The progression of multi-core processors has inspired the development of concurrency libraries that guarantee safety and liveness properties of multiprocessor applications. The difficulty of reasoning about safety and liveness properties in a concurrent environment has led to the development of tools to verify that a concurrent data structure meets a correctness condition or progress guarantee. However, these tools possess shortcomings regarding the ability to verify a composition of data structure operations. Additionally, verification techniques for transactional memory evaluate correctness based on low-level read/write histories, which is not applicable to transactional data structures that use a high-level semantic conflict detection. In my dissertation, I present tools for checking the correctness of multiprocessor programs that overcome the limitations of previous correctness verification techniques. Correctness Condition Specification (CCSpec) is the first tool that automatically checks the correctness of a composition of concurrent multi-container operations performed in a non-atomic manner. Transactional Correctness tool for Abstract Data Types (TxCâ€"ADT) is the first tool that can check the correctness of transactional data structures. TxCâ€"ADT elevates the standard definitions of transactional correctness to be in terms of an abstract data type, an essential aspect for checking correctness of transactions that synchronize only for high-level semantic conflicts. Many practical concurrent data structures, transactional data structures, and algorithms to facilitate non-blocking programming all incorporate helping schemes to ensure that an operation comprising multiple atomic steps is completed according to the progress guarantee. The helping scheme introduces additional interference by the active threads in the system to achieve the designed progress guarantee. Previous progress verification techniques do not accommodate loops whose termination is dependent on complex behaviors of the interfering threads, making these approaches unsuitable. My dissertation presents the first progress verification technique for non-blocking algorithms that are dependent on descriptor-based helping mechanisms.