How to implement a new logic language

- The *quickest* way to implement a new logic language is to write an interpreter in Prolog (e.g., Concurrent Prolog, Still-Life, Timed Gentzen, ...).

- There are several ways to get an *efficient* implementation:

    1. Extend the WAM. This technique has been the most popular (e.g., CLP(R), CLP(FD), ...). It is not the right way for languages very different from Prolog (e.g., λ-Prolog).

    2. Compile the language into an existing implementation of Prolog (e.g., QD-Janus). Modern implementations of Prolog are fast and support advanced control constructs such as coroutining (e.g., SICStus) and first-class suspensions (e.g., ECLiPSe).

    3. If the above is difficult, then try to compile the language into a system that provides useful primitives. For example, λ-Prolog has been implemented with MALI, a memory management library tailored for logic programming.

- To get the *fastest possible* implementation, it is necessary to compile directly to a lower-level language (e.g., C or assembly).