CoCoA-5 - Feature #1112

New function: IsEmpty

27 Oct 2017 11:33 - Anna Maria Bigatti

Status:	Closed	**Start date:**	27 Oct 2017
Priority:	Normal	**Due date:**	
Assignee:	Anna Maria Bigatti	**% Done:**	100%
Category:	CoCoA-5 function: new	**Estimated time:**	2.00 hours
Target version:	CoCoA-5.4.2	**Spent time:**	2.20 hours

Description

add function IsEmpty(L) for LIST.

Related issues:

Related to CoCoA-5 - Slug #31: theValue makes copy	**In Progress**	**15 Nov 2011**

History

#1 - 15 May 2020 10:29 - Anna Maria Bigatti

- Target version changed from CoCoA-5.?.? to CoCoA-5.4.0

#2 - 18 Feb 2021 22:18 - John Abbott

- Status changed from New to Resolved

- % Done changed from 0 to 60

I have just added a first impl to BuiltInFunctions.C.
A couple of trivial tests passed.

Still need to: write tests, write doc, modify existing tests/packages (to use IsEmpty instead of comparing to []).

#3 - 19 Feb 2021 09:50 - John Abbott

The function **IsEmpty** is slower than explicitly testing for equality to the empty list; why should that be?

Here is the simple test I conducted:

```
/**/ L := 1..100;
/**/ t0 := CpuTime();  for j := 1 to 1000000 do if IsEmpty(L) then println "*"; endif; endfor; TimeFrom(t0);
0.500
/**/ t0 := CpuTime();  for j := 1 to 1000000 do if L = [] then println "*"; endif; endfor; TimeFrom(t0);
0.390
```

I tried the test with L := 1..1000000; and the timings were essentially the same.
I suppose I shall have to "play" with the profiler... groan!

#4 - 19 Mar 2021 12:04 - Anna Maria Bigatti

I made a few tests, and the timings appear to be very similar.
It seems that creating a LIST with the [..] syntax is amazingly cheap:

```
/**/ t0 := CpuTime();  for j := 1 to 1000000 do skip; endfor; TimeFrom(t0);
0.162 ~ 0.170
t0 := CpuTime();  for j := 1 to 1000000 do if IsEmpty(L) then println "*"; endif; endfor; TimeFrom(t0);
0.662 ~ 0.682
/**/ t0 := CpuTime();  for j := 1 to 1000000 do if L = [] then println "*"; endif; endfor; TimeFrom(t0);
0.628 ~ 0.676
/**/ t0 := CpuTime();  for j := 1 to 1000000 do if L = [9,8] then println "*"; endif; endfor; TimeFrom(t0);
0.869 ~ 0.951
```

creating [9,8] costs next to nothing.

#5 - 28 May 2021 20:42 - John Abbott

- Related to Slug #31: theValue makes copy added

#6 - 28 May 2021 20:42 - John Abbott

Is this in some way related to issue #31?
If not, delete the link I have just created above.

#7 - 29 May 2021 18:15 - Anna Maria Bigatti

John Abbott wrote:

> Is this in some way related to issue #31?
> If not, delete the link I have just created above.

timings are the same, but it's worth thinking a bit longer

```
/**/ t0 := CpuTime();  for j := 1 to 1000000 do skip; endfor; TimeFrom(t0);
0.172
/**/ t0 := CpuTime();  for j := 1 to 1000000 do if IsEmpty(L) then println "*"; endif; endfor; TimeFrom(t0);
0.640
/**/ t0 := CpuTime();  for j := 1 to 1000000 do if L = [] then println "*"; endif; endfor; TimeFrom(t0);
0.603
/**/ t0 := CpuTime();  for j := 1 to 1000000 do if L = [9,8] then println "*"; endif; endfor; TimeFrom(t0);
0.881
```

#8 - 03 Feb 2022 20:04 - John Abbott

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

#9 - 14 Mar 2023 23:01 - John Abbott

- % Done changed from 60 to 80

Mmm, so what is the conclusion here?
Originally I had hoped/expected that a simple built-in function would be faster (and allocate less memory) than repeatedly creating an empty list. Instead, somehow the function call overhead seems to be greater than any gain.

In the end the difference in speed is pretty unimportant (just rather surprising).

So the real question is: which way gives clearer, more readable code?

```
  if L = [] then ... endif
  if IsEmpty(L) then ... endif
```

Maybe we should also consider a more complicated situation:

```
  if ComplicatedFn(many, args) = [] then ... endif
  if IsEmpty(ComplicatedFn(many, args)) then ... endif
```

Or even (a weird-looking solution)

```
  if [] = ComplicatedFn(many, args) then ... endif
```

It is hard to say. I'm pretty used to **= []** but what about newcomers?

Footnote but how often would one do **IsEmpty(ComplicatedFn(many,args))**?
If it returns a list, we probably want the contents of the list if its not empty.

#10 - 14 Mar 2023 23:07 - John Abbott

Just out of curiosity I made yet another test:

```
EMPTY := [];
t0 := CpuTime();  for j := 1 to 1000000 do if L = EMPTY then println "*"; endif; endfor; TimeFrom(t0);
```

It was almost as "slow" as IsEmpty.

#11 - 15 Mar 2023 06:45 - Anna Maria Bigatti

And also:

```
/**/ t0 := CpuTime();  for j := 1 to 1000000 do if IsEven(N) then println "*"; endif; endfor; TimeFrom(t0);
0.659
/**/ t0 := CpuTime();  for j := 1 to 1000000 do if IsEmpty(L) then println "*"; endif; endfor; TimeFrom(t0);
0.630
```

Now I think the cost is in "function call (variable)"

#12 - 15 Mar 2023 06:56 - Anna Maria Bigatti

- Status changed from Resolved to Feedback

- % Done changed from 80 to 90

In conclusion:
there is no advantage in IsEmpty(L) compared to L = [].
And there is no advantage in removing IsEmpty just because it's not as good as I thought was going to be.

So I would close this issue keeping it, but not encouraging it over "L = []".
I make a minor change in the manual for IsEmpty.

#13 - 22 Dec 2023 19:21 - John Abbott

- Status changed from Feedback to Closed

- % Done changed from 90 to 100

The manual entry looks OK to me; what do you think, Anna?
Each user can decide whether they prefer IsEmpty(L) or L = [].
Closing.