Separation logic, originally developed by O’Hearn and Reynolds [1], is an extension of Hoare logic originally intended for reasoning about programs that use shared mutable data structures. It was based on the concept of separating conjunction, which permits the concise expression of aliasing constraints.

The logic also includes a “frame rule”, which enables local reasoning that is the key to the scalability of proofs. Examples of nontrivial proofs include the Schorr-Waite marking algorithm [2] and the Cheney relocating garbage collector [3].

More recently, by generalizing the concept of storage access to owner-ship and permissions, the logic has been extended to encompass information hiding [4], shared-variable concurrency [5], and numerical permissions [6].

We will survey the current development of separation logic, including, as time permits, extensions to unrestricted address arithmetic, dynamically allocated arrays, recursive procedures, shared-variable concurrency, and read-only sharing. We will also discuss promising future directions.

References


