The author gratefully acknowledges support from the taxpayers of Québec and of Canada who have paid his salary and research grants while this work was developed at McGill University, and from his students (who built the implementations and investigated the data structures and algorithms) and their funding agencies.

Operations on Attributes

The Domain Algebra

- Horizontal (scalar) operations
  - Arithmetic
  - Constants
  - Renaming

- Vertical (aggregation) operations
  - Reduction &
    Equivalence reduction
  - Functional mapping &
    Partial functional mapping

Domain Algebra (principle of closure)

I Horizontal operations

e.g., **let** *TotMk* **be** *Asst* + *Exam*;

NewMk

(*Student*	*Asst*	*Exam*)	*TotMk*
Smith	25.	60.	85.
Jones	28.	66.	94.
Brown	20.	50.	70.
Hung	24.	58.	82.
Raman	24.	66.	90.

N.B. *Virtual* domains (principle of abstraction)

Actualization via Project

let *TotMk* **be** *Asst* + *Exam*;
Result <− [*Student, TotMk*] **in** *NewMk*;

Any expression allowed that can be actualized on each tuple separately.

Domain Algebra

Frequently Asked Questions

- What if *Asst* and *Exam* are also in some other relation?
  Nothing is affected: TotMk could be actualized there, too.

- What if *Asst* and *Exam* each come from different relations?
  Use relational algebra to put the relations together before actualizing.

- What if we update *NewMk* after actualizing *TotMk*?
  TotMk is an actual attribute of Result and unaffected by changes to other relations.

Domain Algebra

Constants

let *One* **be** 1;

Renaming

let *Final* **be** *Exam*;

Course < − [*Student*, *Asst*, *Final*, *TotMk*, *One*] **in** *NewMk*;

Course

(*Student*	*Asst*	*Final*	*TotMk*	*One*)
Smith	25.	60.	85.	1
Jones	28.	66.	94.	1
Brown	20.	50.	70.	1
Hung	24.	58.	82.	1
Raman	24.	66.	90.	1

Domain Algebra

II Vertical operations e.g. of *Reduction*

let *Total* **be red + of** *TotMk*;

let *Count* **be red + of** *One*;

let *AvgMk* **be** *Total*/*Count*;

(**let** *AvgMk* **be** (**red + of** *TotMk*)/(**red + of** 1);)

Course

(*Student*	*Asst*	*Final*	*TotMk*	*One*)	*Total*	*Count*	*AvgMk*
Smith	25.	60.	85.	1	421	5	84.2
Jones	28.	66.	94.	1	421	5	84.2
Brown	20.	50.	70.	1	421	5	84.2
Hung	24.	58.	82.	1	421	5	84.2
Raman	24.	66.	90.	1	421	5	84.2

Vertical operations

Operators: $+, \times$, **max**, **min**, **and**, **or**

e.g., **let** *MaxMk* **be red max of** *TotMk*;

Course					
(*Student*	*Asst*	*Final*	*TotMk*	*One*)	*MaxMk*
Smith	25.	60.	85.	1	94
Jones	28.	66.	94.	1	94
Brown	20.	50.	70.	1	94
Hung	24.	58.	82.	1	94
Raman	24.	66.	90.	1	94

Of course, the attributes outside the (..) are *virtual*, so there is no "waste": the programmer actualizes and controls the waste.

Class <− [*AvgMk, MaxMK*] **in** *Course*;

Course	
(*AvgMk*	*MaxMk*)
84.2	94

Domain Algebra

II Vertical operations

e.g. of *Equivalence Reduction*

let *STot* **be equiv** + **of** *Mark* **by** *S#*;
let *CTot* **be equiv** + **of** *Mark* **by** *C#*;

StuCour

(*S#*	*C#*	*Mark*)	*STot*	*CTot*
1	1	73.	233.	213.
1	2	82.	233.	147.
1	3	78.	233.	78.
2	1	64.	64.	213.
3	1	76.	141.	213.
3	2	65.	141.	147.

Domain algebra—Vertical

e.g. of *Functional Mapping*

let *Cum* **be fun** + **of** *Amount* **order** *Year*

Sales

(*President*	*Year*	*Amount*)	*Cum*
Smith	1994	150	150
Smith	1995	175	325
Smith	1996	200	525
Brown	1996	200	525
Brown	1997	210	735
Brown	1998	225	960

Vertical operations

Operators: $+, -, \times, \div$, **max**, **min**, $||$, **pred**, **succ**

e.g., **let** *Alt* **be fun** $-$ **of** *Num* **order** *Seq*;

(*Num*	*Seq*)	*Alt*
27	1	27
13	2	−14
12	3	26
16	4	−10
9	5	19

Vertical operations

Text		
(*Word*	*Seq*)	*Next*
Algebraic	1	data
data	2	processing
processing	3	techniques
techniques	4	can
can	5	enable
enable	6	applications
applications	7	programmers
programmers	8	to
to	9	work
work	10	with
with	11	units
units	12	of
of	13	data
data	14	larger
larger	15	than
than	16	a
a	17	single
single	18	computer
computer	19	word
word	20	Algebraic

let *Next* **be fun succ of** *Word* **order** *Seq*;

Domain algebra—Vertical

e.g. of *Partial Functional Mapping*

let *DCum* **be par** + **of** *Amount* **order** *Year* **by** *Div*;

DivSales

(*Div*	*Year*	*Amount*)	*DCum*
A	1997	80	80
A	1998	110	190
B	1997	60	60
B	1998	75	135
C	1997	90	90
C	1998	110	200

Combining Relational and Domain Algebras

e.g., Matrix multiplication

A(ValA	I	J)
1	1	1
2	1	3
2	2	2
1	2	3
3	3	1
1	3	3

B(ValB	J	K)
1	1	1
1	1	2
1	2	1
1	3	1
2	3	2
1	3	3

AB(ValAB	I	K)
3	1	1
5	1	2
2	1	3
3	2	1
2	2	2
1	2	3
4	3	1
5	3	2
1	3	3

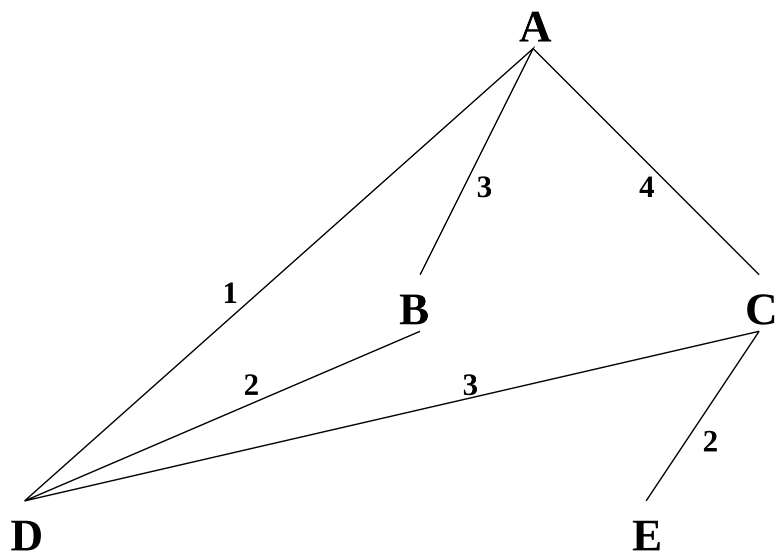
$$A \begin{pmatrix} 1 & 0 & 2 \\ 0 & 2 & 1 \\ 3 & 0 & 1 \end{pmatrix} B \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix} AB \begin{pmatrix} 3 & 5 & 2 \\ 3 & 2 & 1 \\ 4 & 5 & 1 \end{pmatrix}$$

let *ValAB* **be equiv** + **of** *ValA* × *ValB* **by** I, K;
AB <− [*ValAB, I, K*] **in** (A **ijoin** B);

A **ijoin** B

(*ValA*	I	J	K	*ValB*)	*ValA*×*ValB*	*ValAB*
1	1	1	1	1	1	3
1	1	1	2	1	1	:
3	3	1	1	1	3	4
3	3	1	2	1	3	:
2	2	2	1	1	2	
2	1	3	1	1	2	3
2	1	3	2	2	4	:
2	1	3	3	1	2	
1	2	3	1	1	1	
1	2	3	2	2	2	
1	2	3	3	1	1	
1	3	3	1	1	1	4
1	3	3	2	2	2	:
1	3	3	3	1	1	

Bill of Materials



PartOf(A	S	Q)	*Explo*(A	S	Q)	Q''	Q'''
A	B	3	A	B	3		3
A	C	4	A	C	4		4
A	D	1	A	D	1	18	19
B	D	2	B	D	2		2
C	D	3	C	D	3		3
C	E	2	C	E	2		2
			A	E		8	8

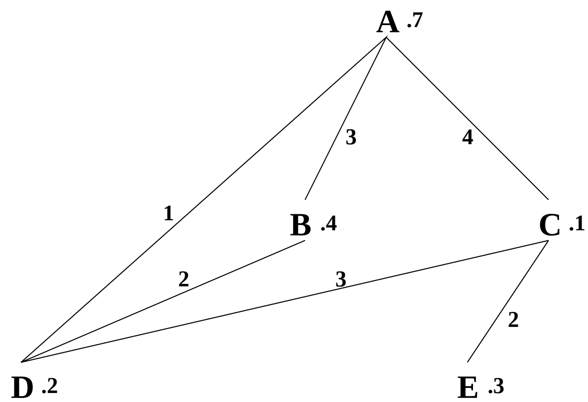
let A' **be** A; **let** S' **be** S; **let** Q' **be** Q;

let Q'' **be equiv** $+$ **of** $Q \times Q'$ **by** A, S';

let Q''' **be** $Q + Q''$; **let** Q **be** Q''';

Explo **is** $[A, S, Q]$ **in** $[A, S, Q''']$ **in** (*PartOf* $[A, S$ **ujoin** $A, S']$
 $[A, S', Q'']$ **in** (*Explo* $[S$ **ijoin** $A']$ $[A', S', Q']$ **in** *PartOf*));

Bill of Materials with Costs



Cost(A	C)
A	.7
B	.4
C	.1
D	.2
E	.3

Explo(A	S	Q)	C	QC	*QCEA*
A	B	3	.4	1.2	7.8
A	C	4	.1	.4	7.8
A	D	19	.2	3.8	7.8
A	E	8	.3	2.4	7.8
B	D	2	.2	.4	.4
C	D	3	.2	.6	1.2
C	E	2	.3	.6	1.2

let A' **be** A; **let** *QC* **be** $Q \times C$;
let *QCEA* **be equiv** + **of** $Q \times C$ **by** A;
let *Tcost* **be** *QCEA*+C;
SubCost <–[A,*QCEA*] **in** (*Explo* [S **ijoin** A'] [A', C] **in** *Cost*);
FinCost <–[A, *Tcost*] **in** (*SubCost* **ujoin** *Cost*);

SubCost **ujoin** *Cost*(A	*QCEA*	C)	*Tcost*
A	7.8	.7	8.5
B	.4	.4	.8
C	1.2	.1	1.3
D		.2	.2
E		.3	.3