

Gdb CS402

Regular Expressions applications in the creation of Compilers for Programming Languages

In programming, a regular expression is an expression that explains a pattern for a string. A string matches a regular expression if that string follows the pattern of that regular expression.

For example, you may want to create an account system, allowing usernames to only have uppercase and lowercase letters, and numbers. While a user is registering, you can check their desired username against a regular expression involving only alphanumeric characters (A-Z, a-z, 0-9). If it matches, then the username is valid to your requests. If it does not, the user has put in a character that does not follow the pattern of the regular expression.

In regular expressions, you can match certain characters, match a certain quantity of characters, match the casing of characters (or just ignore it overall), and plenty more.

A **compiler** is a computer program (or set of programs) that transforms source code written in a programming language (the *source language*) into another computer language (the *target language*, often having a binary form known as *object code*). The most common reason for wanting to transform source code is to create an executable program.

The name "compiler" is primarily used for programs that translate source code from a high-level programming language to a lower level language (e.g., assembly language or machine code). If the compiled program can run on a computer whose CPU or operating system is different from the one on which the compiler runs, the compiler is known as a cross-compiler. A program that translates from a low level language to a higher level one is a *decompiler*. A program that translates between high-level languages is usually called a *language translator*, *source to source translator*, or *language converter*. A *language rewriter* is usually a program that translates the form of expressions without a change of language.

A compiler is likely to perform many or all of the following operations: lexical analysis, preprocessing, parsing, semantic analysis (Syntax-directed translation), code generation, and code optimization.