Matrizenrechnung selbst gemacht

David Vajda

February 11, 2026

1 hinweise

- einteilung:
  1. Matrizenrechnung
  2. MMX
  3. Matrizenrechnung mathematische grundlagen: Prof. luise unger, Fernuni Hagen
  4. MMX (Joachim Rhode), und neue MMX - AVX-512, SSE
  5. Computersysteme 1 und 2: Gepackte und ungepackte BCD Formate sowie: MMX
  6. komplettes lernen des Kurses Mathematische Grundlagen, eben genannt ... (mit allen S"atzen) Steinitz usw.
  7. probierende nicht learn low level probierende matrizenrechnung
  8. wiederholende Aufgabe von mir zur matrizenmultiplikation selber.
- ziel und Hinweise:
  - Normale Deutsche Sprache: ist umgangssprache
  - Keine K"unstliche Expertensprache - keine Ausdrucksfehler bewertung what ever...: kein aggressives Missverst"andnis, dieser Text ist voller expertensprache, blos: aufpassen! wenn man normal redet, dann nicht reden wie der Hofstaat, wo der Hofstaat im Kopf wonders? Kapische?
  - Sonst vollst"andig, sonst alles durcheinander
  - folge: bei Joachim Rhode vorgestellt:
    - `cpuid` befehl, relativ easy, bei mir gestet in Aufgaben
    - `mmx` wird bei Joachim Rhode mit der Verf"ugbarkeit eingef"uhrt
    - `cpuid` und die verf"ugbarkeit von `cpuid` ohne `cpuid` anderes thema
    - trotzdem: bitte ordentlich: also:

2 gepackt und ungepackt,zahlensystem

- Rechenwerkzeug:
  - Handschriftliches Rechnen
  - Mit dem Abakus
  - Mit dem Computer
- Stichwort: Anzahl der Register (Ausdruck???)
  - Computer: General Purpose Register (Allzweckregister) und weitere: (Intel: `ax, bx, dx, cx, si, di, sp, bp`) bzw. andere Architektur meist (`r31, r30, ..., r2, r1, r0`)
  - Abakus: 1 oder 2 oder 3 Abakusse
  - Papier: rechnung, nebenrechnung, mehre (anlehnung: chemie ausdruck???) mehrere Hauptrechnungen, nebenrechnungen
- Nomenklatur, Intel, abh"angig von architektur
  - x86 (16 Bit): 8086/8088, IBM-PC XT/AT klassisch: (`ax, bx, dx, cx, sp, bp, si, di`)
  - `i386`: `eax, ebx, edx, ecx, esp, ebp, esi, edi`
  - `amd64` f"ur intel (core-i7) format: mac os x, z.B. `macho64` und: `rax, rbx, rdx, rcx, rsp, rbp,`
- vorsicht! Breite: Register
  - Byte: 8 Bit (umgangssprache: leitungen: die 0 und 1 tragen, 0V oder 5V, an und aus, H und L)
  - Word: 16 Bit
  - Double Word: 32 Bit
  - Quad Word: 64 Bit
  - Ten Bytes: 10 mal ein Byte.
- vorsicht! Breite, erneut:
  - byte
  - word
  - double word
  - quad word
- vorsicht breite erneut!
  - byte
  - Halbwort
  - Wort
  - Doppelt
  - MIPS32, hat halbwort, ist glaube ich 16 Bit
  - C in x86 hat Turbo C, short ist glaube ich halbwort - ist 16 Bit, so weit ich weiss

  - aber ich weiss: MIPS32, halbwort ist 16 Bit, weil: 32 Bit architektur immer,
  - Doppelwort: MIPS32, ist 64 Bit
  - Aber Intel: Word: 16 Bit, Double Word: 32 Bit, Quadword: 64 Bit

- Erneut:
  - erneut: Ein Byte ist immer 8 Bit!!! Immer
  - Bei intel ist ein Word: immer 16 Bit und bei
  - MIPS32, ist ein Halbwort da und 32 Bit ist ein Wort ...

- Bezeichnungen bei Assembler

```
db define Byte
dw define Word
dd define Double Word
dq define Quad Word
```

- Vorsicht, 2er Potenzen!!!

```
0, 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096, 8129
```

- Vorsicht: LSB und MSB: Least Significant Bit und Most Significant Bit
- Vorsicht: Little Endian und Big Endian: Merksatz: Big Endian Richtig rum, Little Endian Falsch rum
- vorsicht! Muss gewusst werden, ein Register mit 8 Bit hat 2^{256} m"ogliche werte
- Bedeutung von Registern:
  - Abh"angig von Operation: Abakus, mehrere??? z.B. einfaches Addieren, eines notwendig
  - Multiplizieren: 2
  - Z"ahlen: 2 oder 3, bei multilplizieren
- Register: Intel, `CX`: Counter Register...
- Achtung verwendung: Verwendung und Instrument macht den Sinn einer Operation in abh"angigkeit der Operanden und Elemente manchmal hinf"allig oder nicht
- st"andiges Addieren von 0 auf dem papier ist erkennbar
- st"andiges Addieren von 0 auf dem Computer denkbar, leerlaufmodus, cold run, what ever, Server Listening ...
- addieren ohne Counter auf Abakus sinnlos
- vorsicht: Time (UTC usw. von Takt (Computersysteme 1/2)), statistiker, kriegt zahlen rein f"ur abakus, bitte unterscheiden

- vorsicht: wann macht null addieren sinn mit z"ahlen auch bei Abakus: Beispiel: mittel (arithmetisch. usw) durchschnitt: Anzahl der gerauchten Zigaretten, der 30 untersuchten personen, zwischen 0 zigaretten und 30 zigaretten, an einem tag. dann alle addieren, mitzaehlen und das ergebnis durch die anzahl der probanden teilen.
- ohne z"ahler, zweiten abakus ist addieren von 0 sinnlos, wie bei mittelwert aber manchmal notwendig...

2.1 Beispiel: BCD

- TTL Gatter 74xx/54xx/84xx - Commercial, Military, Industrial
- Logikgatter: 7447: ist: BCD-2-7segment anzeige
- Raus gehen: 7 oder 8 bit, nicht als lesbare ziffer, sondern: 7 leitungen fuer 7 LED in 7 Segmentanzeige
- diese sind nummeriert bei 7447 und 7 Segmentanzeige, Kodierer: Zeichen eines vorrats zu den Zeichen eines anderen Zeichenvorrats umsetzen (DIN 44300/181, als Schaltnetz: 44300/91)
- Eingang: vier bit, kodiert, wie zahlen, bin"ar (`0000, 0001, 0010, 0011, 0100, 0101, 0110, 0111,` ...
- vier bit (Vektor von Eingangsvariablen, die sind Boolean)
- Achtung! BCD: heisst, die ziffern `0` bis `9` sind kodiert, aber nicht `hex`, weil bei vier bit, bis `1111`, das sind die zeichen `a, b, c, d, e, f`
- liegen trotzdem nicht als sichtbare arabische ziffern vor
- Gepackt und ungepackt:

```
ungepackt (unpacked)
byte1  byte0
0000 1001 0000 1001

bedeutung: 99d

gepacked (packed)
byte1  byte 0
xxxx xxxx  1001 1001

bedeutung: 99d

ebenso gepackte hexziffern denkbar

byte 1  byte 0
0000 1111  0000 1111
bedeutung: 0xff

gepackt:
```

```
byte1  byte 0
xxxx xxxx  1111 1111
bedeutung: 0xff
```

2.2 zahlensysteme

Beispiele:

- bin"ares Zahlensystem (High und Low, 0 und 1, wahr und falsch) normal, aber als Ziffern
- Un"ares Zahlensystem (Striche: `||||||||||`
- Arabische bzw. indische Ziffern (0, 1, 2, 3, 4, 5, 6, 7, 8, 9)
- r"omische Ziffern: I, II, III, IV, V, VI, VII, VIII, IX, X, ...
- Maya Ziffern: Bitte wikipedia: 20er system: 20 Ziffern, sehen aus wie zahl: hinweise, mischung, wie bei r"omern. bis zu 4 Punkten oben, jeweils 0 bis 3 waagerechte striche unten, jeweils wert 5
- Abakus, hat eigentlich ziffer pro zeile
  - genauere Beispiele:
    - umrechnung: dual (bin"ar) nach Dezimal: `0b1001` $= 1 \cdot 8 + 0 \cdot 4 + 0 \cdot 2 + 2 \cdot 1$. warum: erste

2.3 verschiedene umrechnungswege

2.3.1 erstes beispiel: Bin"ar nach Dezimal

```
1001 1100 1011 0001
```

Mathematisch (kurz nebenbei: hier nicht thema)

$$b_{15} \cdot 2^{15} + b_{14} \cdot 2^{14} + \cdots + b_1 \cdot 2^1 + b_0 \cdot 2^0$$

$$b_{13} \cdot 8192 + b_{12} \cdot 2048 + \cdots + b_1 \cdot 2 + b_0 \cdot 1$$

w"ahrend

$$B = \{b_{15}, b_{14}, \ldots, b_1, b_0\}$$

$$b_n = \{1, 0\}$$

Beispiele eine klassische umrechnung findet statt, derart:

```
1011 0001
1 * 2^7 + 0 * 2^6 + 1 * 2^5 + 1 * 2^4 + 0 * 2^3 + 0 * 2^2 + 0 * 2^1 + 1 * 2^0

das sind:

128 + 32 + 16 + 1

Schriftlich:
```

```
128
 32
 16
  1
----
177
 vermutlich.
```

der andere weg, geht in die andere richtung: und meiner vermutung nach immer dann: wenn das ausgangszahlensystem gr"osser ist als das ziel

```
153 / 2 = 76 Rest 1
76 / 2 = 38 Rest 0
38 / 2 = 19 Rest 0
19 / 2 = 9 Rest 1
9 / 2 = 4 Rest 1
4 / 2 = 2 Rest 0
2 / 2 = 1 Rest 0
1 / 2 = 0 Rest 1

Daraus ergibt sich:

1001 1001
```

- was hier keinen aufschluss gibt, wir muessen von hinten her lesen.
- warum? die zahl 2 die zum letzten mal teilt, ist quasi die die am allermeisten geteilt hat, mit den anderen war sie in jeder division enthalten
- wir m"ussen nun aber kathegorisch unterscheiden: in der bin"ardarstellung ist `10` eine `2d` eine `10` eine `10d` aber eine `1010b`
- kann man den umgekehrschluss schliessen?
- ja: in jedem Zahlensystem ist die Zahl nach allen ziffern, zun"aechst `10`
- kann ich durch die zahl 10 im eigenen zahlensytem teilen: ja: das macht sogar sinn und darauf m"ochte ich jetzt auch hinaus
- ich m"ochte die matrizen am bildschirm darstellen
- dabei ist mir ganz schnell selber gelungen, mit meinem jetzigen wissen, zahlen dizimal dar zu stellen
- indem ich eine zahl, in ihrem eigenen zahlensystem durch ihre eigene `10` teile, erhalte ich eben die reste die ich nachher darstelle
- und warum?
- der Rest ist das zeichen das dargestellt wird, bitte eine besonderheit: `+'0'`. warum? Notwendig zu merkenden ASCII Zeichen

  ```
  NUL STX SOT ETX ETB EOT LF FF HT VT ACK NAK BEL DEC ESC CAN DC1 .. DC4 ...
  ```

- die zeichen `'0' .. '9' a .. z A .. Z` nicht solange wir den assembler nicht selber schreiben
- und was passiert wenn wir eine bin"arzahl durch `1010b` teilen. eben das. indem wir uns std. zifern nicht merken, und buchstaben
- m"ussen wir uns auch im Assembler `10d` nicht merkeh. und doch. indem wir durch: `1010b` teilen was `10d` ist. befolgen wir genau das und
- nebenbei BCD Zahlen...

2.3.2 zweites Beispiel: Dezimal nach Bin"ar

Hier nun ein solcher Qulltext

```
        global   _main
        section  .text
_main:
 mov rsi, num
 mov  rax, [rsi]
 mov rdi, numstr
l1:
 mov rbx, 10
 mov rdx, 0x00
 div rbx
 add rdx, '0'
 mov [rdi], dl
 inc rdi
 cmp  rax, 0x00
 jnz l1

 mov      rax, 0x02000004
        mov      rdi, 0x00000001
        mov      rsi, numstr
        mov      rdx, 6
        syscall
        mov      rax, 0x02000001
        xor      rdi, rdi
        syscall

        section  .data
numstr:    db       0,0,0,0,0,10,0,0,0,0,0,0,0,0,0, 10
num: dq 14823
```

dieser quelltext gibt eine zahl aus, allerdings verkehrt herum, wir m"ussen von hinten her kommen. hier ist es richtig herum

```
        global   _main
        section  .text
_main:
 mov rsi, num
 mov  rax, [rsi]
```

```
 mov rdi, numstr+15
l1:
 mov rbx, 10
 mov rdx, 0x00
 div rbx
 add dl, '0'
 mov [rdi], dl
 dec rdi
 cmp  rax, 0x00
 jnz  l1

 mov  rsi, numstr
l2:
 mov  ah, [rdi]
 add ah, '0'
 mov [rdi], ah
 dec rdi
 cmp rdi, rsi
 jnz l2

 mov      rax, 0x02000004
        mov      rdi, 1
        mov      rsi, numstr
        mov      rdx, 17
        syscall
        mov      rax, 0x02000001
        xor      rdi, rdi
        syscall

        section  .data
numstr:
 db 0x00,0x00,0x00,0x00, 0x00,0x00,0x00,0x00, 0x00,0x00,0x00,0x00, 0x00,0x00,0x00,0x00, 10
num: dq 14823
```

und nun ein assembler spezifischer unterschied:

- hier: linux
- da: Mac OS X
- Microsoft Windows so wie MS DOS wiederum leugnet niemand
- Hier allerdings eine Version f'ur amd64 Linux und Mac OS X auf dem Intel Core i7
- andere Installation
  - Linux:

    ```
    su
    apt-get install ld nasm
    ```

  - Mac OS X:

```
port install nasm ld
# oder
brew install nasm ld
```

- wir arbeiten mit 64 Bit Code, d.h. wir benutzen die Register

  `rax, rbx, rdx, rcx, rbp, rsp, rsi, rdi`

- meiner meinung nach lohnt es sich der architektur an zu passen, ist diese 64 Bit und auf 64 Bit Linux l"auft 32 Bit, w"urde ich 64 Bit benutzen. Damit l"auf die "ubersetzung so:

  ```
  nasm prog.asm -o prog.o
  ld prog.o
  # ergebnis: a.out
  ld prog.o -o prog
  ```

- bei `nasm,ld` gilt, solange einfach angewendet und der Code wirklich 64 Bit ist, geht es ohne weitere angaben unter linux
- der Code darf kein Register `edx ecx` ... enthalten. denn dies ist 32 Bit Code
- und: Die Register `ah, al, bh, bl, dh, dl, ch, cl` existieren trotzdem immer noch - byteweise auch bei 64 Bit und trotz, dessen, dass man `edx` im gegensatz zu `rdx` nicht benutzen kann. auch hier gibt es eine unterteilung, wie die heisst, weiss ich nicht. google abfragen helfen weiter.
- trotzdem geschieht es auch unter.linux etwas anders:

  ```
  nasm -f elf64 prog.asm -o prog.o
  # oder
  nasm -f aout prog.asm -o prog.o
  ld prog.o -o prog
  # der Parameter ist wichtig und gibt die Zielmaschine an
  ```

- bei Mac OS X Lautet der Code so:

  ```
  nasm -f macho64 prog.asm -o prog.o
  ld prog.o
  ```

- wieder spielt die Zielarchitektur eine rolle. diese heisst hier: `macho64` bei linux neben dem `aout`, was `a.out` ergibt, sie finden das format im netz, spielt bei minix eine rolle, wo man sehr leicht lesen kann, im quelltext, wie programme geladen werden, aber auch bei linux - was betriebssysteme sind und den betrieb regeln, wozu es eine einheitliche l"osung gibt, UNIX alleine, aber: Betriebssysteme haben eben etwas mit Zimmeraufr"aumen gemein, sie sind ein System f"ur den betrieb. ich empfehle MINIX quelltexte zu lesen nicht Linux und: Ein Betriebssystem wird gerne von neulingen mit der korrekten sauberen Ordnung gerade im Bereich Hardware gerne verwechselt (wenige Leute koennen nach altem masse uhren herstellen - das bedeutet: es existiert auch hier heute ein irrtum: handarbeit - dem ist oft nicht so - ein Betriebssytem ist eine Art Stapelverarbeitungssystem - .... eine moderne form. ordentlich finde ich MINIX benutzen tue ich Mac OS X und Linux. so. aber: dann existiert noch das `bin` modell, eine art Flat Modell...

- f"ur Linux musste ich den Code etwas "andern, die Maschine hat gemeckert.

```
        global   _start
        section  .text
_start:
 mov rsi, num
 mov  rax, [rsi]
 mov rdi, numstr
l1:
 mov rbx, 10
 mov rdx, 0x00
 div rbx
 add rdx, '0'
 mov [rdi], dl
 inc rdi
 cmp  rax, 0x00
 jnz  l1

 mov      rax, 0x02000004
        mov      rdi, 0x00000001
        mov      rsi, numstr
        mov      rdx, 13
        syscall
        mov      rax, 0x02000001
        xor      rdi, rdi
        syscall

        section  .data
numstr:    db       0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0, 10
num: dq 14823
```

- wichtig sind die unterschiedlichen Systemaufrufe in Linux und in Mac OS X. w"ahrend `MS-DOS` das Interrupt `21h` verwendete, verwendet Linux den Systemaufruf `0x80` und MacOS X:

2.4 Die r"omischen Ziffern zum rechnen

- andere Beispiele sind: die Maya Ziffern
- Der Abakus und die indischen bzw. arabischen zahlen
- die r"omischen ziffern

diese wirken alle, als h"atten sie die stellenschreibweise inbegriffen. nicht zu 100 prozent. die r"omischen ziffern erlauben das zusammenfassen von zahlen zu einem zifferzeichen, die sehr gross sind, die ziffern der maya wirken wie ein zwanziger system, was mit strichen und punkten aehnlich dem roemischen wirkt. die striche dienen dem zaehlen von 5ern und die punkte bis 4. Von Null (Brot - ein Sonderzeichen) bis 19 gibt es 20 Ziffern.
Der Abakus enth"aelt pro zeile mehr oder weniger auch nur eine ziffer. die

arabischen oder indischen sind `0 .. 9`, auf der zeile eines abakus liegen quasi natuerliche zahlen die den peanoschen axiomen gen"ugen w"urden, aber auch der vollst"andigen Induktion 1 zu 1 addiert werden kann. trotzdem liegt pro zeile eine ziffer.
und nicht nur das: der Abakus ist nicht ein bin"ares Zahlensystem hinter diesem oft 6er oder eigentlich bei uns 10er. jede zeile eine ziffer. aber dahinter steckt ein anteil un"arer darstellung. rechts oder links. im bin"aren k"onnen an jeder stelle ein bit gesetzt sein oder nicht, steine hier links oder rechts. f"ur r"omisch gilt

```
IV X III V V  410355
... ..--- waeren 3 * 20^1 + 17 * 20^0 im dezimalen
```

wobei linien unter linien. bei den r"omern l"asst sich hier aber nicht f"ur mich eindeutig sagen, was eigentlich das system ist, bei maya 20er. also:

```
... --- .- .--

3 * 20^3 + 15*20^2 + 6*20^1 + 11*20^0
```

ok

3 zur "ublichen benennenung im Bereich Rechenoperation

```
Summe = Summand1 + Summand2
Differenz = Minuend - Subtrahend
Produkt = Multiplikand * Multiplikator
Quotient = Dividend/Divisor
Quotient = Dividend/Divisor + Rest
Bruch = Zaehler/Nenner
Oben: klein, unten: Gross (oft)
```

4 maxima

4.1 die installation

4.1.1 Debian Linux

```
apt-get install maxima xmaxima
```

4.1.2 Mac OS X

- `port`
- `brew`
- `port`

  ```
  su
  /opt/local/bin/port install maxima
  ```

- "brew"

  ```
  brew install maxima
  ```

4.2 usage

- `maxima` ist open source, `maple` ist kommerziell
- wir brauchen?
  - "offnen
  - speichern
  - oeffnen
  - was wir wissen m"ussen: LISP (programmiersprache), Konsole (maxima nur als Konsole verstehen),
  - Eine Matrix definieren
  - Matrizenmultiplikation mit `. ...`
  - nicht zu verwechseln mit `*`

  konkret:

  - "offnen: ganz normal "uber die konsole:

    ```
    maxima
    ```

  - speichern

    Bitte darauf achten! Gespeichert wird nur der inhalt der ab dem Befehl zum speichern eingegeben wird

  - oeffnen

    ```
    maxima -b file.lsp
    ```

    es ist ratsam, die lisp commands (keine ahnung) in den Plain Text ASCII Editor ein zu geben, die datei mit der Endung meiner meinung nach `lsp` zu speichern f"ur lisp und dann: dies mit maxima wie geshabt zu "offnen
  - was wir wissen m"ussen: LISP (programmiersprache), Konsole (maxima nur als Konsole verstehen),
  - Eine Matrix definieren

    ```
    A : matrix ([a11,a12],[a21,a22]);
    B : matrix ([a,b],[c,d]);
    C: matrix ([0,4,8,2],[23,15,1,4],[9,17,8,3]);
    D: matrix ([d11, d12, d13],[d21,d22,d23],[d31,d32,d33]);
    \end{verbatim
    \item Matrizenmultiplikation mit \verb"." \dots
    \begin{verbatim}
    A . B;
    E = A . B;
    ```

– nicht zu verwechseln mit *

  Bitte unterscheiden!!! Die Matrizenmultiplikation wie sie eigentlich ist unterscheidet sich von der Elementweisen multiplikation. sprich: Zeilenvektor, Spaltenvektor, Komponente und alles was da drueber ist, Matrix usw.

4.3 Beispiele mit maxima

4.3.1 input

```
A: matrix([a,b],[c,d]);
E11: matrix([1,0],[0,0]);
E12: matrix([0,1],[0,0]);
E21: matrix([0,0],[1,0]);
E22: matrix([0,0],[0,1]);
I: matrix([1,0],[0,1]);
J: matrix([0,1],[1,0]);

E11 . A;
E12 . A;
E21 . A;
E22 . A;
A . E11;
A . E12;
A . E21;
A . E22;
E12 . E21 . A . E11;
J . A . E12 . E21;
E12 . E21 . A;
E12 . E21 . A . E12 . E21;
E12 . E21 . A . I;
I . A . E21;
J . A . J;
J . A . J . J;
J . J . A . J . J;
J . A . J . J;
J . J . A . J;
J . A;
J . J . A;
J . J . J . A;
J . J . J . J . A;
A . J;
A . J . J;
A . J . J . J;
A . J . J . J . J;
J . A . J . J . J;
J . A . J . J;
J . A . J;
J . J . A . J;
J . J . A . J . J;
```

```
J . J . A . J . J . J;
J . A . E11 . E22;
J . A . E22 . E11;
J . A . E11 . E12;
J . A . E12 . E11;
J . A . E11 . E21;
J . A . E21 . E11;
J . A . E12 . E21;
J . A . E21 . E12;


J . A;
A . J;

A . E12 . E21;
A . E21 . E12;

J . A . E12 . E21;
A . J . E21 . E12;
A . J . E12 . E21;
J . A . E21 . E12;

E12 . E21 . A . E12;
E12 . A . E12 . E21;
E12 . J . A . E12 . E21;
E12 . A . J . E12 . E21;

J . A;
A . J;

I . A;
A . I;

E12 . E21 . A . E21;
E21 . E12 . A . E12;
E21 . E12 . A . E21;
J . A . E21;

E12 . E21 . A . E12;
E21 . E12 . A . E12;
E12 . E21 . A . E21;
E21 . E12 . A . E21;

E12 . A . E12 . E21;
E12 . A . E21 . E12;
E21 . A . E12 . E21;
E21 . A . E21 . E12;

A . J;
A . J . J;
```

```
A . J . J . J;
A . J . J . J . J;
J . A;
J . J . A;
J . J . J . A;
J . J . J . J . A;

(J . A) . J;

(E12 . A) . E21;
(E21 . A) . E12;
(E12 . A) . E12;
(E21 . A) . E21;
E12 . (A . E21);
E21 . (A . E12);
```

4.3.2 output

```
Maxima 5.47.0 https://maxima.sourceforge.io
using Lisp SBCL 2.6.1
Distributed under the GNU Public License. See the file COPYING.
Dedicated to the memory of William Schelter.
The function bug_report() provides bug reporting information.
(%i1) batch("matrix02102026now.lsp")

read and interpret /Users/davidvajda/Documents/matrix02102026now.lsp
(%i2) A:matrix([a,b],[c,d])
                                   [ a  b ]
(%o2)                              [      ]
                                   [ c  d ]
(%i3) E11:matrix([1,0],[0,0])
                                   [ 1  0 ]
(%o3)                              [      ]
                                   [ 0  0 ]
(%i4) E12:matrix([0,1],[0,0])
                                   [ 0  1 ]
(%o4)                              [      ]
                                   [ 0  0 ]
(%i5) E21:matrix([0,0],[1,0])
                                   [ 0  0 ]
(%o5)                              [      ]
                                   [ 1  0 ]
(%i6) E22:matrix([0,0],[0,1])
                                   [ 0  0 ]
(%o6)                              [      ]
                                   [ 0  1 ]
(%i7) I:matrix([1,0],[0,1])
                                   [ 1  0 ]
(%o7)                              [      ]
                                   [ 0  1 ]
```

```
(%i8) J:matrix([0,1],[1,0])
                                  [ 0  1 ]
(%o8)                             [      ]
                                  [ 1  0 ]
(%i9) E11 . A
                                  [ a  b ]
(%o9)                             [      ]
                                  [ 0  0 ]
(%i10) E12 . A
                                  [ c  d ]
(%o10)                            [      ]
                                  [ 0  0 ]
(%i11) E21 . A
                                  [ 0  0 ]
(%o11)                            [      ]
                                  [ a  b ]
(%i12) E22 . A
                                  [ 0  0 ]
(%o12)                            [      ]
                                  [ c  d ]
(%i13) A . E11
                                  [ a  0 ]
(%o13)                            [      ]
                                  [ c  0 ]
(%i14) A . E12
                                  [ 0  a ]
(%o14)                            [      ]
                                  [ 0  c ]
(%i15) A . E21
                                  [ b  0 ]
(%o15)                            [      ]
                                  [ d  0 ]
(%i16) A . E22
                                  [ 0  b ]
(%o16)                            [      ]
                                  [ 0  d ]
(%i17) E12 . E21 . A . E11
                                  [ a  0 ]
(%o17)                            [      ]
                                  [ 0  0 ]
(%i18) J . A . E12 . E21
                                  [ c  0 ]
(%o18)                            [      ]
                                  [ a  0 ]
(%i19) E12 . E21 . A
                                  [ a  b ]
(%o19)                            [      ]
                                  [ 0  0 ]
(%i20) E12 . E21 . A . E12 . E21
                                  [ a  0 ]
```

```
(%o20)                                [      ]
                                      [ 0  0 ]
(%i21) E12 . E21 . A . I
                                      [ a  b ]
(%o21)                                [      ]
                                      [ 0  0 ]
(%i22) I . A . E21
                                      [ b  0 ]
(%o22)                                [      ]
                                      [ d  0 ]
(%i23) J . A . J
                                      [ d  c ]
(%o23)                                [      ]
                                      [ b  a ]
(%i24) J . A . J . J
                                      [ c  d ]
(%o24)                                [      ]
                                      [ a  b ]
(%i25) J . J . A . J . J
                                      [ a  b ]
(%o25)                                [      ]
                                      [ c  d ]
(%i26) J . A . J . J
                                      [ c  d ]
(%o26)                                [      ]
                                      [ a  b ]
(%i27) J . J . A . J
                                      [ b  a ]
(%o27)                                [      ]
                                      [ d  c ]
(%i28) J . A
                                      [ c  d ]
(%o28)                                [      ]
                                      [ a  b ]
(%i29) J . J . A
                                      [ a  b ]
(%o29)                                [      ]
                                      [ c  d ]
(%i30) J . J . J . A
                                      [ c  d ]
(%o30)                                [      ]
                                      [ a  b ]
(%i31) J . J . J . J . A
                                      [ a  b ]
(%o31)                                [      ]
                                      [ c  d ]
(%i32) A . J
                                      [ b  a ]
(%o32)                                [      ]
                                      [ d  c ]
```

```
(%i33) A . J . J
                                  [ a  b ]
(%o33)                            [      ]
                                  [ c  d ]
(%i34) A . J . J . J
                                  [ b  a ]
(%o34)                            [      ]
                                  [ d  c ]
(%i35) A . J . J . J . J
                                  [ a  b ]
(%o35)                            [      ]
                                  [ c  d ]
(%i36) J . A . J . J . J
                                  [ d  c ]
(%o36)                            [      ]
                                  [ b  a ]
(%i37) J . A . J . J
                                  [ c  d ]
(%o37)                            [      ]
                                  [ a  b ]
(%i38) J . A . J
                                  [ d  c ]
(%o38)                            [      ]
                                  [ b  a ]
(%i39) J . J . A . J
                                  [ b  a ]
(%o39)                            [      ]
                                  [ d  c ]
(%i40) J . J . A . J . J
                                  [ a  b ]
(%o40)                            [      ]
                                  [ c  d ]
(%i41) J . J . A . J . J . J
                                  [ b  a ]
(%o41)                            [      ]
                                  [ d  c ]
(%i42) J . A . E11 . E22
                                  [ 0  0 ]
(%o42)                            [      ]
                                  [ 0  0 ]
(%i43) J . A . E22 . E11
                                  [ 0  0 ]
(%o43)                            [      ]
                                  [ 0  0 ]
(%i44) J . A . E11 . E12
                                  [ 0  c ]
(%o44)                            [      ]
                                  [ 0  a ]
(%i45) J . A . E12 . E11
                                  [ 0  0 ]
```

```
(%o45)                                [      ]
                                      [ 0  0 ]
(%i46) J . A . E11 . E21
                                      [ 0  0 ]
(%o46)                                [      ]
                                      [ 0  0 ]
(%i47) J . A . E21 . E11
                                      [ d  0 ]
(%o47)                                [      ]
                                      [ b  0 ]
(%i48) J . A . E12 . E21
                                      [ c  0 ]
(%o48)                                [      ]
                                      [ a  0 ]
(%i49) J . A . E21 . E12
                                      [ 0  d ]
(%o49)                                [      ]
                                      [ 0  b ]
(%i50) J . A
                                      [ c  d ]
(%o50)                                [      ]
                                      [ a  b ]
(%i51) A . J
                                      [ b  a ]
(%o51)                                [      ]
                                      [ d  c ]
(%i52) A . E12 . E21
                                      [ a  0 ]
(%o52)                                [      ]
                                      [ c  0 ]
(%i53) A . E21 . E12
                                      [ 0  b ]
(%o53)                                [      ]
                                      [ 0  d ]
(%i54) J . A . E12 . E21
                                      [ c  0 ]
(%o54)                                [      ]
                                      [ a  0 ]
(%i55) A . J . E21 . E12
                                      [ 0  a ]
(%o55)                                [      ]
                                      [ 0  c ]
(%i56) A . J . E12 . E21
                                      [ b  0 ]
(%o56)                                [      ]
                                      [ d  0 ]
(%i57) J . A . E21 . E12
                                      [ 0  d ]
(%o57)                                [      ]
                                      [ 0  b ]
```

```
(%i58) E12 . E21 . A . E12
                                  [ 0  a ]
(%o58)                            [      ]
                                  [ 0  0 ]
(%i59) E12 . A . E12 . E21
                                  [ c  0 ]
(%o59)                            [      ]
                                  [ 0  0 ]
(%i60) E12 . J . A . E12 . E21
                                  [ a  0 ]
(%o60)                            [      ]
                                  [ 0  0 ]
(%i61) E12 . A . J . E12 . E21
                                  [ d  0 ]
(%o61)                            [      ]
                                  [ 0  0 ]
(%i62) J . A
                                  [ c  d ]
(%o62)                            [      ]
                                  [ a  b ]
(%i63) A . J
                                  [ b  a ]
(%o63)                            [      ]
                                  [ d  c ]
(%i64) I . A
                                  [ a  b ]
(%o64)                            [      ]
                                  [ c  d ]
(%i65) A . I
                                  [ a  b ]
(%o65)                            [      ]
                                  [ c  d ]
(%i66) E12 . E21 . A . E21
                                  [ b  0 ]
(%o66)                            [      ]
                                  [ 0  0 ]
(%i67) E21 . E12 . A . E12
                                  [ 0  0 ]
(%o67)                            [      ]
                                  [ 0  c ]
(%i68) E21 . E12 . A . E21
                                  [ 0  0 ]
(%o68)                            [      ]
                                  [ d  0 ]
(%i69) J . A . E21
                                  [ d  0 ]
(%o69)                            [      ]
                                  [ b  0 ]
(%i70) E12 . E21 . A . E12
                                  [ 0  a ]
```

```
(%o70)                                [      ]
                                      [ 0  0 ]
(%i71) E21 . E12 . A . E12
                                      [ 0  0 ]
(%o71)                                [      ]
                                      [ 0  c ]
(%i72) E12 . E21 . A . E21
                                      [ b  0 ]
(%o72)                                [      ]
                                      [ 0  0 ]
(%i73) E21 . E12 . A . E21
                                      [ 0  0 ]
(%o73)                                [      ]
                                      [ d  0 ]
(%i74) E12 . A . E12 . E21
                                      [ c  0 ]
(%o74)                                [      ]
                                      [ 0  0 ]
(%i75) E12 . A . E21 . E12
                                      [ 0  d ]
(%o75)                                [      ]
                                      [ 0  0 ]
(%i76) E21 . A . E12 . E21
                                      [ 0  0 ]
(%o76)                                [      ]
                                      [ a  0 ]
(%i77) E21 . A . E21 . E12
                                      [ 0  0 ]
(%o77)                                [      ]
                                      [ 0  b ]
(%i78) A . J
                                      [ b  a ]
(%o78)                                [      ]
                                      [ d  c ]
(%i79) A . J . J
                                      [ a  b ]
(%o79)                                [      ]
                                      [ c  d ]
(%i80) A . J . J . J
                                      [ b  a ]
(%o80)                                [      ]
                                      [ d  c ]
(%i81) A . J . J . J . J
                                      [ a  b ]
(%o81)                                [      ]
                                      [ c  d ]
(%i82) J . A
                                      [ c  d ]
(%o82)                                [      ]
                                      [ a  b ]
```

```
(%i83) J . J . A
                                  [ a  b ]
(%o83)                            [      ]
                                  [ c  d ]
(%i84) J . J . J . A
                                  [ c  d ]
(%o84)                            [      ]
                                  [ a  b ]
(%i85) J . J . J . J . A
                                  [ a  b ]
(%o85)                            [      ]
                                  [ c  d ]
(%i86) (J . A) . J
                                  [ d  c ]
(%o86)                            [      ]
                                  [ b  a ]
(%i87) (E12 . A) . E21
                                  [ d  0 ]
(%o87)                            [      ]
                                  [ 0  0 ]
(%i88) (E21 . A) . E12
                                  [ 0  0 ]
(%o88)                            [      ]
                                  [ 0  a ]
(%i89) (E12 . A) . E12
                                  [ 0  c ]
(%o89)                            [      ]
                                  [ 0  0 ]
(%i90) (E21 . A) . E21
                                  [ 0  0 ]
(%o90)                            [      ]
                                  [ b  0 ]
(%i91) E12 . A . E21
                                  [ d  0 ]
(%o91)                            [      ]
                                  [ 0  0 ]
(%i92) E21 . A . E12
                                  [ 0  0 ]
(%o92)                            [      ]
                                  [ 0  a ]
(%o93)         /Users/davidvajda/Documents/matrix02102026now.lsp
```

5 Die matrizenmultiplikation

5.1 noch ein mal maxima

warum?

- keine rechenfehler

- best"atigung von dem was selber ausgerechnet habe
- f"ur sp"ater: algorithmische untersuchung der m"oglichen variationen (Ausdruck???)

5.1.1 input

```
A: matrix([a,b],[c,d]);
E11: matrix([1,0],[0,0]);
E12: matrix([0,1],[0,0]);
E21: matrix([0,0],[1,0]);
E22: matrix([0,0],[0,1]);
I: matrix([1,0],[0,1]);
J: matrix([0,1],[1,0]);

E11 . A;
E12 . A;
E21 . A;
E22 . A;
A . E11;
A . E12;
A . E21;
A . E22;
E12 . E21 . A . E11;
J . A . E12 . E21;
E12 . E21 . A;
E12 . E21 . A . E12 . E21;
E12 . E21 . A . I;
I . A . E21;
J . A . J;
J . A . J . J;
J . J . A . J . J;
J . A . J . J;
J . J . A . J;
J . A;
J . J . A;
J . J . J . A;
J . J . J . J . A;
A . J;
A . J . J;
A . J . J . J;
A . J . J . J . J;
J . A . J . J . J;
J . A . J . J;
J . A . J;
J . J . A . J;
J . J . A . J . J;
J . J . A . J . J . J;
J . A . E11 . E22;
J . A . E22 . E11;
```

```
J . A . E11 . E12;
J . A . E12 . E11;
J . A . E11 . E21;
J . A . E21 . E11;
J . A . E12 . E21;
J . A . E21 . E12;


J . A;
A . J;

A . E12 . E21;
A . E21 . E12;

J . A . E12 . E21;
A . J . E21 . E12;
A . J . E12 . E21;
J . A . E21 . E12;

E12 . E21 . A . E12;
E12 . A . E12 . E21;
E12 . J . A . E12 . E21;
E12 . A . J . E12 . E21;

J . A;
A . J;

I . A;
A . I;

E12 . E21 . A . E21;
E21 . E12 . A . E12;
E21 . E12 . A . E21;
J . A . E21;

E12 . E21 . A . E12;
E21 . E12 . A . E12;
E12 . E21 . A . E21;
E21 . E12 . A . E21;

E12 . A . E12 . E21;
E12 . A . E21 . E12;
E21 . A . E12 . E21;
E21 . A . E21 . E12;

A . J;
A . J . J;
A . J . J . J;
A . J . J . J . J;
J . A;
```

```
J . J . A;
J . J . J . A;
J . J . J . J . A;

(J . A) . J;

(E12 . A) . E21;
(E21 . A) . E12;
(E12 . A) . E12;
(E21 . A) . E21;
E12 . (A . E21);
E21 . (A . E12);
```

5.1.2 output

```
Maxima 5.47.0 https://maxima.sourceforge.io
using Lisp SBCL 2.6.1
Distributed under the GNU Public License. See the file COPYING.
Dedicated to the memory of William Schelter.
The function bug_report() provides bug reporting information.
(%i1) batch("matrix02102026now.lsp")

read and interpret /Users/davidvajda/Documents/matrix02102026now.lsp
(%i2) A:matrix([a,b],[c,d])
                                    [ a  b ]
(%o2)                               [      ]
                                    [ c  d ]
(%i3) E11:matrix([1,0],[0,0])
                                    [ 1  0 ]
(%o3)                               [      ]
                                    [ 0  0 ]
(%i4) E12:matrix([0,1],[0,0])
                                    [ 0  1 ]
(%o4)                               [      ]
                                    [ 0  0 ]
(%i5) E21:matrix([0,0],[1,0])
                                    [ 0  0 ]
(%o5)                               [      ]
                                    [ 1  0 ]
(%i6) E22:matrix([0,0],[0,1])
                                    [ 0  0 ]
(%o6)                               [      ]
                                    [ 0  1 ]
(%i7) I:matrix([1,0],[0,1])
                                    [ 1  0 ]
(%o7)                               [      ]
                                    [ 0  1 ]
(%i8) J:matrix([0,1],[1,0])
                                    [ 0  1 ]
(%o8)                               [      ]
```

```
                                  [ 1  0 ]
(%i9) E11 . A
                                  [ a  b ]
(%o9)                             [      ]
                                  [ 0  0 ]
(%i10) E12 . A
                                  [ c  d ]
(%o10)                            [      ]
                                  [ 0  0 ]
(%i11) E21 . A
                                  [ 0  0 ]
(%o11)                            [      ]
                                  [ a  b ]
(%i12) E22 . A
                                  [ 0  0 ]
(%o12)                            [      ]
                                  [ c  d ]
(%i13) A . E11
                                  [ a  0 ]
(%o13)                            [      ]
                                  [ c  0 ]
(%i14) A . E12
                                  [ 0  a ]
(%o14)                            [      ]
                                  [ 0  c ]
(%i15) A . E21
                                  [ b  0 ]
(%o15)                            [      ]
                                  [ d  0 ]
(%i16) A . E22
                                  [ 0  b ]
(%o16)                            [      ]
                                  [ 0  d ]
(%i17) E12 . E21 . A . E11
                                  [ a  0 ]
(%o17)                            [      ]
                                  [ 0  0 ]
(%i18) J . A . E12 . E21
                                  [ c  0 ]
(%o18)                            [      ]
                                  [ a  0 ]
(%i19) E12 . E21 . A
                                  [ a  b ]
(%o19)                            [      ]
                                  [ 0  0 ]
(%i20) E12 . E21 . A . E12 . E21
                                  [ a  0 ]
(%o20)                            [      ]
                                  [ 0  0 ]
(%i21) E12 . E21 . A . I
```

```
                                    [ a  b ]
(%o21)                              [      ]
                                    [ 0  0 ]
(%i22) I . A . E21
                                    [ b  0 ]
(%o22)                              [      ]
                                    [ d  0 ]
(%i23) J . A . J
                                    [ d  c ]
(%o23)                              [      ]
                                    [ b  a ]
(%i24) J . A . J . J
                                    [ c  d ]
(%o24)                              [      ]
                                    [ a  b ]
(%i25) J . J . A . J . J
                                    [ a  b ]
(%o25)                              [      ]
                                    [ c  d ]
(%i26) J . A . J . J
                                    [ c  d ]
(%o26)                              [      ]
                                    [ a  b ]
(%i27) J . J . A . J
                                    [ b  a ]
(%o27)                              [      ]
                                    [ d  c ]
(%i28) J . A
                                    [ c  d ]
(%o28)                              [      ]
                                    [ a  b ]
(%i29) J . J . A
                                    [ a  b ]
(%o29)                              [      ]
                                    [ c  d ]
(%i30) J . J . J . A
                                    [ c  d ]
(%o30)                              [      ]
                                    [ a  b ]
(%i31) J . J . J . J . A
                                    [ a  b ]
(%o31)                              [      ]
                                    [ c  d ]
(%i32) A . J
                                    [ b  a ]
(%o32)                              [      ]
                                    [ d  c ]
(%i33) A . J . J
                                    [ a  b ]
(%o33)                              [      ]
```

```
                                    [ c  d ]
(%i34) A . J . J . J
                                    [ b  a ]
(%o34)                              [      ]
                                    [ d  c ]
(%i35) A . J . J . J . J
                                    [ a  b ]
(%o35)                              [      ]
                                    [ c  d ]
(%i36) J . A . J . J . J
                                    [ d  c ]
(%o36)                              [      ]
                                    [ b  a ]
(%i37) J . A . J . J
                                    [ c  d ]
(%o37)                              [      ]
                                    [ a  b ]
(%i38) J . A . J
                                    [ d  c ]
(%o38)                              [      ]
                                    [ b  a ]
(%i39) J . J . A . J
                                    [ b  a ]
(%o39)                              [      ]
                                    [ d  c ]
(%i40) J . J . A . J . J
                                    [ a  b ]
(%o40)                              [      ]
                                    [ c  d ]
(%i41) J . J . A . J . J . J
                                    [ b  a ]
(%o41)                              [      ]
                                    [ d  c ]
(%i42) J . A . E11 . E22
                                    [ 0  0 ]
(%o42)                              [      ]
                                    [ 0  0 ]
(%i43) J . A . E22 . E11
                                    [ 0  0 ]
(%o43)                              [      ]
                                    [ 0  0 ]
(%i44) J . A . E11 . E12
                                    [ 0  c ]
(%o44)                              [      ]
                                    [ 0  a ]
(%i45) J . A . E12 . E11
                                    [ 0  0 ]
(%o45)                              [      ]
                                    [ 0  0 ]
(%i46) J . A . E11 . E21
```

```
                                    [ 0  0 ]
(%o46)                              [      ]
                                    [ 0  0 ]
(%i47) J . A . E21 . E11
                                    [ d  0 ]
(%o47)                              [      ]
                                    [ b  0 ]
(%i48) J . A . E12 . E21
                                    [ c  0 ]
(%o48)                              [      ]
                                    [ a  0 ]
(%i49) J . A . E21 . E12
                                    [ 0  d ]
(%o49)                              [      ]
                                    [ 0  b ]
(%i50) J . A
                                    [ c  d ]
(%o50)                              [      ]
                                    [ a  b ]
(%i51) A . J
                                    [ b  a ]
(%o51)                              [      ]
                                    [ d  c ]
(%i52) A . E12 . E21
                                    [ a  0 ]
(%o52)                              [      ]
                                    [ c  0 ]
(%i53) A . E21 . E12
                                    [ 0  b ]
(%o53)                              [      ]
                                    [ 0  d ]
(%i54) J . A . E12 . E21
                                    [ c  0 ]
(%o54)                              [      ]
                                    [ a  0 ]
(%i55) A . J . E21 . E12
                                    [ 0  a ]
(%o55)                              [      ]
                                    [ 0  c ]
(%i56) A . J . E12 . E21
                                    [ b  0 ]
(%o56)                              [      ]
                                    [ d  0 ]
(%i57) J . A . E21 . E12
                                    [ 0  d ]
(%o57)                              [      ]
                                    [ 0  b ]
(%i58) E12 . E21 . A . E12
                                    [ 0  a ]
(%o58)                              [      ]
```

```
                                    [ 0  0 ]
(%i59) E12 . A . E12 . E21
                                    [ c  0 ]
(%o59)                              [      ]
                                    [ 0  0 ]
(%i60) E12 . J . A . E12 . E21
                                    [ a  0 ]
(%o60)                              [      ]
                                    [ 0  0 ]
(%i61) E12 . A . J . E12 . E21
                                    [ d  0 ]
(%o61)                              [      ]
                                    [ 0  0 ]
(%i62) J . A
                                    [ c  d ]
(%o62)                              [      ]
                                    [ a  b ]
(%i63) A . J
                                    [ b  a ]
(%o63)                              [      ]
                                    [ d  c ]
(%i64) I . A
                                    [ a  b ]
(%o64)                              [      ]
                                    [ c  d ]
(%i65) A . I
                                    [ a  b ]
(%o65)                              [      ]
                                    [ c  d ]
(%i66) E12 . E21 . A . E21
                                    [ b  0 ]
(%o66)                              [      ]
                                    [ 0  0 ]
(%i67) E21 . E12 . A . E12
                                    [ 0  0 ]
(%o67)                              [      ]
                                    [ 0  c ]
(%i68) E21 . E12 . A . E21
                                    [ 0  0 ]
(%o68)                              [      ]
                                    [ d  0 ]
(%i69) J . A . E21
                                    [ d  0 ]
(%o69)                              [      ]
                                    [ b  0 ]
(%i70) E12 . E21 . A . E12
                                    [ 0  a ]
(%o70)                              [      ]
                                    [ 0  0 ]
(%i71) E21 . E12 . A . E12
```

```
                                    [ 0  0 ]
(%o71)                              [      ]
                                    [ 0  c ]
(%i72) E12 . E21 . A . E21
                                    [ b  0 ]
(%o72)                              [      ]
                                    [ 0  0 ]
(%i73) E21 . E12 . A . E21
                                    [ 0  0 ]
(%o73)                              [      ]
                                    [ d  0 ]
(%i74) E12 . A . E12 . E21
                                    [ c  0 ]
(%o74)                              [      ]
                                    [ 0  0 ]
(%i75) E12 . A . E21 . E12
                                    [ 0  d ]
(%o75)                              [      ]
                                    [ 0  0 ]
(%i76) E21 . A . E12 . E21
                                    [ 0  0 ]
(%o76)                              [      ]
                                    [ a  0 ]
(%i77) E21 . A . E21 . E12
                                    [ 0  0 ]
(%o77)                              [      ]
                                    [ 0  b ]
(%i78) A . J
                                    [ b  a ]
(%o78)                              [      ]
                                    [ d  c ]
(%i79) A . J . J
                                    [ a  b ]
(%o79)                              [      ]
                                    [ c  d ]
(%i80) A . J . J . J
                                    [ b  a ]
(%o80)                              [      ]
                                    [ d  c ]
(%i81) A . J . J . J . J
                                    [ a  b ]
(%o81)                              [      ]
                                    [ c  d ]
(%i82) J . A
                                    [ c  d ]
(%o82)                              [      ]
                                    [ a  b ]
(%i83) J . J . A
                                    [ a  b ]
(%o83)                              [      ]
```

```
                                    [ c  d ]
(%i84) J . J . J . A
                                    [ c  d ]
(%o84)                              [      ]
                                    [ a  b ]
(%i85) J . J . J . J . A
                                    [ a  b ]
(%o85)                              [      ]
                                    [ c  d ]
(%i86) (J . A) . J
                                    [ d  c ]
(%o86)                              [      ]
                                    [ b  a ]
(%i87) (E12 . A) . E21
                                    [ d  0 ]
(%o87)                              [      ]
                                    [ 0  0 ]
(%i88) (E21 . A) . E12
                                    [ 0  0 ]
(%o88)                              [      ]
                                    [ 0  a ]
(%i89) (E12 . A) . E12
                                    [ 0  c ]
(%o89)                              [      ]
                                    [ 0  0 ]
(%i90) (E21 . A) . E21
                                    [ 0  0 ]
(%o90)                              [      ]
                                    [ b  0 ]
(%i91) E12 . A . E21
                                    [ d  0 ]
(%o91)                              [      ]
                                    [ 0  0 ]
(%i92) E21 . A . E12
                                    [ 0  0 ]
(%o92)                              [      ]
                                    [ 0  a ]
(%o93)         /Users/davidvajda/Documents/matrix02102026now.lsp
```

5.1.3 was benutze ich f"ur zeichen

- ich benutze, als komponenten `\{a,b,c,d\}`
  und nicht `\{a_{11},a_{12},a_{21},a_{22}`
  aus dem grund, dass bei einer umstellung das visuelle erkenenn nicht gut m"oglich ist

- ich weise darauf hin, dass eine matrix der form

$$\begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$$

in der form

$$\begin{pmatrix} a_{11} & a_{21} \\ a_{12} & a_{22} \end{pmatrix}$$

erscheinen k"onnte?

- frage: ja oder nein?
- antwort: definition matrizenmultiplikation siehe oben: nein: entsprechend

$$a_{ij} \cdot b_{jl}$$

- Array im PC? waagerechte und senkrechte (horizontal und vertikal) im PC nicht wirklich zu unterscheiden
- Gegenbeispiel: monitor: `80x25` (???)
- Matrix (Microarchitektur): Spalte und Zeile im `EEPROM` aber: keine auswirkung auf speicherung
- Normalerweise: Most Significant Index ist: Zeile, Least Significant Index: Spalte
- Blos: bei Matrix hingucken, nicht gut erkennbar und m"oglicherweise nie gesehene alternative f"ur viele von uns.

5.1.4 was auff"allt

- es gibt eine horizontale spiegelung
- es gibt eine vertikale spiegelung
- vermutung: es gibt eine horizontale und. vertikale spiegelung ja: wiederholtes anwenden des gleichen operators, zur gleichen stelle, so weit hier, erst mal nein.
- begriff: vertikale spiegelung
- begriff: horizontale spiegelung
- drehrichtung
- wiederholtes anwenden desselben operators
- R"uckw"artsdrehung
- Vorw"artsdrehung
- Vertikale Spiegelung
- Horizontale
- Gleicher Operator, gleiche Stelle, Invertierung des altes Schrittes:
- Vorsicht: Drehung, gleicher operator:
- Drehsinn: links, rechts

- Drehsinn: uhrzeigersinn, dagegen
- Dreh operator (Ausdruck): wiederholtes anwenden, rad dreht sich
- Spiegellungsoperator, Achse, Achsenspiegelung: Vertikal und Horizontal:
- Spiegelung wiederholtes anwenden, derselben Achsenspiegelung: invertiert die alte.
- Vorsicht: Achsenspiegelung und Punktspiegelung
- Vorsicht: horizontale Achse, Vertikale

5.1.5 wdh: beispiele

```
E12 . E21
E21 . E12
E12 . E21 . A . E12;
E21 . E12 . A . E12;
E12 . E21 . A . E21;
E21 . E12 . A . E21;
```

6 Die Operanden

- Die klassische Matrix A
- Die Matrizen mit einer 1 und sonst 0 (ausdruck, Mathematische Grundlagen, Fernuni Hagen, Prof. Luise Unger): E_{ij}. z.B.: $E_{11}, E_{12}, E_{21}, E_{22}$, in der form:
  - $E_{11} = \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix}$
  - $E_{12} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$
  - $E_{21} = \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}$
  - $E_{11} = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$
- die (ausdruck: mathematische Grundlagen, Fernuni Hagen, Prof. Luise Unger): ich glaube Einheitsmatrix kurs I_m oder $I_m m$

$$I_m = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

- jetzt eine einf"uhrung von mir - eher quatsch:

$$J_m = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{pmatrix}$$

bitte nicht ernst nehmen, noch mal: wichtig: zum einpraegen:

$$I_m = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

neutrales Element der matrizenmultiplikation

- und: wichtig: normale Matrizen immer mit A

6.1 Die Definition dieser Matrizen in maxima

```
A: matrix([a,b],[c,d]);
E11: matrix([1,0],[0,0]);
E12: matrix([0,1],[0,0]);
E21: matrix([0,0],[1,0]);
E22: matrix([0,0],[0,1]);
I: matrix([1,0],[0,1]);
J: matrix([0,1],[1,0]);
```

7 Files

8 hello02072026.asm

```
        global   _main
        section  .text
_main:  mov      rax, 0x02000004
        mov      rdi, 1
        mov      rsi, msg
        mov      rdx, 13
        syscall
        mov      rax, 0x02000001
        xor      rdi, rdi
        syscall

        section  .data
msg:    db       "Hello world!", 10
```

9 matrix02082026.asm

```
        global   _main
        section  .text
_main:
 mov rsi, num
 mov  rax, [rsi]
 mov rdi, numstr+15
l1:
 mov rbx, 10
 mov rdx, 0x00
```

```
 div rbx
 add dl, '0'
 mov [rdi], dl
 dec rdi
 cmp  rax, 0x00
 jnz  l1

 mov  rsi, numstr
l2:
 mov  ah, [rdi]
 add ah, '0'
 mov [rdi], ah
 dec rdi
 cmp rdi, rsi
 jnz l2

 mov      rax, 0x02000004
        mov      rdi, 1
        mov      rsi, numstr
        mov      rdx, 17
        syscall

        movq  xmm0, [A1]
        movq  xmm1, [E1]
 pmulhw xmm0, xmm1
 movq xmm2, xmm0

 movq xmm0, [A2]
 movq xmm1, [E1]
 pmulhw xmm0, xmm1
 paddw xmm3, xmm0

 movq xmm0, [A3]
 movq xmm1, [E1]
 pmulhw xmm0, xmm1
 paddw xmm4, xmm0

 movq xmm0, [A4]
 movq xmm1, [E1]
 pmulhw xmm0, xmm1
 paddw xmm5, xmm0
l3:


        mov      rax, 0x02000001
        xor      rdi, rdi
        syscall

        section  .data
numstr:
```

```
 db 0x00,0x00,0x00,0x00, 0x00,0x00,0x00,0x00, 0x00,0x00,0x00,0x00, 0x00,0x00,0x00,0x00, 10
A: A1: dw 0x0003, 0x0003, 0x0000, 0x0007
 A2: dw 0x0008, 0x0001, 0x0005, 0x0007
 A3: dw 0x0006, 0x0004, 0x0007, 0x0002
 A4: dw 0x0005, 0x0000, 0x0006, 0x0006
B: B1: dw 0x0006, 0x0007, 0x0007, 0x0004
 B2: dw 0x0006, 0x0006, 0x0000, 0x0000
 B3: dw 0x0002, 0x0006, 0x0000, 0x0005
 B4: dw 0x0006, 0x0006, 0x0006, 0x0009
 E1: dw 0x0001, 0x0000, 0x0000, 0x0000
 E1: dw 0x0000, 0x0001, 0x0000, 0x0000
 E1: dw 0x0000, 0x0000, 0x0001, 0x0000
 E1: dw 0x0000, 0x0000, 0x0000, 0x0001
num: dq 14823
```

10 matrix02082026.bak.asm

```
        global   _main
        section  .text
_main:
 mov rsi, num
 mov  rax, [rsi]
 mov rdi, numstr+15
l1:
 mov rbx, 10
 mov rdx, 0x00
 div rbx
 add dl, '0'
 mov [rdi], dl
 dec rdi
 cmp  rax, 0x00
 jnz  l1

 mov  rsi, numstr
l2:
 mov  ah, [rdi]
 add ah, '0'
 mov [rdi], ah
 dec rdi
 cmp rdi, rsi
 jnz l2

 mov      rax, 0x02000004
        mov      rdi, 1
        mov      rsi, numstr
        mov      rdx, 17
        syscall

        movq  xmm0, [A1]
```

```
        movq  xmm1, [M]
        movq  xmm2, [Z]

        pand  xmm0, xmm1
        movq  xmm2, xmm0
l3:


        mov      rax, 0x02000001
        xor      rdi, rdi
        syscall

        section  .data
numstr:
 db 0x00,0x00,0x00,0x00, 0x00,0x00,0x00,0x00, 0x00,0x00,0x00,0x00, 0x00,0x00,0x00,0x00, 10
A: A1: db 0x03, 0x03, 0x00, 0x07, 0x00, 0x00, 0x00, 0x00
 A2: db 0x08, 0x01, 0x05, 0x07, 0x00, 0x00, 0x00, 0x00
 A3: db 0x06, 0x04, 0x07, 0x02, 0x00, 0x00, 0x00, 0x00
 A4: db 0x05, 0x00, 0x06, 0x06, 0x00, 0x00, 0x00, 0x00
B: B1: db 0x06, 0x07, 0x07, 0x04, 0x00, 0x00, 0x00, 0x00
 B2: db 0x06, 0x06, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
 B3: db 0x02, 0x06, 0x00, 0x05, 0x00, 0x00, 0x00, 0x00
 B4: db 0x06, 0x06, 0x06, 0x09, 0x00, 0x00, 0x00, 0x00
 E1: db 0x01, 0x00, 0x00, 0x00,
num: dq 14823
```

11 matrix02082026b.asm

```
        global   _main
        section  .text
_main:
 mov rsi, num
 mov  rax, [rsi]
 mov rdi, numstr+15
l1:
 mov rbx, 10
 mov rdx, 0x00
 div rbx
 add dl, '0'
 mov [rdi], dl
 dec rdi
 cmp  rax, 0x00
 jnz  l1

 mov  rsi, numstr
l2:
 mov  ah, [rdi]
 add ah, '0'
 mov [rdi], ah
```

```
 dec rdi
 cmp rdi, rsi
 jnz l2

 mov      rax, 0x02000004
        mov      rdi, 1
        mov      rsi, numstr
        mov      rdx, 17
        syscall

        movq  xmm0, [A1]
        movq  xmm1, [M]
        movq  xmm2, [Z]

        pand  xmm0, xmm1
        movq  xmm2, xmm0
l3:


        mov      rax, 0x02000001
        xor      rdi, rdi
        syscall

        section  .data
numstr:
 db 0x00,0x00,0x00,0x00, 0x00,0x00,0x00,0x00, 0x00,0x00,0x00,0x00, 0x00,0x00,0x00,0x00, 10
A: A1: db 0x03, 0x03, 0x00, 0x07, 0x00, 0x00, 0x00, 0x00
 A2: db 0x08, 0x01, 0x05, 0x07, 0x00, 0x00, 0x00, 0x00
 A3: db 0x06, 0x04, 0x07, 0x02, 0x00, 0x00, 0x00, 0x00
 A4: db 0x05, 0x00, 0x06, 0x06, 0x00, 0x00, 0x00, 0x00
B: B1: db 0x06, 0x07, 0x07, 0x04, 0x00, 0x00, 0x00, 0x00
 B2: db 0x06, 0x06, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
 B3: db 0x02, 0x06, 0x00, 0x05, 0x00, 0x00, 0x00, 0x00
 B4: db 0x06, 0x06, 0x06, 0x09, 0x00, 0x00, 0x00, 0x00
 E1: db 0x01, 0x00, 0x00, 0x00,
num: dq 14823
```

12 matrix02092026.asm

```
        global   _main
        section  .text
_main:
 mov rsi, num
 mov  rax, [rsi]
 mov rdi, numstr+15
l1:
 mov rbx, 10
 mov rdx, 0x00
 div rbx
```

```
 add dl, '0'
 mov [rdi], dl
 dec rdi
 cmp  rax, 0x00
 jnz  l1

 mov  rsi, numstr
l2:
 mov  ah, [rdi]
 add ah, '0'
 mov [rdi], ah
 dec rdi
 cmp rdi, rsi
 jnz l2

 mov      rax, 0x02000004
        mov      rdi, 1
        mov      rsi, numstr
        mov      rdx, 17
        syscall

        movq  xmm0, [A1]
        movq  xmm1, [M]
        movq  xmm2, [Z]

        pand  xmm0, xmm1
        movq  xmm2, xmm0
l3:


        mov      rax, 0x02000001
        xor      rdi, rdi
        syscall

        section  .data
numstr:
 db 0x00,0x00,0x00,0x00, 0x00,0x00,0x00,0x00, 0x00,0x00,0x00,0x00, 0x00,0x00,0x00,0x00, 10
A: A1: db 0x03, 0x03, 0x00, 0x07, 0x00, 0x00, 0x00, 0x00
 A2: db 0x08, 0x01, 0x05, 0x07, 0x00, 0x00, 0x00, 0x00
 A3: db 0x06, 0x04, 0x07, 0x02, 0x00, 0x00, 0x00, 0x00
 A4: db 0x05, 0x00, 0x06, 0x06, 0x00, 0x00, 0x00, 0x00
B: B1: db 0x06, 0x07, 0x07, 0x04, 0x00, 0x00, 0x00, 0x00
 B2: db 0x06, 0x06, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00
 B3: db 0x02, 0x06, 0x00, 0x05, 0x00, 0x00, 0x00, 0x00
 B4: db 0x06, 0x06, 0x06, 0x09, 0x00, 0x00, 0x00, 0x00
 E1: db 0x01, 0x00, 0x00, 0x00,
num: dq 14823
```

13 matrix02102026.lsp

```
NIL
/* Starts dribbling to matrix02102026.lsp (2026/2/10, 06:06:16).*/
(%o20)                                done
(%i21) save



asdas

incorrect syntax: asdas is not an infix operator

 ^

(%i21)
incorrect syntax: test is not an infix operator
save test.
       ^
(%i21)
incorrect syntax: test is not an infix operator

 ^
(%i21)
incorrect syntax: end of file while scanning expression.

 ^
(%i21)
```

14 matrix02102026now.lsp

```
A: matrix([a,b],[c,d]);
E11: matrix([1,0],[0,0]);
E12: matrix([0,1],[0,0]);
E21: matrix([0,0],[1,0]);
E22: matrix([0,0],[0,1]);
I: matrix([1,0],[0,1]);
J: matrix([0,1],[1,0]);

E11 . A;
E12 . A;
E21 . A;
E22 . A;
A . E11;
A . E12;
A . E21;
A . E22;
E12 . E21 . A . E11;
```

```
J . A . E12 . E21;
E12 . E21 . A;
E12 . E21 . A . E12 . E21;
E12 . E21 . A . I;
I . A . E21;
J . A . J;
J . A . J . J;
J . J . A . J . J;
J . A . J . J;
J . J . A . J;
J . A;
J . J . A;
J . J . J . A;
J . J . J . J . A;
A . J;
A . J . J;
A . J . J . J;
A . J . J . J . J;
J . A . J . J . J;
J . A . J . J;
J . A . J;
J . J . A . J;
J . J . A . J . J;
J . J . A . J . J . J;
J . A . E11 . E22;
J . A . E22 . E11;
J . A . E11 . E12;
J . A . E12 . E11;
J . A . E11 . E21;
J . A . E21 . E11;
J . A . E12 . E21;
J . A . E21 . E12;


J . A;
A . J;

A . E12 . E21;
A . E21 . E12;

J . A . E12 . E21;
A . J . E21 . E12;
A . J . E12 . E21;
J . A . E21 . E12;

E12 . E21 . A . E12;
E12 . A . E12 . E21;
E12 . J . A . E12 . E21;
E12 . A . J . E12 . E21;
```

```
J . A;
A . J;

I . A;
A . I;

E12 . E21 . A . E21;
E21 . E12 . A . E12;
E21 . E12 . A . E21;
J . A . E21;

E12 . E21 . A . E12;
E21 . E12 . A . E12;
E12 . E21 . A . E21;
E21 . E12 . A . E21;

E12 . A . E12 . E21;
E12 . A . E21 . E12;
E21 . A . E12 . E21;
E21 . A . E21 . E12;

A . J;
A . J . J;
A . J . J . J;
A . J . J . J . J;
J . A;
J . J . A;
J . J . J . A;
J . J . J . J . A;

(J . A) . J;

(E12 . A) . E21;
(E21 . A) . E12;
(E12 . A) . E12;
(E21 . A) . E21;
E12 . (A . E21);
E21 . (A . E12);
```

15 matrixMMX02072026.asm

```
    global _main
    section    .data
 num dw 12483
 numstr  db 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 13, 10
    section    .text

    _main:
```

```
    mov rdx, 0x0f
    mov rcx, numstr
    mov rbx, 1
    mov rax, 4
    int 0x80

    mov rbx, 0
    mov rax, 1
    int 80h
```

16 numout02072026.asm

```
        global   _main
        section  .text
_main:
 mov rsi, num
 mov  rax, [rsi]
 mov rdi, numstr
l1:
 mov rbx, 10
 mov rdx, 0x00
 div rbx
 add rdx, '0'
 mov [rdi], dl
 inc rdi
 cmp  rax, 0x00
 jnz l1

 mov      rax, 0x02000004
        mov      rdi, 0x00000001
        mov      rsi, numstr
        mov      rdx, 6
        syscall
        mov      rax, 0x02000001
        xor      rdi, rdi
        syscall

        section  .data
numstr:    db       0,0,0,0,0,10,0,0,0,0,0,0,0,0,0, 10
num: dq 14823
```

17 numoutLSB02072026.asm

```
        global   _main
        section  .text
_main:
 mov rsi, num
```

```
 mov  rax, [rsi]
 mov rdi, numstr+15
l1:
 mov rbx, 10
 mov rdx, 0x00
 div rbx
 add dl, '0'
 mov [rdi], dl
 dec rdi
 cmp  rax, 0x00
 jnz  l1

 mov  rsi, numstr
l2:
 mov  ah, [rdi]
 add ah, '0'
 mov [rdi], ah
 dec rdi
 cmp rdi, rsi
 jnz l2

 mov      rax, 0x02000004
        mov      rdi, 1
        mov      rsi, numstr
        mov      rdx, 17
        syscall
        mov      rax, 0x02000001
        xor      rdi, rdi
        syscall

        section  .data
numstr:
 db 0x00,0x00,0x00,0x00, 0x00,0x00,0x00,0x00, 0x00,0x00,0x00,0x00, 0x00,0x00,0x00,0x00, 10
num: dq 14823
```

18 shmaximaterminal02102026.txt

```
32841
davidvajda@MBP-von-David Documents % nasm -f macho64 numoutLSB02072026.asm
davidvajda@MBP-von-David Documents % ld numoutLSB02072026.o -o numoutLSB02072026 -macosx_v
ld: warning: PIE disabled. Absolute addressing (perhaps -mdynamic-no-pic) not allowed in c
davidvajda@MBP-von-David Documents % ./numout02072026
32841
davidvajda@MBP-von-David Documents % nasm -f macho64 numoutLSB02072026.asm
davidvajda@MBP-von-David Documents % ld numoutLSB02072026.o -o numoutLSB02072026 -macosx_v
ld: warning: PIE disabled. Absolute addressing (perhaps -mdynamic-no-pic) not allowed in c
davidvajda@MBP-von-David Documents % ./numoutLSB02072026
000000000014823
davidvajda@MBP-von-David Documents % maxima
```

```
Maxima 5.47.0 https://maxima.sourceforge.io
using Lisp SBCL 2.6.1
Distributed under the GNU Public License. See the file COPYING.
Dedicated to the memory of William Schelter.
The function bug_report() provides bug reporting information.
(%i1) A: matrix([a,b],[c,d]);
                                  [ a  b ]
(%o1)                             [      ]
                                  [ c  d ]
(%i2) E11: matrix([1,0],[0,0]);
                                  [ 1  0 ]
(%o2)                             [      ]
                                  [ 0  0 ]
(%i3) A . E11;
                                  [ a  0 ]
(%o3)                             [      ]
                                  [ c  0 ]
(%i4) E11 . A;
                                  [ a  b ]
(%o4)                             [      ]
                                  [ 0  0 ]
(%i5) I: matrix([1,0],[0,1]);
                                  [ 1  0 ]
(%o5)                             [      ]
                                  [ 0  1 ]
(%i6) J: matrix([0,1],[1,0]);
                                  [ 0  1 ]
(%o6)                             [      ]
                                  [ 1  0 ]
(%i7) I . A;
                                  [ a  b ]
(%o7)                             [      ]
                                  [ c  d ]
(%i8) A . I;
                                  [ a  b ]
(%o8)                             [      ]
                                  [ c  d ]
(%i9) J . A;
                                  [ c  d ]
(%o9)                             [      ]
                                  [ a  b ]
(%i10) A . J;
                                  [ b  a ]
(%o10)                            [      ]
                                  [ d  c ]
(%i11) E12: matrix([0,1],[0,0]);
                                  [ 0  1 ]
(%o11)                            [      ]
                                  [ 0  0 ]
(%i12) A . E12;
```

```
                                    [ 0  a ]
(%o12)                              [      ]
                                    [ 0  c ]
(%i13) E12 . A;
                                    [ c  d ]
(%o13)                              [      ]
                                    [ 0  0 ]
(%i14) E12 . A . E12;
                                    [ 0  c ]
(%o14)                              [      ]
                                    [ 0  0 ]
(%i15) E11: matrix([1,0],[0,]);

incorrect syntax: Illegal use of delimiter ]
1: matrix([1,0],[0,]
                  ^
(%i15) E11: matrix([1,0],[0,0]);
                                    [ 1  0 ]
(%o15)                              [      ]
                                    [ 0  0 ]
(%i16) E21: matrix([0,0],[1,0]);
                                    [ 0  0 ]
(%o16)                              [      ]
                                    [ 1  0 ]
(%i17) E22: matrix([0,0],[0,1]);
                                    [ 0  0 ]
(%o17)                              [      ]
                                    [ 0  1 ]
(%i18) E11 . A . E11;
                                    [ a  0 ]
(%o18)                              [      ]
                                    [ 0  0 ]
(%i19) E11 . A . E11 . E12;
                                    [ 0  a ]
(%o19)                              [      ]
                                    [ 0  0 ]
(%i20) writefile("matrix02102026.lsp");

NIL
/* Starts dribbling to matrix02102026.lsp (2026/2/10, 06:06:16).*/
(%o20)                               done
(%i21)
```

19 matrix02102026now.lsp.out.txt

```
Maxima 5.47.0 https://maxima.sourceforge.io
using Lisp SBCL 2.6.1
Distributed under the GNU Public License. See the file COPYING.
Dedicated to the memory of William Schelter.
```

```
The function bug_report() provides bug reporting information.
(%i1) batch("matrix02102026now.lsp")

read and interpret /Users/davidvajda/Documents/matrix02102026now.lsp
(%i2) A:matrix([a,b],[c,d])
                                    [ a  b ]
(%o2)                               [      ]
                                    [ c  d ]
(%i3) E11:matrix([1,0],[0,0])
                                    [ 1  0 ]
(%o3)                               [      ]
                                    [ 0  0 ]
(%i4) E12:matrix([0,1],[0,0])
                                    [ 0  1 ]
(%o4)                               [      ]
                                    [ 0  0 ]
(%i5) E21:matrix([0,0],[1,0])
                                    [ 0  0 ]
(%o5)                               [      ]
                                    [ 1  0 ]
(%i6) E22:matrix([0,0],[0,1])
                                    [ 0  0 ]
(%o6)                               [      ]
                                    [ 0  1 ]
(%i7) I:matrix([1,0],[0,1])
                                    [ 1  0 ]
(%o7)                               [      ]
                                    [ 0  1 ]
(%i8) J:matrix([0,1],[1,0])
                                    [ 0  1 ]
(%o8)                               [      ]
                                    [ 1  0 ]
(%i9) E11 . A
                                    [ a  b ]
(%o9)                               [      ]
                                    [ 0  0 ]
(%i10) E12 . A
                                    [ c  d ]
(%o10)                              [      ]
                                    [ 0  0 ]
(%i11) E21 . A
                                    [ 0  0 ]
(%o11)                              [      ]
                                    [ a  b ]
(%i12) E22 . A
                                    [ 0  0 ]
(%o12)                              [      ]
                                    [ c  d ]
(%i13) A . E11
                                    [ a  0 ]
```

```
(%o13)                                [      ]
                                      [ c  0 ]
(%i14) A . E12
                                      [ 0  a ]
(%o14)                                [      ]
                                      [ 0  c ]
(%i15) A . E21
                                      [ b  0 ]
(%o15)                                [      ]
                                      [ d  0 ]
(%i16) A . E22
                                      [ 0  b ]
(%o16)                                [      ]
                                      [ 0  d ]
(%i17) E12 . E21 . A . E11
                                      [ a  0 ]
(%o17)                                [      ]
                                      [ 0  0 ]
(%i18) J . A . E12 . E21
                                      [ c  0 ]
(%o18)                                [      ]
                                      [ a  0 ]
(%i19) E12 . E21 . A
                                      [ a  b ]
(%o19)                                [      ]
                                      [ 0  0 ]
(%i20) E12 . E21 . A . E12 . E21
                                      [ a  0 ]
(%o20)                                [      ]
                                      [ 0  0 ]
(%i21) E12 . E21 . A . I
                                      [ a  b ]
(%o21)                                [      ]
                                      [ 0  0 ]
(%i22) I . A . E21
                                      [ b  0 ]
(%o22)                                [      ]
                                      [ d  0 ]
(%i23) J . A . J
                                      [ d  c ]
(%o23)                                [      ]
                                      [ b  a ]
(%i24) J . A . J . J
                                      [ c  d ]
(%o24)                                [      ]
                                      [ a  b ]
(%i25) J . J . A . J . J
                                      [ a  b ]
(%o25)                                [      ]
                                      [ c  d ]
```

```
(%i26) J . A . J . J
                                    [ c  d ]
(%o26)                              [      ]
                                    [ a  b ]
(%i27) J . J . A . J
                                    [ b  a ]
(%o27)                              [      ]
                                    [ d  c ]
(%i28) J . A
                                    [ c  d ]
(%o28)                              [      ]
                                    [ a  b ]
(%i29) J . J . A
                                    [ a  b ]
(%o29)                              [      ]
                                    [ c  d ]
(%i30) J . J . J . A
                                    [ c  d ]
(%o30)                              [      ]
                                    [ a  b ]
(%i31) J . J . J . J . A
                                    [ a  b ]
(%o31)                              [      ]
                                    [ c  d ]
(%i32) A . J
                                    [ b  a ]
(%o32)                              [      ]
                                    [ d  c ]
(%i33) A . J . J
                                    [ a  b ]
(%o33)                              [      ]
                                    [ c  d ]
(%i34) A . J . J . J
                                    [ b  a ]
(%o34)                              [      ]
                                    [ d  c ]
(%i35) A . J . J . J . J
                                    [ a  b ]
(%o35)                              [      ]
                                    [ c  d ]
(%i36) J . A . J . J . J
                                    [ d  c ]
(%o36)                              [      ]
                                    [ b  a ]
(%i37) J . A . J . J
                                    [ c  d ]
(%o37)                              [      ]
                                    [ a  b ]
(%i38) J . A . J
                                    [ d  c ]
```

```
(%o38)                                [      ]
                                      [ b  a ]
(%i39) J . J . A . J
                                      [ b  a ]
(%o39)                                [      ]
                                      [ d  c ]
(%i40) J . J . A . J . J
                                      [ a  b ]
(%o40)                                [      ]
                                      [ c  d ]
(%i41) J . J . A . J . J . J
                                      [ b  a ]
(%o41)                                [      ]
                                      [ d  c ]
(%i42) J . A . E11 . E22
                                      [ 0  0 ]
(%o42)                                [      ]
                                      [ 0  0 ]
(%i43) J . A . E22 . E11
                                      [ 0  0 ]
(%o43)                                [      ]
                                      [ 0  0 ]
(%i44) J . A . E11 . E12
                                      [ 0  c ]
(%o44)                                [      ]
                                      [ 0  a ]
(%i45) J . A . E12 . E11
                                      [ 0  0 ]
(%o45)                                [      ]
                                      [ 0  0 ]
(%i46) J . A . E11 . E21
                                      [ 0  0 ]
(%o46)                                [      ]
                                      [ 0  0 ]
(%i47) J . A . E21 . E11
                                      [ d  0 ]
(%o47)                                [      ]
                                      [ b  0 ]
(%i48) J . A . E12 . E21
                                      [ c  0 ]
(%o48)                                [      ]
                                      [ a  0 ]
(%i49) J . A . E21 . E12
                                      [ 0  d ]
(%o49)                                [      ]
                                      [ 0  b ]
(%i50) J . A
                                      [ c  d ]
(%o50)                                [      ]
                                      [ a  b ]
```

```
(%i51) A . J
                                    [ b  a ]
(%o51)                              [      ]
                                    [ d  c ]
(%i52) A . E12 . E21
                                    [ a  0 ]
(%o52)                              [      ]
                                    [ c  0 ]
(%i53) A . E21 . E12
                                    [ 0  b ]
(%o53)                              [      ]
                                    [ 0  d ]
(%i54) J . A . E12 . E21
                                    [ c  0 ]
(%o54)                              [      ]
                                    [ a  0 ]
(%i55) A . J . E21 . E12
                                    [ 0  a ]
(%o55)                              [      ]
                                    [ 0  c ]
(%i56) A . J . E12 . E21
                                    [ b  0 ]
(%o56)                              [      ]
                                    [ d  0 ]
(%i57) J . A . E21 . E12
                                    [ 0  d ]
(%o57)                              [      ]
                                    [ 0  b ]
(%i58) E12 . E21 . A . E12
                                    [ 0  a ]
(%o58)                              [      ]
                                    [ 0  0 ]
(%i59) E12 . A . E12 . E21
                                    [ c  0 ]
(%o59)                              [      ]
                                    [ 0  0 ]
(%i60) E12 . J . A . E12 . E21
                                    [ a  0 ]
(%o60)                              [      ]
                                    [ 0  0 ]
(%i61) E12 . A . J . E12 . E21
                                    [ d  0 ]
(%o61)                              [      ]
                                    [ 0  0 ]
(%i62) J . A
                                    [ c  d ]
(%o62)                              [      ]
                                    [ a  b ]
(%i63) A . J
                                    [ b  a ]
```

```
(%o63)                                [      ]
                                      [ d  c ]
(%i64) I . A
                                      [ a  b ]
(%o64)                                [      ]
                                      [ c  d ]
(%i65) A . I
                                      [ a  b ]
(%o65)                                [      ]
                                      [ c  d ]
(%i66) E12 . E21 . A . E21
                                      [ b  0 ]
(%o66)                                [      ]
                                      [ 0  0 ]
(%i67) E21 . E12 . A . E12
                                      [ 0  0 ]
(%o67)                                [      ]
                                      [ 0  c ]
(%i68) E21 . E12 . A . E21
                                      [ 0  0 ]
(%o68)                                [      ]
                                      [ d  0 ]
(%i69) J . A . E21
                                      [ d  0 ]
(%o69)                                [      ]
                                      [ b  0 ]
(%i70) E12 . E21 . A . E12
                                      [ 0  a ]
(%o70)                                [      ]
                                      [ 0  0 ]
(%i71) E21 . E12 . A . E12
                                      [ 0  0 ]
(%o71)                                [      ]
                                      [ 0  c ]
(%i72) E12 . E21 . A . E21
                                      [ b  0 ]
(%o72)                                [      ]
                                      [ 0  0 ]
(%i73) E21 . E12 . A . E21
                                      [ 0  0 ]
(%o73)                                [      ]
                                      [ d  0 ]
(%i74) E12 . A . E12 . E21
                                      [ c  0 ]
(%o74)                                [      ]
                                      [ 0  0 ]
(%i75) E12 . A . E21 . E12
                                      [ 0  d ]
(%o75)                                [      ]
                                      [ 0  0 ]
```

```
(%i76) E21 . A . E12 . E21
                                    [ 0  0 ]
(%o76)                              [      ]
                                    [ a  0 ]
(%i77) E21 . A . E21 . E12
                                    [ 0  0 ]
(%o77)                              [      ]
                                    [ 0  b ]
(%i78) A . J
                                    [ b  a ]
(%o78)                              [      ]
                                    [ d  c ]
(%i79) A . J . J
                                    [ a  b ]
(%o79)                              [      ]
                                    [ c  d ]
(%i80) A . J . J . J
                                    [ b  a ]
(%o80)                              [      ]
                                    [ d  c ]
(%i81) A . J . J . J . J
                                    [ a  b ]
(%o81)                              [      ]
                                    [ c  d ]
(%i82) J . A
                                    [ c  d ]
(%o82)                              [      ]
                                    [ a  b ]
(%i83) J . J . A
                                    [ a  b ]
(%o83)                              [      ]
                                    [ c  d ]
(%i84) J . J . J . A
                                    [ c  d ]
(%o84)                              [      ]
                                    [ a  b ]
(%i85) J . J . J . J . A
                                    [ a  b ]
(%o85)                              [      ]
                                    [ c  d ]
(%i86) (J . A) . J
                                    [ d  c ]
(%o86)                              [      ]
                                    [ b  a ]
(%i87) (E12 . A) . E21
                                    [ d  0 ]
(%o87)                              [      ]
                                    [ 0  0 ]
(%i88) (E21 . A) . E12
                                    [ 0  0 ]
```

```
(%o88)                                [      ]
                                      [ 0   a ]
(%i89) (E12 . A) . E12
                                      [ 0   c ]
(%o89)                                [      ]
                                      [ 0   0 ]
(%i90) (E21 . A) . E21
                                      [ 0   0 ]
(%o90)                                [      ]
                                      [ b   0 ]
(%i91) E12 . A . E21
                                      [ d   0 ]
(%o91)                                [      ]
                                      [ 0   0 ]
(%i92) E21 . A . E12
                                      [ 0   0 ]
(%o92)                                [      ]
                                      [ 0   a ]
(%o93)         /Users/davidvajda/Documents/matrix02102026now.lsp
```