Editorial

We bring the first **Progress in Computing Applications** issue with the papers below.

In the opening paper, "**Abstraction Techniques from Shape Analysis and Program Analysis**," the authors explored two abstractions: shape analysisand a technique derived from program analysis for data type reduction. We examined the relationship between shape analysis and data type reduction. The authors proved that the Abstraction techniques automatically build finite approximations of an infinite-state system.

In the next paper, "**Abstract data type analysis using shape invariants**," the authors studied how shape analysis can be used to analyzeabstract data types. They used the ADT as an example because it is commonly used and found in most major data types libraries, such as STL, Java API, and LEDA. Then, they formalised the idea of the ADT Set using algebras. Finally, they created a parametric shape analysis framework to perform the analyses that prove the compliance of these two implementations.

In the last paper, "**Code generation of compilers for application-specific processors**," the authors presented precise approaches for the general offset assignment problem that arises in the code generation phase of the address code generation of compilers for application-specific processors. The experiments helped study the effect of using more complex memory-addressing capabilities on the address calculation costs of real-world programs. They demonstrated how to incorporate operand reordering techniques for commutative instructions into existing solution approaches.

We will come out with more research in the next issues.

Editors