



Figure 1: Interpretation of the call

Listing 1: Call to a function f

```

1  int f (args) {
    [...]
    return return_expr;
}

6  int main () {
    ...
    v = f (exprs);
    ...
}

```

Interpretation of a Function Call within EVA

Step by Step

Figure 1 outlines each stage of the interpretation of a function call. The magenta lines highlight the functions provided by an abstract domain and dedicated to function calls. The others computations are performed by the standard transfer functions for the evaluation of expressions and interpretation of statements. The teal edges refer to actions written in `transfer_stmt.ml`, while the purple edges refer to actions written in `partitioned_dataflow.ml`. The left line gathers the computations carried out at the call site, while the right line gathers the computations carried out in the body of the called function.

The stages of the interpretation of a function call are as follows:

1. The concrete arguments $exprs$ are evaluated at the call site.
2. `start_call` builds the state at the entry point of function f . This steps includes the entry in scope of the formal arguments $f_{formals} = args$, and their instantiation with the value of the concrete arguments (evaluated at step 1).
3. The local variables of f enter in scope.
4. Standard dataflow analysis of f up to the state inferred before the return statement of f .
5. The special variable RET enters in scope.
6. The special variable RET is assigned to the return value, through a standard assignment.
7. The formal arguments $args$ of the function are evaluated, and the resulting values v_{args} are stored.
8. The formal and local variables leave the scope.
9. `finalize_call` merges the state at the call site and the state at the end of the called function.
10. At the call site, the concrete arguments are reduced to the values v_{args} , provided that $args = exprs$ at the end of the function f . This condition holds if the values of $args$ and $exprs$ have not been modified by f .
11. The assignment of the return value of f , through a regular assignment of the value of RET .
12. The special variable RET leaves the scope.