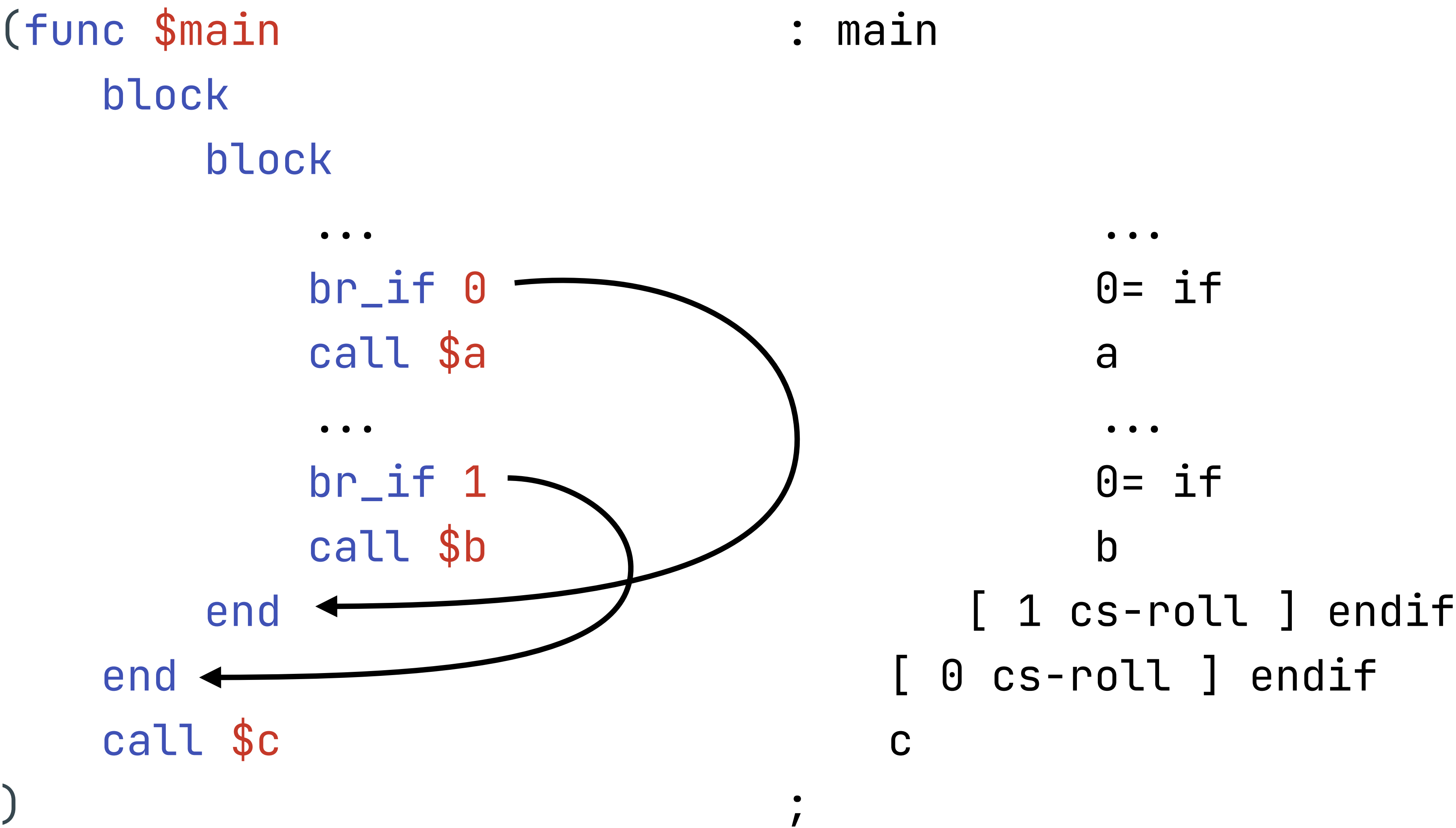


Kontrollfluss



Kontrollfluss mit Schleifen

```
(func $main                                : main
  loop                                     begin
    block
      ...
      br_if 0
      call $a
      ...
      br_if 1
      call $b
    end
  end
  call $c
)                                           ;
```

The diagram illustrates control flow loops in a code snippet. The code is written in a functional style, with function definitions in parentheses and calls in semicolons. The `loop` block contains two `br_if` instructions that branch back to the `loop` label, creating a loop. The first `br_if 0` branches back to the `loop` label, and the second `br_if 1` branches back to the `loop` label. The code is organized into a `block` within the `loop` function, and the `main` function is defined as `(func $main)`. The `begin` and `end` keywords are used to delimit the `loop` block. The `call $a` and `call $b` instructions are used to call functions `$a` and `$b` respectively. The `call $c` instruction is used to call function `$c` after the `loop` block. The `cs-roll` and `cs-drop` instructions are used to roll and drop the stack. The `cs-pick` instruction is used to pick the stack. The `until` keyword is used to indicate the end of the `loop` block. The `endif` keyword is used to end the `if` statement. The `end` keyword is used to end the `loop` block. The `end` keyword is used to end the `main` function. The `;` semicolon is used to end the code snippet.