Method calls

A method call is an expression of the formexpr.method(arguments…) . This expression is roughly equivalent to thefunction call expressionmethod(expr, arguments…) . The differences between the two are: * The expression can be coerced to either a snapshot@expr * or a referenceref expr * , * depending on the method’s signature. * method * must be a trait function.

self and methods

Methods can be defined only in traits and impls. The self parameter in Cairo trait and impl functions represents a value of a specified type, allowing for a more convenient method-like syntax for that type. It must be the first parameter of the function. A function defined in such a way can be called as a method on the value: trait Display { fn display(self: T) -> Array; }

impl DisplayUsize of Display{ fn display(self: usize) -> Array { ... } }

fn bar>(value: T) { ... // Can be called as a method. let c = value.display(); ... }

Search locations for methods

When a method is called, the compiler will search for a trait containing a function with a matching name in the current scope. For each matching trait, it will search for an impl (as specified above). If no traits with relevant impls are found, or if multiple such traits are found, the compiler will throw an error.

7.6 Function calls 7.8 Member access expressions