

Prolog Garbage Collection

Unknown macro: {style}

rite

Unknown macro: {font-family}

Prolog programmers do not explicitly manage memory themselves; an automated garbage collector periodically reclaims memory for the program's use.

New memory is allocated whenever new variables are introduced. The programmer does not have to specify that new memory is allocated.

You can see how many garbage collections it has performed with the clause `statistics`. This has several "compatibility keys," including `—garbage_collection`. We timed and checked garbage collection statistics on a simple recursive function which counts down to zero. With different starting values inserted for `<n>`, we get varied results.

```
run(D,A):- between(1,10,X),
            a(<n>,[ ]),
            statistics(runtime,D),
            statistics(garbage_collection,A),
            write(D), nl, write(A), nl, fail.
a(L,_):- L > 0, K is L - 1, a(K,foo(1,2,3,4,5)).
a(L,_):- L =< 0.
```

The following table has values for each call to `—a` for different starting values.

<code><n></code>	Time, ms	# of garbage collections	Time in GC, ms
100	0	0	0
1000	0	0	0
10000	1190	1	850
100000	20000	16	4000

Unknown macro: {table}