Combined Report of Weeks 1 to 9

Ananth Krishna Kidambi, Guramrit Singh

August 28, 2023

Contents

Week 9

Code Implementation

Example

Consider the set of equations -

$$f0(x0,x1) = \sum_{x2=0} x1((-1)^{x1-x2}\cdot \binom{x1}{x2} \cdot \sum_{x3=0} x0[(-1)^{x0-x3} \cdot \binom{x0}{x3} \cdot f1(x2,x3))$$

$$f1(x2,x3) = \binom{x3}{0} \cdot f1(-1+x2, x3-0) + \binom{x3}{1} \cdot f1(-1+x2, x3-1)$$

$$f1(0,x3) = 1$$

$$f1(x2,0) = 1$$

The corresponding code generated is -

```cpp
#include <iostream>
#include <string>
#include <vector>
#include <cmath>
#include <gmpxx.h>

class cache_elem{
public :
    mpz_class n;
    cache_elem(mpz_class x) : n{x} {}
    cache_elem() : n{-1} {}
};

template <class T> T& get_elem(std::vector<T>& a, size_t n){
    if (n >= a.size()){
        a.resize(n+1);
    }
```

```
18      return a.at(n);
19  }
20
21  mpz_class Binomial(unsigned int n, unsigned int r){
22      mpz_t ans;
23      mpz_init(ans);
24      mpz_bin_uiui(ans, n, r);
25      return mpz_class{ans};
26  }
27
28  mpz_class power(mpz_class x, unsigned int y){
29      mpz_t ans;
30      mpz_init(ans);
31      mpz_pow_ui(ans, x.get_mpz_t(), y);
32      return mpz_class{ans};
33  }
34
35  std::vector<cache_elem> f0_cache;
36  std::vector<std::vector<cache_elem>> f1_cache;
37
38  mpz_class f0(unsigned int x0);
39  mpz_class f1(unsigned int x1, unsigned int x2);
40  mpz_class f1_0x(unsigned int x2);
41  mpz_class f1_x0(unsigned int x1);
42
43  mpz_class f0(unsigned int x0){
44      mpz_class& stored_val = get_elem(f0_cache, x0).n;
45      if (stored_val != -1)
46          return stored_val;
47      if (x0 >= 0){
48          mpz_class ret_val = ([x0](){mpz_class sum{0}; for (unsigned x1 = 0; x1 <= x0; x1+-
49          get_elem(f0_cache, x0).n = ret_val;
50          return ret_val;
51      }
52      exit(1);
53      return -1;
54  }
55  mpz_class f1(unsigned int x1, unsigned int x2){
56      mpz_class& stored_val = get_elem(get_elem(f1_cache, x1), x2).n;
57      if (stored_val != -1)
58          return stored_val;
59      if (x1 >= 1 && x2 >= 1){
60          mpz_class ret_val = (Binomial(x2,0)*f1(x1-1,x2-0))+(Binomial(x2,1)*f1(x1-1,x2-1));
61          get_elem(get_elem(f1_cache, x1), x2).n = ret_val;
62          return ret_val;
```

```
63      }
64      else if (x1 == 0){
65          return  f1_0x( x2);
66      }
67      else if (x2 == 0){
68          return  f1_x0( x1);
69      }
70      exit(1);
71      return -1;
72  }
73  mpz_class f1_0x(unsigned int x2){
74      mpz_class& stored_val = get_elem(get_elem(f1_cache, 0), x2).n;
75      if (stored_val != -1)
76          return stored_val;
77      if (x2 >= 0){
78          mpz_class ret_val = 1;
79          get_elem(get_elem(f1_cache, 0), x2).n = ret_val;
80          return ret_val;
81      }
82      exit(1);
83      return -1;
84  }
85  mpz_class f1_x0(unsigned int x1){
86      mpz_class& stored_val = get_elem(get_elem(f1_cache, x1), 0).n;
87      if (stored_val != -1)
88          return stored_val;
89      if (x1 >= 0){
90          mpz_class ret_val = 1;
91          get_elem(get_elem(f1_cache, x1), 0).n = ret_val;
92          return ret_val;
93      }
94      exit(1);
95      return -1;
96  }
97
98  int main(){
99      std::cout << f0(2048) << std::endl;
100 }
```

Week 8

Code Implementation

Example

Consider the set of equations -

$$f0(x0,x1) = \sum_{x2=0} x1((-1)^(x1-x2)\cdot Binomial(x1,x2)\cdot \sum_{x3=0} x0[(-1)^(x0-x3)\cdot Binomial(x0,x3)\cdot f1(x2,x3))$$

$$f1[x2,x3] = Binomial(x3,0)\cdot f1(-1+x2,x3-0) + Binomial(x3,1)\cdot f1[-1+x2,x3-1]$$

$$f1[0,x3] = 1$$

$$f1[x2,0] = 1$$

The corresponding code generated is -

```cpp
#include <iostream>
#include <string>
#include <vector>
#include <cmath>

class cache_elem{
public :
    int n;
    cache_elem(int x) : n{x} {}
    cache_elem() : n{-1} {}
};

template <class T> T& get_elem(std::vector<T>& a, size_t n){
    if (n >= a.size()){
        a.resize(n+1);
    }
    return a.at(n);
}
```

```
19
20 int Binomial(int n, int r){
21     return round(std::tgamma(n+1)/(std::tgamma(r+1)*std::tgamma(n-r+1)));
22 }
23
24 int power(int x, int y){
25     return round(pow(x, y));
26 }
27
28 std::vector<std::vector<cache_elem>> f0_cache;
29 std::vector<std::vector<cache_elem>> f1_cache;
30
31 int f0(int x0, int x1);
32 int f1(int x2, int x3);
33 int f1_0x(int x3);
34 int f1_x0(int x2);
35
36 int f0(int x0, int x1){
37     int stored_val = get_elem(get_elem(f0_cache, x0), x1).n;
38     if (stored_val != -1)
39         return stored_val;
40     if (x0 >= 0 && x1 >= 0){
41         int ret_val = ([x0,x1](){int sum{0}; for (unsigned x2 = 0; x2 <= x1; x2++){ sum +=
42         get_elem(get_elem(f0_cache, x0), x1).n = ret_val;
43         return ret_val;
44     }
45     return -1;
46 }
47 int f1(int x2, int x3){
48     int stored_val = get_elem(get_elem(f1_cache, x2), x3).n;
49     if (stored_val != -1)
50         return stored_val;
51     if (x2 >= 1 && x3 >= 1){
52         int ret_val = (Binomial(x3,0)*f1(-1+x2,x3-0))+(Binomial(x3,1)*f1(-1+x2,x3-1));
53         get_elem(get_elem(f1_cache, x2), x3).n = ret_val;
54         return ret_val;
55     }
56     else if (x2 == 0){
57         return  f1_0x( x3);
58     }
59     else if (x3 == 0){
60         return  f1_x0( x2);
61     }
62     return -1;
63 }
```

```
int f1_0x(int x3){
    int stored_val = get_elem(get_elem(f1_cache, 0), x3).n;
    if (stored_val != -1)
        return stored_val;
    if (x3 >= 0){
        int ret_val = 1;
        get_elem(get_elem(f1_cache, 0), x3).n = ret_val;
        return ret_val;
    }
    return -1;
}
int f1_x0(int x2){
    int stored_val = get_elem(get_elem(f1_cache, x2), 0).n;
    if (stored_val != -1)
        return stored_val;
    if (x2 >= 0){
        int ret_val = 1;
        get_elem(get_elem(f1_cache, x2), 0).n = ret_val;
        return ret_val;
    }
    return -1;
}

int main(){
    std::cout << f0(3,3) << std::endl;
}
```