Gaston, Maple and Mike

Michael Monagan

Center for Experimental and Constructive Mathematics
Simon Fraser University
British Columbia

 GNOME 2014, Zurich,
July 4th, 2014
Me, Gaston and Maple

May 1982 – Dec 1982 Waterloo, Masters student
Jan 1983 – Aug 1989 Waterloo, PhD student

Gaston gave me this paper for my Masters essay
Shafi Goldswasser and Silvio Micali.
Probabilistic encryption & how to play mental poker keeping
secret all partial information.
STOC '82, June 1982
which we implemented in Maple.
Me, Gaston and Maple

May 1982 – Dec 1982 Waterloo, Masters student
Jan 1983 – Aug 1989 Waterloo, PhD student

Gaston gave me this paper for my Masters essay

Shafi Goldwasser and Silvio Micali.

*Probabilistic encryption & how to play mental poker keeping secret all partial information.* STOC ’82, June 1982

which we implemented in Maple.
Gaston’s number theory package, the first Maple package.

> with(numtheory);
Warning, new definition for order

[ F, M, cyclotomic, divisors, factorset, fermat, ifactor, imagunit,

    isprime, issqrfree, ithprime, jacobi, lambda, legendre, mcombine,

    mersenne, mlog, mroot, msqrt, nextprime, order, phi, prevprime,

    pprimroot, primroot, quadres, rootsunit, safeprime, sigma, tau]

I chose not to pursue cryptography for a PhD.
Life as a graduate student with Gaston ...

First Maple retreat, Sparrow lake, summer, 1983
What was Gaston’s main contribution to Maple?

Maple’s Sum-of-Products representation and hashing of all subexpressions.

What is the most important operation to make efficient?

Polynomial multiplication (and division).

But monomial multiplication cost > 200 cycles.
What was Gaston’s main contribution to Maple?

Maple’s Sum-of-Products representation and hashing of all subexpressions.

\[ 9xy^3z - 4y^3z^2 - 6xy^2z - 8x^3 - 5 \]
What was Gaston’s main contribution to Maple?

Maple’s Sum-of-Products representation and hashing of all subexpressions.

\[
\begin{align*}
&\text{PROD 7} & x & 1 & y & 3 & z & 1 \\
&\text{PROD 5} & y & 3 & z & 2 \\
&\text{PROD 7} & x & 1 & y & 2 & z & 1 \\
&\text{PROD 3} & x & 3 \\
&\text{SUM 11} & 9 & -4 & -6 & -8 & -5 & 1 \\
\end{align*}
\]

\[9 \, xy^3 z - 4 \, y^3 z^2 - 6 \, xy^2 z - 8 \, x^3 - 5\]

What is the most important operation to make efficient?
What was Gaston’s main contribution to Maple?

Maple’s Sum-of-Products representation and hashing of all subexpressions.

\[ 9 \, x y^3 \, z - 4 \, y^3 \, z^2 - 6 \, x y^2 \, z - 8 \, x^3 - 5 \]

What is the most important operation to make efficient?
Polynomial multiplication (and division).
But monomial multiplication cost \( > 200 \) cycles.
Singular’s representation

\[ 9 \, xy^3z - 4 \, y^3z^2 - 6 \, xy^2z - 8 \, x^3 - 5 \]
Our new POLY representation (default in Maple 17)

$$\begin{array}{c|c|c|c}
\text{SEQ} & x & y & z \\
\hline
\text{POLY} & 5131 & 9 & 5032 & -4 & 4121 & -6 & 3300 & -8 & 0000 & -5 \\
\end{array}$$

9 \, xy^3z - 4 \, y^3z^2 - 6 \, xy^2z - 8 \, x^3 - 5.

6 advantages
Our new POLY representation (default in Maple 17)

<table>
<thead>
<tr>
<th>SEQ 4</th>
<th>x</th>
<th>y</th>
<th>z</th>
</tr>
</thead>
<tbody>
<tr>
<td>POLY 12</td>
<td>5131</td>
<td>9</td>
<td>5032</td>
</tr>
</tbody>
</table>

\[ 9 \, xy^3 \, z - 4 \, y^3 z^2 - 6 \, xy^2 z - 8 \, x^3 - 5. \]

6 advantages

1. It’s about 4\( \times \) more compact.
2. Memory access is sequential.
3. Kernel operations become \( O(\#terms), \) some \( O(1). \)
4. Monomial multiplication is one 64 bit integer +
   Monomial comparison is one 64 bit integer \( > \)
5. The simpl table is not filled with PRODs.
6. Division cannot cause exponent overflow in graded lex order.

Michael Monagan

GNOME 2014, Zurich
What will fast multiplication using POLY do for the Maple library?

<table>
<thead>
<tr>
<th></th>
<th>Maple 13</th>
<th>Maple 16 1 core</th>
<th>Maple 16 4 cores</th>
<th>Magma 2.16-8</th>
<th>Singular 3.1.0</th>
<th>Mathematica 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>multiply</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( p_4 := f_4(f_4 + 1) )</td>
<td>95.97</td>
<td>2.14</td>
<td>0.643</td>
<td>13.25</td>
<td>30.64</td>
<td>273.01</td>
</tr>
<tr>
<td>divide</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( q_4 := p_4 / f_4 )</td>
<td>192.87</td>
<td>2.25</td>
<td>0.767</td>
<td>18.54</td>
<td>14.96</td>
<td>228.83</td>
</tr>
<tr>
<td>factor</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( p_4 ) 135751 terms</td>
<td>2953.54</td>
<td>59.29</td>
<td>46.41</td>
<td>332.86</td>
<td>404.86</td>
<td>655.49</td>
</tr>
</tbody>
</table>

Hensel lifting is mostly polynomial multiplication!

\[ f_4 = (1 + x + y + z + t)^{20} + 1 \]

10626 terms

Parallel speedup for \( f_4 \times (f_4 + 1) \) is \( \frac{2.14}{.643} = 3.33 \times \). Why?
What will fast multiplication using POLY do for the Maple library?

Intel Core i7 920 2.66 GHz (4 cores)  

<table>
<thead>
<tr>
<th>Operation</th>
<th>Maple 13</th>
<th>Maple 16</th>
<th>Magma 2.16-8</th>
<th>Singular 3.1.0</th>
<th>Mathematica 7</th>
</tr>
</thead>
<tbody>
<tr>
<td>multiply</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$p_4 := f_4(f_4 + 1)$</td>
<td>95.97</td>
<td>2.14</td>
<td>0.643</td>
<td>13.25</td>
<td>30.64</td>
</tr>
<tr>
<td>divide</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$q_4 := p_4/f_4$</td>
<td>192.87</td>
<td>2.25</td>
<td>0.767</td>
<td>18.54</td>
<td>14.96</td>
</tr>
<tr>
<td>factor</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$p_4$ 135751 terms</td>
<td>2953.54</td>
<td>59.29</td>
<td>46.41</td>
<td>332.86</td>
<td>404.86</td>
</tr>
</tbody>
</table>

Hensel lifting is mostly polynomial multiplication!

$f_4 = (1 + x + y + z + t)^{20} + 1$ 10626 terms

Parallel speedup for $f_4 \times (f_4 + 1)$ is $2.14 / .643 = 3.33 \times$. Why?
Conversion overhead between POLY and SUM of PRODs!
After brainstorming with Roman, I asked Laurent if we could make POLY the default in Maple. Maple 17 uses POLY if all monomials in a polynomial with integer coefficients fit in 64 bits - otherwise we use SUM-of-PRODs. Conversions between POLY and SUM-of-PRODs are automatic and invisible to the Maple user.
So we coded POLY for each kernel routine. Faster at everything except op, map, etc.

<table>
<thead>
<tr>
<th>command</th>
<th>Maple 16</th>
<th>Maple 17</th>
<th>speedup</th>
<th>notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>coeff(f, x, 20)</td>
<td>2.140 s</td>
<td>0.005 s</td>
<td>420x</td>
<td>terms easy to locate</td>
</tr>
<tr>
<td>coeffs(f, x)</td>
<td>0.979 s</td>
<td>0.119 s</td>
<td>8x</td>
<td>reorder exponents and radix</td>
</tr>
<tr>
<td>degree(f, x)</td>
<td>0.073 s</td>
<td>0.003 s</td>
<td>24x</td>
<td>stop early using monomial degree</td>
</tr>
<tr>
<td>diff(f, x)</td>
<td>0.956 s</td>
<td>0.031 s</td>
<td>30x</td>
<td>terms remain sorted</td>
</tr>
<tr>
<td>eval(f, x = 6)</td>
<td>3.760 s</td>
<td>0.175 s</td>
<td>21x</td>
<td>use Horner form recursively</td>
</tr>
<tr>
<td>expand(2 * x * f)</td>
<td>1.190 s</td>
<td>0.066 s</td>
<td>18x</td>
<td>terms remain sorted</td>
</tr>
<tr>
<td>indets(f)</td>
<td>0.060 s</td>
<td>0.000 s</td>
<td>→ O(1)</td>
<td>first word in dag</td>
</tr>
<tr>
<td>op(f)</td>
<td>0.634 s</td>
<td>2.420 s</td>
<td>0.26x</td>
<td>has to construct old structure</td>
</tr>
<tr>
<td>for t in f do</td>
<td>0.646 s</td>
<td>2.460 s</td>
<td>0.26x</td>
<td>has to construct old structure</td>
</tr>
<tr>
<td>taylor(f, x, 50)</td>
<td>0.668 s</td>
<td>0.055 s</td>
<td>12x</td>
<td>get coefficients in one pass</td>
</tr>
<tr>
<td>type(f, polynom)</td>
<td>0.029 s</td>
<td>0.000 s</td>
<td>→ O(n)</td>
<td>type check variables only</td>
</tr>
<tr>
<td>f;</td>
<td>0.162 s</td>
<td>0.000 s</td>
<td>→ O(n)</td>
<td>evaluate the variables</td>
</tr>
</tbody>
</table>

For \( f \) with \( n = 3 \) variables and \( t = 10^6 \) terms created by

\[
f := \text{expand}\left(\text{mul}(\text{randpoly}(v,\text{degree}=100,\text{dense}),v=[x,y,z])\right);
\]
\begin{tabular}{|c|c|c|c|}
\hline
\text{multiply} & \text{Maple 16} & \text{Maple 17} \\
& 1 core & 4 cores & 1 core & 4 cores \\
\hline
\( p_4 := f_4(f_4 + 1) \) & 2.140 & 0.643 & 1.770 & 0.416 \\
\hline
\text{factor} & & & \\
\( p_4 \) 135751 terms & 59.27 & 46.41 & 24.35 & 12.65 \\
\hline
\end{tabular}

Intel Core i5 750 2.66 GHz 4 cores. Real times in seconds.

\[ f_4 = (1 + x + y + z + t)^{20} + 1 \]

10626 terms

Parallel speedup for \( f_4 \times (f_4 + 1) \) is \( 1.77/0.416 = 4.2\times \). How?
Joris van der Hoven: Do you use the extra bits for the total degree?
My answer: No, because ...
I changed my mind. Roman Pearce recoded everything for Maple 18.

<table>
<thead>
<tr>
<th>$n$</th>
<th>per variable</th>
<th>total degree</th>
<th>$V_n = \det n \times n$ Vandermonde</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>#bits  maxdeg</td>
<td>#bits  maxdeg</td>
<td>deg Maple 16 17 18</td>
</tr>
<tr>
<td>7</td>
<td>8    255</td>
<td>8    255</td>
<td>21  0.012s 0.005 0.004</td>
</tr>
<tr>
<td>8</td>
<td>7    127</td>
<td>8    255</td>
<td>28  0.093s 0.027 0.026</td>
</tr>
<tr>
<td>9</td>
<td>6    63</td>
<td>10   1023</td>
<td>36  1.35 s 0.218 0.150</td>
</tr>
<tr>
<td>10</td>
<td>5    31</td>
<td>14   16383</td>
<td>45  15.95s 25.44 1.57</td>
</tr>
<tr>
<td>11</td>
<td>5    31</td>
<td>9    511</td>
<td>55  – – 18.87</td>
</tr>
<tr>
<td>12</td>
<td>4    15</td>
<td>16   65535</td>
<td>66  236.4</td>
</tr>
<tr>
<td>13</td>
<td>4    15</td>
<td>12   4095</td>
<td>78  –</td>
</tr>
<tr>
<td>14</td>
<td>4    15</td>
<td>8    255</td>
<td>91  –</td>
</tr>
<tr>
<td>15</td>
<td>4    15</td>
<td>4    15</td>
<td>105</td>
</tr>
<tr>
<td>16</td>
<td>3    7</td>
<td>16   65535</td>
<td>120</td>
</tr>
</tbody>
</table>
Maple retreat, Sparrow lake, circa 1992

Thank you Gaston for Waterloo, Zurich and Maple. Mike.
Given a polynomial $f(x_1, x_2, ..., x_n)$, we store $f$ using POLY if

1. $f$ is expanded and has integer coefficients,
2. $d > 1$ and $t > 1$ where $d = \text{deg} \ f$ and $t = \#\text{terms}$,
3. we can pack all monomials of $f$ into one 64 bit word, i.e. if $d < 2^b$ where $b = \lfloor \frac{64}{n+1} \rfloor$

Otherwise we use the sum-of-products representation.

- The representation is invisible to the Maple user. Conversions are automatic.
- POLY polynomials will be displayed in sorted order.
- Packing is fixed by $n = \#\text{variables}$.
- Maple 18 uses remaining bits for total degree.
Parallel multiplication using a binary heap.

**Target architecture**

One thread per core.

Threads write to a finite circular buffer.

 Threads try to acquire global heap as buffer fills up to balance load.