{n}{d}$$

$$n = \sum_{d|n} \phi(d)$$

$$\phi(mn) = \phi(m)\phi(n), \quad (m,n) = 1.$$

$$\phi(p^a) = p^a - p^{a-1}$$

$$x^{\phi(m)} \equiv 1 \pmod{m}, \quad (x,m) = 1.$$

Identity function

$$I(n) := \left\lfloor \frac{1}{n} \right\rfloor = \begin{cases} 1, & \text{if } n = 1, \\ 0, & \text{otherwise} \end{cases}$$

Radical/conductor

$$\kappa(n) := \prod_{p|n} p$$

von Mangoldt's function

$$\Lambda(n) := \begin{cases} \log p, & \text{if } n = p^m,\ m \geq 1, \\ 0, & \text{otherwise} \end{cases}$$

Liouville's function

$$\lambda(n) := \begin{cases} 1, & \text{if } n = 1, \\ (-1)^{\alpha_1 + \alpha_2 + \dots + \alpha_k}, & \text{otherwise} \end{cases}$$

Unit function

$$u(n) := 1$$

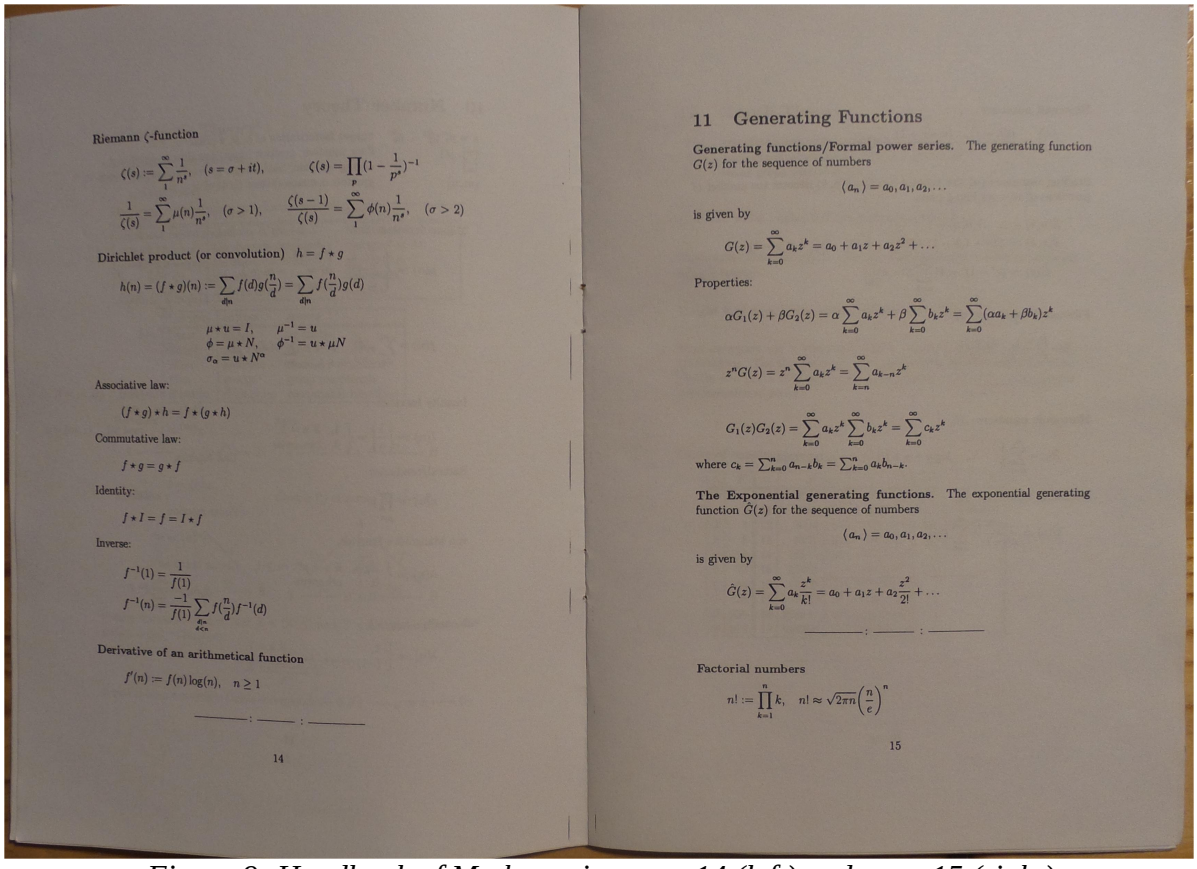
Power function

$$N^a(n) := n^a$$

Divisor functions

$$\sigma_a(n) := \sum_{d|n} d^a$$

Figure 8: Handbook of Mathematics page 12 (left) and page 13 (right)



Riemann ζ-function

$$\zeta(s) := \sum_1^\infty \frac{1}{n^s}, \quad (s = \sigma + it), \qquad \zeta(s) = \prod_p (1 - \frac{1}{p^s})^{-1}$$

$$\frac{1}{\zeta(s)} = \sum_1^\infty \mu(n) \frac{1}{n^s}, \quad (\sigma > 1), \qquad \frac{\zeta(s-1)}{\zeta(s)} = \sum_1^\infty \phi(n) \frac{1}{n^s}, \quad (\sigma > 2)$$

Dirichlet product (or convolution) $h = f * g$

$$h(n) = (f * g)(n) := \sum_{d|n} f(d) g(\tfrac{n}{d}) = \sum_{d|n} f(\tfrac{n}{d}) g(d)$$

$$\mu * u = I, \qquad \mu^{-1} = u$$
$$\phi = \mu * N, \qquad \phi^{-1} = u * \mu N$$
$$\sigma_a = u * N^a$$

Associative law:

$$(f * g) * h = f * (g * h)$$

Commutative law:

$$f * g = g * f$$

Identity:

$$f * I = f = I * f$$

Inverse:

$$f^{-1}(1) = \frac{1}{f(1)}$$

$$f^{-1}(n) = \frac{-1}{f(1)} \sum_{\substack{d|n \\ d<n}} f(\tfrac{n}{d}) f^{-1}(d)$$

Derivative of an arithmetical function

$$f'(n) := f(n) \log(n), \quad n \geq 1$$

———— : ———— : ————

11 Generating Functions

Generating functions/Formal power series. The generating function $G(z)$ for the sequence of numbers

$$\langle a_n \rangle = a_0, a_1, a_2, \dots$$

is given by

$$G(z) = \sum_{k=0}^\infty a_k z^k = a_0 + a_1 z + a_2 z^2 + \dots$$

Properties:

$$\alpha G_1(z) + \beta G_2(z) = \alpha \sum_{k=0}^\infty a_k z^k + \beta \sum_{k=0}^\infty b_k z^k = \sum_{k=0}^\infty (\alpha a_k + \beta b_k) z^k$$

$$z^n G(z) = z^n \sum_{k=0}^\infty a_k z^k = \sum_{k=n}^\infty a_{k-n} z^k$$

$$G_1(z) G_2(z) = \sum_{k=0}^\infty a_k z^k \sum_{k=0}^\infty b_k z^k = \sum_{k=0}^\infty c_k z^k$$

where $c_k = \sum_{k=0}^n a_{n-k} b_k = \sum_{k=0}^n a_k b_{n-k}$.

The Exponential generating functions. The exponential generating function $\hat{G}(z)$ for the sequence of numbers

$$\langle a_n \rangle = a_0, a_1, a_2, \dots$$

is given by

$$\hat{G}(z) = \sum_{k=0}^\infty a_k \frac{z^k}{k!} = a_0 + a_1 z + a_2 \frac{z^2}{2!} + \dots$$

———— : ———— : ————

Factorial numbers

$$n! := \prod_{k=1}^n k, \quad n! \approx \sqrt{2\pi n} \left(\frac{n}{e}\right)^n$$

Figure 9: Handbook of Mathematics page 14 (left) and page 15 (right)

Binomial numbers

$$\binom{n}{r} := \frac{n(n-1)\dots(n-r+1)}{r!}, \quad G(z) := (1+z)^n$$

Stirling numbers (of the second kind) $S(n,k)$ denotes the number of partitions of an n-set into k parts.

$$S(n,1) = 1, \quad S(n,n) = 1,$$
$$S(n,k) := S(n-1,k-1) + kS(n-1,k), \quad (2 \le k \le n-1)$$
$$\hat{G}(z) := (e^z - 1)^k = k! \sum_{n=k}^{\infty} S(n,k)\frac{z^n}{n!}$$

Fibonacci numbers F_n

$$F_1 = 1, \quad F_2 = 1,$$
$$F_n := F_{n-1} + F_{n-2}, \quad (n > 2)$$
$$G(z) := \frac{1}{1-z-z^2}$$

Harmonic numbers H_n

$$H_n := \sum_{k=1}^{n} \frac{1}{k}, \quad H_n = \log n + \gamma + O(\frac{1}{n}), \quad \gamma \approx 0.577216$$

Bernoulli numbers B_n

$$\hat{G}(z) := \frac{z}{e^z - 1} = \sum_{n=0}^{\infty} B_n \frac{z^n}{n!}$$

12 Graph Theory

K_n complete graph with n vertices.
$K_{m,n}$ complete bipartite graph with $m+n$ vertices
W_n wheel graph with $V = \mathbf{n+1}$

Eulerian walk: exists if G has at most 2 odd vertices.

Chromatic Number $\chi(G)$**:** the least k for which there is a vertex-colouring using k colours.

Height h **of** m**-ary tree with** l **leaves:** $h \ge \lceil \log_m l \rceil$.

Free category on a directed graph. Given a directed graph G with no relations. The free category on G, denoted G has the vertices of G as objects and paths of G as arrows. Composition of arrows is given by the concatenation of paths.

Table of constants

n	2^n	n	2^n	n	2^n	n	2^n
0	1	10	1024	20	1048576	30	1073741824
1	2	11	2048	21	2097152	31	2147483648
2	4	12	4096	22	4194304	32	4294967296
3	8	13	8192	23	8388608		
4	16	14	16384	24	16777216		
5	32	15	32768	25	33554432		
6	64	16	65536	26	67108864		
7	128	17	131072	27	134217728		
8	256	18	262144	28	268435456		
9	512	19	524288	29	536870912		

Figure 10: Handbook of Mathematics page 16 (left) and page 17 (right)

13 Laws of Boolean Algebra

No.	Law	Name	Huntington's Postulates
$B1$	$X + 0 = X$	identity	Postulate 2
$B2$	$X \cdot 1 = X$	identity	Postulate 2
$B3$	$X + 1 = 1$	zero	Theorem 2
$B4$	$X \cdot 0 = 0$	zero	Theorem 2
$B5$	$X + X = X$	idempotency	Theorem 1
$B6$	$X \cdot X = X$	idempotency	Theorem 1
$B7$	$X + \overline{X} = 1$	converse	Postulate 5
$B8$	$X \cdot \overline{X} = 0$	converse	Postulate 5
$B9$	$\overline{\overline{X}} = X$	double negation	Theorem 3 (involution)
$B10$	$X + Y = Y + X$	commutativity	Postulate 3
$B11$	$X \cdot Y = Y \cdot X$	commutativity	Postulate 3
$B12$	$X + (Y + Z) = (X + Y) + Z$	associativity	Theorem 4
$B13$	$X \cdot (Y \cdot Z) = (X \cdot Y) \cdot Z$	associativity	Theorem 4
$B14$	$X(Y + Z) = XY + XZ$	distributivity	Postulate 4
$B15$	$X + YZ = (X + Y)(X + Z)$	distributivity	Postulate 4
$B16$	$\overline{X+Y} = \overline{X} \cdot \overline{Y}$	DeMorgan's Law	Theorem 5
$B17$	$\overline{X \cdot Y} = \overline{X} + \overline{Y}$	DeMorgan's Law	Theorem 5
	$X + X \cdot Y = X$	absorption	Theorem 6
	$X \cdot (X + Y) = X$	absorption	Theorem 6

Part II
Technology Supplement

14 68000 Programmers Model

General Purpose Registers :
Longword Data Registers D0 ... D7
Longword Address Registers A0 ... A7

Special Purpose Registers :
Program Counter PC
Instruction Register IR
Word Status Register SR made up of System and User Bytes.
Least Significant Byte of SR: Condition Code Register: CCR
Longword Stack Pointer SP, equal to A7
Stack Pointer has System SSP and User USP versions.

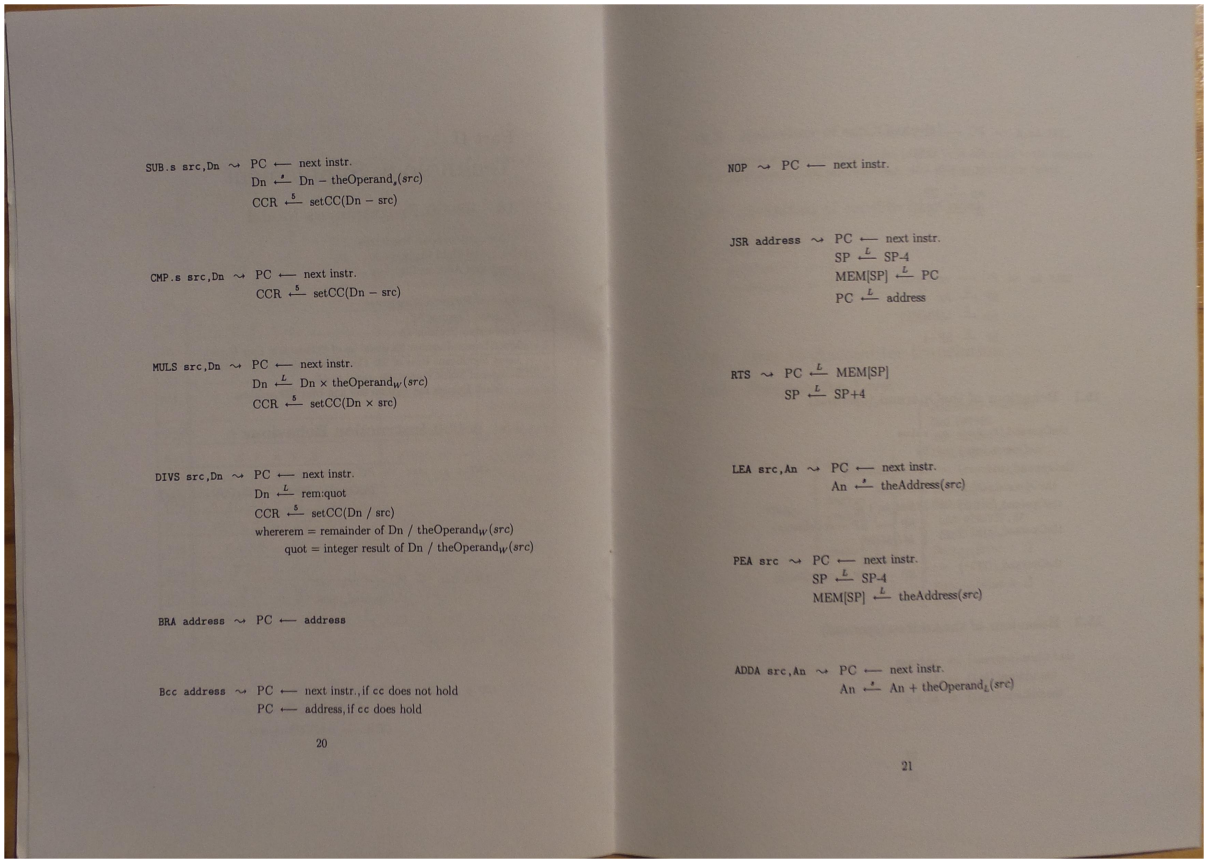
15 68000 Instruction Behaviour

`MOVE.s src,dst` $\rightsquigarrow$ PC $\longleftarrow$ next instr.
theOperand$_s$(dst) $\xleftarrow{s}$ theOperand$_s$(src)
CCR $\xleftarrow{5}$ setCC(src)

`CLR.s dst` $\rightsquigarrow$ PC $\longleftarrow$ next instr.
theOperand$_s$(dst) $\xleftarrow{s}$ 0
CCR $\xleftarrow{5}$ setCC(0)

`ADD.s src,Dn` $\rightsquigarrow$ PC $\longleftarrow$ next instr.
Dn $\xleftarrow{s}$ Dn + theOperand$_s$(src)
CCR $\xleftarrow{5}$ setCC(Dn + src)

Figure 11: Handbook of Mathematics page 18 (left) and page 19 (right)



SUB.s src,Dn ⇝ PC ⟵ next instr.
Dn $\xleftarrow{s}$ Dn − theOperand$_s$(*src*)
CCR $\xleftarrow{5}$ setCC(Dn − src)

CMP.s src,Dn ⇝ PC ⟵ next instr.
CCR $\xleftarrow{5}$ setCC(Dn − src)

MULS src,Dn ⇝ PC ⟵ next instr.
Dn $\xleftarrow{L}$ Dn × theOperand$_W$(*src*)
CCR $\xleftarrow{5}$ setCC(Dn × src)

DIVS src,Dn ⇝ PC ⟵ next instr.
Dn $\xleftarrow{L}$ rem:quot
CCR $\xleftarrow{5}$ setCC(Dn / src)
whererem = remainder of Dn / theOperand$_W$(*src*)
quot = integer result of Dn / theOperand$_W$(*src*)

BRA address ⇝ PC ⟵ address

Bcc address ⇝ PC ⟵ next instr.,if cc does not hold
PC ⟵ address,if cc does hold

20

NOP ⇝ PC ⟵ next instr.

JSR address ⇝ PC ⟵ next instr.
SP $\xleftarrow{L}$ SP-4
MEM[SP] $\xleftarrow{L}$ PC
PC $\xleftarrow{L}$ address

RTS ⇝ PC $\xleftarrow{L}$ MEM[SP]
SP $\xleftarrow{L}$ SP+4

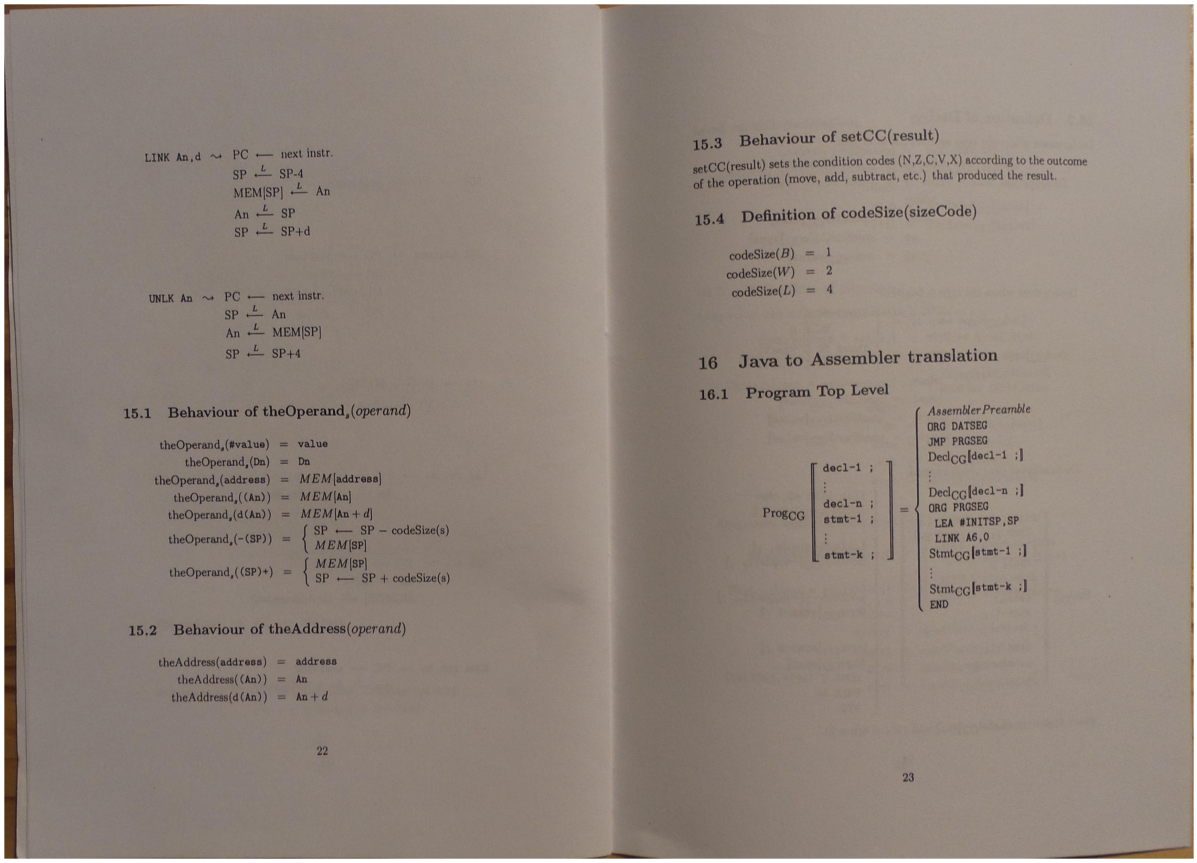
LEA src,An ⇝ PC ⟵ next instr.
An $\xleftarrow{s}$ theAddress(*src*)

PEA src ⇝ PC ⟵ next instr.
SP $\xleftarrow{L}$ SP-4
MEM[SP] $\xleftarrow{L}$ theAddress(*src*)

ADDA src,An ⇝ PC ⟵ next instr.
An $\xleftarrow{s}$ An + theOperand$_L$(*src*)

21

Figure 12: Handbook of Mathematics page 20 (left) and page 21 (right)



LINK An,d ⇝ PC ⟵ next instr.
SP $\xleftarrow{L}$ SP-4
MEM[SP] $\xleftarrow{L}$ An
An $\xleftarrow{L}$ SP
SP $\xleftarrow{L}$ SP+d

UNLK An ⇝ PC ⟵ next instr.
SP $\xleftarrow{L}$ An
An $\xleftarrow{L}$ MEM[SP]
SP $\xleftarrow{L}$ SP+4

15.1 Behaviour of theOperand$_s$(*operand*)

$$
\begin{aligned}
\text{theOperand}_s(\texttt{\#value}) &= \texttt{value}\\
\text{theOperand}_s(\texttt{Dn}) &= \texttt{Dn}\\
\text{theOperand}_s(\texttt{address}) &= \mathit{MEM}[\texttt{address}]\\
\text{theOperand}_s(\texttt{(An)}) &= \mathit{MEM}[\texttt{An}]\\
\text{theOperand}_s(\texttt{d(An)}) &= \mathit{MEM}[\texttt{An}+d]\\
\text{theOperand}_s(\texttt{-(SP)}) &= \begin{cases}\texttt{SP} \longleftarrow \texttt{SP} - \text{codeSize(s)}\\ \mathit{MEM}[\texttt{SP}]\end{cases}\\
\text{theOperand}_s(\texttt{(SP)+}) &= \begin{cases}\mathit{MEM}[\texttt{SP}]\\ \texttt{SP} \longleftarrow \texttt{SP} + \text{codeSize(s)}\end{cases}
\end{aligned}
$$

15.2 Behaviour of theAddress(*operand*)

$$
\begin{aligned}
\text{theAddress}(\texttt{address}) &= \texttt{address}\\
\text{theAddress}(\texttt{(An)}) &= \texttt{An}\\
\text{theAddress}(\texttt{d(An)}) &= \texttt{An}+d
\end{aligned}
$$

22

15.3 Behaviour of setCC(result)

setCC(result) sets the condition codes (N,Z,C,V,X) according to the outcome of the operation (move, add, subtract, etc.) that produced the result.

15.4 Definition of codeSize(sizeCode)

$$
\begin{aligned}
\text{codeSize}(B) &= 1\\
\text{codeSize}(W) &= 2\\
\text{codeSize}(L) &= 4
\end{aligned}
$$

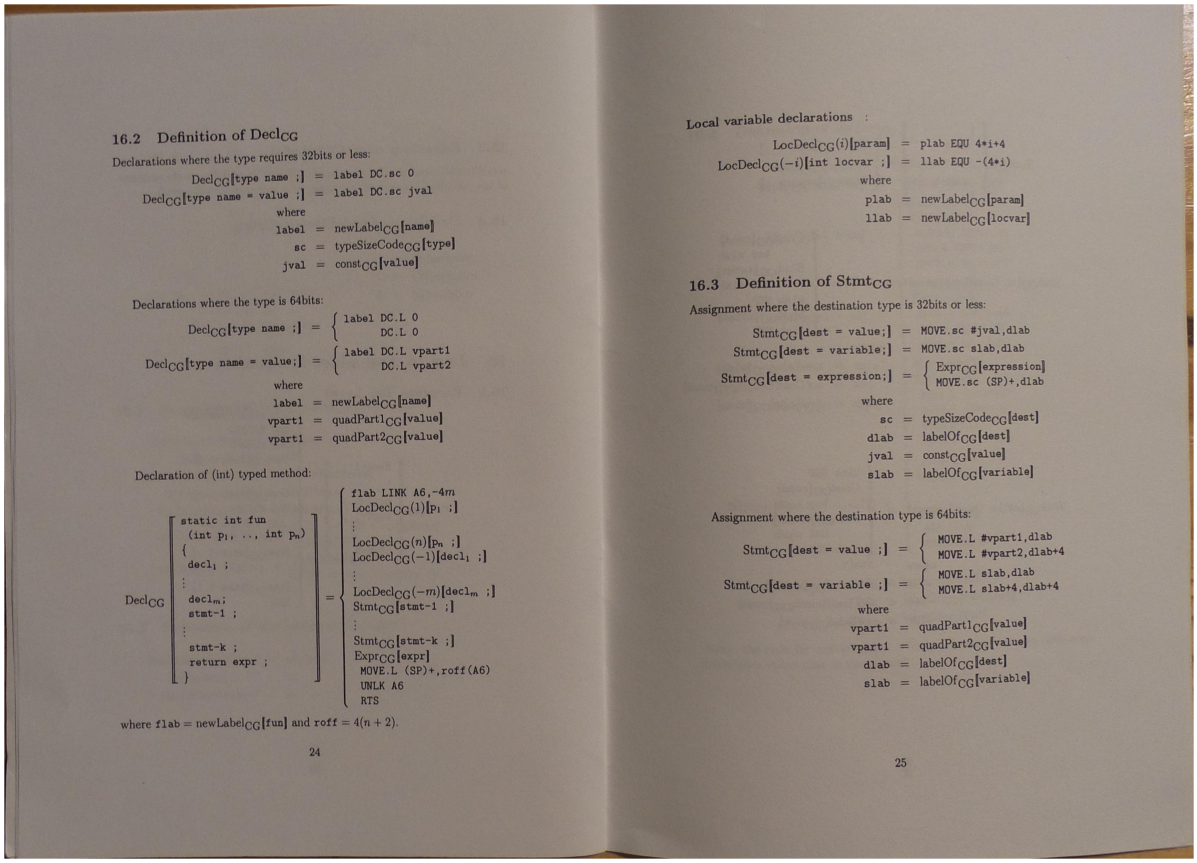
16 Java to Assembler translation

16.1 Program Top Level

$$
\text{Prog}_{CG}\left[\!\!\left[\begin{array}{l}\texttt{decl-1 ;}\\ \vdots\\ \texttt{decl-n ;}\\ \texttt{stmt-1 ;}\\ \vdots\\ \texttt{stmt-k ;}\end{array}\right]\!\!\right] = \left\{\begin{array}{l}\textit{AssemblerPreamble}\\ \texttt{ORG DATSEG}\\ \texttt{JMP PRGSEG}\\ \text{Decl}_{CG}\texttt{[decl-1 ;]}\\ \vdots\\ \text{Decl}_{CG}\texttt{[decl-n ;]}\\ \texttt{ORG PRGSEG}\\ \texttt{\ LEA \#INITSP,SP}\\ \texttt{\ LINK A6,0}\\ \text{Stmt}_{CG}\texttt{[stmt-1 ;]}\\ \vdots\\ \text{Stmt}_{CG}\texttt{[stmt-k ;]}\\ \texttt{END}\end{array}\right.
$$

23

Figure 13: Handbook of Mathematics page 22 (left) and page 23 (right)



16.2 Definition of Decl_{CG}

Declarations where the type requires 32bits or less:

$$\begin{aligned}
\text{Decl}_{CG}\texttt{[type name ;]} &= \texttt{label DC.sc 0} \\
\text{Decl}_{CG}\texttt{[type name = value ;]} &= \texttt{label DC.sc jval} \\
&\text{where} \\
\texttt{label} &= \text{newLabel}_{CG}\texttt{[name]} \\
\texttt{sc} &= \text{typeSizeCode}_{CG}\texttt{[type]} \\
\texttt{jval} &= \text{const}_{CG}\texttt{[value]}
\end{aligned}$$

Declarations where the type is 64bits:

$$\begin{aligned}
\text{Decl}_{CG}\texttt{[type name ;]} &= \left\{ \begin{array}{l} \texttt{label DC.L 0} \\ \texttt{\quad DC.L 0} \end{array} \right. \\
\text{Decl}_{CG}\texttt{[type name = value;]} &= \left\{ \begin{array}{l} \texttt{label DC.L vpart1} \\ \texttt{\quad DC.L vpart2} \end{array} \right. \\
&\text{where} \\
\texttt{label} &= \text{newLabel}_{CG}\texttt{[name]} \\
\texttt{vpart1} &= \text{quadPart1}_{CG}\texttt{[value]} \\
\texttt{vpart1} &= \text{quadPart2}_{CG}\texttt{[value]}
\end{aligned}$$

Declaration of (int) typed method:

$$\text{Decl}_{CG}\left[\!\left[\begin{array}{l} \texttt{static int fun} \\ \texttt{\quad (int } p_1\texttt{, .., int } p_n\texttt{)} \\ \texttt{\{} \\ \texttt{\quad } decl_1\texttt{ ;} \\ \vdots \\ \texttt{\quad } decl_m\texttt{;} \\ \texttt{\quad stmt-1 ;} \\ \vdots \\ \texttt{\quad stmt-k ;} \\ \texttt{\quad return expr ;} \\ \texttt{\}} \end{array} \right]\!\right] = \left\{ \begin{array}{l} \texttt{flab LINK A6,-}4m \\ \text{LocDecl}_{CG}(1)\texttt{[}p_1\texttt{ ;]} \\ \vdots \\ \text{LocDecl}_{CG}(n)\texttt{[}p_n\texttt{ ;]} \\ \text{LocDecl}_{CG}(-1)\texttt{[}decl_1\texttt{ ;]} \\ \vdots \\ \text{LocDecl}_{CG}(-m)\texttt{[}decl_m\texttt{ ;]} \\ \text{Stmt}_{CG}\texttt{[stmt-1 ;]} \\ \vdots \\ \text{Stmt}_{CG}\texttt{[stmt-k ;]} \\ \text{Expr}_{CG}\texttt{[expr]} \\ \texttt{\quad MOVE.L (SP)+,roff(A6)} \\ \texttt{\quad UNLK A6} \\ \texttt{\quad RTS} \end{array} \right.$$

where $\texttt{flab} = \text{newLabel}_{CG}\texttt{[fun]}$ and $\texttt{roff} = 4(n+2)$.

Local variable declarations :

$$\begin{aligned}
\text{LocDecl}_{CG}(i)\texttt{[param]} &= \texttt{plab EQU 4*i+4} \\
\text{LocDecl}_{CG}(-i)\texttt{[int locvar ;]} &= \texttt{llab EQU -(4*i)} \\
&\text{where} \\
\texttt{plab} &= \text{newLabel}_{CG}\texttt{[param]} \\
\texttt{llab} &= \text{newLabel}_{CG}\texttt{[locvar]}
\end{aligned}$$

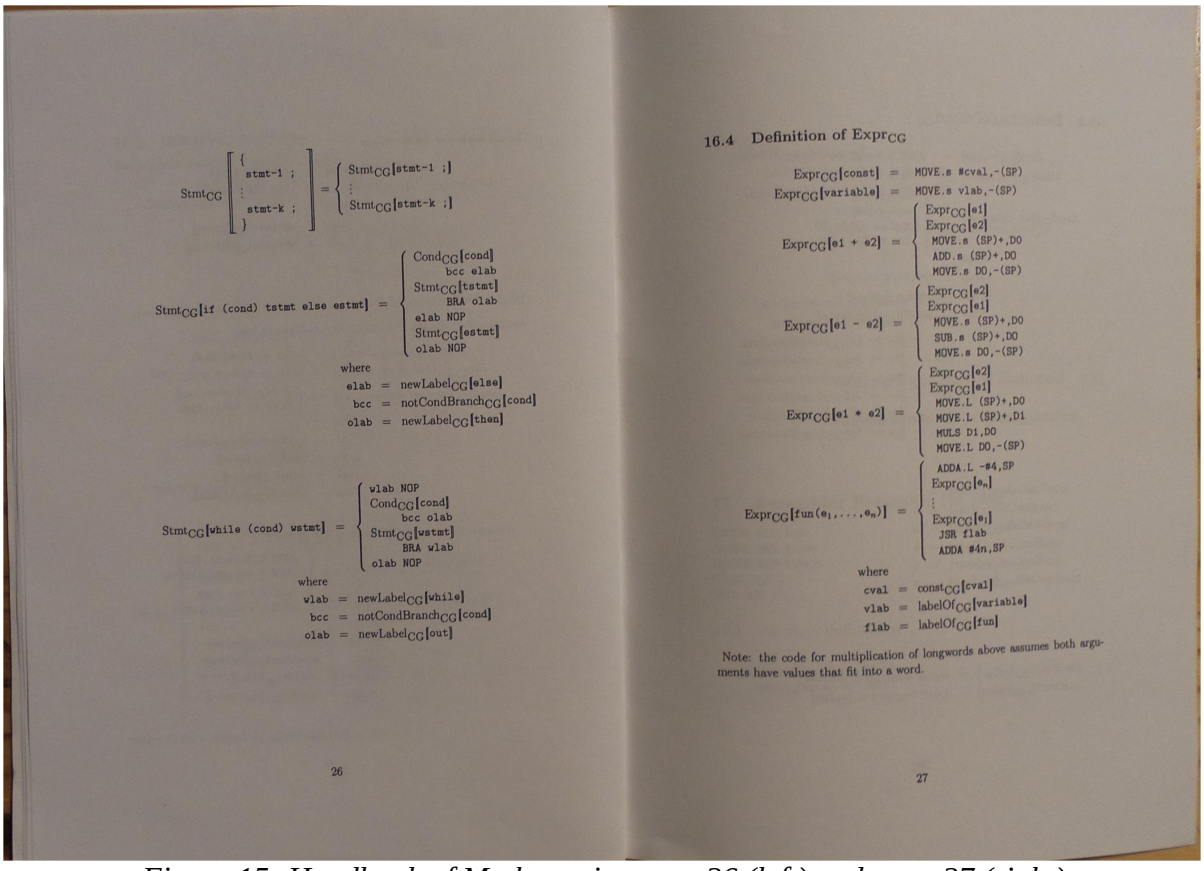
16.3 Definition of Stmt_{CG}

Assignment where the destination type is 32bits or less:

$$\begin{aligned}
\text{Stmt}_{CG}\texttt{[dest = value;]} &= \texttt{MOVE.sc \#jval,dlab} \\
\text{Stmt}_{CG}\texttt{[dest = variable;]} &= \texttt{MOVE.sc slab,dlab} \\
\text{Stmt}_{CG}\texttt{[dest = expression;]} &= \left\{ \begin{array}{l} \text{Expr}_{CG}\texttt{[expression]} \\ \texttt{MOVE.sc (SP)+,dlab} \end{array} \right. \\
&\text{where} \\
\texttt{sc} &= \text{typeSizeCode}_{CG}\texttt{[dest]} \\
\texttt{dlab} &= \text{labelOf}_{CG}\texttt{[dest]} \\
\texttt{jval} &= \text{const}_{CG}\texttt{[value]} \\
\texttt{slab} &= \text{labelOf}_{CG}\texttt{[variable]}
\end{aligned}$$

Assignment where the destination type is 64bits:

$$\begin{aligned}
\text{Stmt}_{CG}\texttt{[dest = value ;]} &= \left\{ \begin{array}{l} \texttt{MOVE.L \#vpart1,dlab} \\ \texttt{MOVE.L \#vpart2,dlab+4} \end{array} \right. \\
\text{Stmt}_{CG}\texttt{[dest = variable ;]} &= \left\{ \begin{array}{l} \texttt{MOVE.L slab,dlab} \\ \texttt{MOVE.L slab+4,dlab+4} \end{array} \right. \\
&\text{where} \\
\texttt{vpart1} &= \text{quadPart1}_{CG}\texttt{[value]} \\
\texttt{vpart1} &= \text{quadPart2}_{CG}\texttt{[value]} \\
\texttt{dlab} &= \text{labelOf}_{CG}\texttt{[dest]} \\
\texttt{slab} &= \text{labelOf}_{CG}\texttt{[variable]}
\end{aligned}$$



$$\text{Stmt}_{CG}\left[\!\left[\begin{array}{l} \texttt{\{} \\ \texttt{\quad stmt-1 ;} \\ \vdots \\ \texttt{\quad stmt-k ;} \\ \texttt{\}} \end{array} \right]\!\right] = \left\{ \begin{array}{l} \text{Stmt}_{CG}\texttt{[stmt-1 ;]} \\ \vdots \\ \text{Stmt}_{CG}\texttt{[stmt-k ;]} \end{array} \right.$$

$$\text{Stmt}_{CG}\texttt{[if (cond) tstmt else estmt]} = \left\{ \begin{array}{l} \text{Cond}_{CG}\texttt{[cond]} \\ \texttt{\quad bcc elab} \\ \text{Stmt}_{CG}\texttt{[tstmt]} \\ \texttt{\quad BRA olab} \\ \texttt{elab NOP} \\ \text{Stmt}_{CG}\texttt{[estmt]} \\ \texttt{olab NOP} \end{array} \right.$$

where

$$\begin{aligned}
\texttt{elab} &= \text{newLabel}_{CG}\texttt{[else]} \\
\texttt{bcc} &= \text{notCondBranch}_{CG}\texttt{[cond]} \\
\texttt{olab} &= \text{newLabel}_{CG}\texttt{[then]}
\end{aligned}$$

$$\text{Stmt}_{CG}\texttt{[while (cond) wstmt]} = \left\{ \begin{array}{l} \texttt{wlab NOP} \\ \text{Cond}_{CG}\texttt{[cond]} \\ \texttt{\quad bcc olab} \\ \text{Stmt}_{CG}\texttt{[wstmt]} \\ \texttt{\quad BRA wlab} \\ \texttt{olab NOP} \end{array} \right.$$

where

$$\begin{aligned}
\texttt{wlab} &= \text{newLabel}_{CG}\texttt{[while]} \\
\texttt{bcc} &= \text{notCondBranch}_{CG}\texttt{[cond]} \\
\texttt{olab} &= \text{newLabel}_{CG}\texttt{[out]}
\end{aligned}$$

16.4 Definition of Expr_{CG}

$$\begin{aligned}
\text{Expr}_{CG}\texttt{[const]} &= \texttt{MOVE.s \#cval,-(SP)} \\
\text{Expr}_{CG}\texttt{[variable]} &= \texttt{MOVE.s vlab,-(SP)} \\
\text{Expr}_{CG}\texttt{[e1 + e2]} &= \left\{ \begin{array}{l} \text{Expr}_{CG}\texttt{[e1]} \\ \text{Expr}_{CG}\texttt{[e2]} \\ \texttt{\quad MOVE.s (SP)+,D0} \\ \texttt{\quad ADD.s (SP)+,D0} \\ \texttt{\quad MOVE.s D0,-(SP)} \end{array} \right. \\
\text{Expr}_{CG}\texttt{[e1 - e2]} &= \left\{ \begin{array}{l} \text{Expr}_{CG}\texttt{[e2]} \\ \text{Expr}_{CG}\texttt{[e1]} \\ \texttt{\quad MOVE.s (SP)+,D0} \\ \texttt{\quad SUB.s (SP)+,D0} \\ \texttt{\quad MOVE.s D0,-(SP)} \end{array} \right. \\
\text{Expr}_{CG}\texttt{[e1 * e2]} &= \left\{ \begin{array}{l} \text{Expr}_{CG}\texttt{[e2]} \\ \text{Expr}_{CG}\texttt{[e1]} \\ \texttt{\quad MOVE.L (SP)+,D0} \\ \texttt{\quad MOVE.L (SP)+,D1} \\ \texttt{\quad MULS D1,D0} \\ \texttt{\quad MOVE.L D0,-(SP)} \end{array} \right. \\
\text{Expr}_{CG}\texttt{[fun(}e_1\texttt{,...,}e_n\texttt{)]} &= \left\{ \begin{array}{l} \texttt{\quad ADDA.L -\#4,SP} \\ \text{Expr}_{CG}\texttt{[}e_n\texttt{]} \\ \vdots \\ \text{Expr}_{CG}\texttt{[}e_1\texttt{]} \\ \texttt{\quad JSR flab} \\ \texttt{\quad ADDA \#}4n\texttt{,SP} \end{array} \right.
\end{aligned}$$

where

$$\begin{aligned}
\texttt{cval} &= \text{const}_{CG}\texttt{[cval]} \\
\texttt{vlab} &= \text{labelOf}_{CG}\texttt{[variable]} \\
\texttt{flab} &= \text{labelOf}_{CG}\texttt{[fun]}
\end{aligned}$$

Note: the code for multiplication of longwords above assumes both arguments have values that fit into a word.

Figure 15: Handbook of Mathematics page 26 (left) and page 27 (right)

16.5 Definition of Cond_{CG}

$$\text{Cond}_{CG}[\texttt{value}] = \texttt{MOVE.B \#cval,D0}$$
$$\text{Cond}_{CG}[\texttt{variable}] = \texttt{MOVE.B vlab,D0}$$
$$\text{Cond}_{CG}[\texttt{e1 relop e2}] = \begin{cases} \text{Expr}_{CG}[\texttt{e2}] \\ \text{Expr}_{CG}[\texttt{e1}] \\ \texttt{MOVE.s (SP)+,D0} \\ \texttt{CMP.s (SP)+,D0} \end{cases}$$

where

$\texttt{relop} \in \{\texttt{==}, \texttt{!=}, \texttt{<}, \texttt{<=}, \texttt{>}, \texttt{>=}\}$
$\texttt{cval} = \text{const}_{CG}[\texttt{value}]$
$\texttt{vlab} = \text{labelOf}_{CG}[\texttt{variable}]$

16.6 Definition of Auxillary Functions

$$\text{newLabel}_{CG}[\texttt{name}] = \text{new label for } [\texttt{name}]$$

$$\text{typeSizeCode}_{CG}[\texttt{float}] = \texttt{L}$$
$$\text{typeSizeCode}_{CG}[\texttt{int}] = \texttt{L}$$
$$\text{typeSizeCode}_{CG}[\texttt{short}] = \texttt{W}$$
$$\text{typeSizeCode}_{CG}[\texttt{byte}] = \texttt{B}$$
$$\text{typeSizeCode}_{CG}[\texttt{char}] = \texttt{B}$$
$$\text{typeSizeCode}_{CG}[\texttt{boolean}] = \texttt{B}$$

$$\text{quadPart1}_{CG}[\texttt{value}] = \text{higher longword of } [\texttt{value}]$$
$$\text{quadPart2}_{CG}[\texttt{value}] = \text{lower longword of } [\texttt{value}]$$

$$\text{labelOf}_{CG}[\texttt{globalvar}] = \texttt{label}$$
$$\text{labelOf}_{CG}[\texttt{param}] = \texttt{label(A6)}$$
$$\text{labelOf}_{CG}[\texttt{localvar}] = \texttt{label(A6)}$$

where

$\texttt{label} = \text{label assigned by newLabel}_{CG}$

$$\text{notCondBranch}_{CG}[\texttt{const}] = \texttt{BEQ}$$
$$\text{notCondBranch}_{CG}[\texttt{var}] = \texttt{BEQ}$$
$$\text{notCondBranch}_{CG}[\texttt{e1 == e2}] = \texttt{BNE}$$
$$\text{notCondBranch}_{CG}[\texttt{e1 != e2}] = \texttt{BEQ}$$
$$\text{notCondBranch}_{CG}[\texttt{e1 < e2}] = \texttt{BGE}$$
$$\text{notCondBranch}_{CG}[\texttt{e1 <= e2}] = \texttt{BGT}$$
$$\text{notCondBranch}_{CG}[\texttt{e1 > e2}] = \texttt{BLE}$$
$$\text{notCondBranch}_{CG}[\texttt{e1 >= e2}] = \texttt{BLT}$$

17 Assembler Preamble for Java

```
DATSEG EQU $400
PRGSEG EQU $1000
INITSP EQU $1F000
FALSE EQU 0
TRUE EQU 1
```

17.1 const_{CG} definitions

$$\text{const}_{CG}[\texttt{false}] = \texttt{FALSE}$$
$$\text{const}_{CG}[\texttt{true}] = \texttt{TRUE}$$

Figure 16: Handbook of Mathematics page 28 (left) and page 29 (right)

18 Peephole Optimisations

In what follows, multiple instructions are given on a single line, seperated by semi-colons, so

```
instr1 ;label2 instr2 ; instr3
```

would appear in an assembly file as

```
        instr1
label2  instr2
        instr3
```

1. Redundant Push-Pop `MOVE.s src,-(SP) ; MOVE.s (SP+),dst` ⇝ `MOVE.s src,dst`
2. Redundant Move `MOVE.s src,src` ⇝ `NOP` or nothing
3. Labelled No-Op. `label NOP ; instr` ⇝ `label instr`
4. Move Zero. `MOVE.s #0,dest` ⇝ `CLR.s dest`

Figure 17: Handbook of Mathematics page 30 (left)