GENERATING FUNCTIONS AND LOGIC IN INTRODUCTORY DISCRETE MATHEMATICS WITH MAPLE

Dr. Joel C. Fowler
Mathematics Department, Southern Polytechnic State University
1100 South Marietta Parkway
Marietta, GA 30060-2896
jfowler@spsu.edu

We consider three areas of Discrete Mathematics where Maple can be useful: Symbolic Logic Computations, Solving Counting Problems with Generating Functions, and Game Computations in Two-Person, Progressively Finite, Impartial Games of Perfect Information.

Logic Problems

Maple's truth table capabilities can be used to evaluate logical arguments without the need for large truth tables.

Example 1) The deficit is too large only if the dollar is weak. If fuel prices are high then the dollar is weak or fuel demand is high. Whenever the deficit is too large, fuel prices are high and taxes are too high. But fuel prices are high, although taxes are not. Hence fuel demand must be high. Is this a logically sound argument?

Solution: Let: dw = dollar is weak, fh = fuel prices are high, dl = deficit is large, th = taxes are too high, and dh = demand is high for fuel. Execution of:

```
> with(Logic):
> TruthTable(((dl &implies dw) &and (fh &implies (dw &or dh)) &and (dl &implies (fh &and th)) &and (fh) &and (&not th)) &implies (dh), [dw,fh,dl,th,dh], form = MOD2);
> Tautology(((dl &implies dw) &and (fh &implies (dw &or dh)) &and ((fh &and cl) &implies dl) &and (&not cl) &and (&not dl) &and (fh) ) &implies (dh));
```

shows that the argument is not valid and fails precisely when the dollar is weak, fuel prices are high, the deficit isn't too large, taxes aren't too high, and fuel demand isn't high.

Counting Problems

The easily accessible polynomial operations of Maple can be used to introduce generating functions as a tool much earlier than is typical.

Example 2) How many solutions in nonnegative integers are there to the equation:
$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 = 10$ with all x_i no more than 5? Equivalently we could ask how many ways are there to distribute 10 identical candybars to 7 different locations with

no location receiving more than 5, or how many ways are there to make an unordered selection of 10 poker chips from a collection of 35 chips, 5 each of red, blue, white, yellow, green, black, and orange?

Solution: These can all be solved by finding the coefficient of x^{10} in the expansion of $(1 + x + x^2 + \ldots + x^5)^7$. Via any of the following Maple statements, the answer is 6538.

```
> coeff( (1+x+x^2+x^3+x^4+x^5)^7,x^10);
> coeff( taylor(((1 - x^6)/(1 - x))^7, x, 11), x^10);
> coeftayl( ((1 - x^6)/(1 - x))^7 , x=0, 10);
```

We note that the statement: coeff(((1 - x^6)/(1 - x))^7, x^10) does not produce an answer.

Example 3) How many unordered 7 card hands are possible from 5 decks of cards?

Solution: This is equivalent to finding the number of solutions in nonnegative integers to the equation $x_1 + x_2 + x_3 + \ldots + x_{52} = 7$ with all x_i no more than 5. The number of solutions is the coefficient of x^7 in the expansion of: $(1 + x + x^2 + \ldots + x^5)^{52}$. Via:

```
> coeff( (1+x+x^2+x^3+x^4+x^5)^52,x^7);
```

the answer is: 300,671,384

Example 4) What is the probability of rolling a sum of 30 with 10 dice?

Solution: We find the number of ways by counting the solutions in integers to the equation: $x_1 + x_2 + x_3 + \ldots + x_{10} = 30$ with all x_i from 1 to 6 and then dividing by 6^{10}. Via:

```
> coeff(sum(x^i,i=1..6)^10,x^30);
> evalf[7](coeff(sum(x^i,i=1..6)^10,x^30)/6^10);
```

we have the answer as 2,930,455 / 60,466,176 ~ .04846 .

Example 5) Find the number of non-negative integer solutions to the equation: $x_1 + x_2 + x_3 + \ldots + x_7 = n$ with x_1 even, x_2 odd, x_3 and x_4 less than 6, and x_5 at least 4.

Solution: The number of solutions is the coefficient of x^n in the expansion of:

$$(1 + x^2 + x^4 + \ldots)(x + x^3 + x^5 + \ldots)(1 + x + x^2 + \ldots + x^5)^2(x^4 + x^5 + x^6 + \ldots)(1 + x + x^2 + \ldots)^2$$

$$= \frac{1}{1-x^2}\cdot\frac{x}{1-x^2}\cdot\frac{1-x^6}{1-x}\cdot\frac{x^4}{1-x}\cdot\frac{1}{(1-x)^2} = \frac{x^5(1-x^6)}{(1-x)^6(1+x)^2}$$

. Via higher level Maple:

```
> with(genfunc);
```

```
> seq( [i,rgf_term(x^5*(1-x^6)^2 / ((1-x^2)^2*(1-x)^5), x, i)],i=0..8),[n,expand(rgf_expand(rem( x^5*(1-x^6)^2 , (1-x^2)^2*(1-x)^5, x) / ((1-x^2)^2*(1-x)^5), x, n))];
```

we have the number of solutions is: $\frac{3}{8}n^4 - \frac{39}{4}n^3 + \frac{849}{8}n^2 - \frac{2223}{4}n + 1156$, for n>8; and

0, 0, 0, 0, 0, 1, 5, 17, 45 for n=0, 1, 2, 3, 4, 5, 6, 7, 8 respectively.

Two-Person Games

We will be concerned with two player, deterministic, impartial, progressively finite games with no ties. The winner is the last player to make a legal move. It is well known that in games of this type:

1) A winning strategy must exist for either the first or second player.
2) A unique set, A, of **safe** positions exists such that all winning positions are in A, every move from a position in A is to a position not in A, and from every position not in A, a move exists to a position in A.
3) A winning strategy is to make a move to a safe position at the first opportunity, and thereafter always make a move to safe positions.
4) Safe positions are those of **Grundy-Sprague** number 0. These numbers are defined by: Let GS(P) = 0 for all winning positions P, and thereafter, for a position P, whose successor positions are $X_1, X_2, \ldots, X_n$, GS(P) = minimum non-negative integer not in { $GS(X_1)$, $GS(X_2), \ldots, GS(X_n)$ } = mex({ $GS(X_1), GS(X_2), \ldots, GS(X_n)$ }).
5) The sum of games $g_1, g_2, \ldots, g_k$ is the game in which a legal move consists of a single move in any one of the games. The GS number of the game sum is given by: $GS(g_1) +_d GS(g_2) +_d \ldots +_d GS(g_k)$. $n +_d m$, the **digital** or **Nim - sum**, is the number whose binary representation is the mod 2 vector sum of the binary representations of n and m.

A wide variety of games are known and studied. In teaching this material the frequently onerous computations can obscure the theory and reduce the enjoyment of playing the games. Maple procedures can be written for computing values for various games. These can be used to study properties of particular games. The following are just a few examples.

Example 6) Subtraction Games. A game is played with a single pile of counters. At each turn a player may remove any number of counters in a fixed non-empty set of positive integers, S. The last player to remove a counter wins.

```
> SubtractGS := proc(S::list, n::integer ) description "Grundy-Sprague Numbers for Subtraction Games";
Temp(0) := 0; j := - max(op(S))-1 ; while j < n do j := j+1;
        if (j > 0) then
                Temp(j) := min( op({seq(i,i=0..nops(S)+1)} minus {seq(Temp(j-S[i]),i=1..nops(S))}))
        else Temp(j) := sign(j)-1
```

```
        end if;  end do; Temp(n)
end proc;
```

This procedure can be used to generate the sequence of Grundy-Sprague numbers for various subtraction sets to form conjectures about their behavior. Consider, for example, the GS sequence for S = { 1, k } for k even and k odd. Then prove that the behavior you see always occurs.

Example 7) Digital Sum Computation

```
> DS := proc(n::nonnegint , m::nonnegint) description "Digital/Nim-Sum";
        Value :=0 ; nv:= n ; mv:= m ; count := 0 ;
                while ((nv > 0) or (mv > 0)) do
                        Value := Value + ((nv + mv) mod 2) * (2^count);
                        count := count + 1; nv := trunc(nv/2); mv := trunc(mv/2);
                end do; Value
end proc:
```

```
>  DSlist := proc(S::list) description "Find Digital (Nim) Sum of a list of values";
Sum :=0 ; count := 0 ;
while count < nops(S) do Sum := DS(Sum,S[count+1]) ; count := count+1 ; end do ;
Sum
end proc:
```

Example 8) Kayles. A Kayles position consists of several separated groupings of pins in a horizontal line. At each turn a player may remove one or two adjacent pins from any position in any one of the groupings. The last player to remove a pin wins. We can use Maple to find GS numbers for Kayles.

```
> KaylesGS := proc(S::list)   description "Grundy-Sprague Numbers for Kayles";
Temp(0):= 0 ; Temp(1):= 1 ; Temp(2):=2 ; j:= 2;
        while ( j < max(op(S)) ) do j:=j+1; Temp(j) := min( op( {seq(i,i=0..(j+1))} minus
        ({seq(DS(Temp(i-1),Temp(j-i)),i=1..(floor((j+1)/2)))} union
        {seq(DS(Temp(i-2),Temp(j-i)),i=2..(floor(j/2)+1))}) ) );
        end do;
Sum :=0 ; count := 0 ;
while count < nops(S) do Sum := DS(Sum,Temp(S[count+1]));count := count+1;end do;
Sum
end proc;
```

Again we can investigate the sequence of GS numbers. For example, it appears that a single solid line is never safe since the GS numbers always seem to be positive. A proof of this fact can then be constructed. Also, once we have the GS numbers, we can use Maple for the more time consuming task of finding best plays from a given position.

```
>ABestPlayKayles := proc(S::list) description "Find a Winning move from a given Kayles
Game position";
Value := KaylesGS(S); Ans := S ;
location:=1 ; while ( DS(Value,KaylesGS([S[location]])) > KaylesGS([S[location]]) ) do
location:=location+1; end do;
        split:=1 ; while ( ( DS(KaylesGS([split-1]),KaylesGS([S[location]-split])) <>
        DS(Value,KaylesGS([S[location]])) ) and ( split < floor((S[location]+1)/2)+1 ) )
        do split:=split+1; end do;
        if(split<floor((S[location]+1)/2)+1) then Ans[location] := [split-1,S[location]-split]
        else
        split := 2 ; while ( ( DS(KaylesGS([split-2]),KaylesGS([S[location]-split])) <>
        DS(Value,KaylesGS([S[location]])) ) and ( split < floor(S[location]/2)+2 ) )
        do split:=split+1; end do; if ( split < floor(S[location]/2)+2) then
        Ans[location] := [split-2,S[location]-split] end if;
        end if; Ans
end proc:
```

Example 9) Grundy's Game is played with several piles of counters. At each turn a player may divide any pile into two unequal sized piles. The last player able to divide a pile wins.

```
> GrundyGameGS := proc(S::list )   description "Grundy-Sprague Numbers for Grundy's
Game";
        Temp(0):= 0 ; Temp(1):= 0 ; Temp(2):=0 ; j:= 2;
        while ( j < max(op(S)) ) do j:= j+1; Temp(j) := min( op( {seq(i,i=0..(j+1))} minus
        {seq(DS(Temp(i),Temp(j-i)),i=1..(ceil(j/2)-1))} ) );
        end do; Sum :=0 ; count := 0 ;
while count<nops(S) do Sum := DS(Sum,Temp(S[count+1]));count :=count+1 ; end do ;
Sum
end proc;
```

Using this procedure we can form conjectures about various game situations. For example, the two-pile position [k, k+1] appears to be unsafe for all k>1. Can you prove this?