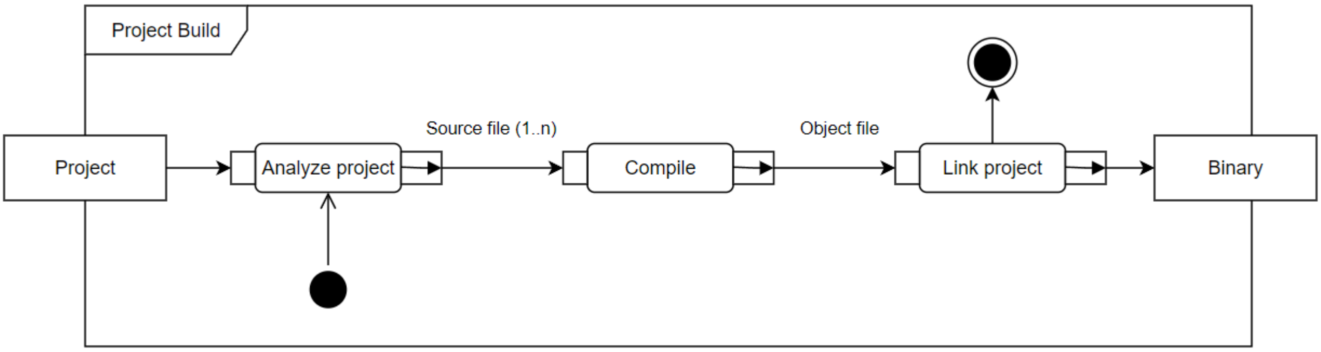
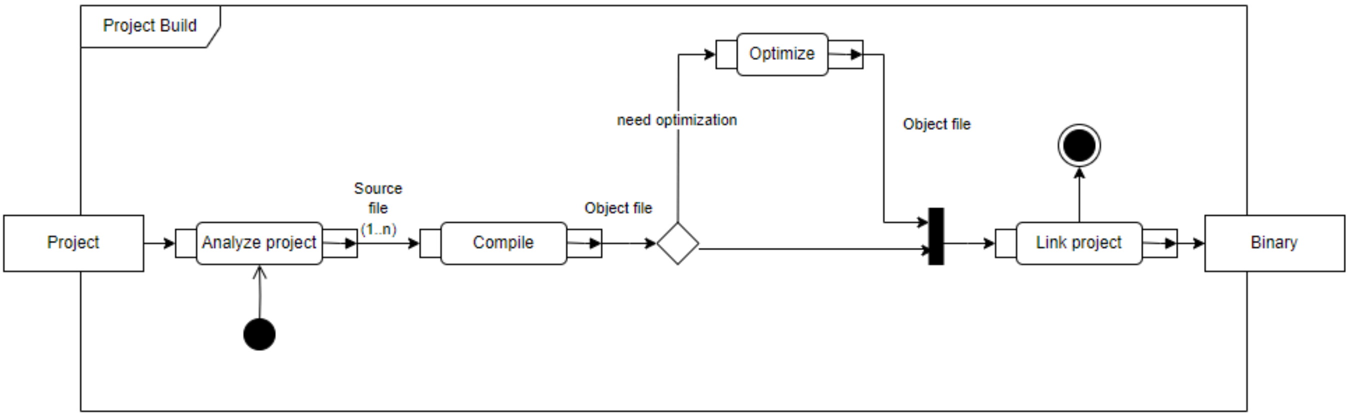
A



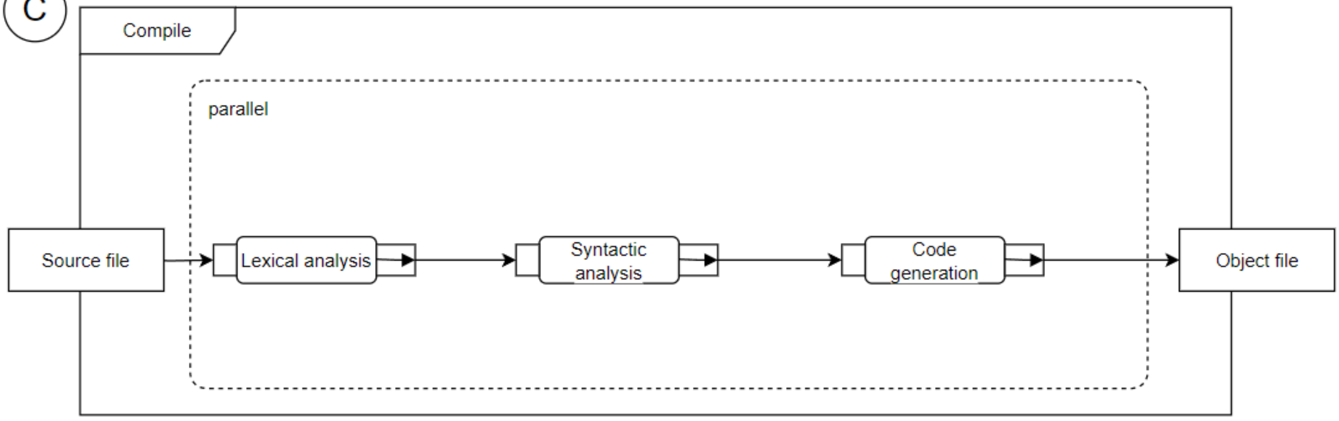
A - The start point in this diagram is the action “Analyze project”, an object “Project” is shown as a rectangle and if “flows” to the start point. As a result of “Analyze project” there are one or more source files and they go to “Compile” action. After compiling source files we have an object file, so it is a result of “Compile” action. After that the flow of control from “Compile” action goes to “Link project”, which is the final node, so here any actions stop. The result of “Link project” is an object called “Binary”.

B



B - In this part of the task we have optional action “Optimize”, so there is a decision node (diamond shape), which leads to “Optimize”, if we need optimization, otherwise to “Link project” action. The result of “Optimize” is also an object file. In the end we need to join control flows as they lead to the same action “Link project”.

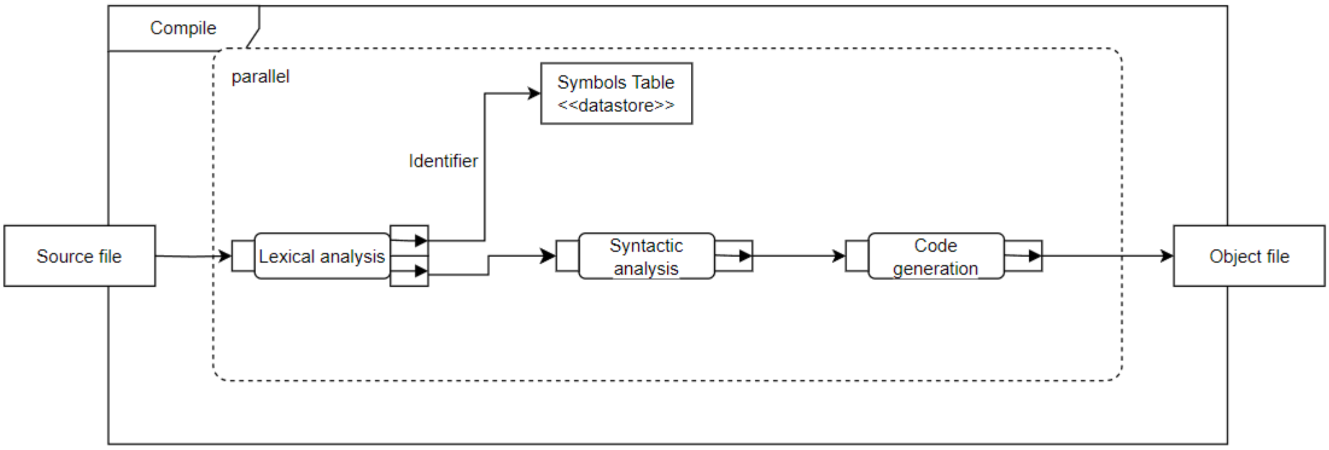
C



C - Here we expand “Compile”, so we have an expansion region with “parallel” on the top. “Lexical analysis” action gets the source file as an entrance, after that control flows to

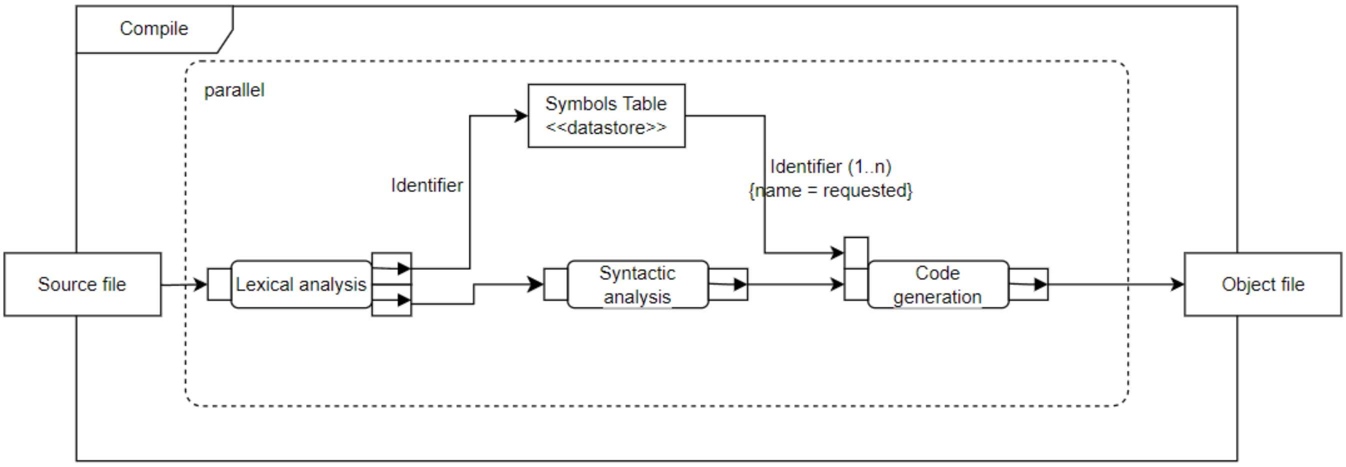
“Syntactic analysis” and after the “Code generation” action. As a result of completing the final there is an “Object file”.

D



D - While “Lexical analysis” is launched action “Symbol table” is completing, so there is an object “Symbol Table”. It is not said that “Symbol table” goes somewhere, so there is no arrows leading from it.

E



E - In this task identifiers (there can be one or more) from “Symbols Table” go to “Code generation” action. Only requested names could be passed to “Code generation”.

F - Number of tokens depends on many things, for instance, on programming language, size of the program or its complexity. So it is extremely difficult to say the exact number of tokens.