Certified Parsing

Background

Parsing is the act of transforming plain text into some structure that can be analysed by computers for further processing. One might think that parsing has been studied to death and after *yacc* and *lex* no new results can be obtained in this area. However recent results and novel approaches make it increasingly clear, that this is not true anymore.

We propose to approach the subject of parsing from a certification point of view. Parsers are increasingly part of certified compilers, like CompCert, which are guaranteed to be correct and bug-free. Such certified compilers are important in areas where software just cannot fail. However, so far the parsers of these compilers have been left out of the certification. This is because parsing algorithms are often adhoc and their semantics is not clearly specified. Unfortunately, this means parsers can harbour errors that potentially invalidate the whole certification and correctness of the compiler. In this project, we like to change that.

Only in the last few years, theorem provers have become good enough for establishing the correctness of some standard lexing and parsing algorithms. For this, the algorithms need to be formulated in way so that it is easy to reason about them. In earlier work about lexing and regular languages, the authors showed that this precludes algorithms working over graphs. However regular languages can be formulated and reasoned about entirely in terms regular expressions, which can be easily represented in theorem provers. This work uses the device of derivatives of regular languages. We like to extend this work to parsers and grammars. The aim is to come up with elegant and useful parsing algorithms whose correctness and absence of bugs can be certified in a theorem prover.