

## Topic 31: The Compiler Design Handbook - Chapter 6 (GarbageCollection Techniques)

Lukáš Macko, Stanislav Melo

### Incremental and concurrent collectors

Garbage collector is a form of memory management, that reclaims memory occupied by objects no longer used in the program. Main program is suspended during collection. This behaviour is not desirable in most of applications. Some of them need guaranteed bound on the length of pause.

That is where incremental and concurrent collectors are used.

Incremental collector runs on the same thread as main program and their execution is interleaved. Concurrent collector runs on separate thread.

Besides concurrent or incremental approach they can be divided into moving and nonmoving collectors. Moving collectors divide the heap into two parts, and copy reachable objects from one part to another. This approach prevents fragmentation problems, but it is more complicated. Nonmoving collectors do not move or copy reachable objects.

The main problem these collectors face is, that program may change object, already traced by collector, that it points to untraced object. Therefore the untraced object will not be traced and will be collected as garbage. That is because collector can be interrupted by program at any time changing some objects. Both approaches deal with this in different ways. Nonmoving collectors can be based on *snapshot-at-beginning* or *incremental update* method. These methods use write-barriers to ensure correctness in different ways. Moving collectors are based on *Baker's copying method*, *Appel-Ellis-Li collector* or *Nettles' replicating collector*. These methods use read-barriers or virtual-memory page protection in addition to write-barriers.

In the presentation, we will give an introduction to the tricolor abstraction for indicating the state of tracing the objects. We will demonstrate the problem of collecting objects, that program still needs, on example. Then, we will discuss *snapshot-at-beginning* and *incremental update* methods and how they solve this problem. At the end, we will focus on Baker's copying method. We will describe the concepts of the method, how it prevents the collecting of reachable objects and its drawbacks.