QL language reference

Learn all about QL, the powerful query language that underlies the code scanning tool CodeQL.

- About the QL language: QL is the powerful query language that underlies CodeQL, which is used to analyze code.
- Predicates: Predicates are used to describe the logical relations that make up a QL program.
- Queries: Queries are the output of a QL program. They evaluate to sets of results.
- Types: QL is a statically typed language, so each variable must have a declared type.
- Modules: Modules provide a way of organizing QL code by grouping together related types, predicates, and other modules.
- Signatures: Signatures provide a typing mechanism to parameters of parameterized modules.
- Aliases: An alias is an alternative name for an existing QL entity.
- Variables: Variables in QL are used in a similar way to variables in algebra or logic. They represent sets of values, and those values are usually restricted by a formula.
- Expressions: An expression evaluates to a set of values and has a type.
- Formulas: Formulas define logical relations between the free variables used in expressions.
- Annotations: An annotation is a string that you can place directly before the declaration of a QL entity or name.
- Recursion: QL provides strong support for recursion. A predicate in QL is said to be recursive if it depends, directly or indirectly, on itself.
- Lexical syntax: The QL syntax includes different kinds of keywords, identifiers, and comments.
- Name resolution: The QL compiler resolves names to program elements.
- Evaluation of QL programs: A QL program is evaluated in a number of different steps.
- QL language specification: A formal specification for the QL language. It provides a comprehensive reference for terminology, syntax, and other technical details about QL.