

**Safe and efficient storage allocation for iterators**

R A Finkel and H Schmidt

**ABSTRACT**

Iterators are control structures in programming languages that generalise **for** loops for Algol 60. They are like functions, but they may suspend on return and resume on subsequent activation, maintaining state between activations and generating a sequence of return values used by successive iterations of a generalised loop construct. Iterators have aspects of coroutines and streams. Multi-iterators support traversing several data structures simultaneously in one loop and have traditionally been implemented by heap allocation of frames.

This paper studies various forms of iterators from a programming language semantics and implementation viewpoint. We discuss runtime storage for the frames of iterators, showing how heap allocation can always be avoided, albeit with a potential cost in dead frames on the stack frames and prove their safety.

These results apply both to pure iterators and to the more general and novel kinds of iterators that we introduce.