Subject: Set Operations on A and B
Posted by mxd1007 on Sun, 02 Oct 2005 03:30:22 GMT
View Forum Message <> Reply to Message

Hello

I'm doing a programming project in computational Graph Theory. I'm trying to implement an algorithm to color the vertices of a graph, a pentagon, and will be using a greedy coloring algorithm as detailed below:

```
 ********************************************************** ******************
GreedyColor( G = (Vertices, Edges))

global color

k = 0
 for i = 0 to n -1
    h = 0
    while h < k and A[i] INTERSECTION ColorClass[h] not eqaul to NULL
do h = h+1

    if h = k then
      k = k+1
      ColorClass[h] = NULL

    ColorClass[h] = ColorClass[h] UNION {i}
    color[i] = h

return(k)
 ********************************************************** ******************
```

Graph coloring is where you color the vertices of a graph with a particular color, using a different color for vertices that ARE connected to each other. So for a regular pentagon, you can apply at least 3 different colors(for set of vertices (0,1,2,3,4), 0 = red, 1 = blue, 2 = red, 3 = blue, 4 = green

the actual k-coloring is stored in the array color

ColorClass is defined as:
ColorClass[h]={i is an element of Vertices : color[i] = h}

UNION and INTERSECTION is of course the typical union and interesection relating to sets

I have no problem implementing the pseudocode as is except for the intersection and union computations. I did find a nice website on this

by Fanning Consulting:

http://www.dfanning.com/tips/set_operations.html

so I used the setuUnion and setIntersection funtions within my program.
when ready to compile, I got this error:

```
IDL> graphcolor
% HISTOGRAM: Expression must be an array in this context: A.
% Execution halted at: SETINTERSECTION    36
/Applications/rsi/graphcolor.pro
%                GRAPHCOLOR       76
/Applications/rsi/graphcolor.pro
%                $MAIN$
```

My question and confusion is about this A array since I'm passing an array(actually a sub array) into the function setIntersection. Or are these set functions useless for my application to graph coloring? any suggestions would be appreciated. I do think these set functions are very useful, just that I don't understand why it can't be applied to my arrays I'm passing in.

~Michael

My code as follows(I indicated with ** where the error happened in SetIntersection function):

```
FUNCTION SetUnion, a,b

   ;A union NULL = a
   IF a[0] LT 0 THEN RETURN, b

   ;B union NULL = b
   IF b[0] LT 0 THEN RETURN, a

   ;Return combined set
   RETURN, Where(Histogram([a,b], OMin = omin)) + omin

END



FUNCTION SetIntersection, a,b

;only need intersection of ranges
```

```
minab = Min(a, Max=maxa) > Min(b, Max=maxb)

maxab=maxa < maxb

;If either set is empty, or their ranges don't intersect:
;result = NULL

IF maxab LT minab OR maxab LT 0 THEN RETURN, -1
** r = WHERE((Histogram(a, Min=minab, Max=maxab) NE 0) AND $
        (Histogram(b, Min=minab, Max=maxab) NE 0), count)

IF count EQ 0 THEN RETURN, -1 ELSE RETURN, r + minab
END



PRO GRAPHCOLOR


;construct an array to hold the set of vertices adjacent
;to any particular vertex. for example for vertex 0,
; vertices 1 and 4 would be a set of vertices adjacent
;to vertex 0, hence a[0] = [1,4]

setA = [[1,4], [0,2], [1,3], [2,4], [0,3]]

;create a color array to hold colors of vertices
;in a undirected regular pentagon graph

color = BYTARR(5,1)
colorClass = BYTARR(5, 1)

k = 0

FOR i = 0, 4 DO BEGIN

   h = 0

   WHILE ( h < k AND setIntersection(setA[i], colorClass[h])$
         NE -1) DO BEGIN


      h = h + 1

      IF ( h EQ k) THEN BEGIN

         k = k+1
```

```
        colorClass[h] = NULL

      ENDIF

      colorClass[h] = setUnion(colorClass[h], i)

      color[i] = h

   ENDWHILE

ENDFOR

print, k

END
```

Subject: Re: Set Operations on A and B
Posted by mxd1007@cs.rit.edu on Mon, 03 Oct 2005 01:36:46 GMT
View Forum Message <> Reply to Message

I did fix the error, by deleting setA[i] in :

WHILE (h < k AND setIntersection(setA[i], colorClass[h])$

and replacing it with newSetA, where newSetA is defined right before
the WHILE loop:

```
newsetA = EXTRAC(setA, 0, i, 2, 1)
WHILE ( h < k AND setIntersection(setA[i], colorClass[h])$
...
....
....
```

however I get an error stating:

```
IDL> graphcolor
% HISTOGRAM: Expression must be an array in this context: B.
% Execution halted at: SETINTERSECTION    36
/Applications/rsi/graphcolor.pro
%                GRAPHCOLOR       76
/Applications/rsi/graphcolor.pro
%                $MAIN$
```

which refers to Colorclass[h]. So I guess since this is only a
numerical value, I need to somehow convert it to an array.
ColorClass[h] is the set of vertices with color h, and is constructed
as follows:

ColorClass[h] = {i is an element of the set of vertices : color[i] = h}
for 0 <= h <= k-1

Subject: Re: Set Operations on A and B
Posted by btt on Mon, 03 Oct 2005 14:57:49 GMT
View Forum Message <> Reply to Message

mxd1007@cs.rit.edu wrote:

> which refers to Colorclass[h]. So I guess since this is only a
> numerical value, I need to somehow convert it to an array.
> ColorClass[h] is the set of vertices with color h, and is constructed
> as follows:
>
> ColorClass[h] = {i is an element of the set of vertices : color[i] = h}
> for 0 <= h <= k-1
>

Hi,

My first guess is that IDL is changing your selection from a subset of
an array to a scalar. This happens when you extract one element from an
array. Use REFORM to force IDL to treat your value as an array of one
element.

IDL> ColorClass = BINDGEN(12)
IDL> i = 7

Now extract the ith element - raw and reformed...

IDL> help, ColorClass[i], REFORM(ColorClass[i],1)
<Expression> BYTE = 7
<Expression> BYTE = Array[1]

Try it with your set operators.

Cheers,
Ben

Subject: Re: Set Operations on A and B
Posted by mxd1007 on Sat, 08 Oct 2005 01:22:08 GMT
View Forum Message <> Reply to Message

Ben Tupper wrote:

> mxd1007@cs.rit.edu wrote:
>
>> which refers to Colorclass[h]. So I guess since this is only a
>> numerical value, I need to somehow convert it to an array.
>> ColorClass[h] is the set of vertices with color h, and is constructed
>> as follows:
>>
>> ColorClass[h] = {i is an element of the set of vertices : color[i] = h}
>> for 0 <= h <= k-1
>>
>
> Hi,
>
> My first guess is that IDL is changing your selection from a subset of
> an array to a scalar. This happens when you extract one element from an
> array. Use REFORM to force IDL to treat your value as an array of one
> element.
>
> IDL> ColorClass = BINDGEN(12)
> IDL> i = 7
>
> Now extract the ith element - raw and reformed...
>
> IDL> help, ColorClass[i], REFORM(ColorClass[i],1)
> <Expression> BYTE = 7
> <Expression> BYTE = Array[1]
>
> Try it with your set operators.
>
> Cheers,
> Ben

Thanks Ben

I changed the set of adjacent vertices to a larger graph, 11 vertices,
and the structure of the graph is as follows, where i is the vertex and
A[i] is the set of vertices adjacent to vertex i, however A[i] had to
be a struct instead of an array

setA = {vertex0:[1,5,8],vertex1:[0,2,7],vertex2:[,3,6,7},$
vertex3:[2,4,10],vertex4:[3,5,10],vertex5:[0,4,6,8],$
vertex6:[2,5,9],vertex7:[1,2,9,10],vertex8:[0,5,9,10],$
vertex9:[6,7,8,10],vertex10:[3,4,7,8,9]}

knowing that this graph can have at least 3 different colors on
vertices(a color must not be connected to the same color on any vertex,
so vertex 0 would be green, vertex 1 would be red, vertex 2 would be

blue, any adjacent vertices have to be of different colors) so I constructed an array called colorClass,

```
 colorclass=make_array(3,1, value=-1)
```

where initially this array has to be NULL or empty.

so to pass in colorClass[h] into the setIntersection function, I used the reform function,

```
k=0
FOR i = 0, 10 DO BEGIN
h = 0

;retrieve subarray from structure setA
newSetA = setA.(i)

sizeColorClass = SIZE(colorClass[h], /n_dimensions)

WHILE( h LT k AND SetIntersection(newSetA, REFORM(colorClass[h],$
      sizeColorClass) NE -1) DO.....
```

my question is, colorClass[h], h = 0,1,.....10 is going to change in size, perhaps colorclass[0] = {0,2,4}, colorClass[1] = {1,3,5,8}, colorClass[2] = {6,7,9,10} but the highest h will be is 2, so with that, am I passing in the correct numeric value for the reform fucntion for colorClass? I keep getting this error and trying to figure out how to pass colorClass[h] to my setIntersection function where colorClass[h] could be an empty set, a set with 1 elements, or 2 elements, or 3 elements, etc etc...

REFORM: New subscripts must not change the number elements in <INT Array[1]>.

Subject: Re: Set Operations on A and B
Posted by btt on Mon, 10 Oct 2005 13:46:16 GMT

View Forum Message <> Reply to Message

mxd1007@cs.rit.edu wrote:

```
>
>  setA = {vertex0:[1,5,8],vertex1:[0,2,7],vertex2:[,3,6,7},$
>         vertex3:[2,4,10],vertex4:[3,5,10],vertex5:[0,4,6,8],$
>         vertex6:[2,5,9],vertex7:[1,2,9,10],vertex8:[0,5,9,10],$
>         vertex9:[6,7,8,10],vertex10:[3,4,7,8,9]}
>
```

snip

>
> ;retrieve subarray from structure setA
> newSetA = setA.(i)
>
> sizeColorClass = SIZE(colorClass[h], /n_dimensions)
>
> WHILE(h LT k AND SetIntersection(newSetA, REFORM(colorClass[h],$
>   sizeColorClass) NE -1) DO.....
>

Hello,

Two things, I think, will help here.

(1) You probably want change the call to size so that it returns a vector of dimensions - not a scalar of the number of dimensions. I think your REFORM statement will work if you do since it will work with

 sizeColorClass = SIZE(colorClass[h], /dimensions)

(2) I would seriously be looking transforming this whole thing into an object system. Make each vertex an object. Then you have two options - have each vertex know something about its adjacency or have another type of object provide supervision/navigation for you. I did something similar with a quadtree object system a long time ago. Once I had my head wrapped around the object part - the quadtree navigation was a snap. And a bonus was waaaay less coding than that without objects.

Cheers,
Ben
