All the rest ...

Martin Sulzmann

All the rest

What else there is

- Effects and optimizations.
- Liveness analysis.
- Static single assignment.
- Register allocation.
- Memory ordering.

The Effect Monster

Example

- Consider

```
x := f(1) + 3;
y := 4 + f(1);
```

- Can we rewrite the above to the following?

```
t := f(1);
x := t + 3;
y := 4 + t;
```

- This optimization is only valid assuming that function `f` is side-effect free!

Liveness Analysis

Objective

Eliminate *dead* variable assignments.

Example

```
x:= y + x;    (1)
y:= y + z;    (2)
x:= y + 1;    (3)
```

- There is no path between (1) and (3) on which `x` is alive.
- Hence, can eliminate (1).

Liveness Analysis (2)

Definition

A variable `x` is *alive* at a certain program point L if we find a program path starting with L such that

- `x` is used along L, and
- `x` is not reassigned along L.

[

Static Liveness Analysis]

- Variant of a data-flow analysis.
- Approximate the behavior of program.
- Will be necessarly an overapproximation (due to static).

Static Single Assignment

Static Single Assignment (SSA)

Each variable defined exactly once but possibly used many times.

Non-SSA Example

```
x = 1
y = x + 2
x = x * 2
z = y * x
```

SSA Example

```
x = 1
y = x + 2
x2 = x * 2
z = y * x2
```

Static Single Assignment (2)

Joins

```
x = ..
if x > 5 {
  y = x * 2
} else {
  z = 5
}
w = phi(y,z)
```

- "phi" function chooses the argument based on which control-flow path is taken.

Static Single Assignment (3)

Summary

- Non-trivial to achieve SSA form.
- Many analyses/optimizations become easier once we reach SSA form.

Register Allocation

Objective

- Ensure fast access to values. Memory slow (unless cached).
- Registers superfast. Limited amount of registers.
- How to maximize amount of registers?

Register Allocation

- Must assign registers to variables.
- Two variables can share the same register if not alive at the same time.
- Based on liveness analysis build graph where "live at the same time" variables are connected.
- Finding register-to-variable allocation reduces to the graph coloring problem.

Memory Ordering

Out-of order execution

- To fully utilize a modern CPU, a compiler my reorder (independent) memory operations.
- Statements as seen in a specific order may not be executed in that order!

```
x := ...
y := ...
```

- Has effect on the program's semantics in a concurrent setting.
- Memory fences prevent a compiler to make such optimizations.