1 Background

This chapter provides background information for the work that follows throughout the paper. We first discuss the Silver attribute grammar system [**2**], which is used to implement the extensions to the host language. We follow this with a discussion of `ABLE`C [**7**], the extensible version of C utilized in this work.

1.1 Silver

Silver [**2**], created by Van Wyk et al. is an attribute grammar specification system. Furthermore, Silver is extensible, allowing us to add both general features (pattern matching, for instance) and domain-specific features to Silver. This gives us an attribute grammar specification system with a rich set of language features we can utilize in developing new extensions.

Silver has several nice features useful in generating new language extensions. First and foremost, Silver allows for *forwarding* [**3**] to implement new extensions in cost-effective ways. Forwarding allows language designers to utilize some form of inheritance within their language extensions, saving designers significant time and effort in creating new language features.

1.2 Extensible Programming and Able-C

One of the primary programming languages that is utilized in modern computing when speed or low-level control is vitally important is the C programming language. Unfortunately, C lacks many of the features of more modern programming languages, often making it cumbersome to work with in certain applications.

One way that some have tried to improve upon the C language is through the use of extensions. Writing extensions to the C language, however, can be quite difficult, often involving many complications. For instance, the Cilk extension was introduced to C to allow

for easier parallel programming. However, the original implementation of Cilk5 utilized its own type-checker, despite not changing the underlying C type system [**1**, **p.14**].

This difficulty in extending the C language was one of the motivations behind creating the `ABLE`C language. Built utilizing Silver, `ABLE`C is an extensible C pre-processor, conforming to the C11 standard [**7**]. It takes an "extended" version of C and translates it back into plain C, performing transformations and analyses as it does so.