2. NORMAL FORMS AND DATA STRUCTURES

Reference: Chapter 3 of

Geddes, Czapor, and Labahn

NORMAL FORMS

Three Levels of Abstraction

(i) The *object* level

abstract level where elements of a domain are considered as primitive objects

(ii) The *form* level

here we are concerned with how an object is represented in terms of some chosen 'basic symbols'

e.g., distinguish the following in Z[x,y]:

$$a(x,y) = 12x^2y - 4xy + 9x - 3$$

$$a(x,y) = (3x - 1)\,(4xy + 3)$$

$$a(x,y) = (12y)x^2 + (-4y + 9)x - 3$$

(iii) The *data structure* level

concerned with the organization of computer memory used in representing an object in a particular form

e.g., linked-lists versus static arrays versus dynamic vectors

The Problem of Simplification

Why simplify expressions?

(i) computer resources (space and time) may be wasted storing and manipulating unsimplified expressions

memory space is always finite

(ii) for human readability, want results to be expressed in simple form

Example: fully expanded polynomials

The following would simplify to zero:

$$(12x^2y - 4xy + 9x - 3) - (3x - 1)(4xy + 3)$$

But consider

$$(x + y)^{1000} - y^{1000}$$

- expanded form of this polynomial will contain a thousand terms

The expression

$$x^{1000} - y^{1000}$$

which is in expanded form is 'simpler' than a corresponding factored form in which $(x - y)$ is factored out

Zero Equivalence

Special case of general simplification problem in which we are concerned with recognizing when an expression is equivalent to zero

Example:

$$\log\left(\tan\left(\frac{x}{2}\right) + \sec\left(\frac{x}{2}\right)\right) - \operatorname{arcsinh}\left(\frac{\sin(x)}{1+\cos(x)}\right),$$

$$-1 \le x \le 1$$

(this is equal to 0)

- nontrivial transformations required

In 'sufficiently rich' class of expressions, zero equivalence problem is recursively undecidable

In many classes of expressions of practical interest, the problem can be solved

- **polynomials**
- **rational functions**
- **polynomials modulo side relations**

Normal Form and Canonical Form

E a set of expressions with σ an equivalence relation defined on E

σ partitions E into equivalence classes

Transformation functions f: E → E

Definition:

Let E be a set of expressions and let σ be an equivalence relation on E. A *normal function* for [E; σ] is a computable function f: E → E which satisfies the following properties:

(i) f(*a*) σ *a* for all *a in* E

(ii) *a* σ 0 *implies* f(*a*) ≡ f(0) for all *a in* E. □

Definition:

Let [E; σ] be as above. A *canonical function* for [E; σ] is a normal function f: E → E which satisfies the additional property:

(iii) *a* σ *b* *implies* f(*a*) ≡ f(*b*) for all *a*,*b in* E. □

Representation Issues

OBJECT LEVEL

↓

FORM LEVEL A:
normal/canonical forms

↓

FORM LEVEL B:
recursive/distrib. representation

↓

FORM LEVEL C:
sparse/dense representation

↓

FORM LEVEL D:
zero exponent representation

↓

DATA STRUCTURE LEVEL

namely the choice between recursive representation and distributive representation. In the *recursive representation* a polynomial $a(x_1, \ldots, x_v) \in D[x_1, \ldots, x_v]$ is represented as

$$a(x_1, \ldots, x_v) = \sum_{i=0}^{\deg_1(a(\mathbf{x}))} a_i(x_2, \ldots, x_v)\, x_1^i$$

(i.e. as an element of the domain $D[x_2, \ldots, x_v][x_1]$) where, recursively, the polynomial coefficients $a_i(x_2, \ldots, x_v)$ are represented as elements of the domain $D[x_3, \ldots, x_v][x_2]$, and so on so that ultimately the polynomial $a(x_1, \ldots, x_v)$ is viewed as an element of the domain $D[x_v][x_{v-1}] \cdots [x_1]$. An example of a polynomial from the domain $\mathbf{Z}[x,y,z]$ expressed in the recursive representation is:

$$a(x,y,z) = (3y^2+(-2z^3)y+5z^2)x^2 + 4x + ((-6z+1)y^3+3y^2+(z^4+1)). \tag{3.8}$$

In the *distributive representation* a polynomial $a(\mathbf{x}) \in D[\mathbf{x}]$ is represented as

$$a(\mathbf{x}) = \sum_{\mathbf{e} \in \mathbf{N}^v} a_{\mathbf{e}} \mathbf{x}^{\mathbf{e}}$$

where $a_{\mathbf{e}} \in D$. For example, the polynomial $a(x,y,z) \in \mathbf{Z}[x,y,z]$ given in (3.8) could be expressed in the distributive representation as

$$a(x,y,z) = 3x^2y^2-2x^2yz^3+5x^2z^2+4x-6y^3z+y^3+3y^2+z^4+1. \tag{3.9}$$

Normal Forms for Polynomials

Definition:

An *expanded normal form* **for polynomial expressions can be specified by the normal function**

f_1: (i) multiply out all products

(ii) collect terms of the same degree

An *expanded canonical form* **for polynomial expressions can be specified by the canonical function**

f_2: apply f_1, then

(iii) rearrange the terms into descending orde

Also, can define *factored normal form*

f_3 **- leave factors that exist, expanding each factor to ensure zero recognition**

And a *factored canonical form*

f_4 **- fully factor the expression**

Example 3.1. Let E be the domain $\mathbf{Z}[x]$ of univariate polynomials over the integers. Consider the normal functions f_1 and f_2 specified as follows:

f_1: (i) multiply out all products of polynomials;

(ii) collect terms of the same degree.

f_2: (i) multiply out all products of polynomials;

(ii) collect terms of the same degree;

(iii) rearrange the terms into descending order of their degrees.

Then f_1 is a normal function which is not a canonical function and f_2 is a canonical function. A normal form for polynomials in $\mathbf{Z}[x]$ corresponding to f_1 is

$$a_1x^{e_1} + a_2x^{e_2} + \cdots + a_mx^{e_m} \quad \text{with } e_i \neq e_j \text{ when } i \neq j.$$

A canonical form for polynomials in $\mathbf{Z}[x]$ corresponding to f_2 is

$$a_1x^{e_1} + a_2x^{e_2} + \cdots + a_mx^{e_m} \text{ with } e_i < e_j \text{ when } i > j. \quad \bullet$$

)efinition 3.5. A *factored normal form* for polynomial expressions in a domain)$[x_1, \ldots, x_v]$ can be specified by the normal function

f_3: if the expression is in the product form $\prod_{i=1}^{k} p_i$, $p_i \in D[x_1, \ldots, x_v]$ for $i = 1,2, \ldots, k$, where no p_i is itself in product form, then replace the expression by $\prod_{i=1}^{k} f_2(p_i)$ where f_2 is the canonical function defined in Definition 3.4 and where the latter product is understood to be zero if any of its factors is zero.

\ *factored canonical form* for polynomial expressions in a domain $D[x_1, \ldots, x_v]$ (assuming 1at D is a UFD) can be specified by the canonical function

f_4: apply f_3 and if the result is nonzero then factorize each $f_2(p_i)$ into its unit normal factorization (according to Definition 2.13) and collect factors to obtain the unit normal factorization of the complete expression (made unique by imposing a pre-specified ordering on the factors).

●

Example 3.3. Let $a(x,y) \in \mathbf{Z}[x,y]$ be the expression

$$a(x,y) = ((x^2 - xy + x) + (x^2 + 3)(x - y + 1)) \cdot$$
$$((y^3 - 3y^2 - 9y - 5) + x^4 (y^2 + 2y + 1)).$$

Using the distributive representation for writing polynomials, an expanded normal forr obtained by applying f_1 of Definition 3.4 to $a(x,y)$ might be (depending on the order i which the multiplication algorithm produces the terms):

$$f_1(a(x,y)) = 5x^2y^3 + 3x^2y^2 - 13x^2y - 10x^2 + 3x^6y + 2x^6 - xy^4 + 7xy^3$$
$$- 3xy^2 - 31xy - x^5y^3 + 2x^5y^2 + 7x^5y - 20x + 4x^5 + x^3y^3$$
$$- 3x^3y^2 - 9x^3y - 5x^3 + x^7y^2 + 2x^7y + x^7 - x^2y^4 - x^6y^3$$
$$- 3y^4 + 12y^3 + 18y^2 - 12y - 3x^4y^3 - 3x^4y^2 + 3x^4y - 15 + 3x^4.$$

The expanded canonical form obtained by applying f_2 of Definition 3.4 to $a(x,y)$ is

$$f_2(a(x,y)) = x^7y^2 + 2x^7y + x^7 - x^6y^3 + 3x^6y + 2x^6 - x^5y^3 + 2x^5y^2$$
$$+ 7x^5y + 4x^5 - 3x^4y^3 - 3x^4y^2 + 3x^4y + 3x^4 + x^3y^3 - 3x^3y^2$$
$$- 9x^3y - 5x^3 - x^2y^4 + 5x^2y^3 + 3x^2y^2 - 13x^2y - 10x^2 - xy^4$$
$$+ 7xy^3 - 3xy^2 - 31xy - 20x - 3y^4 + 12y^3 + 18y^2 - 12y - 15.$$

Applying, respectively, f_3 and f_4 of Definition 3.5 to $a(x,y)$ yields the factored normal form

$$f_3(a(x,y)) = (x^3 - x^2y + 2x^2 - xy + 4x - 3y + 3) \cdot$$
$$(x^4y^2 + 2x^4y + x^4 + y^3 - 3y^2 - 9y - 5)$$

and the factored canonical form

$$f_4(a(x,y)) = (x - y + 1)(x^2 + x + 3)(x^4 + y - 5)(y + 1)^2.$$

On the other hand, the canonical function f_2 would "simplify" th

expression

$$a(x,y) = (x-y)(x^{19} + x^{18}y + x^{17}y^2 + x^{16}y^3 + x^{15}y^4 + x^{14}y^5 + x^{13}y^6 + x^{12}y^7 + x^{11}y^8 + x^{10}y^9 + x^9y^{10} + x^8y^{11} + x^7y^{12} + x^6y^{13} + x^5y^{14} + x^4y^{15} + x^3y^{16} + x^2y^{17} + xy^{18} + y^{19})$$

into the expression

$$f_2(a(x,y)) = x^{20} - y^{20}$$

while the normal function f_3 would leave $a(x,y)$ unchanged.

Normal Forms for Rational Functions

Normalizing operations:

- form common denominator
- remove GCD from numer and denom
- make denominator unit normal
- put numer and denom into a normal form

The latter operation allows some choices, thus leading to different normal forms:

- expanded/expanded
- factored/factored
- factored/expanded
- expanded/factored

The *expanded/factored* normal form has been found to be particularly useful

expanded / expanded

Definition 3.6. An *expanded canonical form* for rational expressions in a field $D(x_1, \ldots, x_\nu)$ can be specified by the canonical function

f_5: (i) [form common denominator] put the expression into the form a/b where $a, b \in D[x_1, \ldots, x_\nu]$ by performing the arithmetic operations according to equations (2.42)-(2.43);

(ii) [satisfy condition (2.44): remove GCD] compute $g = \mathrm{GCD}(a, b) \in D[x_1, \ldots, x_\nu]$ (e.g. by using Algorithm 2.3) and replace the expression a/b by a'/b' where $a = a'g$ and $b = b'g$;

(iii) [satisfy condition (2.45): unit normalize] replace the expression a'/b' by a''/b'' where $a'' = a' \cdot (\mathrm{u}(b'))^{-1}$ and $b'' = b' \cdot (\mathrm{u}(b'))^{-1}$;

(iv) [satisfy condition (2.46): make polynomials canonical] replace the expression a''/b'' by $f_2(a'')/f_2(b'')$ where f_2 is the canonical function of Definition 3.4.

●

As in the case of polynomials, it can be useful to consider non-canonical normal forms for rational functions (and indeed more general forms which are neither canonical nor normal). We will not set out formal definitions of normal forms for rational expressions but several possible normal forms can be outlined as follows:

factored/factored: numerator and denominator both in factored normal form;

factored/expanded: numerator in factored normal form and denominator in expanded canonical form;

expanded/factored: numerator in expanded canonical form and denominator in factored normal form.

In this notation the expanded canonical form of Definition 3.6 would be denoted as *expanded/expanded*. In the above we are assuming that conditions (2.44) and (2.45) are satisfied but that condition (2.46) is not necessarily satisfied. Noting that to satisfy condition

DATA STRUCTURES

Multiprecision Integers

$$d = \sum_{i=0}^{\ell-1} d_i \beta^i \quad (\ell \text{ is the length})$$

for some choice of base β.

Typical choices: $\beta = 2^{31}$ (32-bit word)

$\beta = 10^9$ (32-bit word)

$\beta = 10^4$ (fits in half of 32-bit word)

Data Structure Choices

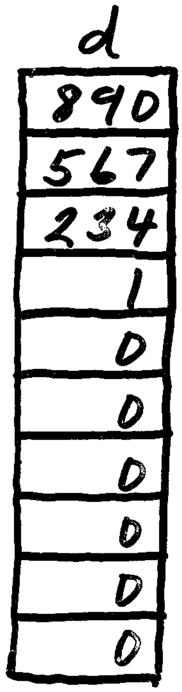
- linked lists (used by LISP-based systems)
- static arrays (ALTRAN)
- dynamic vectors (Maple)

Example: $d = 1234567890$

using base $\beta = 10^3$

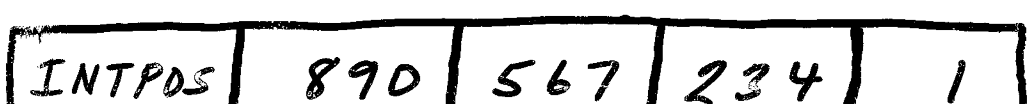
Linked List

Static Array of Length 10



(Fixed-length array for each integer)

Dynamic Vector



HEADER encodes:
- type (positive integer)
- length ($\ell + 1 = 5$)
- simplification status
- garbage collection status

Multivariate Polynomials

Linked List: (Recursive Representation)

COEF_LINK	EXPONENT	NEXT_LINK

with a **header node**:

INDET_LINK	FIRST_LINK

$$D[x_1, \ldots, x_v] = D[x_2, \ldots, x_v][x_1]$$

"main variable"

Example:

$$A(x,y,z) = 3x^2y^2 - 2x^2yz^3 + 5x^2z^2 + 4x - z^4 + 1 \in \underline{Z}[x,y,z]$$

$$= (3y^2 + (-2z^3)y + 5z^2)x^2 + (4)x + (-z^4 + 1) \in \underline{Z}[z][y][x]$$

Example 3.4. Let $a(x,y,z) \in \mathbf{Z}[x,y,z]$ be the polynomial

$$a(x,y,z) = 3x^2y^2 - 2x^2yz^3 + 5x^2z^2 + 4x - z^4 + 1$$

or, in recursive representation,

$$a(x,y,z) = (3y^2 + (-2z^3)y + 5z^2)x^2 + 4x + (-z^4 + 1).$$

Using the linked list data structure just described, the recursive form of the polynomi $a(x,y,z)$ is represented as shown in Figure 3.2.

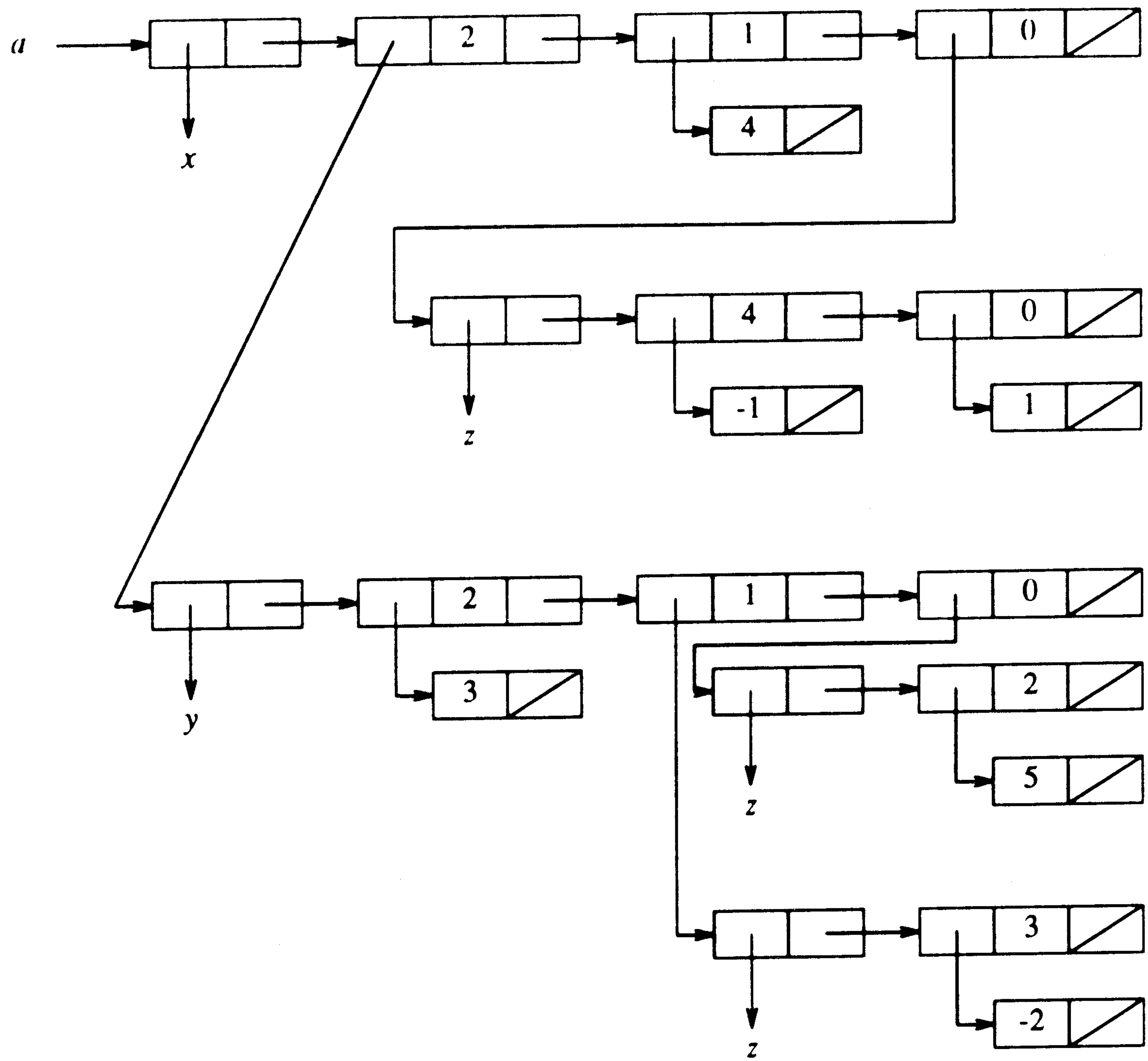
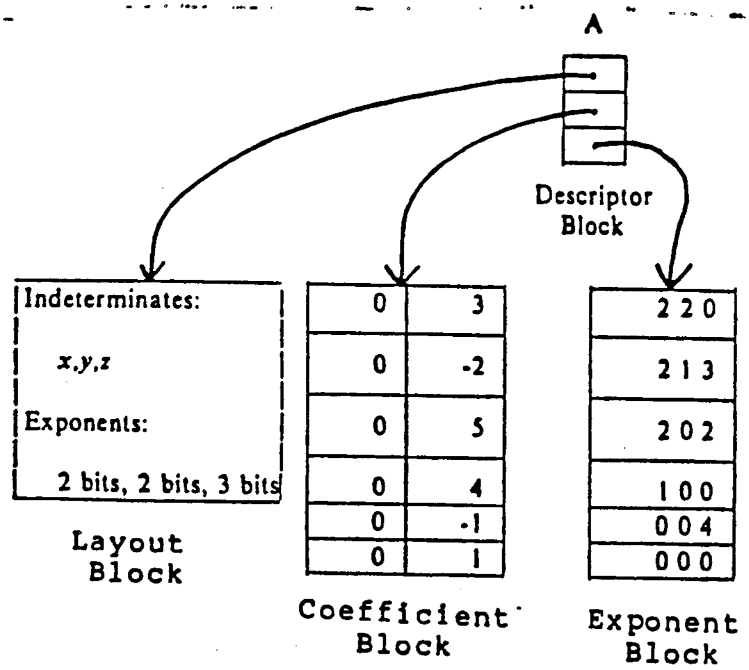


Figure 3.2. A linked list representation.

Example 3.5.

Let $A(x,y,z) \in Z[x,y,z]$ be the polynomial given in Example 3.4. Using the descriptor block data structure just described, suppose that the declared maximum degrees are degree 2 in x, degree 3 in y, and degree 4 in z. Suppose further that multiprecision integers are represented using base $\beta = 10^3$ and with pre-specified length $l = 2$. Then the polynomial $A(x,y,z)$ is represented as follows.

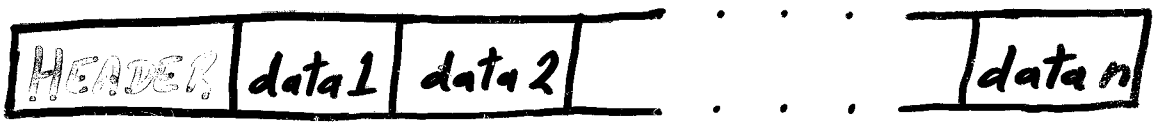


The layout block illustrated in this example indicates that the information stored in the actual layout block would include pointers to the names of the indeterminates and also a specification of the fact that each vector of three exponents is packed into one computer word, with the exponent of x occupying 2 bits, the exponent of y occupying 2 bits, and the exponent of z occupying 3 bits. In practice there is also a guard bit in front of each exponent (to facilitate performing arithmetic operations on the exponents) so this specification implies that the computer word consists of at least 10 bits. The coefficient block illustrated here reflects the specification of $l = 2$ words for each multiprecision intege: although $l = 1$ would have sufficed in this particular example. □

DYNAMIC VECTORS

Examples of Maple's Data Representation

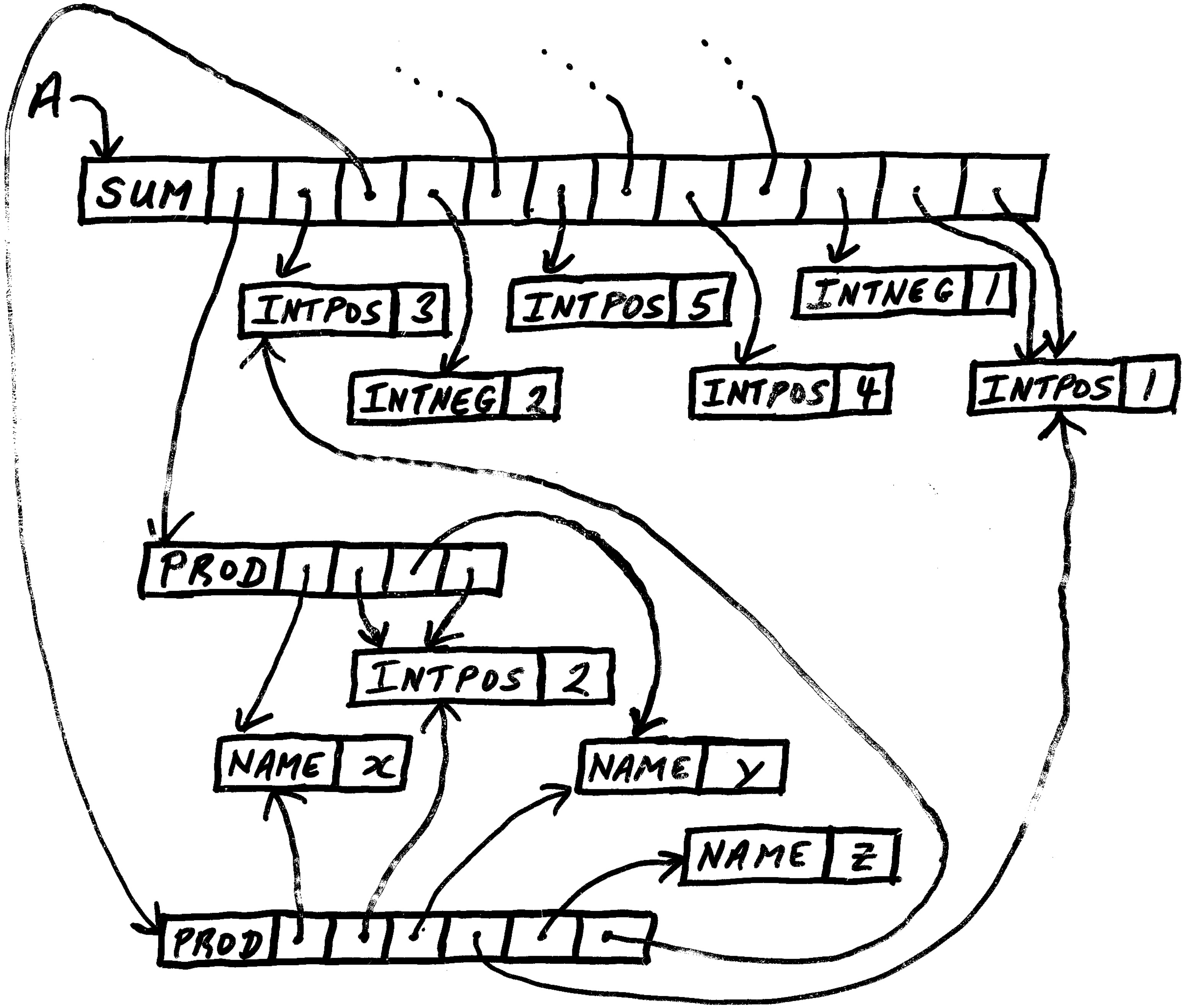
General format:



HEADER field: encodes

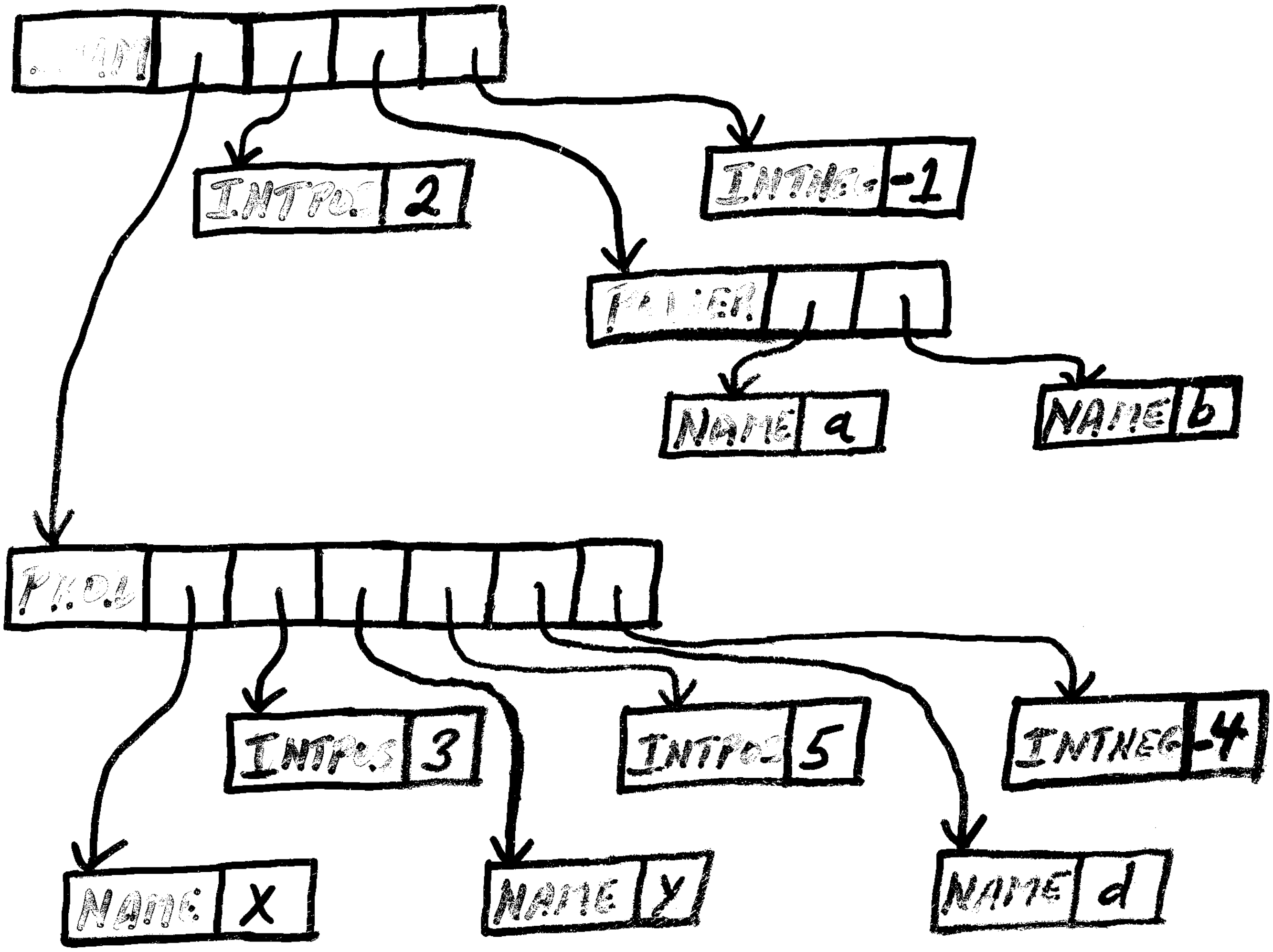
- type
- length n+1
- 1 bit simplification status
- 2 bits garbage collection status

Dynamic Vectors: (Distributive Representation)



$$A(x, y, z) = 3x^2y^2 - 2x^2yz^3 + 5x^2z^2 + 4x - z^4 + 1$$

e.g. $$\frac{2\,x^3 y^5}{d^4} - a^b$$



An Example using dismantle

$$r := \frac{2 \cdot x^3 \cdot y^5}{d^4} - a^b$$

$$r := \frac{2\, x^3\, y^5}{d^4} - a^b$$

dismantle(r)

```
SUM(5)
   PROD(7)
      NAME(4): x
      INTPOS(2): 3
      NAME(4): y
      INTPOS(2): 5
      NAME(4): d
      INTNEG(2): -4
   INTPOS(2): 2
   POWER(3)
      NAME(4): a
      NAME(4): b
   INTNEG(2): -1
```

dismantle$(2 \cdot x + 1)$

```
SUM(5)
   NAME(4): x
   INTPOS(2): 2
   INTPOS(2): 1
   INTPOS(2): 1
```