A Framework for Pen-Based Mathematical Computing

Stephen M. Watt

Ontario Research Centre for Computer Algebra
University of Western Ontario

`www.orcca.on.ca/PenMath`

IAMC Beijing July 24 2005

Western
125 YEARS
est. 1878

ORCCA
Ontario Research Centre for Computer Algebra

ORCCA

Ontario Research Centre for Computer Algebra

- **Symbolic computation group:**

  Western: Corless, Jeffrey, Moreno Maza, Reid, Watt
  Waterloo: Geddes, Giesbrecht, Labahn, Storjohan
  Associate: Carette, Devitt, Kotsireas, Nedialkov, Wolf, Zima

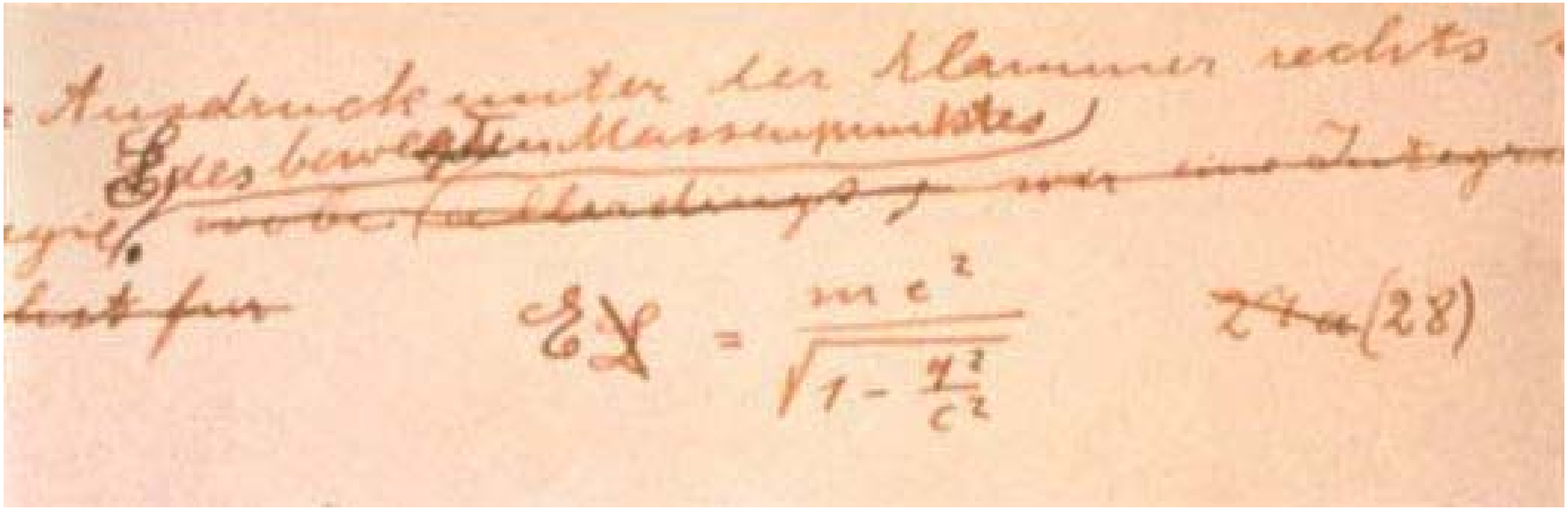
- **Problems:**

  Classical CA: linear algebra, $\int$ and DEs, polynomial systems,…
  CAS: Maple, Aldor/libalgebra, …
  SNAP: gcd, decomposition, factorization, …
  Prog Lang support: type systems, compilers, mem. Management, …
  MKM and Interfaces: MathML, data conv, web services, *pen*, …

Outline

- *Introduction*
- *Approach*
- *Components*
- *Cheating*
- *Portability*
- *Prototype*
- *Conclusions*

Introduction



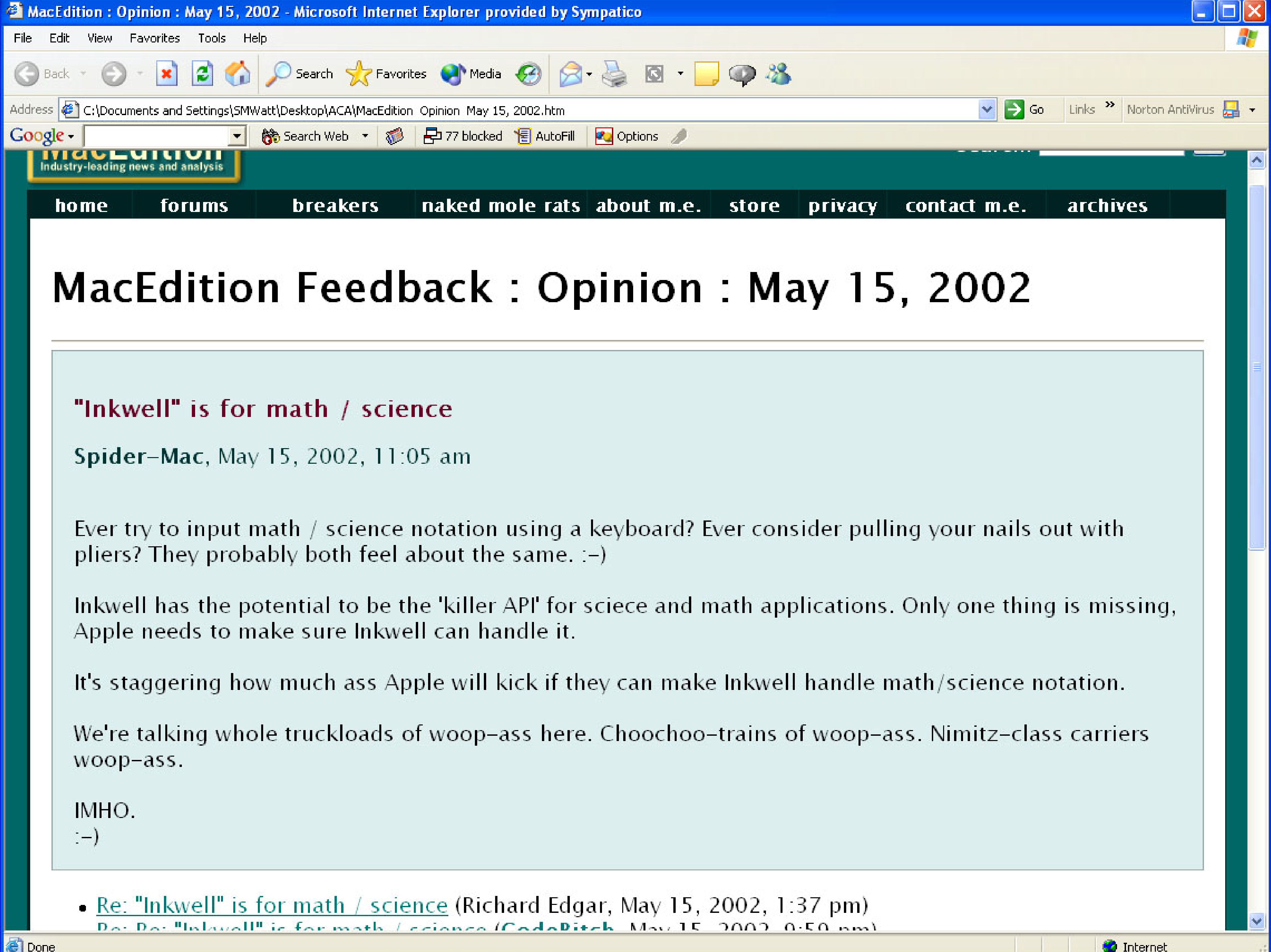
Ausdruck unter der Klammer rechts

E des bewegten Massenpunktes

ergie ~~[illegible]~~

~~ist für~~

$$\mathfrak{E} = \frac{mc^2}{\sqrt{1-\frac{q^2}{c^2}}} \qquad (28)$$


MacEdition : Opinion : May 15, 2002 - Microsoft Internet Explorer provided by Sympatico
File Edit View Favorites Tools Help
Back
Search
Favorites
Media
Address
C:\Documents and Settings\SMWatt\Desktop\ACA\MacEdition Opinion May 15, 2002.htm
Go
Links
Norton AntiVirus
Google
Search Web
77 blocked
AutoFill
Options
Industry-leading news and analysis
home
forums
breakers
naked mole rats
about m.e.
store
privacy
contact m.e.
archives
Done
Internet

MacEdition Feedback : Opinion : May 15, 2002

"Inkwell" is for math / science

Spider-Mac, May 15, 2002, 11:05 am

Ever try to input math / science notation using a keyboard? Ever consider pulling your nails out with pliers? They probably both feel about the same. :-)

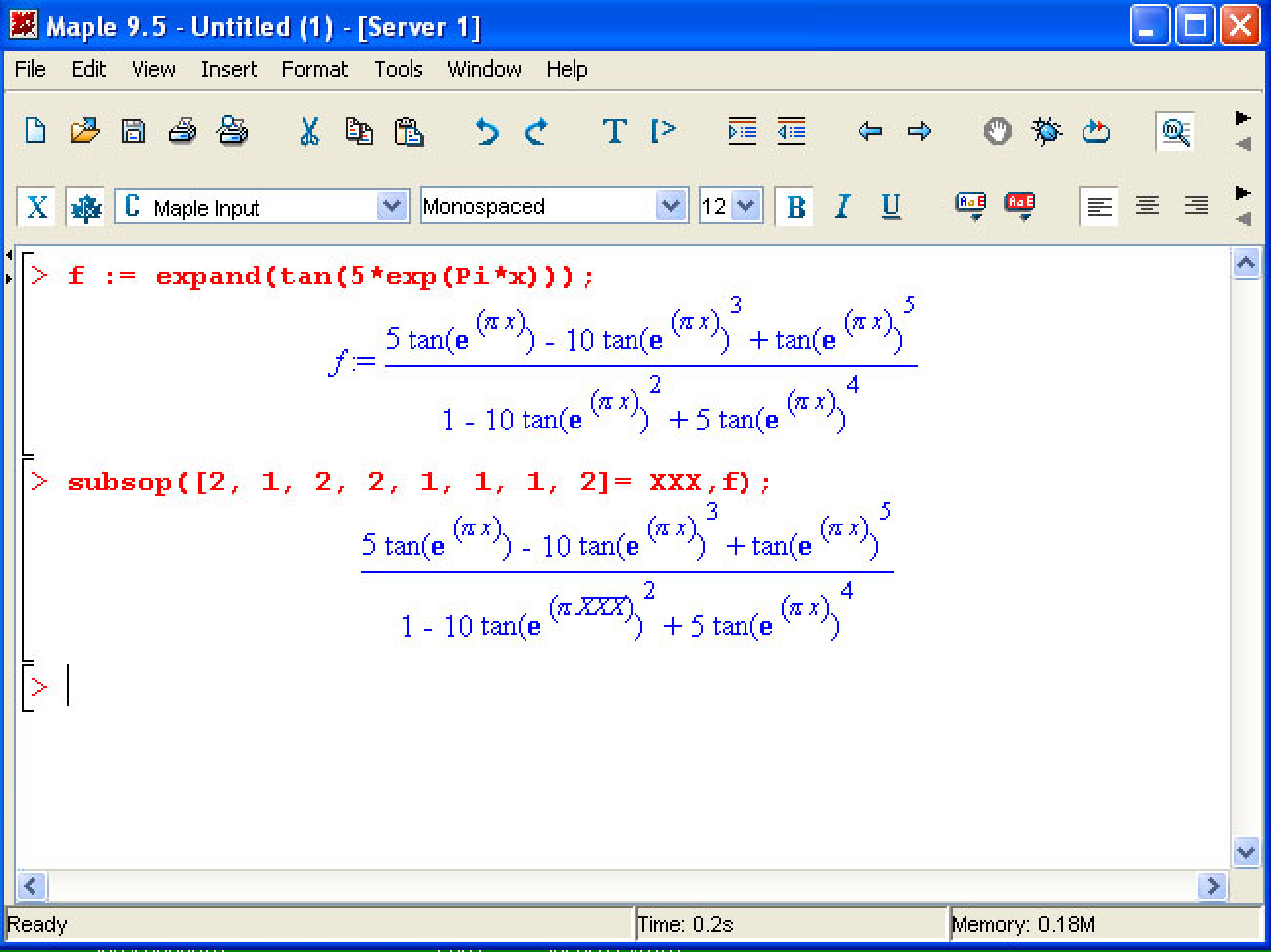
Inkwell has the potential to be the 'killer API' for sciece and math applications. Only one thing is missing, Apple needs to make sure Inkwell can handle it.

It's staggering how much ass Apple will kick if they can make Inkwell handle math/science notation.

We're talking whole truckloads of woop-ass here. Choochoo-trains of woop-ass. Nimitz-class carriers woop-ass.

IMHO.
:-)

- Re: "Inkwell" is for math / science (Richard Edgar, May 15, 2002, 1:37 pm)
- Re: Re: "Inkwell" is for math / science (CodeBitch, May 15, 2002, 9:59 pm)



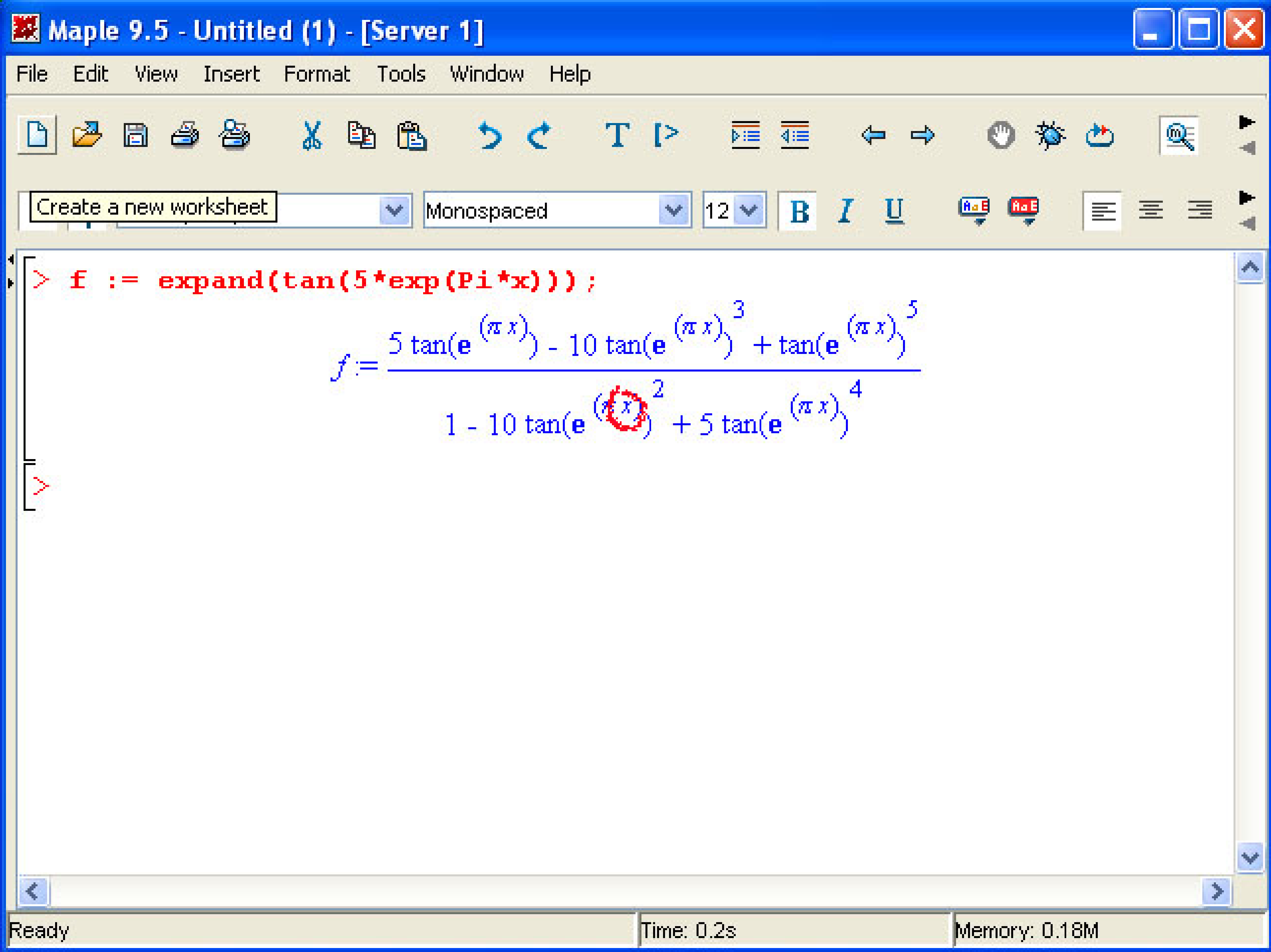
```
> f := expand(tan(5*exp(Pi*x)));
```

$$f := \frac{5\tan(\mathrm{e}^{(\pi x)}) - 10\tan(\mathrm{e}^{(\pi x)})^3 + \tan(\mathrm{e}^{(\pi x)})^5}{1 - 10\tan(\mathrm{e}^{(\pi x)})^2 + 5\tan(\mathrm{e}^{(\pi x)})^4}$$

```
> subsop([2, 1, 2, 2, 1, 1, 1, 2]= XXX,f);
```

$$\frac{5\tan(\mathrm{e}^{(\pi x)}) - 10\tan(\mathrm{e}^{(\pi x)})^3 + \tan(\mathrm{e}^{(\pi x)})^5}{1 - 10\tan(\mathrm{e}^{(\pi XXX)})^2 + 5\tan(\mathrm{e}^{(\pi x)})^4}$$

```
> 
```



```
> f := expand(tan(5*exp(Pi*x)));
```

$$f := \frac{5\tan\left(e^{(\pi x)}\right) - 10\tan\left(e^{(\pi x)}\right)^{3} + \tan\left(e^{(\pi x)}\right)^{5}}{1 - 10\tan\left(e^{(\pi x)}\right)^{2} + 5\tan\left(e^{(\pi x)}\right)^{4}}$$

```
>
```

Ready

Time: 0.2s

Memory: 0.18M

Tablet PC

Meeting
This is priority!
Numbers for February
List of invoices

Long-Term Goals

- Enter and manipulate math naturally by pen
- Support high-powered math transforms
- Support collaboration
- Do so portably, across applications and platforms

Previous Efforts

- Graduate student creates rudimentary version of each element (character recognition, spatial grouping, math semantics)
- Recognize $a^2 + b^2 = c^2$
- Leave “a few remaining cases” to “future work”
- Repeat…
- Exceptions
  Suzuki et al -- Infty project
  XThink – elementary math
  Fateman et al -- OCR

Approach

- Architect for a large problem, with many interacting components
- Combination of drawing and handwriting requiring special APIs
- Recognize that each component requires research related to mathematical nature
- Use CA for expression transformation
- Recognize on-going hardware evolution

- Project with Western, Waterloo, Maple, Microsoft

Typical Handwriting Recognition

- Segment input to words
- Break words into glyph candidates
- Compute possibilities for each glyph
- Dictionary determines most likely alternatives

Pen-based Mathematics System

- Mathematics input:
  -- character recognition,
  -- layout parsing,
  -- linear parsing
- Mathematics editing:
  -- subexpression selection,
  -- searching and linking,
  -- expression re-arrangement,
  -- expression transformation
  e.g. **`expand(sin(a+b))`**
  or **`factor(p)`**

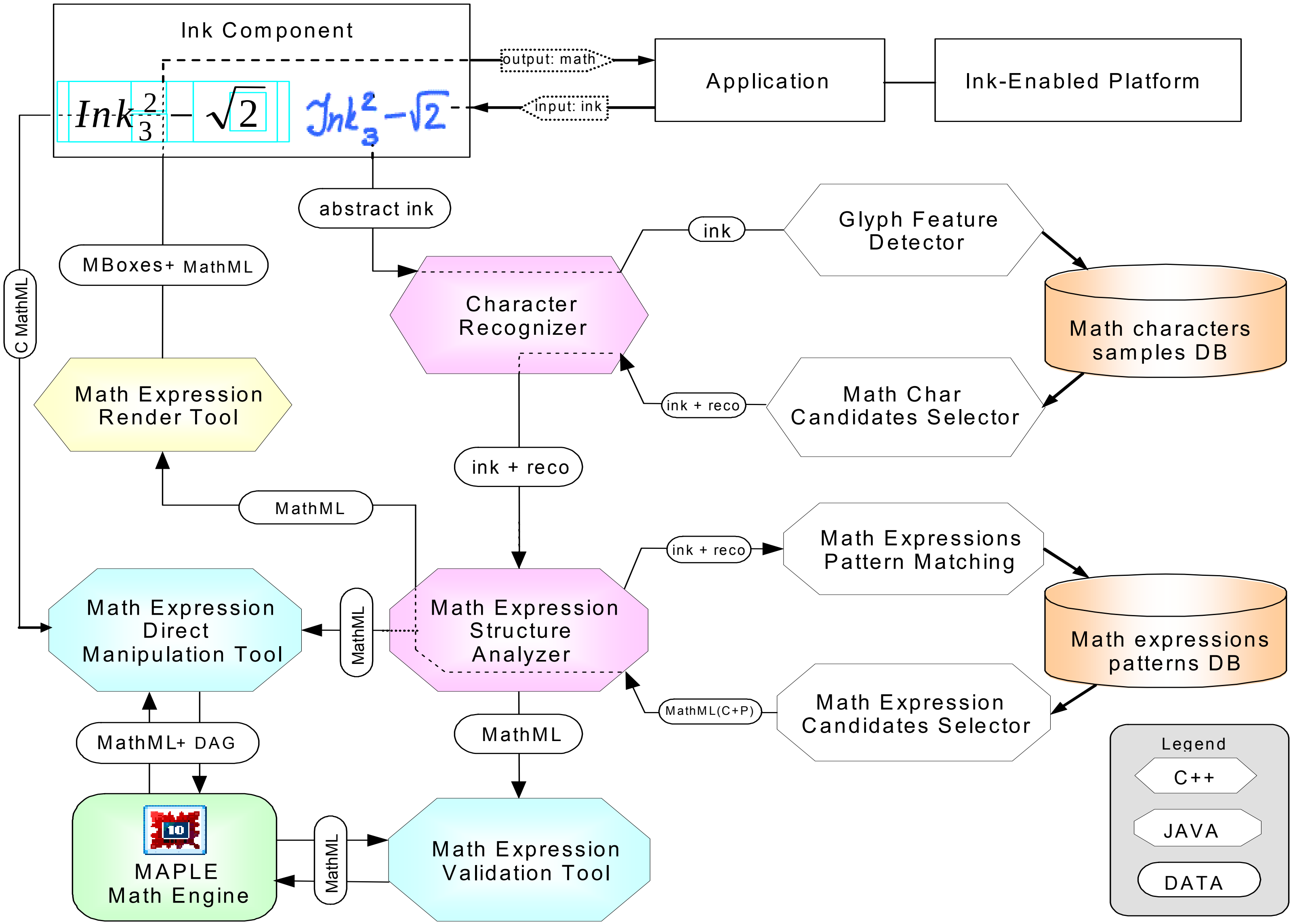
- Sketching
- Re-winding and re-playing derivations
- Visual scenario/case organization
- Spreadsheet-like recalculation
- Collaboration

Early Projects at ORCCA

- CrossPad
  w Louie (2000) Off-line analysis

- Pocket PC
  w Wan (2001) Elastic matching, alternative prompting
- Single-line expression grouping
  w So (2003)
- Notation Selection Tool
  w Liu, Smirnova (2000-2003)
- Expression Transformation
  w Huerter Li Rodionov Smirnova So (1999-2004)
  TeX ↔ MathML ↔ OpenMath ↔ Maple

Components and relations


Ink Component
$Ink_3^2 - \sqrt{2}$
output: math
input: ink
Application
Ink-Enabled Platform
abstract ink
MBoxes+ MathML
C MathML
ink
Glyph Feature Detector
Character Recognizer
Math characters samples DB
ink + reco
Math Char Candidates Selector
Math Expression Render Tool
ink + reco
MathML
ink + reco
Math Expressions Pattern Matching
Math Expression Direct Manipulation Tool
MathML
Math Expression Structure Analyzer
Math expressions patterns DB
MathML(C+P)
Math Expression Candidates Selector
MathML+ DAG
MathML
10
MAPLE Math Engine
MathML
Math Expression Validation Tool
Legend
C++
JAVA
DATA

Character Recognition

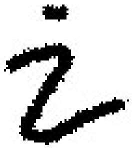
- Large vocabulary of mathematical symbols
- Usual trade off is #symbols vs accuracy

ɑ vs α vs proportionality …

- ~ 2000 named entities in MathML
- Trick of using special alphabet doesn’t work

- Stronger feature identification (w X. Xie)
- Heavier use of context (w So)

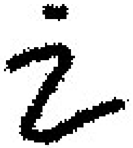
Ambiguities



Ambiguities

$$\sum_{i} i^2$$

Ambiguities



Ambiguities

$$\dot{z} + z = \sin \omega t$$

Problems

Why is math different?

- The set of symbols is large.
- No specific stroke order and stroke number.
- Spatial relation gives complex context-sensitive two dimensional rules.

Recognition in Large Symbol Sets

- Vendor APIs insufficient
  – limited to Roman or Chinese/Japanese

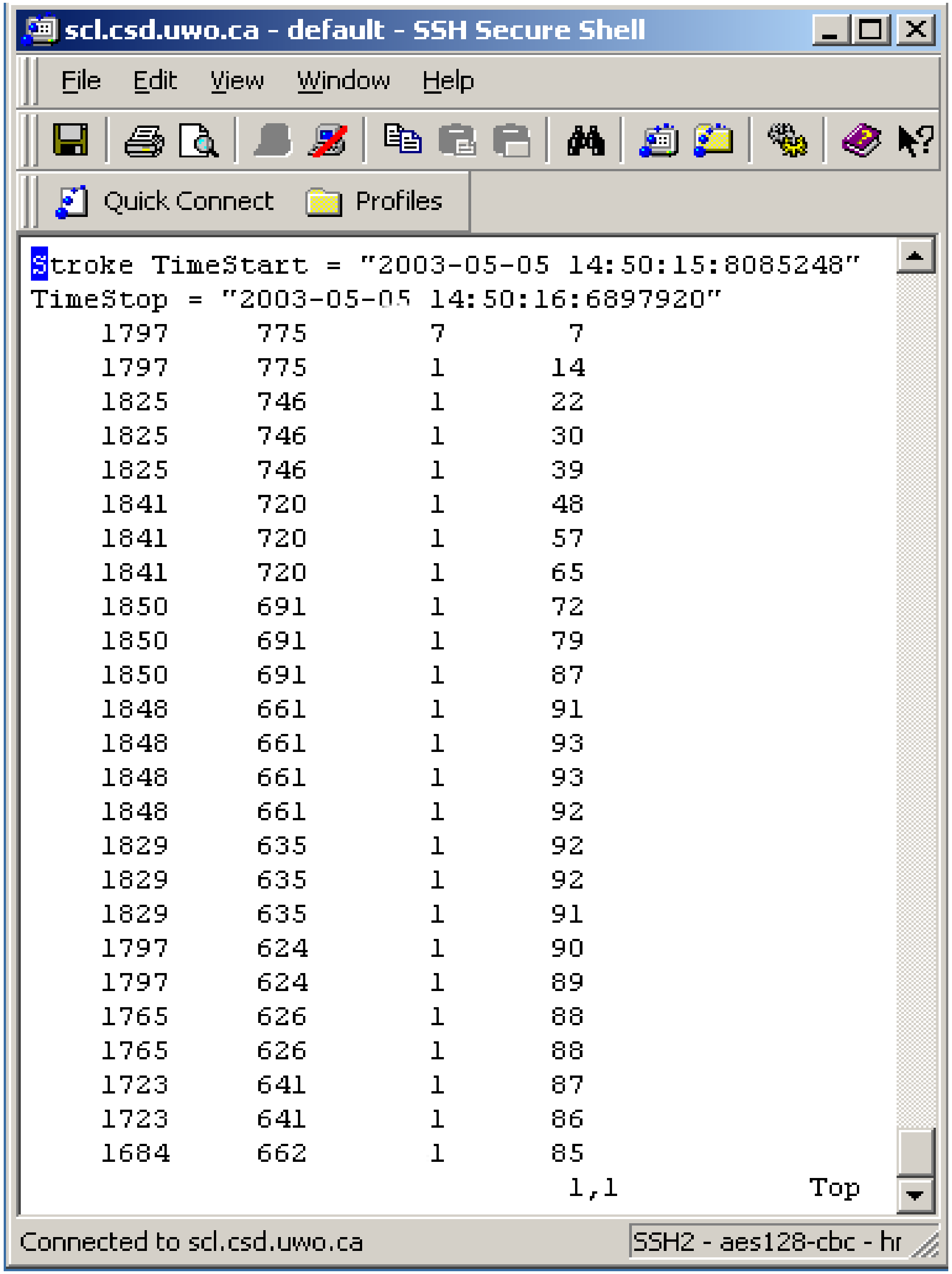
- Normalize input characters (size, slant, jitter)

- Detect specific features (direction, cusps, crossings,…)

- Elastic match within equivalence class of few entries

Data Collection

- Math survey
  - IBM Cross Pad Data
  - Tablet PC Data
- UniPen Data
- 240 symbols and a number of formulas.

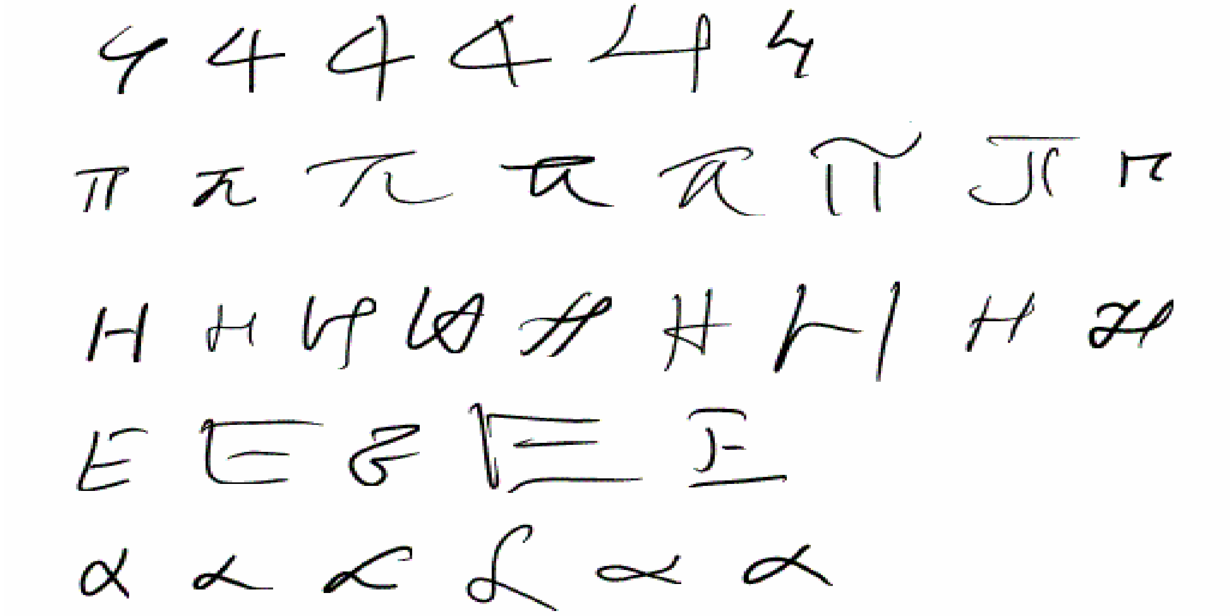


```
Stroke TimeStart = "2003-05-05 14:50:15:8085248"
TimeStop = "2003-05-05 14:50:16:6897920"
    1797     775       7       7
    1797     775       1      14
    1825     746       1      22
    1825     746       1      30
    1825     746       1      39
    1841     720       1      48
    1841     720       1      57
    1841     720       1      65
    1850     691       1      72
    1850     691       1      79
    1850     691       1      87
    1848     661       1      91
    1848     661       1      93
    1848     661       1      93
    1848     661       1      92
    1829     635       1      92
    1829     635       1      92
    1829     635       1      91
    1797     624       1      90
    1797     624       1      89
    1765     626       1      88
    1765     626       1      88
    1723     641       1      87
    1723     641       1      86
    1684     662       1      85
                               1,1           Top
```

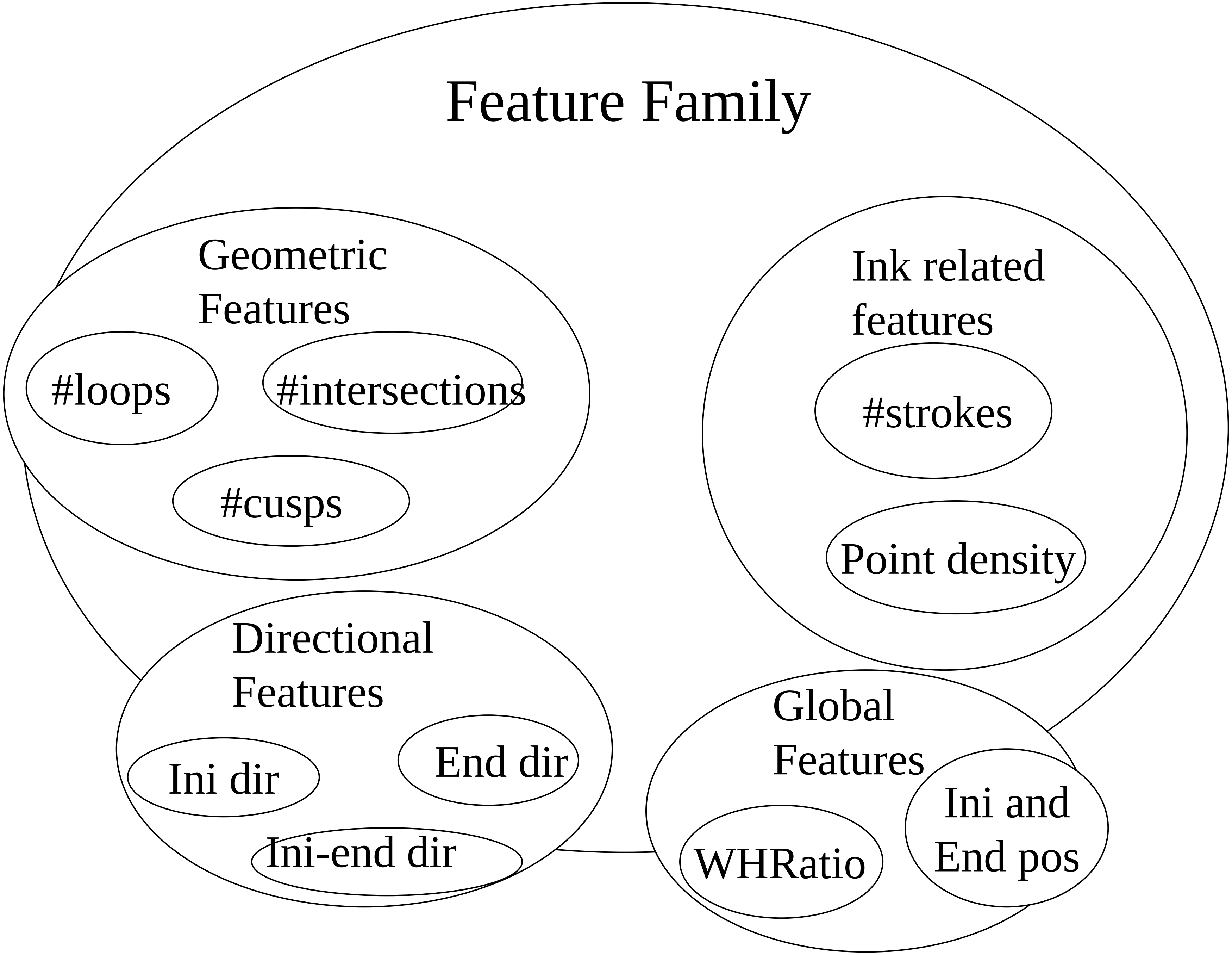
Connected to scl.csd.uwo.ca

SSH2 - aes128-cbc - hr

Variance Analysis

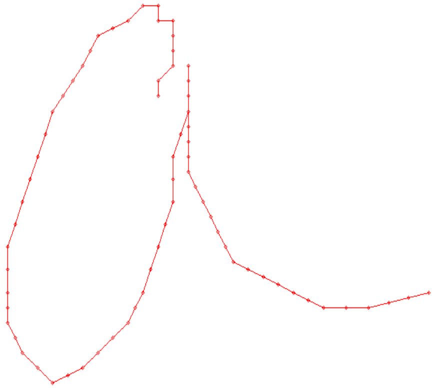


- Identify allomorphs θ θ ε ∈ x x

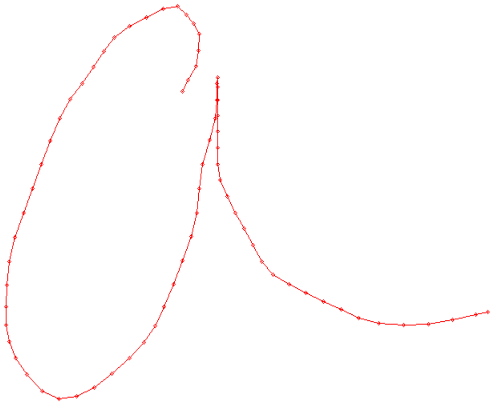

Feature Family
Geometric Features
#loops
#intersections
#cusps
Ink related features
#strokes
Point density
Directional Features
Ini dir
End dir
Ini-end dir
Global Features
WHRatio
Ini and End pos

Preprocessing

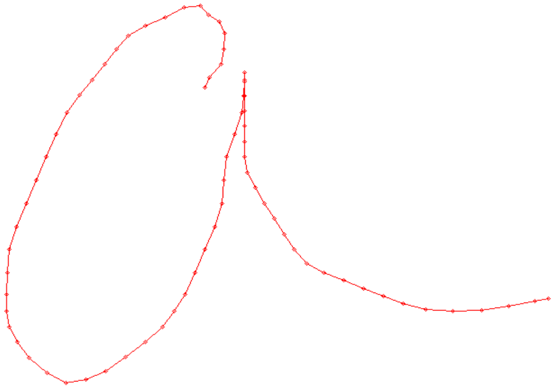
- Re-sample for device independence, writing speed, computation cost.
- Smoothing remove noise.
- Size normalization



Before Smoothing



Average Smoothing



Gaussian Smoothing

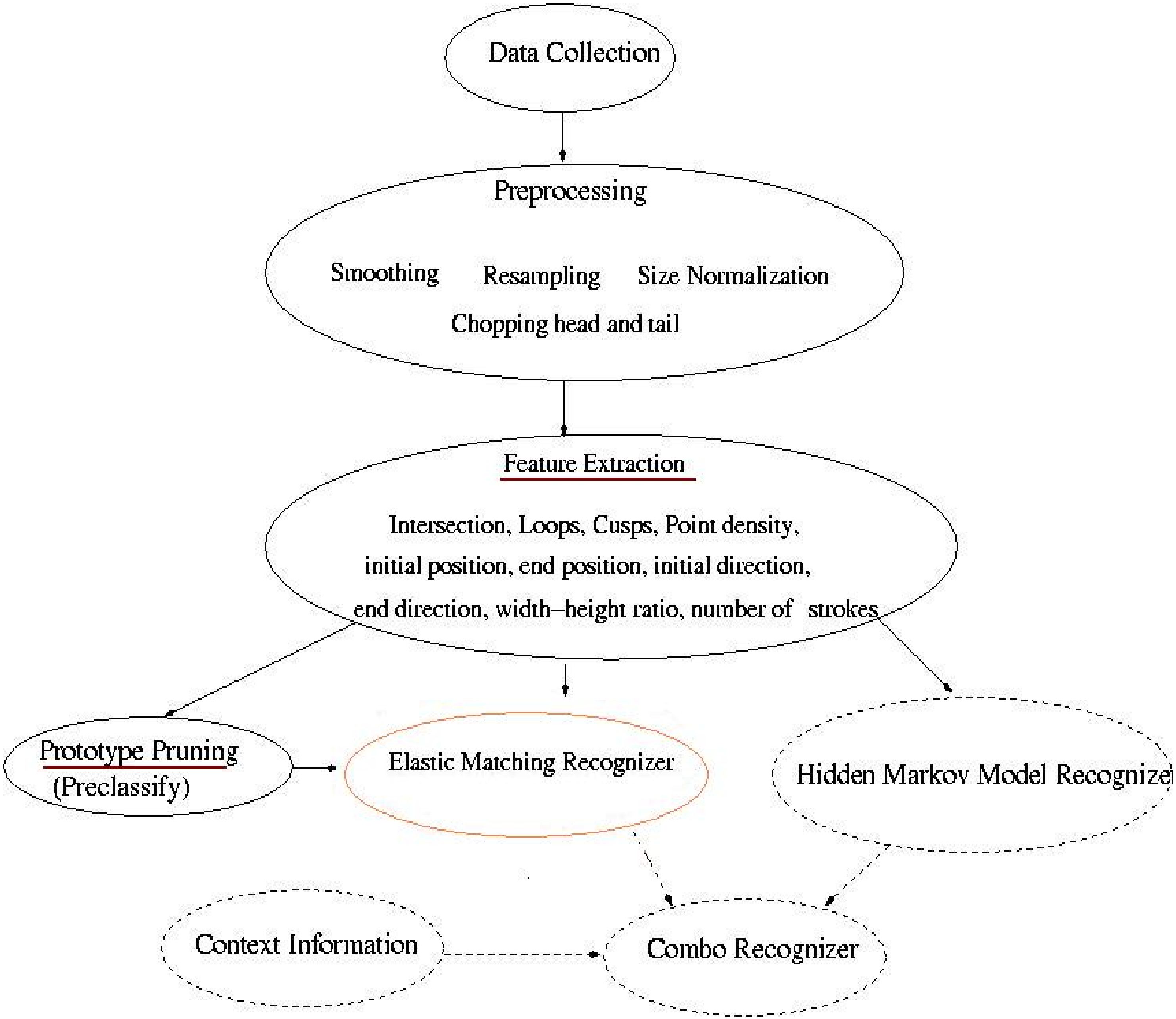
Feature Extraction

- Features split characters into equivalence classes

- Choose features on quality of separation vs cost

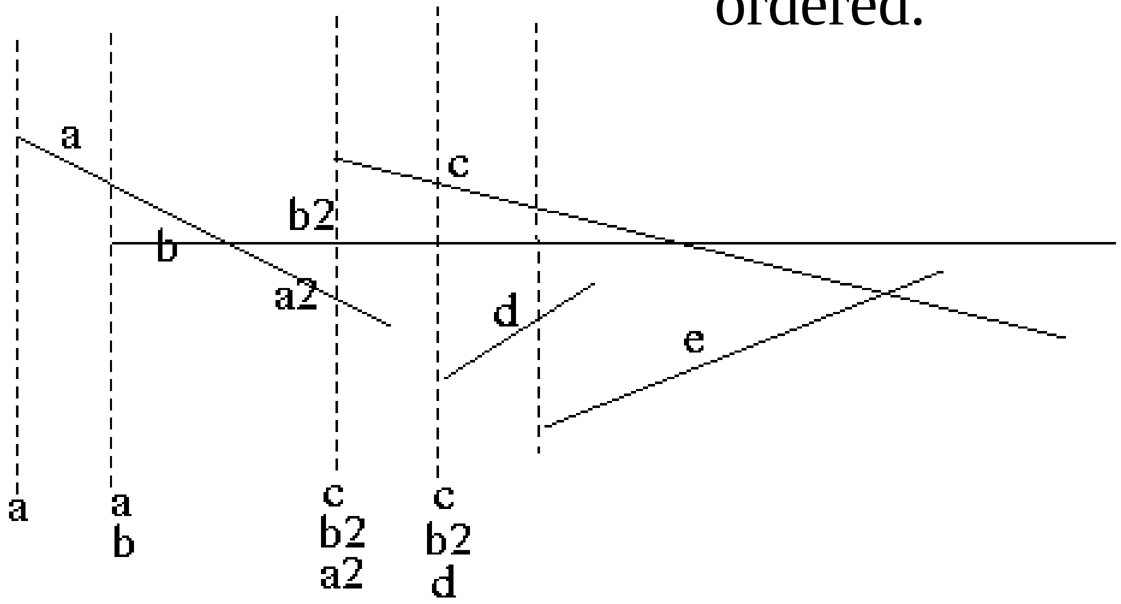
- Use to prune the set of character possibilities

- Use elastic matching on pruned prototype set


Data Collection
Preprocessing
Smoothing
Resampling
Size Normalization
Chopping head and tail
Feature Extraction
Intersection, Loops, Cusps, Point density,
initial position, end position, initial direction,
end direction, width–height ratio, number of strokes
Prototype Pruning
(Preclassify)
Elastic Matching Recognizer
Hidden Markov Model Recognize
Context Information
Combo Recognizer

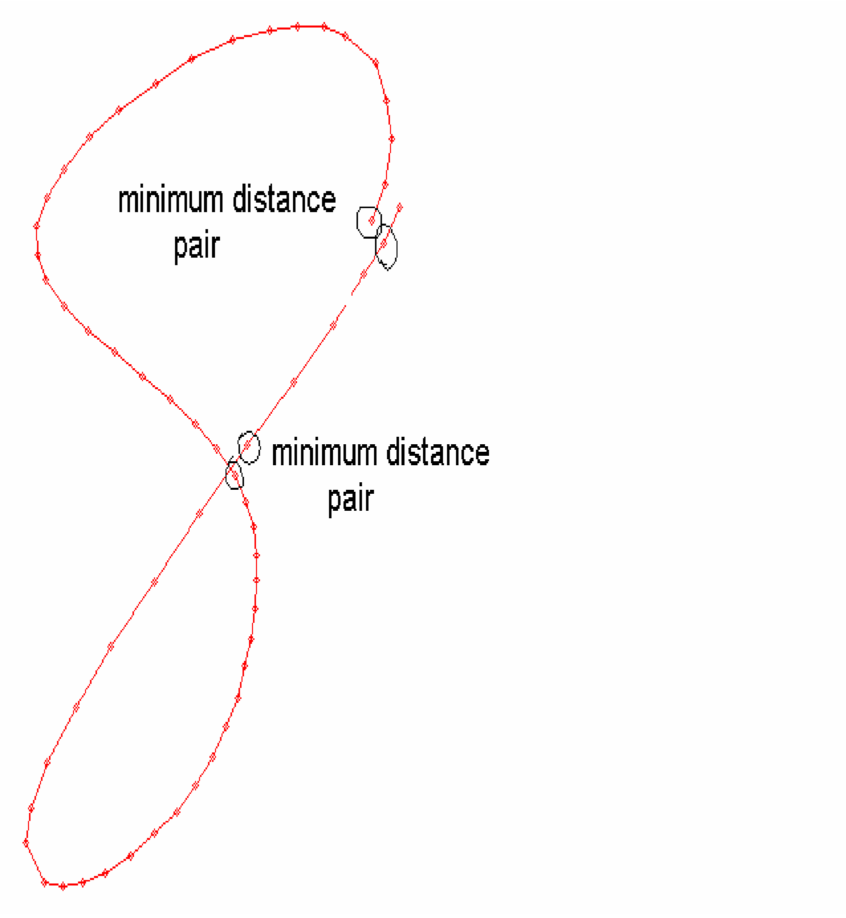
Intersections

Line segments are ordered.

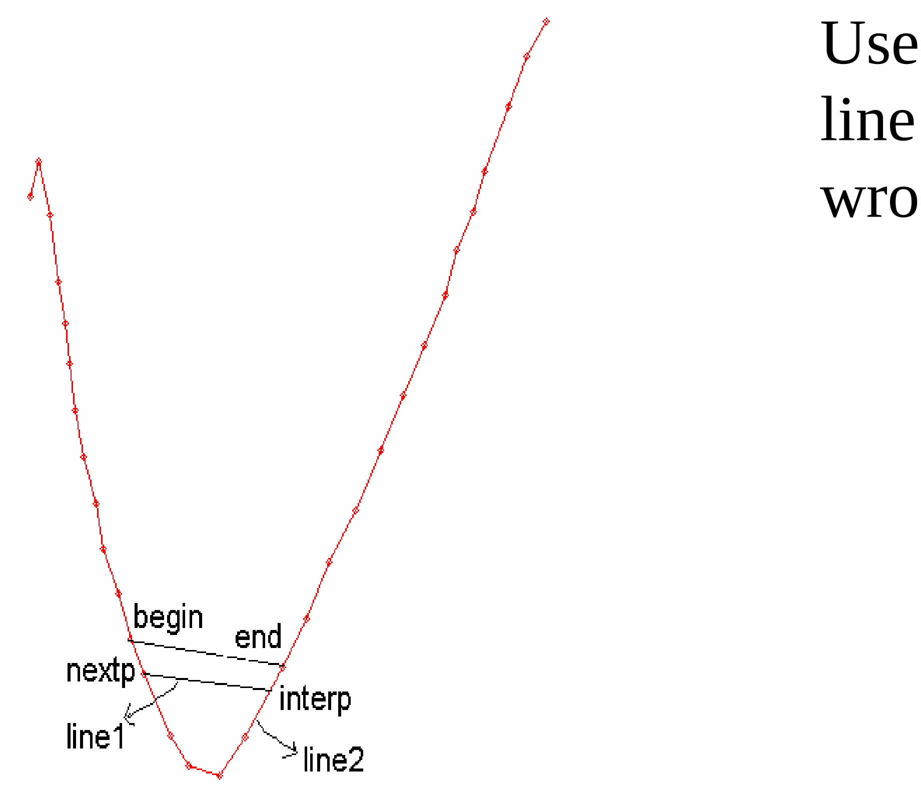


Modified Sweepline Algorithm

Loops

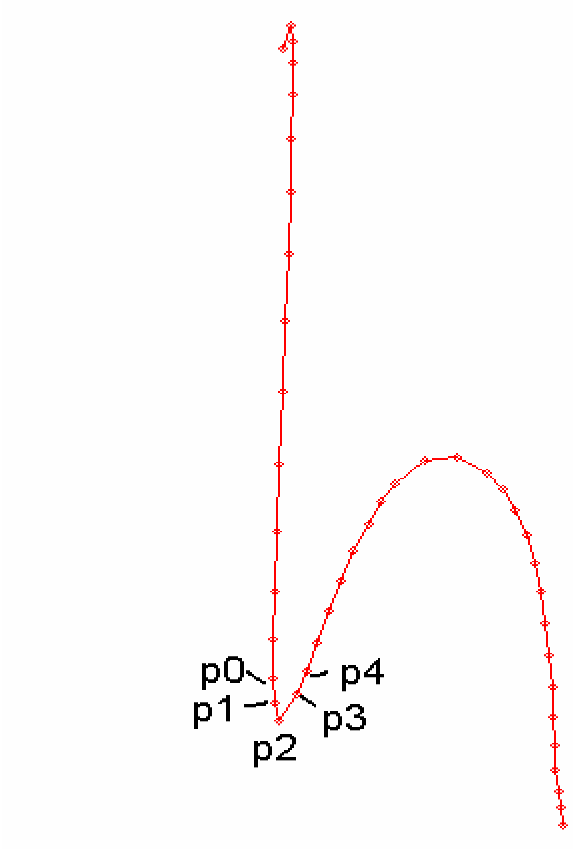


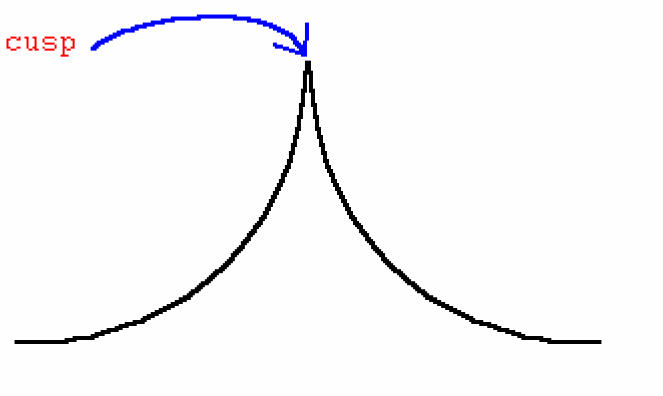
Minimum distance pair: a pair of points with minimum non-local distance

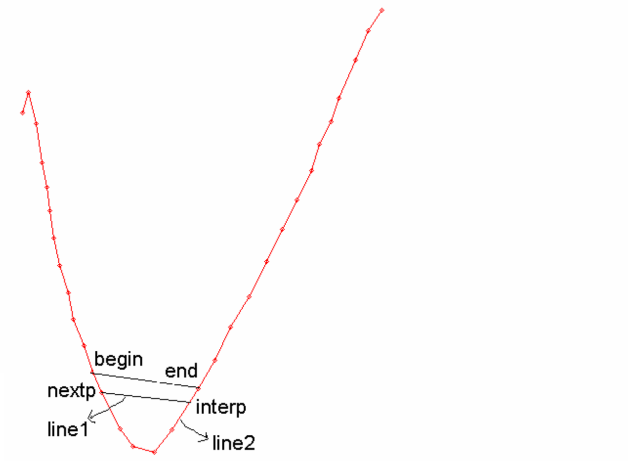


Use parallel line to filter the wrong loops

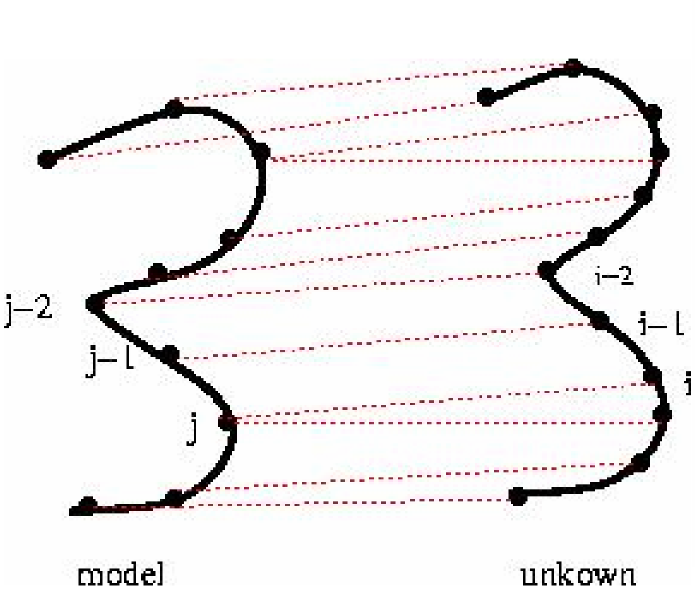
Cusps







Elastic Matching



$$D(i,j) = \delta(i,j) + \begin{cases} \sum_{k=0}^{j-1} \delta(0,k) & \text{if } i = 0 \\ \sum_{k=0}^{i-1} \delta(k,0) & \text{if } j = 0 \\ \min \begin{cases} D(i-1,j) \\ D(i-1,j-1) \end{cases} & \text{if } i > 0, j = 1 \\ \min \begin{cases} D(i-1,j-1) \\ D(i-1,j-2) \end{cases} & \text{if } i > 0, j > 1 \end{cases}$$

$$\delta(i,j) = (x_i - x_j)^2 + (y_i - y_j)^2 + C\,|\phi_i - \phi_j|$$

Without Features

Experiment	# prototypes	Recog.Rate(%)
P1:T1,2,3,4	227	81.8
P1,2:T1,2,3,4	454	90.1
P1,2,3:T1,2,3,4	681	93.9
P1,2,3,4:T1,2,3,4	908	94.8

With Features

Experiment	# prototypes	Candidate prototypes	Percent. Pruned	Recog. Rate(%)
P1:T1,2,3,4	227	26	88.5	76.0
P1,2:T1,2,3,4	366	38	89.6	85.5
P1,2,3:T1,2,3,4	495	52	89.5	90.0
P1,2,3,4:T1,2,3,4	575	60	89.6	91.9

Comparison

experiment	# prototype		Candidate prototypes		Percentage Pruned		Recog. Rate(%)	
	J.K's	Our	J.K's	Our	J.K's	Our	J.K's	Our
P1-4:T1-4	121	169	47	24	61.5	85.8	99.0	97.6
P1-4:T1-4	122	288	92	288	N/A	N/A	99.0	99.7

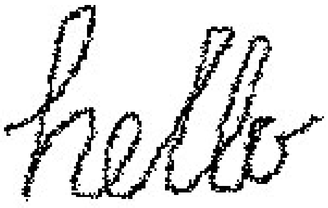
JK = J.Kurtzberg

Conclusion

- We have made progress in handwritten mathematical symbol recognition area by using feature sets to prune the prototypes.
- We have attempted to identify these features, and analyzed thousands of handwriting samples.
- Our recognizer can recognize digits, English letters, Greek letters, most of the common mathematical operators and notations.
- Accuracy and speed are improved comparing with a recognizer in the literature.

Dictionary-based methods

- Use word database to disambiguate.
- Database has “hello” but not “hdb” or “heUo”



- We can greatly restrict the set of symbols considered using knowledge of the mathematical context.

$$\sin(\omega\ t + k\ x) \quad vs \quad s\ i\ n(w\ t\ t\ k\ x)$$

Dictionary-based methods

- Build an *(h,k)* frequency table
- Collect all sub-expressions of height *h* and length *k*

- Replace level-1 sub-expressions by symbols and repeat

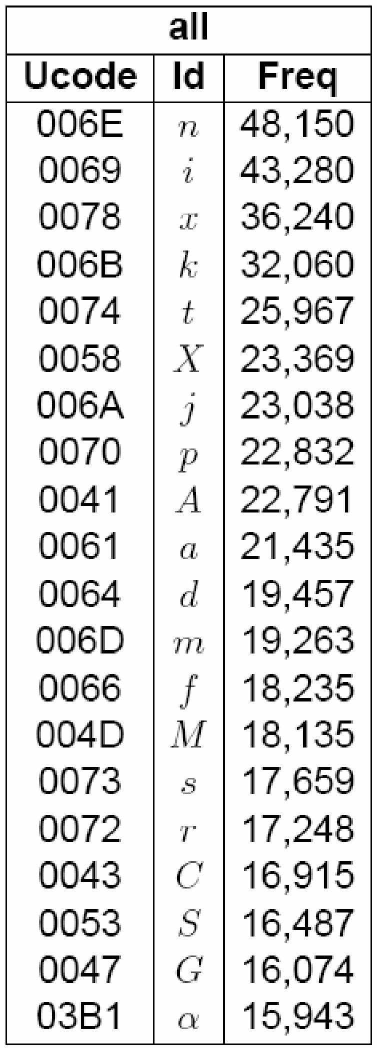
```
sqrt(sin(x)^2 + cos(y)^2)        sin(x)    cos(y)
                                 sin(x)^2  cos(y)^2
                                 sqrt(A^2 + B^2)
```

- Optionally group symbols by category, e.g.
  Greek *vs* Roman letters
  Letter ranges x, y, z
  Capital letters, …

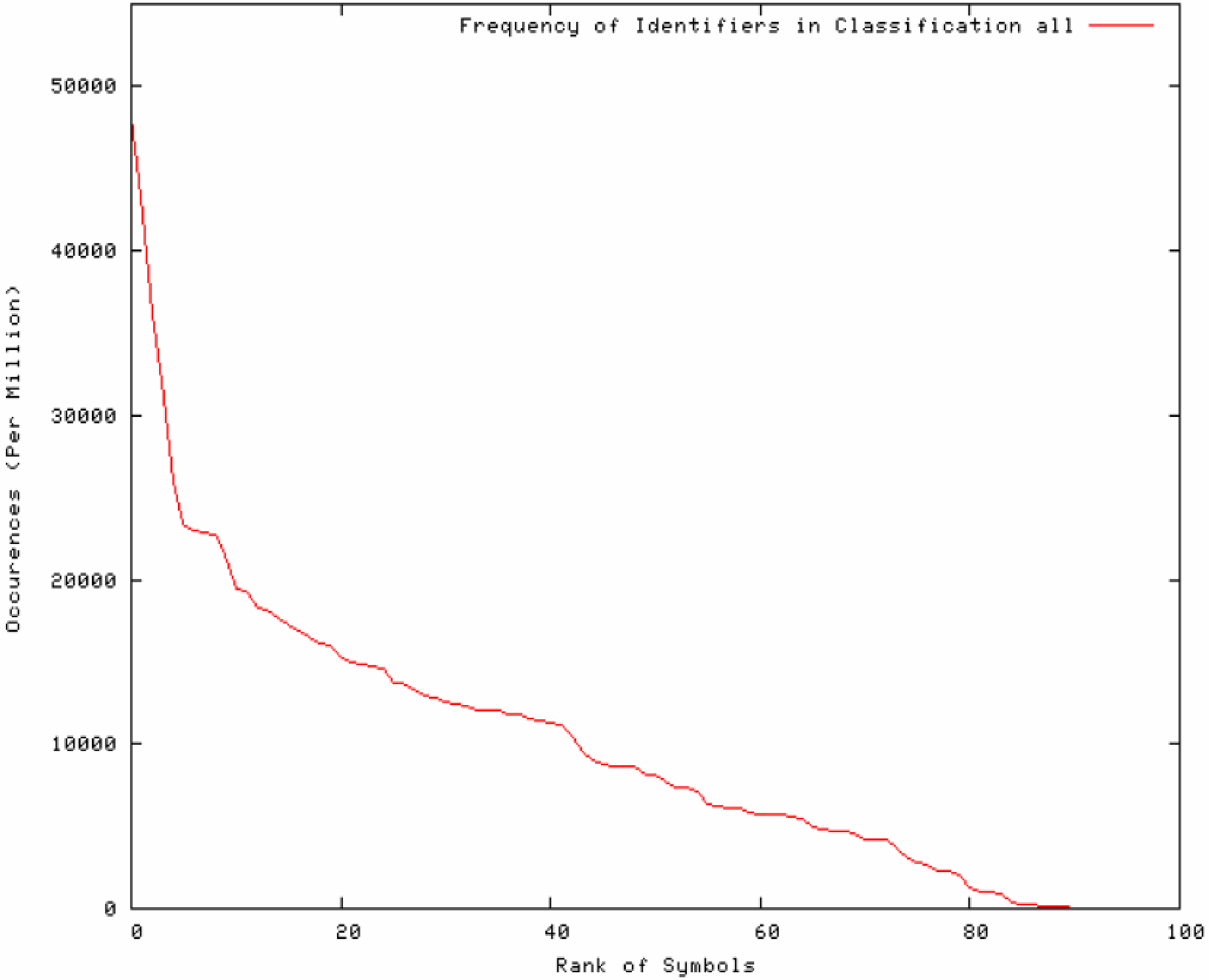
Dictionary-based methods

- Analyzed 20,000 articles from different MR classifications to build database of typical subexpressions
- Step 1: Develop profile of typical expressions by area
- Step 2: Identify user context by expressions used then disambiguate accordingly

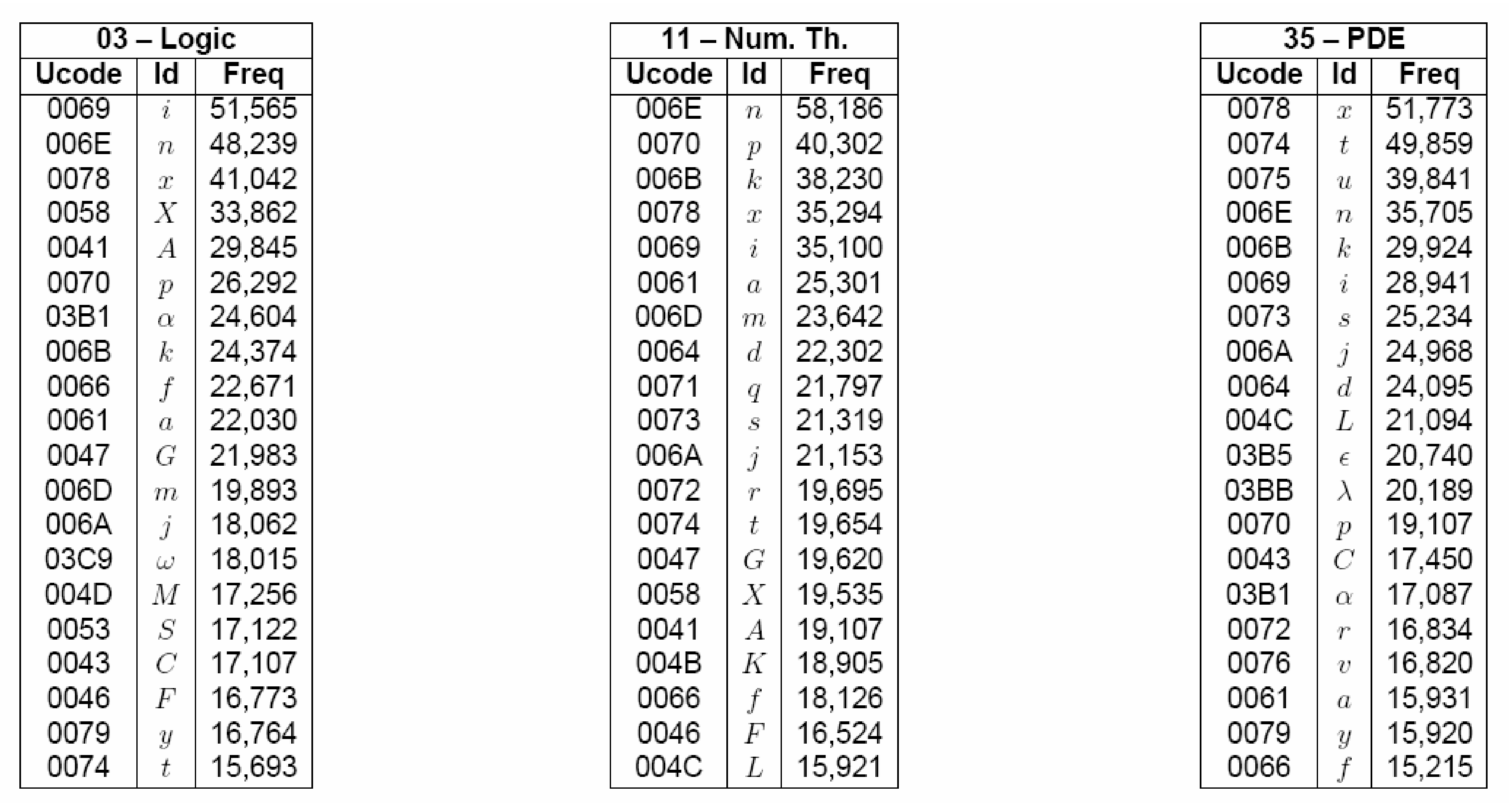
Frequency of Symbols



all		
Ucode	Id	Freq
006E	n	48,150
0069	i	43,280
0078	x	36,240
006B	k	32,060
0074	t	25,967
0058	X	23,369
006A	j	23,038
0070	p	22,832
0041	A	22,791
0061	a	21,435
0064	d	19,457
006D	m	19,263
0066	f	18,235
004D	M	18,135
0073	s	17,659
0072	r	17,248
0043	C	16,915
0053	S	16,487
0047	G	16,074
03B1	α	15,943



Id Frequencies in 3 Classifications

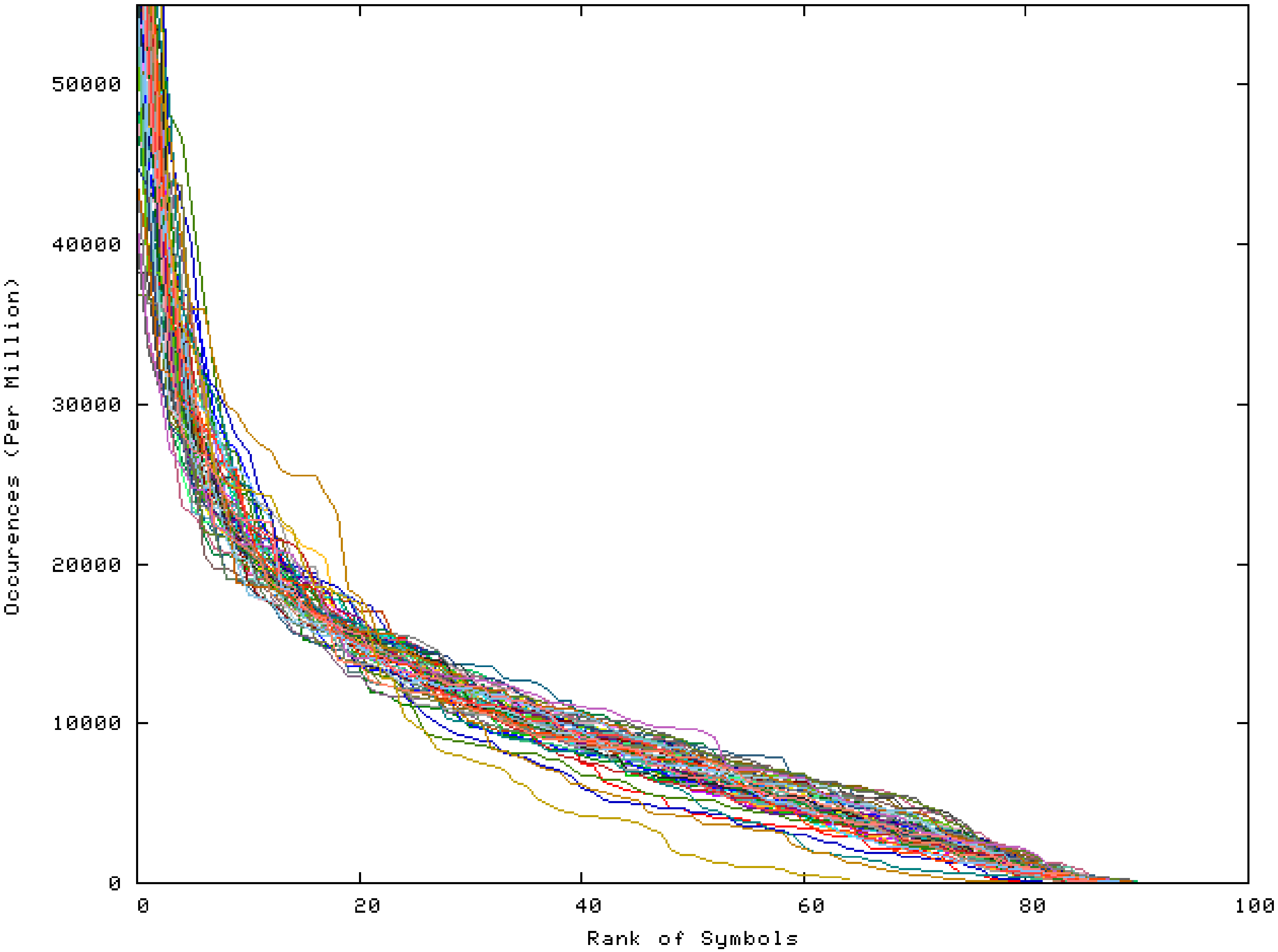


03 – Logic		
Ucode	Id	Freq
0069	i	51,565
006E	n	48,239
0078	x	41,042
0058	X	33,862
0041	A	29,845
0070	p	26,292
03B1	α	24,604
006B	k	24,374
0066	f	22,671
0061	a	22,030
0047	G	21,983
006D	m	19,893
006A	j	18,062
03C9	ω	18,015
004D	M	17,256
0053	S	17,122
0043	C	17,107
0046	F	16,773
0079	y	16,764
0074	t	15,693

11 – Num. Th.		
Ucode	Id	Freq
006E	n	58,186
0070	p	40,302
006B	k	38,230
0078	x	35,294
0069	i	35,100
0061	a	25,301
006D	m	23,642
0064	d	22,302
0071	q	21,797
0073	s	21,319
006A	j	21,153
0072	r	19,695
0074	t	19,654
0047	G	19,620
0058	X	19,535
0041	A	19,107
004B	K	18,905
0066	f	18,126
0046	F	16,524
004C	L	15,921

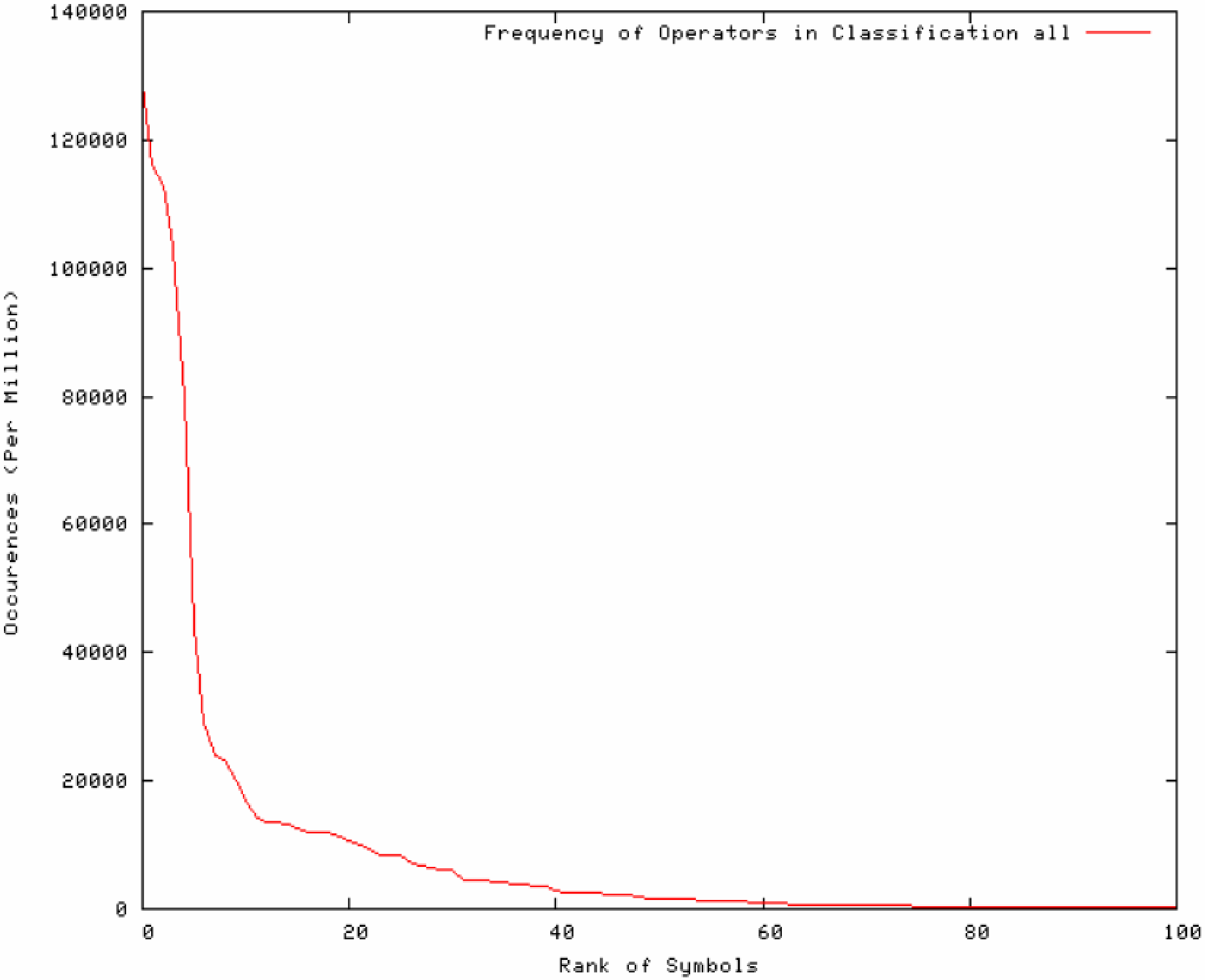
35 – PDE		
Ucode	Id	Freq
0078	x	51,773
0074	t	49,859
0075	u	39,841
006E	n	35,705
006B	k	29,924
0069	i	28,941
0073	s	25,234
006A	j	24,968
0064	d	24,095
004C	L	21,094
03B5	ϵ	20,740
03BB	λ	20,189
0070	p	19,107
0043	C	17,450
03B1	α	17,087
0072	r	16,834
0076	v	16,820
0061	a	15,931
0079	y	15,920
0066	f	15,215

Id Freq from All Classifications



Frequency of Operators

all		
Ucode	Op	Freq
003D	=	128,715
002D	−	116,064
002C	,	112,818
2061		103,090
002B	+	79,404
2208	∋	43,942
002A	∗	29,210
2192	→	23,818
002F	/	23,405
2264	≤	20,088
02DC	˜	16,875
2297	⊗	14,242
2211	∑	13,560
003E	>	13,528
221E	∞	13,138
00AF	¯	12,451
003C	<	12,058
22EF	⋯	12,005
2202	∂	11,940
00D7	×	11,294



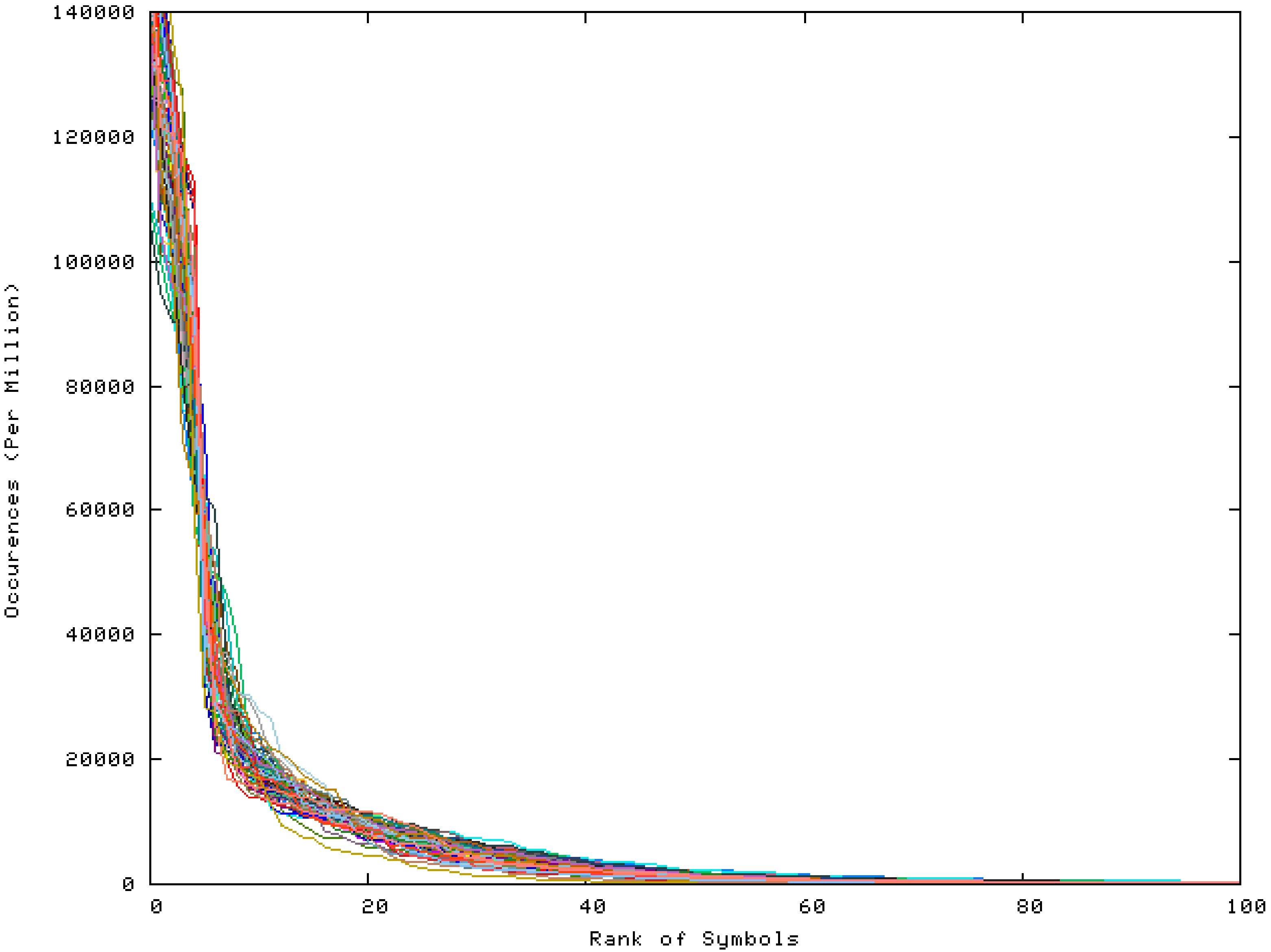
Op Frequencies in 3 Classifications

03 – Logic		
Ucode	Op	Freq
003D	=	121,806
2061		115,262
002C	,	100,880
2208	∋	77,021
002D	−	60,732
002B	+	60,121
002A	∗	32,796
003C	<	28,345
02C9	ˉ	25,805
2192	→	24,370
2264	≤	24,242
002F	/	14,626
2026	…	13,495
222A	∪	12,654
2229	∩	12,483
2286	⊆	12,330
003E	>	11,784
2223	∣	9,883
22EF	⋯	9,781
02DC	˜	9,428

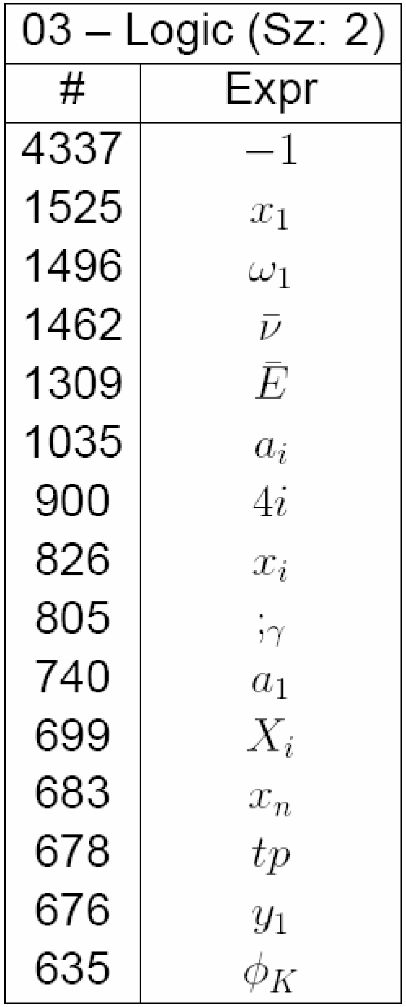
11 – Num. Th.		
Ucode	Op	Freq
003D	=	130,735
002D	−	128,330
2061		112,484
002C	,	104,964
002B	+	94,172
002F	/	40,239
2208	∋	39,319
2211	∑	20,165
2264	≤	19,574
2192	→	18,481
002A	∗	17,757
00AF	¯	14,708
221E	∞	14,627
003E	>	12,926
22EF	⋯	12,358
02DC	˜	12,209
2265	≥	11,963
2113	ℓ	10,997
003C	<	10,151
00D7	×	10,144

35 – PDE		
Ucode	Op	Freq
002D	−	138,603
002C	,	111,176
2061		103,527
003D	=	103,376
002B	+	97,579
2208	∋	38,370
2264	≤	34,575
2202	∂	28,815
002F	/	25,985
221E	∞	23,460
222B	∫	23,196
02DC	˜	19,545
003C	<	16,453
2207	∇	15,387
003E	>	15,256
002A	∗	14,470
2192	→	14,381
22C5	⋅	12,669
2211	∑	12,394
2265	≥	11,531

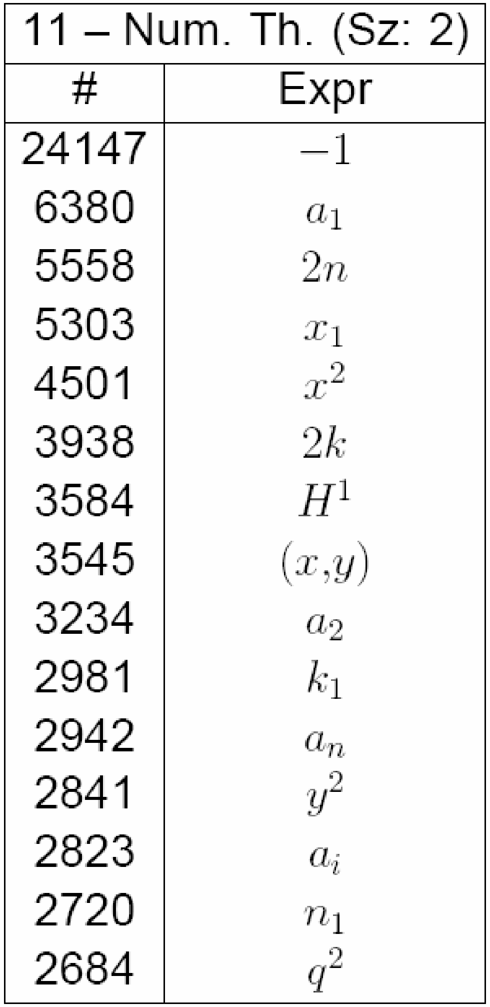
Op Freq from All Classifications



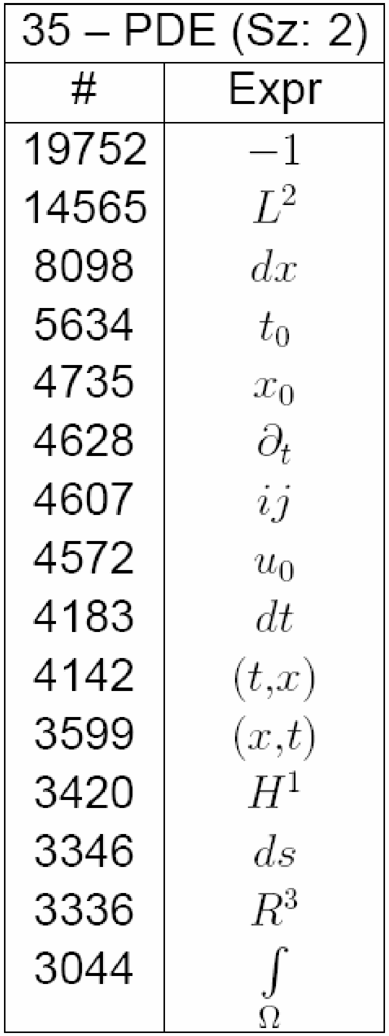
Most Popular Expressions of Size 2



03 – Logic (Sz: 2)	
#	Expr
4337	-1
1525	x_1
1496	ω_1
1462	$\bar{\nu}$
1309	$\bar{E}$
1035	a_i
900	$4i$
826	x_i
805	$;\gamma$
740	a_1
699	X_i
683	x_n
678	tp
676	y_1
635	ϕ_K

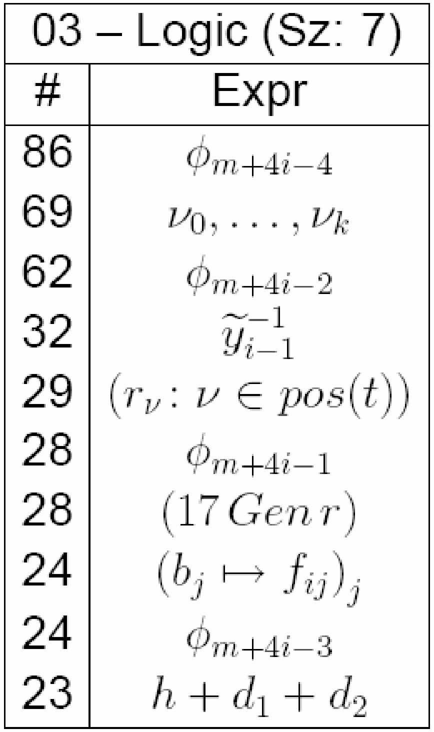


11 – Num. Th. (Sz: 2)	
#	Expr
24147	-1
6380	a_1
5558	$2n$
5303	x_1
4501	x^2
3938	$2k$
3584	H^1
3545	(x,y)
3234	a_2
2981	k_1
2942	a_n
2841	y^2
2823	a_i
2720	n_1
2684	q^2

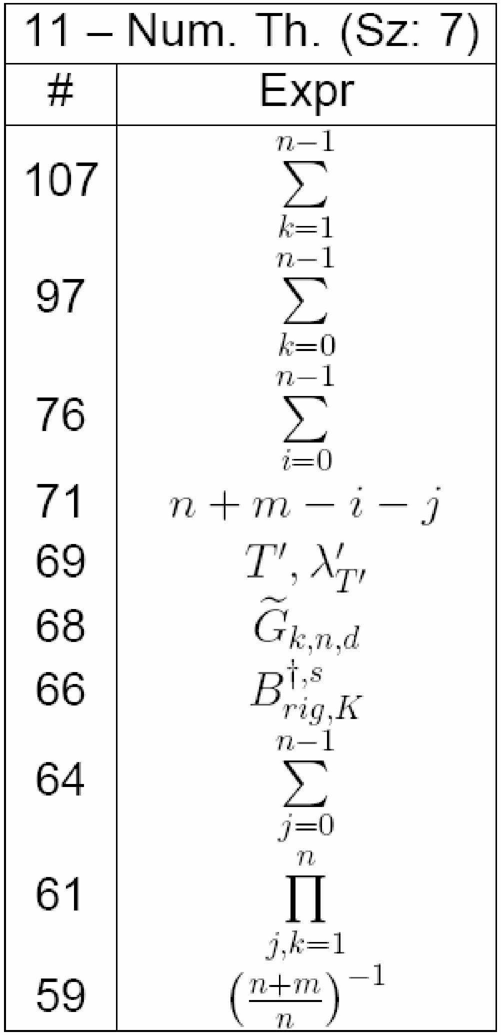


35 – PDE (Sz: 2)	
#	Expr
19752	-1
14565	L^2
8098	dx
5634	t_0
4735	x_0
4628	∂_t
4607	ij
4572	u_0
4183	dt
4142	(t,x)
3599	(x,t)
3420	H^1
3346	ds
3336	R^3
3044	$\int_\Omega$

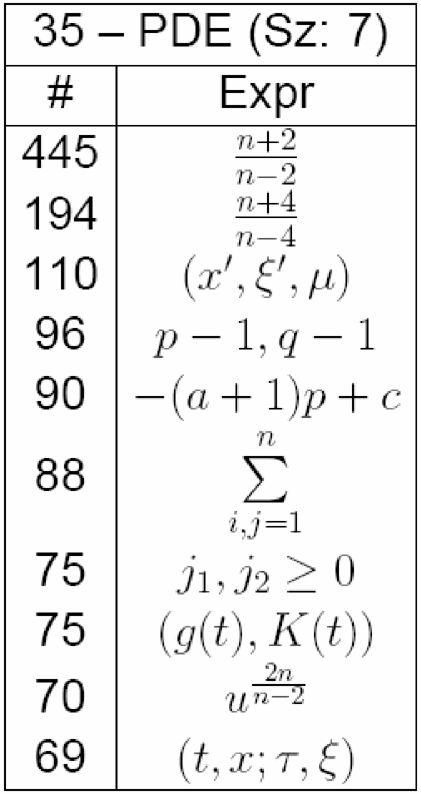
Most Popular Expressions of Size 7



03 – Logic (Sz: 7)	
#	Expr
86	ϕ_{m+4i-4}
69	$\nu_0, \ldots, \nu_k$
62	ϕ_{m+4i-2}
32	$\widetilde{y}_{i-1}^{-1}$
29	$(r_\nu : \nu \in pos(t))$
28	ϕ_{m+4i-1}
28	$(17\, Gen\, r)$
24	$(b_j \mapsto f_{ij})_j$
24	ϕ_{m+4i-3}
23	$h + d_1 + d_2$



11 – Num. Th. (Sz: 7)	
#	Expr
107	$\sum_{k=1}^{n-1}$
97	$\sum_{k=0}^{n-1}$
76	$\sum_{i=0}^{n-1}$
71	$n + m - i - j$
69	$T', \lambda'_{T'}$
68	$\widetilde{G}_{k,n,d}$
66	$B_{rig,K}^{\dagger,s}$
64	$\sum_{j=0}^{n-1}$
61	$\prod_{j,k=1}^{n}$
59	$\binom{n+m}{n}^{-1}$



35 – PDE (Sz: 7)	
#	Expr
445	$\frac{n+2}{n-2}$
194	$\frac{n+4}{n-4}$
110	(x', ξ', μ)
96	$p-1, q-1$
90	$-(a+1)p + c$
88	$\sum_{i,j=1}^{n}$
75	$j_1, j_2 \geq 0$
75	$(g(t), K(t))$
70	$u^{\frac{2n}{n-2}}$
69	$(t, x; \tau, \xi)$

Expression Analysis and Transformation

- Understanding expression arrangement and re-arrangement

$ax + by +$
$cz + wt$

$$\begin{bmatrix} ax & by \\ cz & wt \end{bmatrix}$$

$a\,x = b\,y$
$\quad = c\,z - wt$

$\mathrm{u}_2\mathrm{v}_1(\mathrm{a+b+c+z})$ $\mathrm{u}_2\mathrm{F}_1(\mathrm{a, b, c; z})$

$(x+y)^2$

Expression Transformation

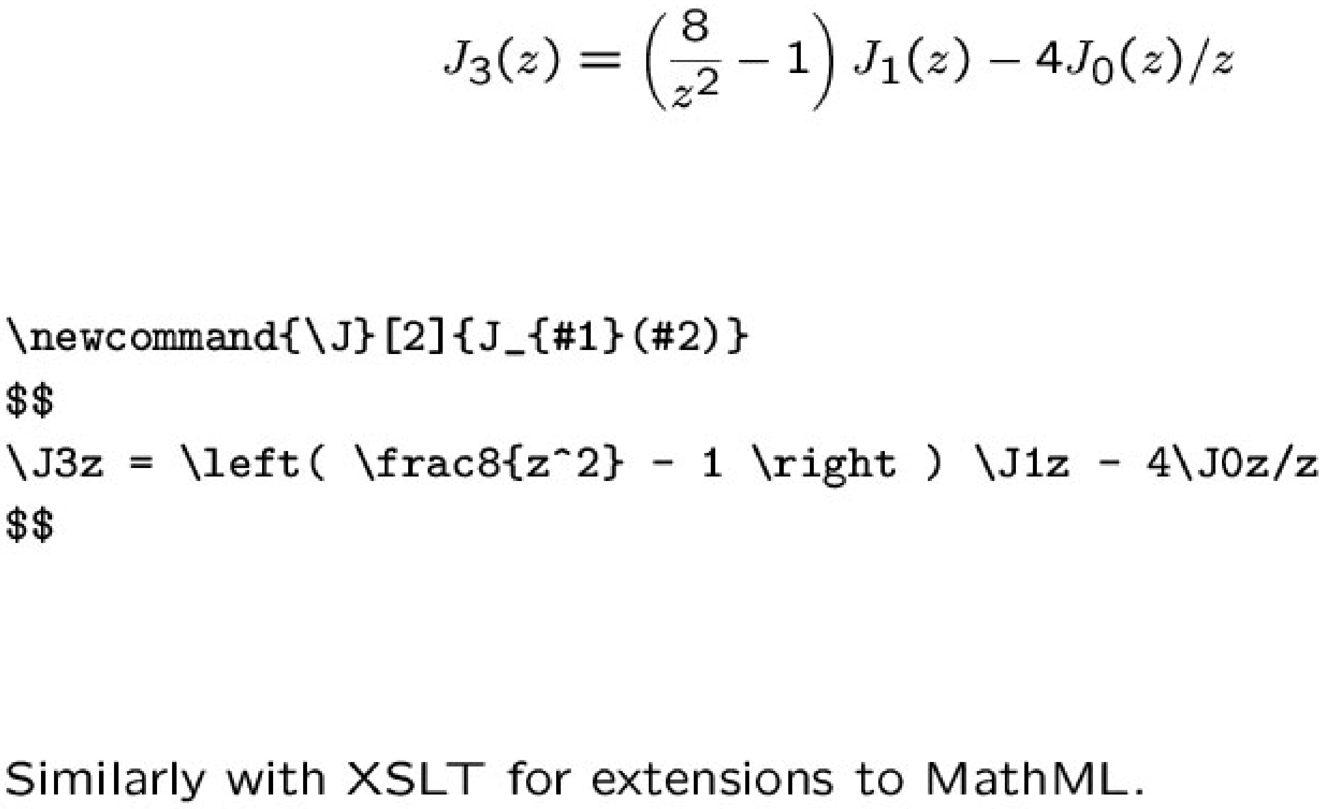
- TeX ↔ MathML ↔ OpenMath ↔ Maple

- Naïve approach to TeX → MathML translation:
  * Macro expansion:
  TeX → Low-level TeX
  * Translate:
  Low-level TeX → Low-level Presentation MathML

- Resulting MathML has correct visual structure,
  but has lost all the implicit semantics

Implicit Semantics

Conversion must know about macros



$$J_3(z) = \left(\frac{8}{z^2} - 1\right) J_1(z) - 4J_0(z)/z$$

```
\newcommand{\J}[2]{J_{#1}(#2)}
$$
\J3z = \left( \frac8{z^2} - 1 \right ) \J1z - 4\J0z/z
$$
```

Similarly with XSLT for extensions to MathML.

Less Naïve Approach

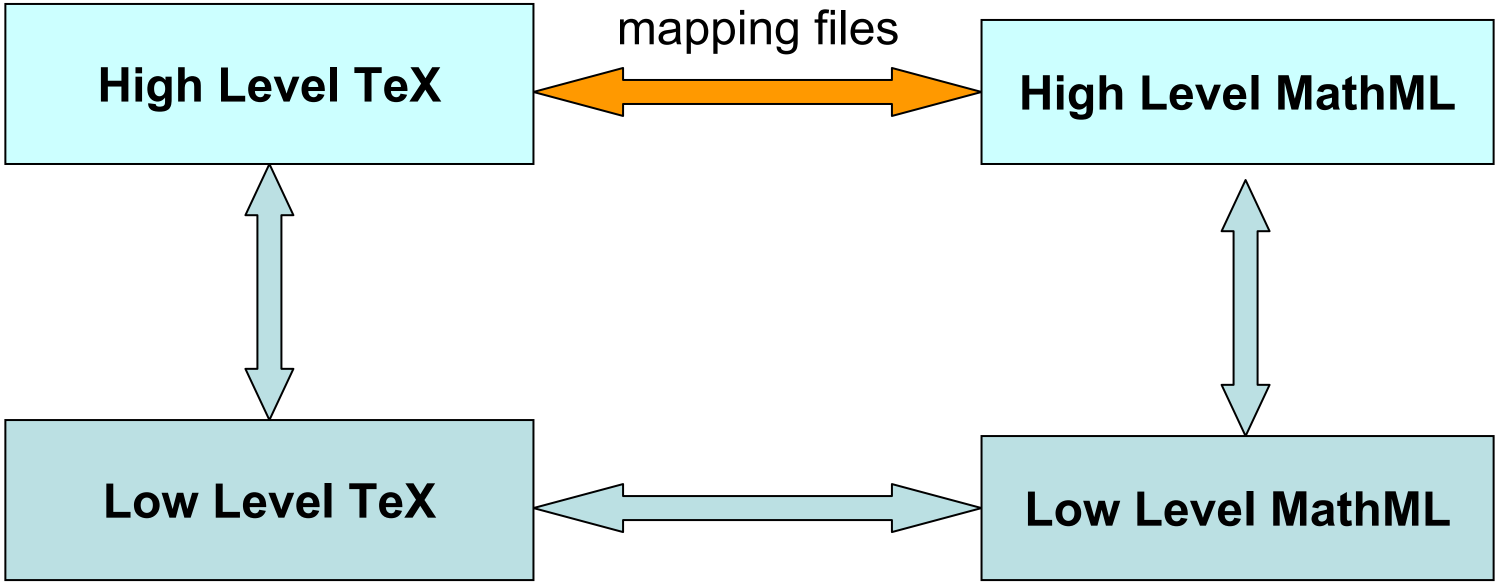
- Mapping file: associates TeX macros with XSLT templates,

  e.g. `\J{u}{z}` ↔ `<apply> <xmml:J/> u z </apply>`

- Converter uses mapping file rules to short-circuit detailed translation

- Mapping file can insert additional explicit semantics, e.g. OpenMath

TeX/MathML Conversion



High Level TeX

mapping files

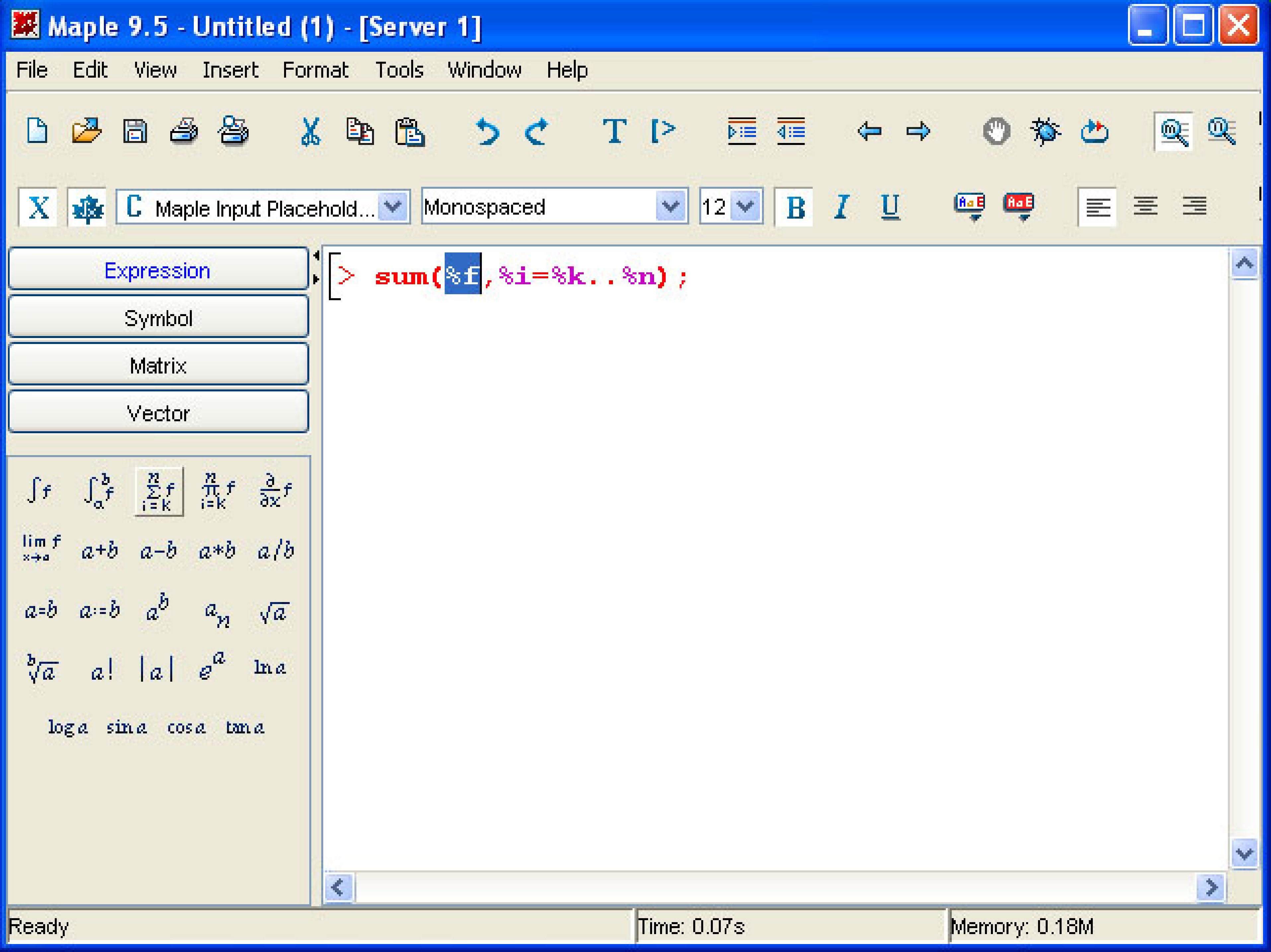
High Level MathML

Low Level TeX

Low Level MathML

Cheating

Do you want to wash those dishes
or do you just want to get them clean?


Maple 9.5 - Untitled (1) - [Server 1]
File
Edit
View
Insert
Format
Tools
Window
Help
Maple Input Placehold...
Monospaced
12
Expression
Symbol
Matrix
Vector
Ready
Time: 0.07s
Memory: 0.18M

```
> sum(%f,%i=%k..%n);
```

Bootstrapping techniques

- Initially deal with a limited range of possibilities, requiring palette selection of others

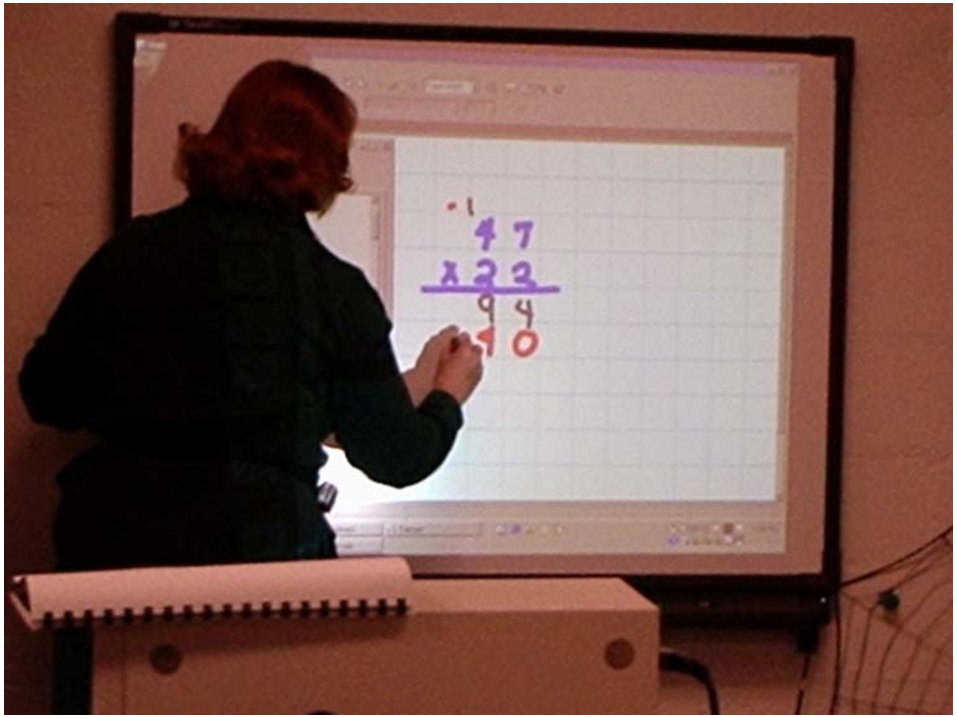
- Prompting areas: $x^2y^{\blacksquare} \;\; \blacksquare$

- Build in explicit knowledge of some domains

- Disallow certain selections

```
      a1|    a2|    a3|
a0 + ---- + ---- + ---- + . . .
     |b1    |b2    |b3
```

Portability



Goals

- Platform Portability
  - *Across platforms* and applications
  - *Over time* for evolving platforms and applications
- Digital Ink Portability
  - can be achieved with InkML
  - Wrappers for device-specific ink interfaces
- Mathematical Data Portability
  - OpenMath
  - MathML

Our Architectural Approach

Invariant Components with Replaceable Glue

* Parts remaining invariant:

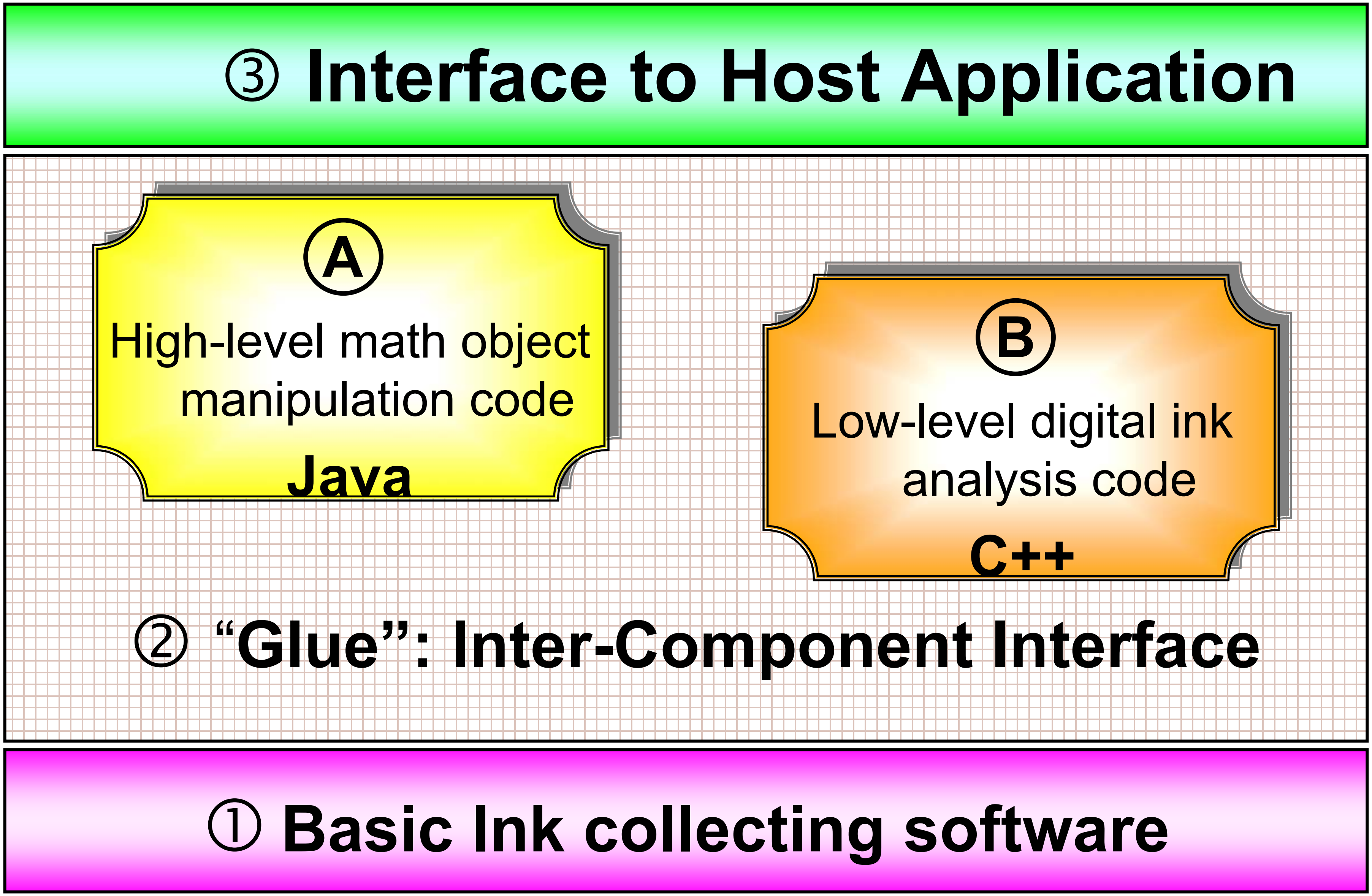
A. High-level math object manipulation code

B. Low-level digital ink analysis code

* Parts depending on host system:

1. Basic ink collecting software:
   supports abstract ink representation
2. “Glue” – Inter-component interface:
   links (A) and (B) with (1) and (3)
3. Interface code:
   embeds pen-based math input in host application

Framework Components



③ Interface to Host Application

Ⓐ

High-level math object manipulation code

Java

Ⓑ

Low-level digital ink analysis code

C++

② "Glue": Inter-Component Interface

① Basic Ink collecting software

Implementation Languages

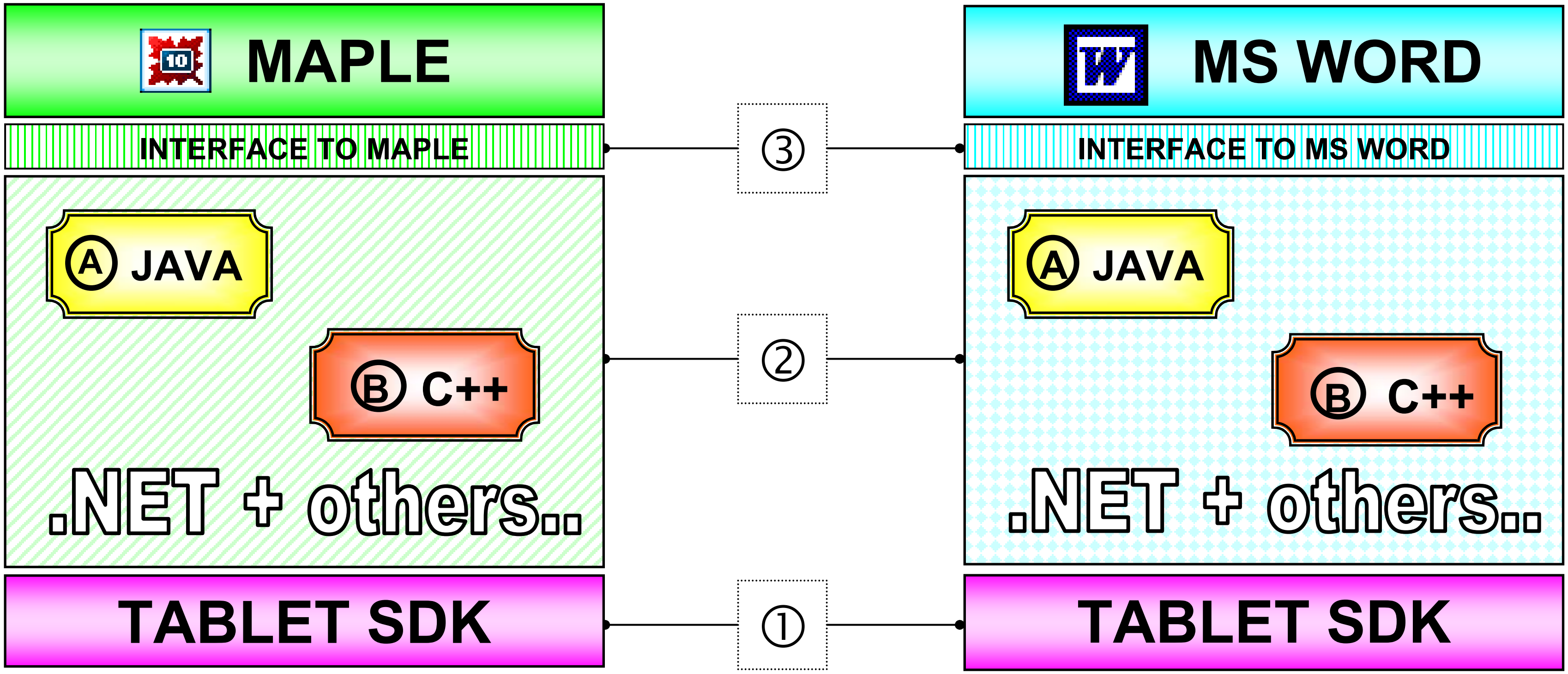
- **C#**
  - **assignment**: ink collecting and processing,
  - **example of use**: connecting to Tablet SDK
- **C++**
  - **assignment**: low-level intensive computations
  - **example of use**: character recognizer, glyph feature determiner
- **Java**
  - **assignment**: high-level code for connecting with mathematical engine
  - **example of use**: math expression manipulation

Instantiating the Architecture

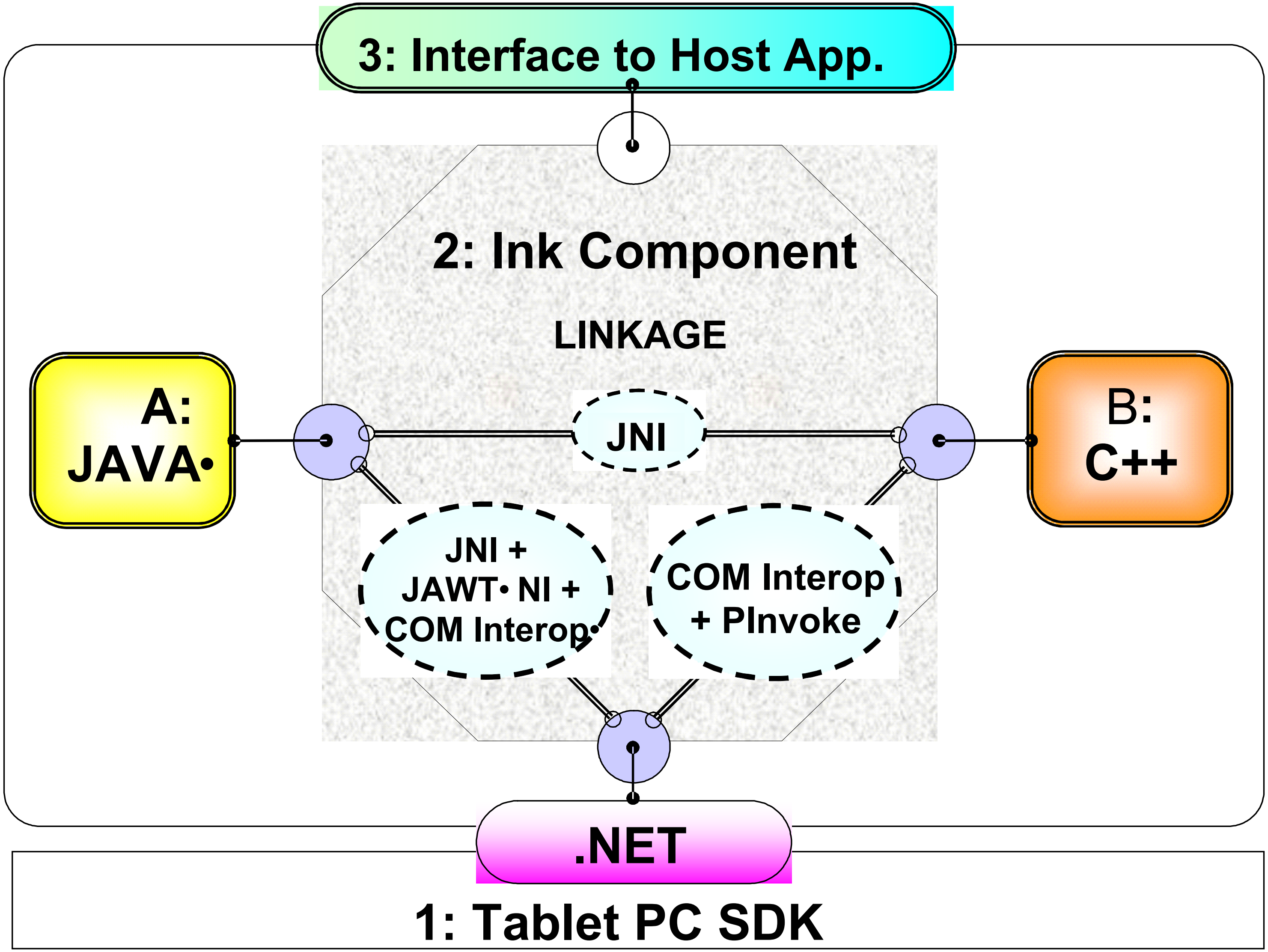
Instantiation for Tablet PC:

- ① For **basic ink software**
  we used .NET-based Tablet PC SDK
- ② Specially designed **linkage mechanism**
  - a number of .NET technologies (C#, managed C++),
  - COM interoperability features and
  - Java Native Interface (as described further)
- ③ **Interface to the hosting application**
  vary depending on the application

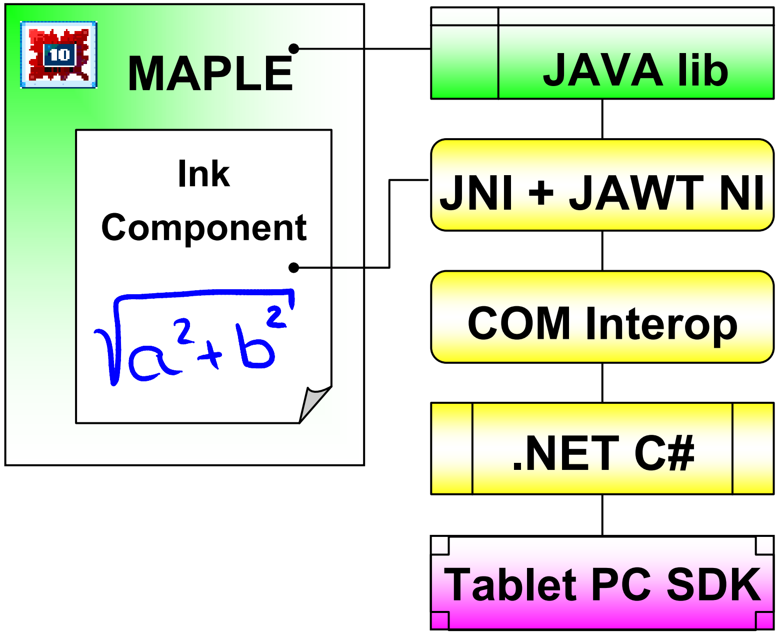
Testing the Framework



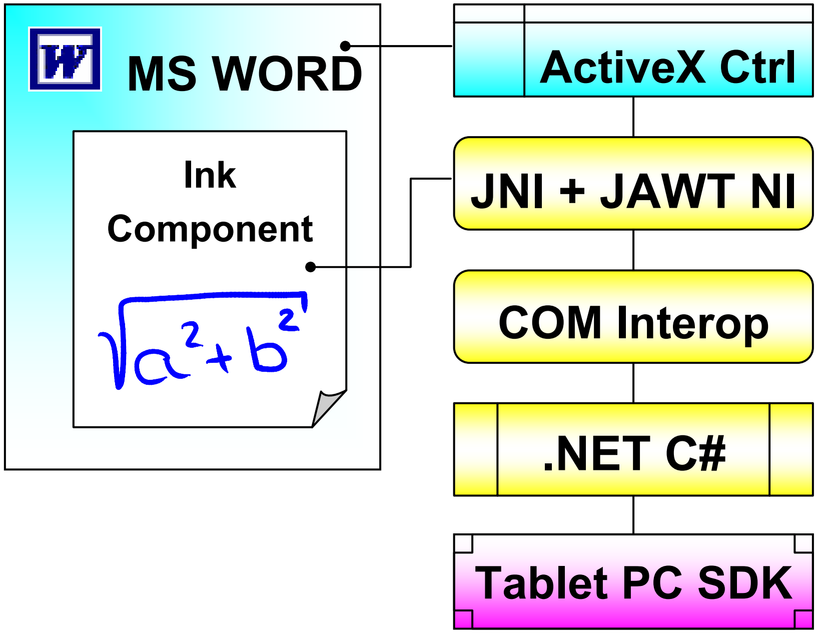
Linkage for the Test Framework



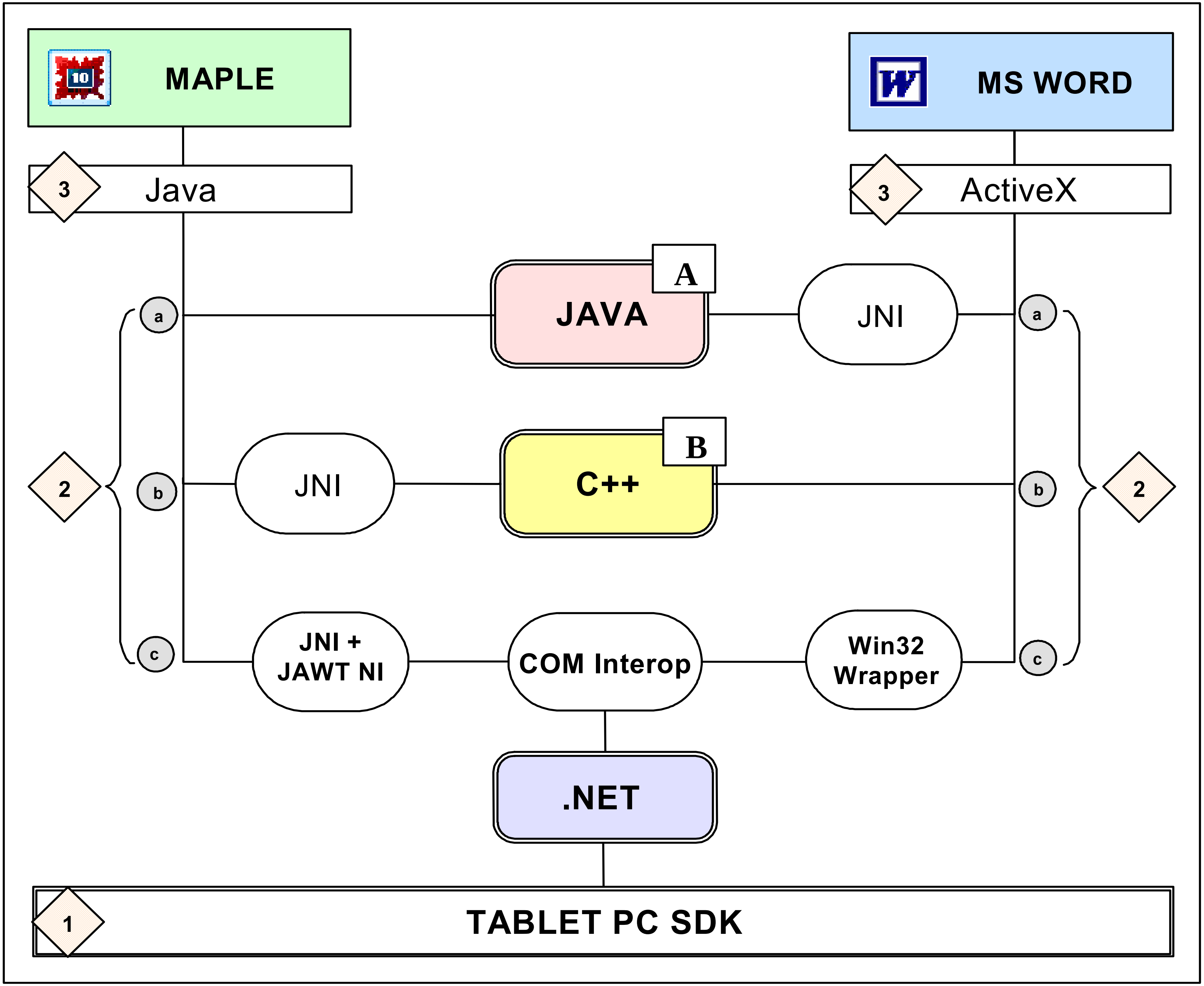
Interface to Host Application



- Java library
- accessing .NET control
- through JNI



- ActiveX control
- accessing .NET control
- via Win32 C++ Wrapper

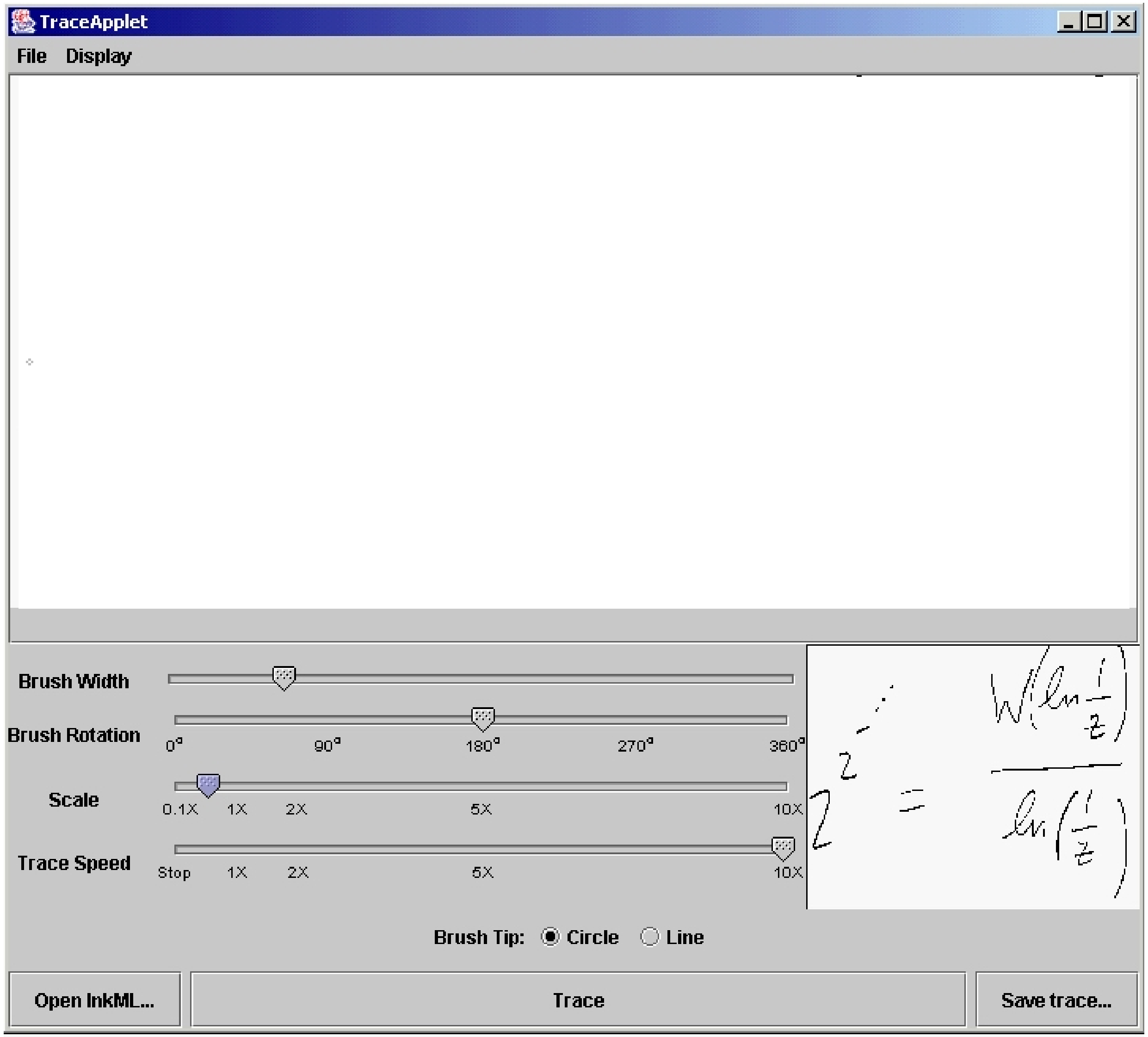

MAPLE
MS WORD
3
Java
3
ActiveX
A
JAVA
JNI
a
a
2
b
JNI
B
C++
b
2
c
JNI +
JAWT NI
COM Interop
Win32
Wrapper
c
.NET
1
TABLET PC SDK

InkML

Forthcoming standard for digital ink from W3C

```
<?xml version="1.0"?>
<ink>
    <defs>
        <traceFormat id="MSTabletInk">
            <regularChannels>
                <channel name="X" type="integer"/>
                <channel name="Y" type="integer"/>
                <channel name="F" type="integer"/>
            </regularChannels> ...
        </traceFormat>
    </defs>
    <traceGroup>
        <desc>Lambert W example</desc>
        <trace start="1123890433611">
            1030 985  13
            1024 970  32
            1024 970  47
            1024 960  63
            1024 960  75
            ...
```

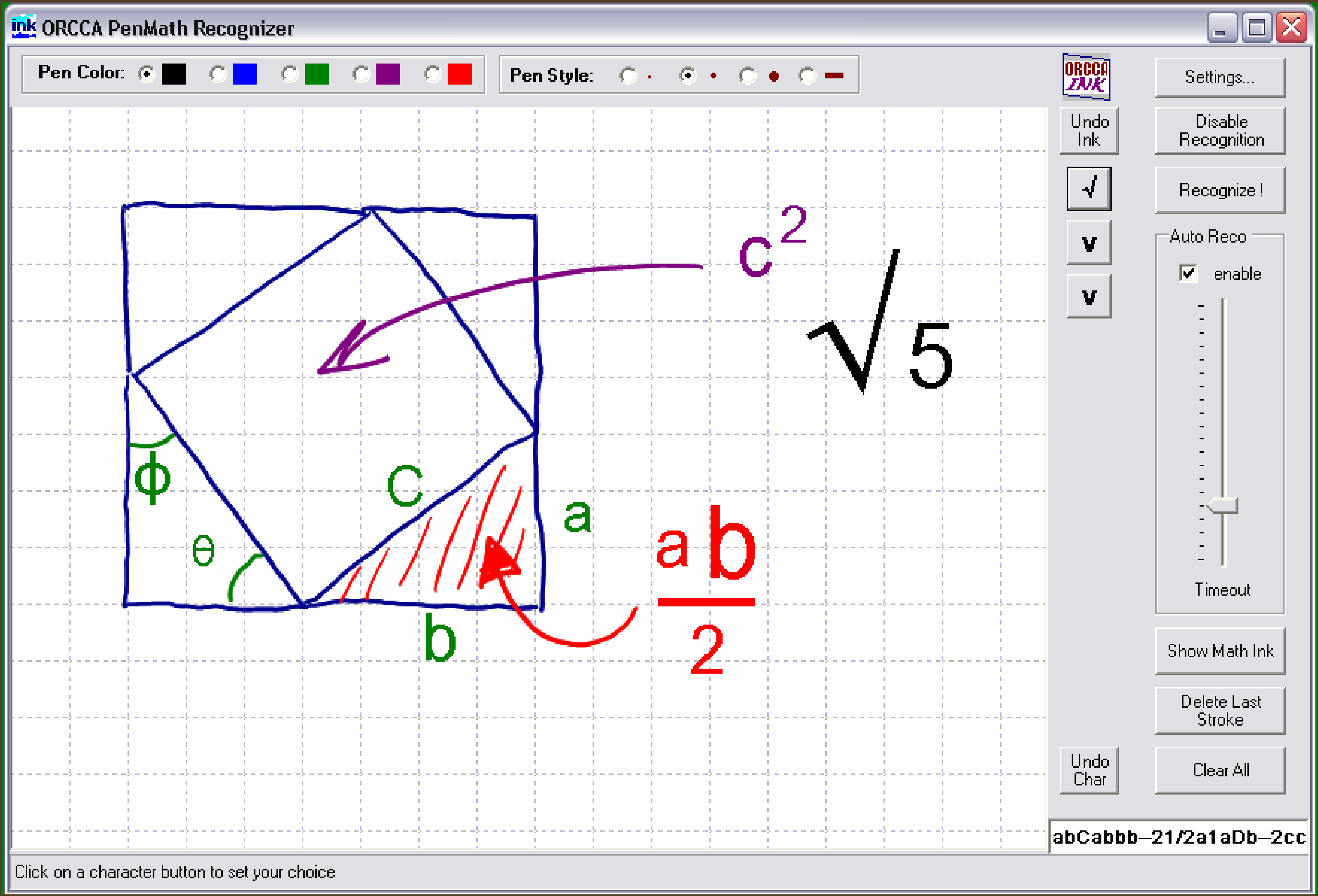
InkML



Brush Tip: ● Circle ○ Line

$$z^{z^{z^{\cdot^{\cdot^{\cdot}}}}} = \frac{W\left(\ln \frac{1}{z}\right)}{\ln\left(\frac{1}{z}\right)}$$

Prototype



ORCCA PenMath Recognizer

Pen Color: ● ○ ○ ○ ○

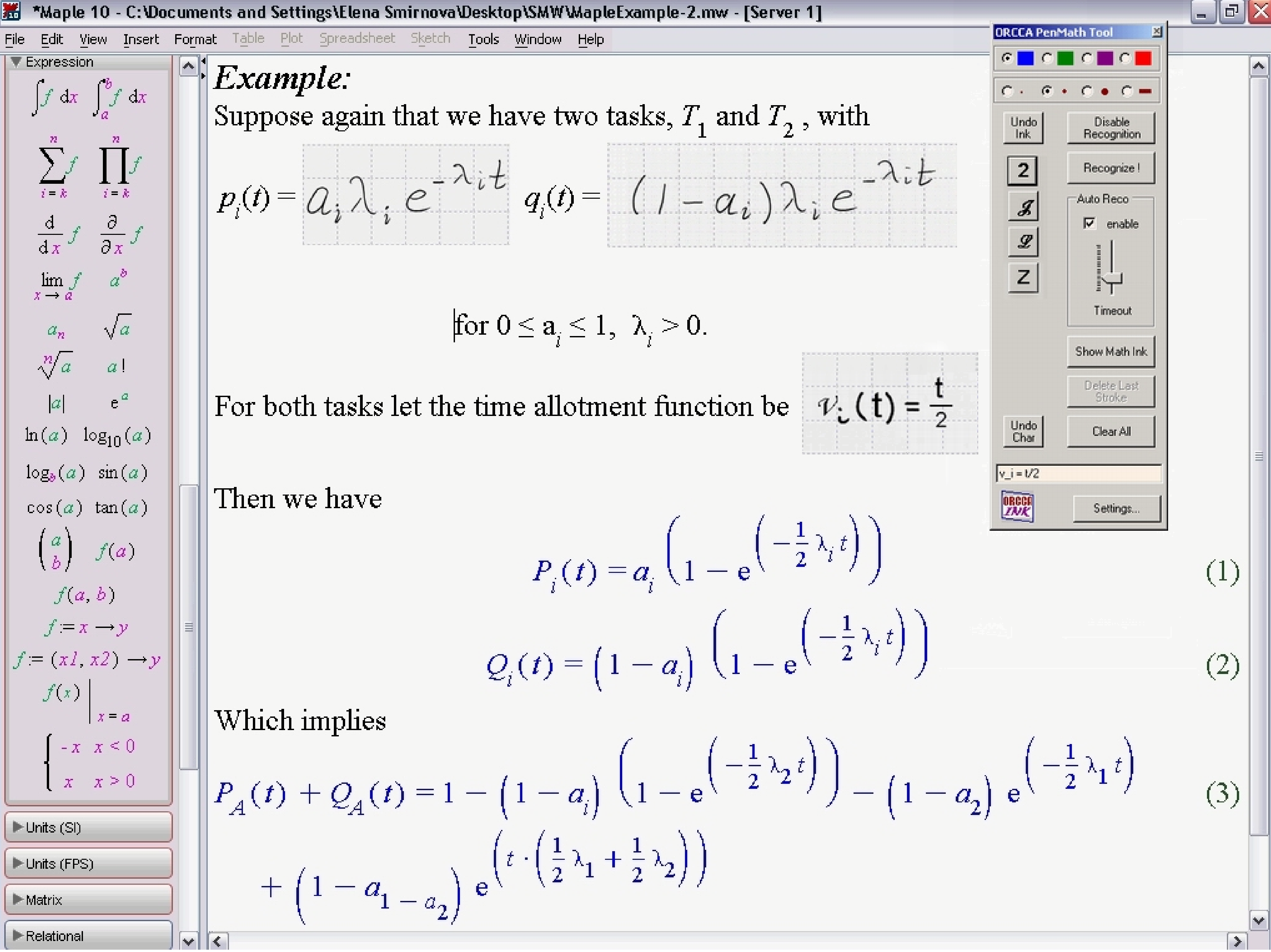
Pen Style: ○ ● ○ ○

Auto Reco

☑ enable

abCabbb–21/2a1aDb–2cc

Click on a character button to set your choice



Example:

Suppose again that we have two tasks, T_1 and T_2, with

$$p_i(t) = a_i \lambda_i e^{-\lambda_i t} \qquad q_i(t) = (1 - a_i)\lambda_i e^{-\lambda_i t}$$

$$\text{for } 0 \le a_i \le 1,\ \lambda_i > 0.$$

For both tasks let the time allotment function be $v_i(t) = \frac{t}{2}$

v_i = t/2

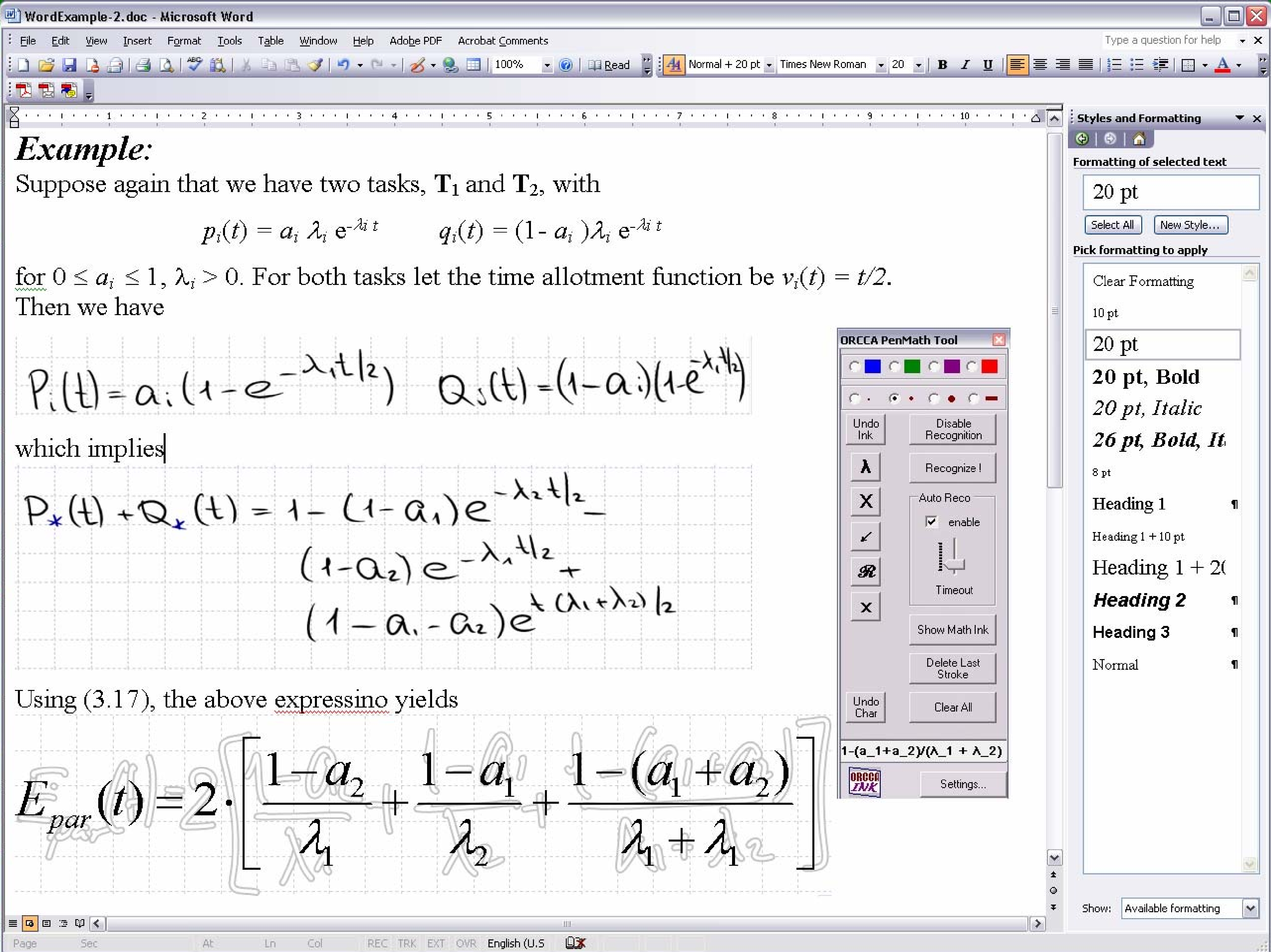
Then we have

$$P_i(t) = a_i \left(1 - e^{\left(-\frac{1}{2}\lambda_i t\right)}\right) \tag{1}$$

$$Q_i(t) = \left(1 - a_i\right)\left(1 - e^{\left(-\frac{1}{2}\lambda_i t\right)}\right) \tag{2}$$

Which implies

$$P_A(t) + Q_A(t) = 1 - \left(1 - a_i\right)\left(1 - e^{\left(-\frac{1}{2}\lambda_2 t\right)}\right) - \left(1 - a_2\right) e^{\left(-\frac{1}{2}\lambda_1 t\right)} + \left(1 - a_{1 - a_2}\right) e^{\left(t \cdot \left(\frac{1}{2}\lambda_1 + \frac{1}{2}\lambda_2\right)\right)} \tag{3}$$



1-(a_1+a_2)/(λ_1 + λ_2)

Example:

Suppose again that we have two tasks, $\mathbf{T}_1$ and $\mathbf{T}_2$, with

$$p_i(t) = a_i\ \lambda_i\ e^{-\lambda i\ t} \qquad q_i(t) = (1- a_i\ )\lambda_i\ e^{-\lambda i\ t}$$

for $0 \le a_i \le 1$, $\lambda_i > 0$. For both tasks let the time allotment function be $v_i(t) = t/2$.
Then we have

$$P_i(t) = a_i(1-e^{-\lambda_i t/2}) \qquad Q_j(t) = (1-a_i)(1-e^{-\lambda_i t/2})$$

which implies

$$P_*(t) + Q_*(t) = 1-(1-a_1)e^{-\lambda_2 t/2} - (1-a_2)e^{-\lambda_1 t/2} + (1-a_1-a_2)e^{t(\lambda_1+\lambda_2)/2}$$

Using (3.17), the above expressino yields

$$E_{par}(t) = 2\cdot\left[\frac{1-a_2}{\lambda_1}+\frac{1-a_1}{\lambda_2}+\frac{1-(a_1+a_2)}{\lambda_1+\lambda_1}\right]$$

Architectural Issues

- Coupling between components
  written variously in C++, Java, Maple

- Feed-back between components

- GUI human-factors issues
  fewest pen movements to accomplish task

- Collaborative back-plane
  multiple displays, multiple pens, shared math objects

Conclusions

- Math can be a “killer app” for pen-based computing
- Many have stood on the toes of giants,
  or at least on each other’s
- Must plan for a complex project with many components
- Building on experience with PocketPC, MathML, Maple
- Work has been underway for just over one year --
  we will see what comes out…
- Hopefully Spider-Mac can put away his pliers!