Sparse Polynomial Multiplication and Division in Maple 14

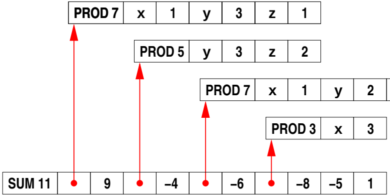
Michael Monagan and Roman Pearce

Simon Fraser University

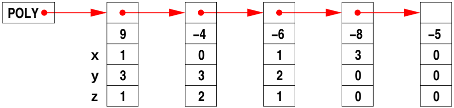
June 2010

Supported by MITACS NCE of Canada and Maplesoft

Polynomial Data Structures

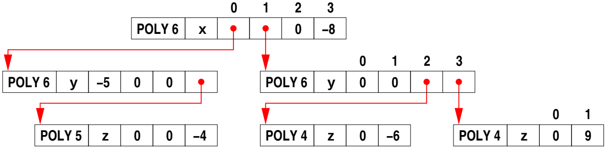


Maple



Singular

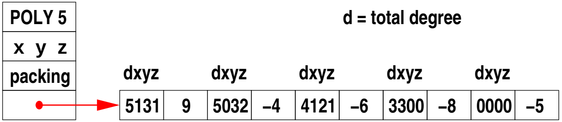
$$9xy^3z - 4y^3z^2 - 6xy^2z - 8x^3 - 5$$



Pari

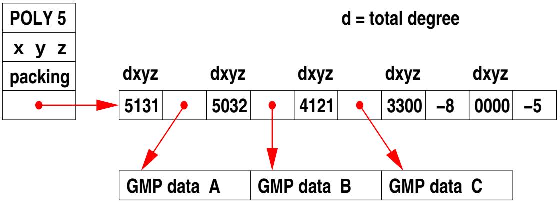
$$(-5y - 4z^2y^3) + (-6zy^2 + 9zy^3)x - 8x^3$$

Our Packed Array Representation



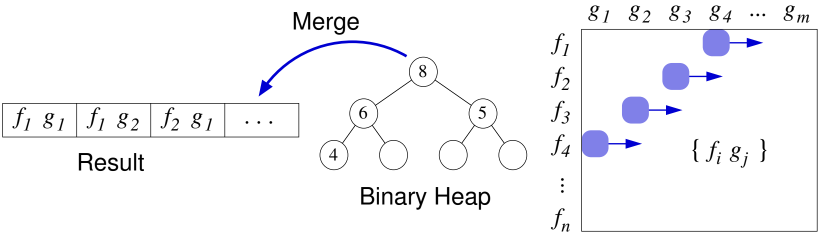
$$9xy^3z - 4y^3z^2 - 6xy^2z - 8x^3 - 5$$

encode $x^i y^j z^k$ as $[(i + j + k), i, j, k]$ in a 64 bit int (grlex)



Multiplication and Division Using a Heap

$f \cdot g = \sum f_i \cdot g$ simultaneous n-ary merge



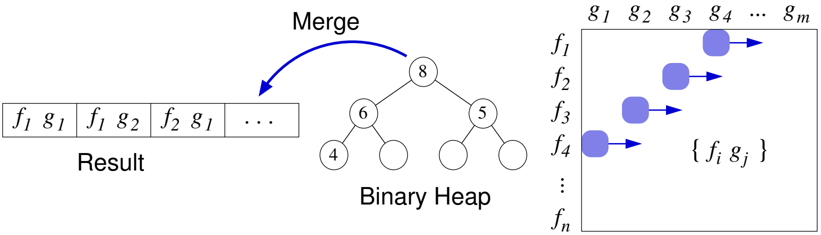
multiply in $O(\#f \#g \log(\min(\#f, \#g)))$ comparisons

Monagan & Pearce: $f - \sum_{i=1}^{\#g} q_i \cdot g - \sum_{i=2}^{\#g} g_i \cdot (q_{\#g+1} + \dots)$

- $f \div g = q$ in $O(\#f + \#q\#g \log(\min(\#q, \#g)))$ comparisons
- $O(\#f + \#q\#g)$ multiplications, $O(\#q)$ divisions & gcds in $\mathbb{Z}$

Multiplication and Division Using a Heap

$f \cdot g = \sum f_i \cdot g$ simultaneous n-ary merge

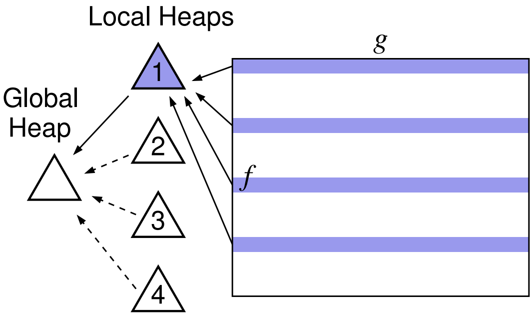


multiply in $O(\#f\#g \log(\min(\#f, \#g)))$ comparisons

Monagan & Pearce: $f - \sum_{i=1}^{\#g} q_i \cdot g - \sum_{i=2}^{\#g} g_i \cdot (q_{\#g+1} + \dots)$

- $f \div g = q$ in $O(\#f + \#q\#g \log(\min(\#q, \#g)))$ comparisons
- $O(\#f + \#q\#g)$ multiplications, $O(\#q)$ divisions & gcds in $\mathbb{Z}$

Parallel Algorithms



- distribute partial products $f_i \cdot g$ to cores
- threads take turns merging results in global heap
- superlinear speedup in practice (Core i3/i5/i7 & Phenom)
- parallel multiplication (ISSAC '09) now in Maple 14
- parallel division (PASCO '10) recently completed

Maple Integration

```
if #f > 2 and #g > 2 and (#f #g > 1500) then

‘expand/bigprod‘ := proc(a,b)  # multiply two large sums
   ...  # type check, variables, degree
   A := sdmp:-Import(a, plex(vars), pack=k);
   B := sdmp:-Import(b, plex(vars), pack=k);
   C := sdmp:-Multiply(A,B);
   return sdmp:-Export(C);

‘expand/bigdiv‘ := proc(a,b,q)  # exact division over Z
   ...  # type check, variables, degree
   A := sdmp:-Import(a, grlex(vars), pack=k);
   B := sdmp:-Import(b, grlex(vars), pack=k);
   if sdmp:-Divide(A,B,’Q’) then
     q := sdmp:-Export(Q);
     return true;
   else
     return false;
```

Benchmarks

Intel Core i7 920 2.66 GHz (4 cores, no hyperthreading) (64 bit Linux)
See paper for Singular, Mathematica, and Pari timings.

	Maple 13	Maple 14 1 core	Maple 14 4 cores		Magma 2.16-8	Trip 1.0 (RS)	Trip 1.0 (RD)
multiply $p_3 := f_3(f_3+1)$	26.76s	0.532s	0.166s	28%	4.09s	0.376s	0.314s
divide $q_3 := p_3/f_3$	24.74s	0.73s	0.74s	5%	4.31s	1.880s	1.722s
factor $\#p_3 = 38711$	391.44s	17.15s	15.59s		117.53s		

$f_1 = (1+x+y+z)^{20}+1$
1771 terms

$\mathbf{f_3 = (1+x+y+z)^{30}+1}$
5456 terms

$f_2 = (1+x^2+y^2+z^2)^{20}+1$
1771 terms

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