Makanin's Algorithm

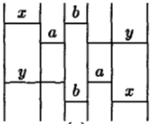
April 2, 2023

Basic definitions

- $\rightarrow$ Variables : $\nu = \{v_1, v_2 ... v_r\}$
- $\rightarrow$ Constants : $\mathcal{C} = \{a_1, a_2 .... a_r\}$
- $\rightarrow$ Word : $\mathcal{W}$ = Finite sequence of elements of $\mathcal{C} \cup \nu$
- $\rightarrow$ Word length $= |\mathcal{W}|$
- $\rightarrow$ Exponent of periodicity of $(\mathcal{W})$: maximal number 'p' such that $\mathcal{W} = uw^p z$
- $\rightarrow$ Word equation$(\mathcal{E})$: $w_1 = w_2$ for some words w_1 and w_2
- $\rightarrow$ Length of word equation : $|\mathcal{E}| = |w_1| + |w_2|$
- $\rightarrow$ Unifier of $\mathcal{E}$: Sequence of words U $= u_1, u_2 .... u_k$ such that if we replace the variables $x_1, x_2 .... x_k$ with corresponding element of unifier (i.e. replace x_1 by u_1); w_1 becomes equal to w_2.
- $\rightarrow$ Exponent of periodicity of U : maximal exponent of periodicity of words U_i

Graphical Representation

$$\mathcal{E} : xaby = ybax$$



(a)

→ In this representation the equation has a solution if there is a way to overlap both sides (top and bottom) such that the word between the boundaries are same

→ The vertical lines will be called boundaries.

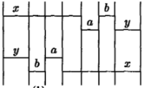
→ The length of horizontal lines for constant is always 1 and is unknown for variables.

→ The way to solve is to replace equals by equals like $y = xa$; then replace all occurrences of y and guess the boundaries again until we have solved.

→ Basic idea of algorithm is to guess the boundaries, replace from left to right; then guess boundaries again and so on.

Problems with this

- → The number of occurrences of some variables starts growing after replacement
- → What to do in cases where there is no evident replacement



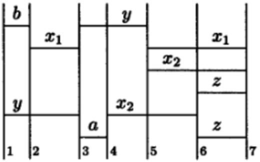
(b)

- → You can go on forever (cf.the equation $xa == ax$).

Solution

→ Limit the number of occurrences of a variable in an equation to 2. This avoids the problems of variables growing after replacement.

$$bxyx = yaxz \rightarrow \begin{cases} bx_1yx_1 = yax_2z \\ x_1 = x_2 \end{cases}$$



Generalized Equations Definition

→ The idea is to encode graphical representations into an equation.

(1) Two finite sets $\mathcal{C}$ and $\mathcal{X}$, the *labels*.
(2) A finite linear ordered set $(BD, \preceq)$, the *boundaries*.
(3) A finite set BS of *bases*. A *base bs* has the form $(t, (e_1, \ldots, e_n))$, where $n \geq 2$, $t \in \mathcal{C} \cup \mathcal{X}$, and $E_{bs} = (e_1, \ldots, e_n)$ is a sequence of boundaries ordered by $\preceq$.

subject to the following conditions:[2]

(C1) For each $x \in \mathcal{X}$, there are exactly two bases with label x, called *duals*, and (abusing notation) denoted by x and $\bar{x}$ respectively. Also, their respective boundary sequences $E_x, E_{\bar{x}}$ must have the same length.
(C2) For each base bs with $t \in \mathcal{C}$, the boundary sequence E_{bs} has exactly two elements and they are consecutive in the order $\preceq$.

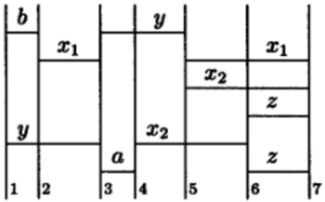
→ We are using a new set of variables χ instead of ν because the χ will also include all those variables that we get after limiting the variables to 2. (Like x_1 and x_2 in previous example)

More definitions

- $\rightarrow$ Base $bs = (t, E_{bs})$ is Constant base if $t \in \mathcal{C}$ and Variable base if $t \in \chi$
- $\rightarrow$ Left Boundary : Left(bs) = First element of E_{bs}
- $\rightarrow$ Right Boundary : Right(bs) = Last element of E_{bs}
- $\rightarrow$ Column : Pair of boundaries (i,j) with i
- $\rightarrow$ Column(i,i) is empty column.
- $\rightarrow$ Column (i,i+1) is called indecomposable.
- $\rightarrow$ Column of base bs = (Left(bs),Right(bs))
- $\rightarrow$ A base is empty if its column is empty.
- $\rightarrow$ A generalised equation is solved if all variable bases are empty.
- $\rightarrow$ Letters z, y, z will be used as meta variables for variable bases.
- $\rightarrow$ Letters i , j , . . . , will denote boundaries.

Example

GEN(bxyx = yaxz)



$\mathcal{C} = \{a, b\}$
$\chi = \{x_1, x_2, y, z\}$
BD = {1,2...7}
BS = {(b,(1,2)), (a, (3, 4)), (x_1, (2, 3)), (x_1, (5, 7)),
(y, (3, 5)), (y, (1, 3)), (x_2, (4, 6)), (x_2, (5, 7)),
(z,(6, 7)), (z, (6, 7))}

Definition 3

Definition 3. A *unifier* of GE is a function U that assigns to each indecomposable column of GE a word over $\mathcal{C} \cup \mathcal{V}$ (extend it by concatenation to all non-empty columns of GE) with the following properties:

(1) For each constant base bs of label c, $U(\mathrm{col}(bs)) = c$.
(2) For every pair of dual variables $x, \bar{x}$, and for every $e_j \in E_x$, $U(e_1, e_j) = U(\bar{e}_1, \bar{e}_j)$ (recall $\bar{e}_1, \bar{e}_j \in E_{\bar{x}}$). In particular $U(\mathrm{col}(x)) = U(\mathrm{col}(\bar{x}))$.

U is *strict* if $U(i, i+1)$ is non-empty for every $i \in BD$. The *index* of U is the number $|U(b_1, b_M)|$, where b_1 is the first and b_M the last element of BD. The *exponent of periodicity* of U is the maximal exponent of periodicity of the words $U(\mathrm{col}(x))$, where x is a variable base.

Definition 4

Definition 4. For a generalized equation GE, and $c \in \mathcal{C}$, the *associated system of linear Diophantine equations*, $L(GE, c)$, is defined by:

(1) A variable Z_i for each indecomposable column $(i, i+1)$ of GE.
(2) For each pair of dual variables bases $(x, (e_1, \ldots, e_n))$ and $(x, (\bar{e}_1, \ldots, \bar{e_n}))$ define $(n-1)$ equations, for $j = 1, \ldots, n-1$:

$$\sum_{e_j \preceq i \prec e_{j+1}} Z_i = \sum_{\bar{e}_j \preceq i \prec \bar{e}_{j+1}} Z_i$$

(3) For each constant base $(t, (i, i+1))$, define the equation $Z_i = 1$ if $t = c$ and $Z_i = 0$ if $t \neq c$.

Lemma 5

Lemma 5. *If GE has a unifier, then $L(GE, c)$ is solvable for each $c \in \mathcal{C}$.*

Proof. Let U be a unifier of GE and $c \in \mathcal{C}$. Define $Z_i = |U(i, i+1)| - D_c$ where D_c is the number of occurrences of constants different from c in the word $U(i, i+1)$. Using the fact that U is a unifier, it is easy to check that this is a solution to $L(GE, c)$. □

Checking solvability of systems of linear Diophantine systems is decidable, although expensive (NP-complete). A generalized equation GE whose system $L(GE, c)$ is solvable for all $c \in \mathcal{C}$ is called *admissible*.

Lemma 6

Lemma 6. *There exists an algorithm* GEN *which for every word equation* $\mathcal{E}$ *outputs a finite set* $\text{GEN}(\mathcal{E})$ *of generalized equations with the following properties:*

(1) $\mathcal{E}$ *has a unifier with exponent of periodicity* p *if and only if some* $GE \in \text{GEN}(\mathcal{E})$ *has a strict unifier with exponent of periodicity* p.
(2) *For each* $GE \in \text{GEN}(\mathcal{E})$, *every boundary is the right or left boundary of a base. Also, every boundary sequence contains exactly these two boundaries.*
(3) *For* $GE \in \text{GEN}(\mathcal{E})$, *the number of bases of* GE *does not exceed* $2|\mathcal{E}|$.
(4) *Every* $GE \in \text{GEN}(\mathcal{E})$ *is admissible.* □

Defintions

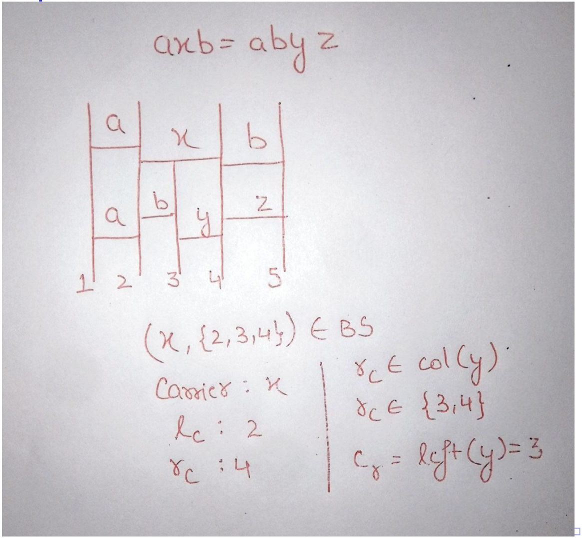
- → From the graphical representation, the naive idea was to pick the leftmost biggest variable (*carrier*) and move all its columns to the position of its *dual*.

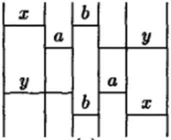
Definition 7. The *carrier* of GE, denoted x_c, is the non-empty variable base with smallest left boundary. If there is more than one, x_c is the one with largest right boundary. If there is still more than one, choose one among them randomly. We will denote $l_c = \text{LEFT}(x_c)$ and $r_c = \text{RIGHT}(x_c)$.

The *critical boundary* of GE is defined as $cr = \min\{\text{LEFT}(y) : r_c \in \text{col}(y)\}$ if the set is non-empty, and $cr = r_c$ if not.

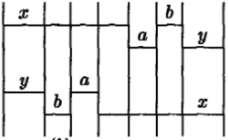
- → If any variable column has r_c in it, then c_r is the left boundary of that variable column.

Examples


axb = abyz
a
x
b
a
b
y
z
1 2 3 4 5
(x, {2,3,4}) ∈ BS
Carrier : x
l_c : 2
r_c : 4
r_c ∈ col(y)
r_c ∈ {3,4}
c_r = left(y) = 3



(a)



(b)

- → Carrier : y
- → l_c : 1
- → r_c : 3
- → c_r : 3

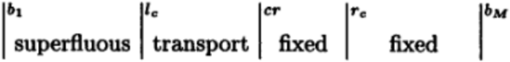
- → Carrier : x
- → l_c : 1
- → r_c : 5
- → c_r : 4

More definitions

Definition 8. Let bs be base of GE, bs is not the carrier. Then

(1) bs is *superfluous* if $\mathrm{col}(bs) = (i, i) \prec l_c$.
(2) bs is *transport* if $l_c \preceq \mathrm{LEFT}(bs) \prec cr$ or $\mathrm{col}(bs) = (cr, cr)$.
(3) bs is *fixed* if it is not superfluous and not transport.

Note that all variable bases with $\mathrm{LEFT}(x) \prec l_c$ are empty by definition of the carrier. Also, each base—except the carrier—is exactly one of these: superfluous, transport or fixed, depending on what region of the diagram below its left boundary is:



→ We have to move the transport bases. We haven't defined where to move them yet

Notation. For each boundary $l_c \preceq i \preceq r_c$ in BD, let us introduce a new symbol i^{tr} (which will indicate the place where the boundary i should go) and denote $\mathrm{tr}(E_x) = \mathrm{tr}(e_1, \ldots, e_n) = (e_1^{tr}, \ldots, e_n^{tr})$.

Definition 9. A *print* of GE is a linear order $\preceq$ on the set $BD \cup \{i^{tr} : i \in [l_c, r_c]\}$ satisfying the following conditions:

(1) $\preceq$ extends the order of BD and $j^{tr} \prec k^{tr}$ for $l_c \preceq j \prec k \preceq r_c$.
(2) $\mathrm{tr}(E_c) = \bar{E}_c$. (The structure of the carrier overlaps its dual.)
(3) If x is transport, $\bar{x}$ fixed, then if for some $e_i \in E_x$, $e_i^{tr} = \bar{e}_i$, then $\mathrm{tr}(E_x) = E_{\bar{x}}$. (The order $\preceq$ is consistent with the boundary sequence information.)
(4) If $(c, (i, j))$ is a constant base, then i, j (and also i^{tr}, j^{tr} if $i, j \in [l_c, r_c]$) are consecutive in the order $\preceq$. (Constants are preserved.)

Recap

(3) A finite set BS of *bases*. A *base bs* has the form $(t, (e_1, \ldots, e_n))$, where $n \geq 2$, $t \in \mathcal{C} \cup \mathcal{X}$, and $E_{bs} = (e_1, \ldots, e_n)$ is a sequence of boundaries ordered by $\preceq$.

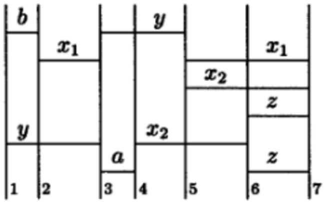
subject to the following conditions:[2]

(C1) For each $x \in \mathcal{X}$, there are exactly two bases with label x, called *duals*, and (abusing notation) denoted by x and $\bar{x}$ respectively. Also, their respective boundary sequences $E_x, E_{\bar{x}}$ must have the same length.

(C2) For each base bs with $t \in \mathcal{C}$, the boundary sequence E_{bs} has exactly two elements and they are consecutive in the order $\preceq$.

Recap

GEN(bxyx = yaxz)



$$BS = \{(b,(1,2)),(a,(3,4)),(x_1,(2,3)),(x_1,(5,7)),$$
$$(y,\ (3,\ 5)),\ (y,\ (1,\ 3)),\ (x_2,(4,6)),(x_2,(5,7)),$$
$$(z,(6,\ 7)),\ (z,\ (6,\ 7))\}$$

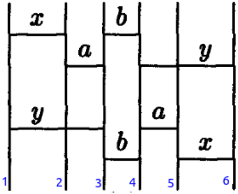
$E_y = \{1,2,3\}$

$E_{\overline{y}} = \{3,4,5\}$

Algorithm

- $\rightarrow$ Takes a Generalised Equation (GE)
- $\rightarrow$ Labels the bases as superfluous, transport or fixed
- $\rightarrow$ Comes up with a print($\leq$) which is a guess as to where the transport base can go. This guess can be any linear order of boundaries but it should satisfy the conditions in definition 9.
- $\rightarrow$ TRANSPORT (GE,$\leq$)
- $\rightarrow$ TRANSPORT (GE,$\leq$) is a Generalised equation too so we can do the same procedure on it now until all bases are empty.
- $\rightarrow$

Transportation

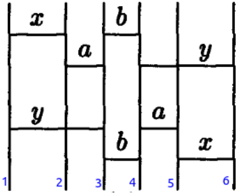


$l_c : 1$ $\quad r_c : 3$ $\quad c_r : 3$

(1) $\preceq$ extends the order of BD and $j^{tr} \prec k^{tr}$ for $l_c \preceq j \prec k \preceq r_c$.

$$1^{tr} < 2^{tr} < 3^{tr}$$

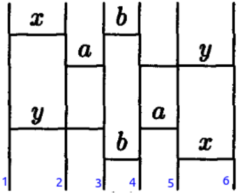
Transportation



$l_c : 1$ $r_c : 3$ $c_r : 3$

(2) $\mathrm{tr}(E_c) = \bar{E}_c$. **(The structure of the carrier overlaps its dual.)**

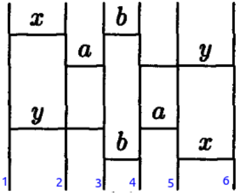
Transportation



$l_c : 1$ $r_c : 3$ $c_r : 3$

(3) If x is transport, $\bar{x}$ fixed, then if for some $e_i \in E_x$, $e_i^{tr} = \bar{e}_i$, then $\mathrm{tr}(E_x) = E_{\bar{x}}$. (The order $\preceq$ is consistent with the boundary sequence information.)

Transportation



$l_c : 1$ $r_c : 3$ $c_r : 3$

(4) If $(c, (i, j))$ is a constant base, then i, j (and also i^{tr}, j^{tr} if $i, j \in [l_c, r_c]$) are consecutive in the order $\preceq$. (Constants are preserved.)