Match expressions

A matchexpression is used for pattern matching, it allows you to match a value against a set of patterns and execute code based on the matching pattern.

Syntax

A match expression starts with thekeyword match followed by an expression and a list of arms. Each arm begins with apattern which is followed by the⇒ operator, an expression, and a comma. Ablock expression is typically used if the logic in the arm is more than a single expression. The pattern in a given arm may introducevariables that can be used in expression of the said arm. If the pattern contains variables that are unused in the expression, then the wildcard_ can be used in their place.

Semantics

A match expression is evaluated by evaluating the expression and then matching the value against the patterns in the arms in order. The first arm whose pattern matches the value of the expression is executed and the result of the expression in the arm is returned as the result of the match. All the arms that don’t break the control flow must evaluate to the same type. An arm breaks the control flow if itreturns orpanics .

Limitations

Currently two kinds of match expression are supported:

Match on an Enum.

match enum_var { variant_0(a, b, c) => { / *code* / } variant_1(_) => { / *code* / } ... variant_k(a, c) => { / *code* / } } Whereenum_var is an instance of someenum andvariant_0, variant_1 , ...,variant_k are all the variants of the said enum, order in the way that they were declared.

Match on an felt252.

match felt_var { 0 => { / *code* / } _ => { / *code* / } } Wherefelt_var is afelt252 and the match patterns are theliteral '0' and wildcard '_' (which matches any value).

7.11 If expressions ð§ 7.13 For loop expressions ð§