Section 12:

Peephole Optimization

CS 164 @ UC Berkeley, Fall 2023

Follows Chapter 8.7 of *Compilers: Principles, Techniques, and Tools* (the "Dragon Book") closely.

AST → AST

```
(let
  ((time-limit
    (* 5 (* seconds-per-minute milliseconds-per-second))))
  ...)
```

AST → AST

```
(let
  ((time-limit
    (* 5 (* seconds-per-minute milliseconds-per-second))))
  ...)
```

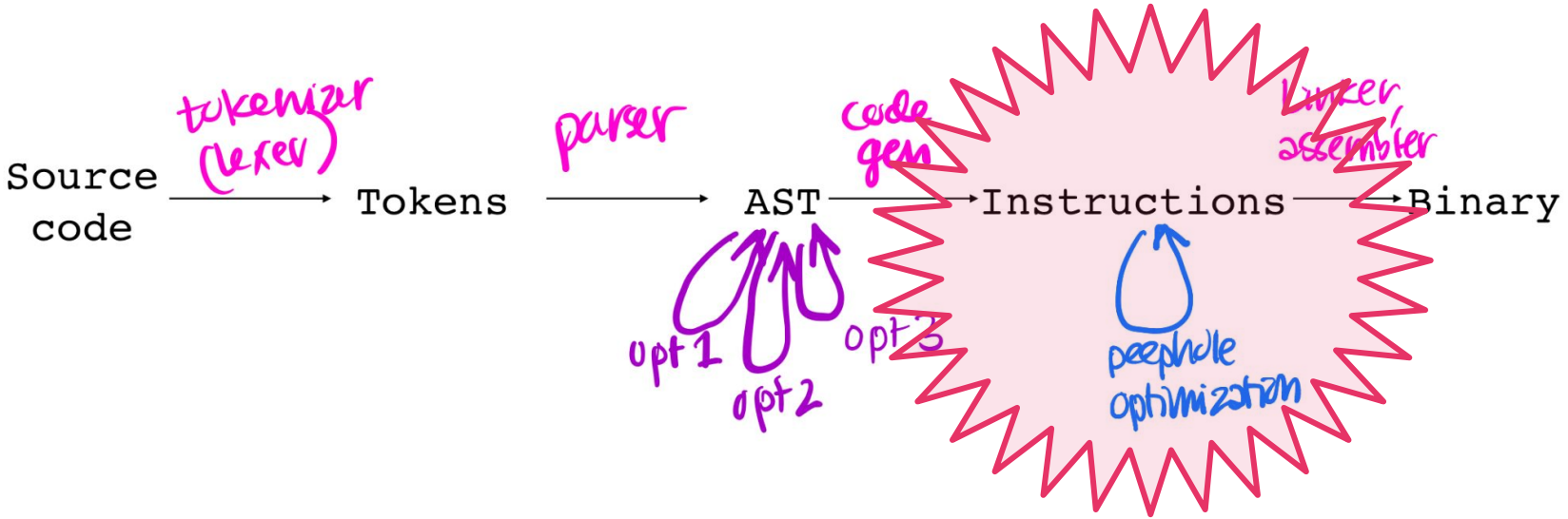
```
(let
  ((time-limit
    300000))
  ...)
```

~~AST → AST~~ Instruction → Instruction?

```
(let
  ((time-limit
    (* 5 (* seconds-per-minute milliseconds-per-second))))
  ...)
```

```
(let
  ((time-limit
    300000))
  ...)
```

Peephole Optimization (Instruction → Instruction)



```
default rel
global _lisp_entry
extern _lisp_error
extern _read_num
extern _print_value
extern _print_newline
    section .text
_function_f_18913907:
    mov rax, 12
    mov QWORD [-8 + rsp], rax
    mov rax, 16
    cmp QWORD [-8 + rsp], rax
    mov rax, 0
    setl al
    shl rax, 7
    or rax, 31
    cmp rax, 31
    je _else__1
_then__0:
    mov rax, 12
    mov QWORD [-8 + rsp], rax
    mov rax, 16
    cmp QWORD [-8 + rsp], rax
    mov rax, 0
    setl al
    shl rax, 7
    or rax, 31
    cmp rax, 31
    je _else__4
_then__3:
    mov rax, 28
    jmp _continue__5
    mov rax, r8
    mul rax, 1
    mov r8, rax
    ret
_else__4:
    mov rax, 28
_continue__5:
    jmp _continue__2
_else__1:
    mov rax, 28
_continue__2:
    mov QWORD [-8 + rsp], rax
    mov rax, 0
    add rax, QWORD [-8 + rsp]
    mov QWORD [-8 + rsp], rdi
    mov rdi, rax
    add rsp, -8
    call _print_value
    sub rsp, -8
    mov rdi, QWORD [-8 + rsp]
    mov rax, 159
    ret
_lisp_entry:
    add rsp, -8
    call _function_f_18913907
    sub rsp, -8
    ret
```

```
default rel
global _lisp_entry
extern _lisp_error
extern _read_num
extern _print_value
extern _print_newline
    section .text
_function_f_18913907:
    mov rax, 12
    mov QWORD [-8 + rsp], rax
    mov rax, 16
    cmp QWORD [-8 + rsp], rax
    mov rax, 0
    setl al
    shl rax, 7
    or rax, 31
    cmp rax, 31
    je _else__1
_then__0:
    mov rax, 12
    mov QWORD [-8 + rsp], rax
    mov rax, 16
    cmp QWORD [-8 + rsp], rax
    mov rax, 0
    setl al
    shl rax, 7
    or rax, 31
    cmp rax, 31
    je _else__4
_then__3:
    mov rax, 28
    jmp _continue__5
    mov rax, r8
    mul rax, 1
    mov r8, rax
    ret
_else__4:
    mov rax, 28
_continue__5:
    jmp _continue__2
_else__1:
    mov rax, 28
_continue__2:
    mov QWORD [-8 + rsp], rax
    mov rax, 0
    add rax, QWORD [-8 + rsp]
    mov QWORD [-8 + rsp], rdi
    mov rdi, rax
    add rsp, -8
    call _print_value
    sub rsp, -8
    mov rdi, QWORD [-8 + rsp]
    mov rax, 159
    ret
_lisp_entry:
    add rsp, -8
    call _function_f_18913907
    sub rsp, -8
    ret
```

```
default rel
global _lisp_entry
extern _lisp_error
extern _read_num
extern _print_value
extern _print_newline
    section .text
_function_f_18913907:
    mov rax, 12
    mov QWORD [-8 + rsp], rax
    mov rax, 16
    cmp QWORD [-8 + rsp], rax
    mov rax, 0
    setl al
    shl rax, 7
    or rax, 31
    cmp rax, 31
    je _else__1
```

```
_then__0:
    mov rax, 12
    mov QWORD [-8 + rsp], rax
    mov rax, 16
    cmp QWORD [-8 + rsp], rax
    mov rax, 0
    setl al
    shl rax, 7
    or rax, 31
    cmp rax, 31
    je _else__4
_then__3:
    mov rax, 28
    jmp _continue__5
    mov rax, r8
    mul rax, 1
    mov r8, rax
    ret
```

```
_else__4:
    mov rax, 28
_continue__5:
    jmp _continue__2
_else__1:
    mov rax, 28
_continue__2:
    mov QWORD [-8 + rsp], rax
    mov rax, 0
    add rax, QWORD [-8 + rsp]
    mov QWORD [-8 + rsp], rdi
    mov rdi, rax
    add rsp, -8
    call _print_value
    sub rsp, -8
    mov rdi, QWORD [-8 + rsp]
    mov rax, 159
    ret
_lisp_entry:
    add rsp, -8
    call _function_f_18913907
    sub rsp, -8
    ret
```

```
default rel
global _lisp_entry
extern _lisp_error
extern _read_num
extern _print_value
extern _print_newline
    section .text
_function_f_18913907:
    mov rax, 12
    mov QWORD [-8 + rsp], rax
    mov rax, 16
    cmp QWORD [-8 + rsp], rax
    mov rax, 0
    setl al
    shl rax, 7
    or rax, 31
    cmp rax, 31
    je _else__1
_then__0:
    mov rax, 12
    mov QWORD [-8 + rsp], rax
    mov rax, 16
    cmp QWORD [-8 + rsp], rax
    mov rax, 0
    setl al
    shl rax, 7
    or rax, 31
    cmp rax, 31
    je _else__4
_then__3:
    mov rax, 28
    jmp _continue__5
    mov rax, r8
    mul rax, 1
    mov r8, rax
    ret
_else__4:
    mov rax, 28
_continue__5:
    jmp _continue__2
_else__1:
    mov rax, 28
_continue__2:
    mov QWORD [-8 + rsp], rax
    mov rax, 0
    add rax, QWORD [-8 + rsp]
    mov QWORD [-8 + rsp], rdi
    mov rdi, rax
    add rsp, -8
    call _print_value
    sub rsp, -8
    mov rdi, QWORD [-8 + rsp]
    mov rax, 159
    ret
_lisp_entry:
    add rsp, -8
    call _function_f_18913907
    sub rsp, -8
    ret
```

```
default rel
global _lisp_entry
extern _lisp_error
extern _read_num
extern _print_value
extern _print_newline
    section .text
_function_f_18913907:
    mov rax, 12
    mov QWORD [-8 + rsp], rax
    mov rax, 16
    cmp QWORD [-8 + rsp], rax
    mov rax, 0
    setl al
    shl rax, 7
    or rax, 31
    cmp rax, 31
    je _else__1
```

```
_then__0:
    mov rax, 12
    mov QWORD [-8 + rsp], rax
    mov rax, 16
    cmp QWORD [-8 + rsp], rax
    mov rax, 0
    setl al
    shl rax, 7
    or rax, 31
    cmp rax, 31
    je _else__4
_then__3:
    mov rax, 28
    jmp _continue__5
    mov rax, r8
    mul rax, 1
    mov r8, rax
    ret
```

```
_else__4:
    mov rax, 28
_continue__5:
    jmp _continue__2
_else__1:
    mov rax, 28
_continue__2:
    mov QWORD [-8 + rsp], rax
    mov rax, 0
    add rax, QWORD [-8 + rsp]
    mov QWORD [-8 + rsp], rdi
    mov rdi, rax
    add rsp, -8
    call _print_value
    sub rsp, -8
    mov rdi, QWORD [-8 + rsp]
    mov rax, 159
    ret
_lisp_entry:
    add rsp, -8
    call _function_f_18913907
    sub rsp, -8
    ret
```

```
default rel
global _lisp_entry
extern _lisp_error
extern _read_num
extern _print_value
extern _print_newline
    section .text
_function_f_18913907:
    mov rax, 12
    mov QWORD [-8 + rsp], rax
    mov rax, 16
    cmp QWORD [-8 + rsp], rax
    mov rax, 0
    setl al
    shl rax, 7
    or rax, 31
    cmp rax, 31
    je _else__1
```

```
_then__0:
    mov rax, 12
    mov QWORD [-8 + rsp], rax
    mov rax, 16
    cmp QWORD [-8 + rsp], rax
    mov rax, 0
    setl al
    shl rax, 7
    or rax, 31
    cmp rax, 31
    je _else__4
_then__3:
    mov rax, 28
    jmp _continue__5
    mov rax, r8
    mul rax, 1
    mov r8, rax
    ret
```

```
_else__4:
    mov rax, 28
_continue__5:
    jmp _continue__2
_else__1:
    mov rax, 28
_continue__2:
    mov QWORD [-8 + rsp], rax
    mov rax, 0
    add rax, QWORD [-8 + rsp]
    mov QWORD [-8 + rsp], rdi
    mov rdi, rax
    add rsp, -8
    call _print_value
    sub rsp, -8
    mov rdi, QWORD [-8 + rsp]
    mov rax, 159
    ret
_lisp_entry:
    add rsp, -8
    call _function_f_18913907
    sub rsp, -8
    ret
```

```
default rel
global _lisp_entry
extern _lisp_error
extern _read_num
extern _print_value
extern _print_newline
    section .text
_function_f_18913907:
    mov rax, 12
    mov QWORD [-8 + rsp], rax
    mov rax, 16
    cmp QWORD [-8 + rsp], rax
    mov rax, 0
    setl al
    shl rax, 7
    or rax, 31
    cmp rax, 31
    je _else__1
```

```
_then__0:
    mov rax, 12
    mov QWORD [-8 + rsp], rax
    mov rax, 16
    cmp QWORD [-8 + rsp], rax
    mov rax, 0
    setl al
    shl rax, 7
    or rax, 31
    cmp rax, 31
    je _else__4
_then__3:
    mov rax, 28
    jmp _continue__5
    mov rax, r8
    mul rax, 1
    mov r8, rax
    ret
```

```
_else__4:
    mov rax, 28
_continue__5:
    jmp _continue__2
_else__1:
    mov rax, 28
_continue__2:
    mov QWORD [-8 + rsp], rax
    mov rax, 0
    add rax, QWORD [-8 + rsp]
    mov QWORD [-8 + rsp], rdi
    mov rdi, rax
    add rsp, -8
    call _print_value
    sub rsp, -8
    mov rdi, QWORD [-8 + rsp]
    mov rax, 159
    ret
_lisp_entry:
    add rsp, -8
    call _function_f_18913907
    sub rsp, -8
    ret
```

```
default rel
global _lisp_entry
extern _lisp_
extern _read_
extern _print
extern _print
    section .
_function_f_1
    mov rax,
    mov QWORD
    mov rax,
    cmp QWORD
    mov rax,
    setl al
    shl rax,
    or rax, 3
    cmp rax, 31
    je _else__1
```

```
_then__0:
    mov rax, 12
    mov r8, rax
    ret
```

```
_else__4:
    mov rax, 28
_continue__5:
    jmp _continue__2
_else__1:
    mov rax, 28
_continue__2:
    mov QWORD [-8 + rsp], rax
    mov rax, 0
    add rax, QWORD [-8 + rsp]
    mov QWORD [-8 + rsp], rdi
    mov rdi, rax
    add rsp, -8
    call _print_value
    sub rsp, -8
    mov rdi, QWORD [-8 + rsp]
    mov rax, 159
    ret
_lisp_entry:
    add rsp, -8
    call _function_f_18913907
    sub rsp, -8
    ret
```


```
_continue__2:
    mov QWORD [-8 + rsp], rdi
    mov rdi, rax
```

```
default rel
global _lisp_entry
extern _lisp_error
extern _read_num
extern _print_value
extern _print_newline
    section .text
_function_f_18913907:
    mov rax, 12
    mov QWORD [-8 + rsp], rax
    mov rax, 16
    cmp QWORD [-8 + rsp], rax
    mov rax, 0
    setl al
    shl rax, 7
    or rax, 31
    cmp rax, 31
    je _else__1
_then__0:
    mov rax, 12
    mov QWORD [-8 + rsp], rax
    mov rax, 16
    cmp QWORD [-8 + rsp], rax
    mov rax, 0
    setl al
    shl rax, 7
    or rax, 31
    cmp rax, 31
    je _else__4
_then__3:
    mov rax, 28
    jmp _continue__5
    mov rax, r8
    mul rax, 1
    mov r8, rax
    ret
_else__4:
    mov rax, 28
_continue__5:
    jmp _continue__2
_else__1:
    mov rax, 28
_continue__2:
    mov QWORD [-8 + rsp], rax
    mov rax, 0
    add rax, QWORD [-8 + rsp]
    mov QWORD [-8 + rsp], rdi
    mov rdi, rax
    add rsp, -8
    call _print_value
    sub rsp, -8
    mov rdi, QWORD [-8 + rsp]
    mov rax, 159
    ret
_lisp_entry:
    add rsp, -8
    call _function_f_18913907
    sub rsp, -8
    ret
```

Activity:

Come Up With Peephole Optimizations!

Classic Peephole Optimizations

Classic Peephole Optimizations

- **Algebraic simplifications**
  - `x = x + 0` `x = x * 1`

```
default rel
global _lisp_entry
extern _lisp_error
extern _read_num
extern _print_value
extern _print_newline
    section .text
_function_f_18913907:
    mov rax, 12
    mov QWORD [-8 + rsp], rax
    mov rax, 16
    cmp QWORD [-8 + rsp], rax
    mov rax, 0
    setl al
    shl rax, 7
    or rax, 31
    cmp rax, 31
    je _else__1
_then__0:
    mov rax, 12
    mov QWORD [-8 + rsp], rax
    mov rax, 16
    cmp QWORD [-8 + rsp], rax
    mov rax, 0
    setl al
    shl rax, 7
    or rax, 31
    cmp rax, 31
    je _else__4
_then__3:
    mov rax, 28
    jmp _continue__5
    mov rax, r8
    mul rax, 1
    mov r8, rax
    ret
_else__4:
    mov rax, 28
_continue__5:
    jmp _continue__2
_else__1:
    mov rax, 28
_continue__2:
    mov QWORD [-8 + rsp], rax
    mov rax, 0
    add rax, QWORD [-8 + rsp]
    mov QWORD [-8 + rsp], rdi
    mov rdi, rax
    add rsp, -8
    call _print_value
    sub rsp, -8
    mov rdi, QWORD [-8 + rsp]
    mov rax, 159
    ret
_lisp_entry:
    add rsp, -8
    call _function_f_18913907
    sub rsp, -8
    ret
```

```
default rel
global _lisp_entry
extern _lisp_error
extern _read_num
extern _print_value
extern _print_newline
    section .text
_function_f_18913907:
    mov rax, 12
    mov QWORD [-8 + rsp], rax
    mov rax, 16
    cmp QWORD [-8 + rsp], rax
    mov rax, 0
    setl al
    shl rax, 7
    or rax, 31
    cmp rax, 31
    je _else__1
_then__0:
    mov rax, 12
    mov QWORD [-8 + rsp], rax
    mov rax, 16
    cmp QWORD [-8 + rsp], rax
    mov rax, 0
    setl al
    shl rax, 7
    or rax, 31
    cmp rax, 31
    je _else__4
_then__3:
    mov rax, 28
    jmp _continue__5
    mov rax, r8
    mul rax, 1
    mov r8, rax
    ret
_else__4:
    mov rax, 28
_continue__5:
    jmp _continue__2
_else__1:
    mov rax, 28
_continue__2:
    mov QWORD [-8 + rsp], rax
    mov rax, 0
    add rax, QWORD [-8 + rsp]
    mov QWORD [-8 + rsp], rdi
    mov rdi, rax
    add rsp, -8
    call _print_value
    sub rsp, -8
    mov rdi, QWORD [-8 + rsp]
    mov rax, 159
    ret
_lisp_entry:
    add rsp, -8
    call _function_f_18913907
    sub rsp, -8
    ret
```

Classic Peephole Optimizations

- **Algebraic simplifications**
  - `x = x + 0` `x = x * 1`

Classic Peephole Optimizations

- **Algebraic simplifications**
  - `x = x + 0` `x = x * 1`
- **Eliminating redundant loads and stores**
  -
```
LD R0, a
ST a, R0
```

```
default rel
global _lisp_entry
extern _lisp_error
extern _read_num
extern _print_value
extern _print_newline
    section .text
_function_f_18913907:
    mov rax, 12
    mov QWORD [-8 + rsp], rax
    mov rax, 16
    cmp QWORD [-8 + rsp], rax
    mov rax, 0
    setl al
    shl rax, 7
    or rax, 31
    cmp rax, 31
    je _else__1
_then__0:
    mov rax, 12
    mov QWORD [-8 + rsp], rax
    mov rax, 16
    cmp QWORD [-8 + rsp], rax
    mov rax, 0
    setl al
    shl rax, 7
    or rax, 31
    cmp rax, 31
    je _else__4
_then__3:
    mov rax, 28
    jmp _continue__5
    mov rax, r8
    mul rax, 1
    mov r8, rax
    ret
_else__4:
    mov rax, 28
_continue__5:
    jmp _continue__2
_else__1:
    mov rax, 28
_continue__2:
    mov QWORD [-8 + rsp], rax
    mov rax, 0
    add rax, QWORD [-8 + rsp]
    mov QWORD [-8 + rsp], rdi
    mov rdi, rax
    add rsp, -8
    call _print_value
    sub rsp, -8
    mov rdi, QWORD [-8 + rsp]
    mov rax, 159
    ret
_lisp_entry:
    add rsp, -8
    call _function_f_18913907
    sub rsp, -8
    ret
```

```
default rel
global _lisp_entry
extern _lisp_error
extern _read_num
extern _print_value
extern _print_newline
    section .text
_function_f_18913907:
    mov rax, 12
    mov QWORD [-8 + rsp], rax
    mov rax, 16
    cmp QWORD [-8 + rsp], rax
    mov rax, 0
    setl al
    shl rax, 7
    or rax, 31
    cmp rax, 31
    je _else__1
_then__0:
    mov rax, 12
    mov QWORD [-8 + rsp], rax
    mov rax, 16
    cmp QWORD [-8 + rsp], rax
    mov rax, 0
    setl al
    shl rax, 7
    or rax, 31
    cmp rax, 31
    je _else__4
_then__3:
    mov rax, 28
    jmp _continue__5
    mov rax, r8
    mul rax, 1
    mov r8, rax
    ret
_else__4:
    mov rax, 28
_continue__5:
    jmp _continue__2
_else__1:
    mov rax, 28
_continue__2:
    mov QWORD [-8 + rsp], rax
    mov rax, 0
    add rax, QWORD [-8 + rsp]
    mov QWORD [-8 + rsp], rdi
    mov rdi, rax
    add rsp, -8
    call _print_value
    sub rsp, -8
    mov rdi, QWORD [-8 + rsp]
    mov rax, 159
    ret
_lisp_entry:
    add rsp, -8
    call _function_f_18913907
    sub rsp, -8
    ret
```

Classic Peephole Optimizations

- **Algebraic simplifications**
  - `x = x + 0` `x = x * 1`
- **Eliminating redundant loads and stores**
  -
```
LD R0, a
ST a, R0
```

Classic Peephole Optimizations

- **Algebraic simplifications**
  - `x = x + 0` `x = x * 1`
- **Eliminating redundant loads and stores**
  - ```
    LD R0, a
    ST a, R0
    ```
- **Eliminating unreachable code**
  - ```
    jmp L1
    mov rax, 0
    L1:
      …
    ```

Classic Peephole Optimizations

- **Algebraic simplifications**
  - `x = x + 0` `x = x * 1`
- **Eliminating redundant loads and stores**
  - ```
    LD R0, a
    ST a, R0
    ```
- **Eliminating unreachable code**
  - ```
    jmp L1
    mov rax, 0
    L1:
      …
    ```

```
default rel
global _lisp_entry
extern _lisp_error
extern _read_num
extern _print_value
extern _print_newline
    section .text
_function_f_18913907:
    mov rax, 12
    mov QWORD [-8 + rsp], rax
    mov rax, 16
    cmp QWORD [-8 + rsp], rax
    mov rax, 0
    setl al
    shl rax, 7
    or rax, 31
    cmp rax, 31
    je _else__1
_then__0:
    mov rax, 12
    mov QWORD [-8 + rsp], rax
    mov rax, 16
    cmp QWORD [-8 + rsp], rax
    mov rax, 0
    setl al
    shl rax, 7
    or rax, 31
    cmp rax, 31
    je _else__4
_then__3:
    mov rax, 28
    jmp _continue__5
    mov rax, r8
    mul rax, 1
    mov r8, rax
    ret
_else__4:
    mov rax, 28
_continue__5:
    jmp _continue__2
_else__1:
    mov rax, 28
_continue__2:
    mov QWORD [-8 + rsp], rax
    mov rax, 0
    add rax, QWORD [-8 + rsp]
    mov QWORD [-8 + rsp], rdi
    mov rdi, rax
    add rsp, -8
    call _print_value
    sub rsp, -8
    mov rdi, QWORD [-8 + rsp]
    mov rax, 159
    ret
_lisp_entry:
    add rsp, -8
    call _function_f_18913907
    sub rsp, -8
    ret
```

```
default rel
global _lisp_entry
extern _lisp_error
extern _read_num
extern _print_value
extern _print_newline
    section .text
_function_f_18913907:
    mov rax, 12
    mov QWORD [-8 + rsp], rax
    mov rax, 16
    cmp QWORD [-8 + rsp], rax
    mov rax, 0
    setl al
    shl rax, 7
    or rax, 31
    cmp rax, 31
    je _else__1
```

```
_then__0:
    mov rax, 12
    mov QWORD [-8 + rsp], rax
    mov rax, 16
    cmp QWORD [-8 + rsp], rax
    mov rax, 0
    setl al
    shl rax, 7
    or rax, 31
    cmp rax, 31
    je _else__4
_then__3:
    mov rax, 28
    jmp _continue__5
    mov rax, r8
    mul rax, 1
    mov r8, rax
    ret
```

```
_else__4:
    mov rax, 28
_continue__5:
    jmp _continue__2
_else__1:
    mov rax, 28
_continue__2:
    mov QWORD [-8 + rsp], rax
    mov rax, 0
    add rax, QWORD [-8 + rsp]
    mov QWORD [-8 + rsp], rdi
    mov rdi, rax
    add rsp, -8
    call _print_value
    sub rsp, -8
    mov rdi, QWORD [-8 + rsp]
    mov rax, 159
    ret
_lisp_entry:
    add rsp, -8
    call _function_f_18913907
    sub rsp, -8
    ret
```

Classic Peephole Optimizations

- **Algebraic simplifications**
  - `x = x + 0` `x = x * 1`
- **Eliminating redundant loads and stores**
  - ```
    LD R0, a
    ST a, R0
    ```
- **Eliminating unreachable code**
  - ```
    jmp L1
    mov rax, 0
    L1:
      …
    ```

Classic Peephole Optimizations

- **Algebraic simplifications**
  - `x = x + 0` `x = x * 1`
- **Eliminating redundant loads and stores**
  - ```
    LD R0, a
    ST a, R0
    ```
- **Eliminating unreachable code**
  - ```
    jmp L1
    mov rax, 0
    L1:
      …
    ```
- **Flow-of-control optimizations**
  - ```
    jmp L1
    …
    L1:
      jmp L2
    ```
    ```

Classic Peephole Optimizations

- **Algebraic simplifications**
  - `x = x + 0` `x = x * 1`
- **Eliminating redundant loads and stores**
  - ```
    LD R0, a
    ST a, R0
    ```
- **Eliminating unreachable code**
  - ```
    jmp L1
    mov rax, 0
    L1:
      …
    ```
- **Flow-of-control optimizations**
  - ```
    jmp L1
    …
    L1:
      jmp L2
    ```
  - ```
    jmp L2
    …
    L1:
      jmp L2
    ```

```
default rel
global _lisp_entry
extern _lisp_error
extern _read_num
extern _print_value
extern _print_newline
    section .text
_function_f_18913907:
    mov rax, 12
    mov QWORD [-8 + rsp], rax
    mov rax, 16
    cmp QWORD [-8 + rsp], rax
    mov rax, 0
    setl al
    shl rax, 7
    or rax, 31
    cmp rax, 31
    je _else__1
_then__0:
    mov rax, 12
    mov QWORD [-8 + rsp], rax
    mov rax, 16
    cmp QWORD [-8 + rsp], rax
    mov rax, 0
    setl al
    shl rax, 7
    or rax, 31
    cmp rax, 31
    je _else__4
_then__3:
    mov rax, 28
    jmp _continue__5
    mov rax, r8
    mul rax, 1
    mov r8, rax
    ret
_else__4:
    mov rax, 28
_continue__5:
    jmp _continue__2
_else__1:
    mov rax, 28
_continue__2:
    mov QWORD [-8 + rsp], rax
    mov rax, 0
    add rax, QWORD [-8 + rsp]
    mov QWORD [-8 + rsp], rdi
    mov rdi, rax
    add rsp, -8
    call _print_value
    sub rsp, -8
    mov rdi, QWORD [-8 + rsp]
    mov rax, 159
    ret
_lisp_entry:
    add rsp, -8
    call _function_f_18913907
    sub rsp, -8
    ret
```

```
default rel
global _lisp_entry
extern _lisp_error
extern _read_num
extern _print_value
extern _print_newline
    section .text
_function_f_18913907:
    mov rax, 12
    mov QWORD [-8 + rsp], rax
    mov rax, 16
    cmp QWORD [-8 + rsp], rax
    mov rax, 0
    setl al
    shl rax, 7
    or rax, 31
    cmp rax, 31
    je _else__1
```

```
_then__0:
    mov rax, 12
    mov QWORD [-8 + rsp], rax
    mov rax, 16
    cmp QWORD [-8 + rsp], rax
    mov rax, 0
    setl al
    shl rax, 7
    or rax, 31
    cmp rax, 31
    je _else__4
_then__3:
    mov rax, 28
    jmp _continue__5
    mov rax, r8
    mul rax, 1
    mov r8, rax
    ret
```

```
_else__4:
    mov rax, 28
_continue__5:
    jmp _continue__2
_else__1:
    mov rax, 28
_continue__2:
    mov QWORD [-8 + rsp], rax
    mov rax, 0
    add rax, QWORD [-8 + rsp]
    mov QWORD [-8 + rsp], rdi
    mov rdi, rax
    add rsp, -8
    call _print_value
    sub rsp, -8
    mov rdi, QWORD [-8 + rsp]
    mov rax, 159
    ret
_lisp_entry:
    add rsp, -8
    call _function_f_18913907
    sub rsp, -8
    ret
```

What is one advantage of

Instruction → Instruction

(peephole) optimization?

What is one advantage of

AST → AST optimization?