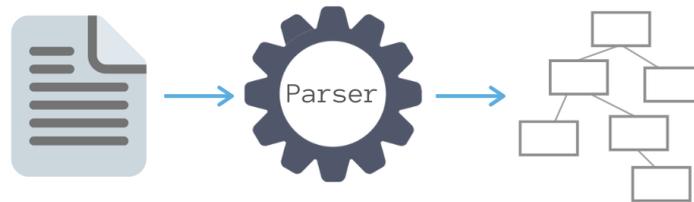




## Multi-Machine Parsing

What if your next parser was a graph of parsing machines working together?



If you've ever written a parser you must be thinking "Wait, this far from revolutionary..."

The truth is that even though people writing parsers by hand have been doing this for ages, parser generators are yet to catch up. In fact, some people argue that it is precisely the inflexibility of parser generators that forces them to write all their parsers by hand, which is an arduous and repetitive task that gets boring second time round you're doing it. That is, unless you're paid by the line.

There have been parser generators that introduced a series of semantic hacks to overcome the most common limitations of parsing machines, such as the ANTLR project, but a definitive solution is yet to be proposed.

The goal of this thesis is to continue the development of a parser generator that takes extended grammar as input and produces a parser utilizing multiple parsing machines – algorithms, if you like – at once. A first version of this parser generator has already been developed in FS22. You could either extend on this project (in C++) or start from ground up in the language of your choice.

There is a conceptual level to this project, and then there is the practical part.

- *Bachelor's* students will work on a smaller project mostly on the practical side.
- *Master's* students will work on an larger project, both coding and advancing our work on the higher level of abstraction.

**Candidate Profile.** A good candidate for the project is interested in programming languages, parsing, or compiler design. He/She is further a competent programmer in their best language.

**Interested? Please contact us to learn more!**

### Contact

- Peter Belcak: [belcak@ethz.ch](mailto:belcak@ethz.ch), ETZ G61.3