

CoCoA-5 - Feature #1372

New function: find ?

29 Nov 2019 17:21 - Anna Maria Bigatti

Status:	In Progress	Start date:	29 Nov 2019
Priority:	Normal	Due date:	
Assignee:	Anna Maria Bigatti	% Done:	10%
Category:	CoCoA-5 function: new	Estimated time:	0.00 hour
Target version:	CoCoA-5.4.2	Spent time:	1.00 hour
Description Should we add the function find (similar to C++ find)? This is a quick and easy prototype (should be easy to write a more efficient version in BuiltinFunctions) <pre>define find(f, L) for i := 1 to len(L) do if L[i] = f then return i; endif; endfor; return 0; enddefine; -- find</pre>			
Related issues: Related to CoCoA-5 - Feature #1535: New functions: argmin, argmax In Progress 11 Nov 2020			

History

#1 - 16 Jan 2020 12:17 - John Abbott

This is probably a very useful issue, but I prefer to add new functions after the next release (unless this fn is needed for the next release?)
Suggest: postpone.

#2 - 14 Feb 2020 17:35 - John Abbott

If the list L contains entries of different types then the fn defn given above might produce an error (since operator= will throw).

Perhaps the test L[j] = f should be inside a try block, so that exceptions do not escape...

Here is a failing example (I expect):

```
GF3 := NewZZmod(3);
GF5 := NewZZmod(5);
L := [one(GF3), one(GF5)];
find(one(GF5), L); --> error!
```

#3 - 15 Feb 2020 10:23 - John Abbott

find may also behave in a "slightly unexpected" way:

```
P := QQ[x,y];
L := [one(P), 1];
```

```
find(1, L); --> which index?
```

Since $\text{one}(P) = 1$ gives true, I presume the excerpt above will produce the index 1 (even though the value in position 2 is a "perfect match").

To be fair, the "troublesome" examples I have given arise from lists having mixed types of entry. Really the find function makes sense only for homogeneous lists (whose entries are "of the same type").

#4 - 17 Feb 2020 13:55 - John Abbott

I still prefer postponing this issue to the next version, so that we can discuss better the semantics, and when/if errors should be signalled.

If it is really needed for Vietnam, we can put in a first (KISS?) version.

#5 - 26 Feb 2020 17:19 - Anna Maria Bigatti

- Assignee set to Anna Maria Bigatti

- Target version changed from CoCoA-5.3.0 to CoCoA-5.4.0

#6 - 03 Mar 2020 22:20 - John Abbott

If we do add a find function, it may be nice to allow it to take a list and a predicate...

```
L := [3, 4, 5];
find(L, IsEven);
2
```

#7 - 19 Apr 2021 13:48 - John Abbott

An important feature of the C++ version of find is that it advances an iterator -- effectively it is "find next".

In CoCoA-5 we do not have iterators. If we want to imitate the C++ behaviour then probably the args should be a list and a start index. Or else we could have FindFirst and FindAll.

A design advantage to having find accept a predicate (fn which returns bool) is that it should be clearer to the user what the meaning of the example in comment 3 above is.

Also, I think the predicate version could delegate the responsibility of handling errors to the predicate: for instance

```
L := [2, "one"];
define Is1(X) return X=1; enddefine;
find(L, Is1); --> ERROR
define Is1_better(X) return type(X)=INT and X=1; enddefine;
find(L, Is1_better); --> returns 2
```

DISADVANTAGE of the predicate version is that it cannot be implemented in C++ (at least not easily).

#8 - 04 Nov 2021 23:01 - John Abbott

- Target version changed from CoCoA-5.4.0 to CoCoA-5.4.2

#9 - 04 Nov 2021 23:01 - John Abbott

- Related to Feature #1535: New functions: *argmin*, *argmax* added