Maple Program to Compute Conjugate Sets and mark Primitives PhL 7.2.13

This is all done and installed as Chapter 5 (k), I like it.

I didn't realize I was so close to having this capability! Looking at section below Galois (5.24) I have almost the whole thing done! What I need basically is to mark which ones are primitives and to make Maple filter out stuff which I did by hand.

There are two filters required:

(1) if an exponent set replicates another one, throw the new one out
(2) throw out duplicate elements within a set.

But not quite there yet. I have a series of things to learn how to do.

Here is how to take the exponents, get them into a sequence joe and then a list and **sort** them into a list bob.

```
joe := e[1]:
for n from 2 to m do
   joe := joe,e[n];
od;
```

$$joe := 30, 29$$
$$joe := 30, 29, 27$$
$$joe := 30, 29, 27, 23$$
$$joe := 30, 29, 27, 23, 15$$

```
bob := sort([joe]);
```

$$bob := [15, 23, 27, 29, 30]$$

How do I remove duplicate elements from a list? Answer: convert it to a set and then back to a list.

1. Conversion between Set and Sequence

```
seta := {1,2,3,4};
```

$$seta := \{1, 2, 3, 4\}$$

```
seqa := op(seta);
```

$$seqa := 1, 2, 3, 4$$

```
setaa := {seqa};
```

$$setaa := \{1, 2, 3, 4\}$$

2. Conversion between List and Sequence

```
lista := [a,b,c,d];
```

$$lista := [a, b, c, d]$$

```
seqa := op(lista);
```

$$seqa := a, b, c, d$$

```
listaa := [seqa];
```

$$listaa := [a, b, c, d]$$

3. Conversion between Set and List. Note how to filter duplicate items!

```
lista := [x,y,z,t];
```

$$lista := [x, y, z, t]$$

```
seta := convert(lista,set);
```

$$seta := \{x, y, t, z\}$$

```
listaa := convert(seta,list);
```

$$listaa := [x, y, t, z]$$

```
lista := [x,y,y,t];
```

$$lista := [x, y, y, t]$$

```
seta := convert(lista,set);
```

$$seta := \{x, y, t\}$$

```
listaa := convert(seta,list);
```

$$listaa := [x, y, t]$$

4. Convert between Array and a List

```
arraya := array([1,2,3,4]);
```

$$arraya := [1, 2, 3, 4]$$

```
type(arraya,list);type(arraya,array);
```

$$false$$

$$true$$

```
lista := convert(arraya,list);
```

$$lista := [1, 2, 3, 4]$$

```
type(lista,list);type(lista,array);
```

$$true$$

$$false$$

```
arrayaa := array(lista);
```

$$arrayaa := [1, 2, 3, 4]$$

```
type(arrayaa,list);type(arrayaa,array);
```

$$false$$

$$true$$

I am now very close to what I want. For example

```
restart;
p := 2: m:= 5:
smax := 2^(m-1)-1:
for s from 1 to smax by 2 do
  for n from 1 to m do            #create array e of exponents
    e[n] := (2^n*s) mod (2^m-1);
  od;
  eset := convert(e,set);        #convert array e into a set, dupes removed
  if gcd(s,31) = 1 then flag := "primitive" else flag := "  " fi;
od;
```

$$eset := \{1, 2, 4, 8, 16\}$$
$$eset := \{3, 6, 12, 17, 24\}$$
$$eset := \{5, 9, 10, 18, 20\}$$
$$eset := \{7, 14, 19, 25, 28\}$$
$$eset := \{5, 9, 10, 18, 20\}$$
$$eset := \{11, 13, 21, 22, 26\}$$
$$eset := \{11, 13, 21, 22, 26\}$$
$$eset := \{15, 23, 27, 29, 30\}$$

It only remains to somehow throw out the duplicate sets.

OK, I have now advanced a bit to this point:

```
first := true;
for s from 1 to smax by 2 do
  for n from 1 to m do            #create array e of exponents
    e[n] := (2^n*s) mod (2^m-1);
  od;
  eset := convert(e,set);        #convert array e into a set, dupes removed
  if first = true then seta := {eset};first := false;
    else eset1 := {eset}; seta := seta union eset1; fi
  #seta := seta union eset;
  #if gcd(s,31) = 1 then flag := "primitive" else flag := "  " fi;
od:
```

$$first := true$$

```
seta;
```

$$\{\{5, 9, 10, 18, 20\}, \{15, 23, 27, 29, 30\}, \{11, 13, 21, 22, 26\}, \{3, 6, 12, 17, 24\}, \{1, 2, 4, 8, 16\}, \{7, 14, 19, 25, 28\}\}$$

I maintain a set of sets, and as a result, duplicate sets of exponents are automatically removed.

Now all that remains is to get a primitive p symbol in there somehow. OK, here we are

```
first := true;
for s from 1 to smax by 2 do
  for n from 1 to m do           #create array e of exponents
    e[n] := (2^n*s) mod (2^m-1);
  od;
  eset := convert(e,set);       #convert array e into a set, dupes removed
  if gcd(s,p^m-1) = 1 then eset := eset union {"p"};  fi;
  if first = true then seta := {eset};first := false;
    else eset1 := {eset}; seta := seta union eset1; fi
  #seta := seta union eset;
  #if gcd(s,31) = 1 then flag := "primitive" else flag := "  " fi;
od:
```

$$first := true$$

```
seta;
```

$$\{\{"p", 1, 2, 4, 8, 16\}, \{"p", 3, 6, 12, 17, 24\}, \{"p", 5, 9, 10, 18, 20\}, \{"p", 7, 14, 19, 25, 28\}, \{"p", 11, 13, 21, 22, 26\}, \{"p", 15, 23, 27, 29, 30\}\}$$

I find that the sets are not always sorted, so want to convert to a list and then I can use the sort command because you cannot sort a set.

I have moved a long quite a bit now. One bug is that Prog sometimes goes to the start of my list!

OK, it is now all done and I pronounce it good! It has been installed.