Tour of common optimizations

Simple example

```plaintext
foo(z) {
    x := 3 + 6;
    y := x - 5
    return z * y
}
```

Another example

```plaintext
x := a + b;
...
y := a + b;
```

Another example

```plaintext
if (..) {
    x := a + b;
} else {
    x := a + b;
}
```

Another example

```plaintext
a := 0
if (..) {
    x := a + b;
} else {
    y := a + b;
}
```
Another example

```plaintext
if {...} {
  x := a + b;
}
...  \[ t := a + b \]  
\text{Partial Redundancy Elimination, DFE}

y := a + b;
```

Another example

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

What if we run CSE now?

```plaintext
x := \textbf{E}
...
```

Another example

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

What if we run CSE now?

```plaintext
x := E
...
\emptyset \rightarrow x
```

Another example

```plaintext
x := y^n x
...
```

Another example

```plaintext
x := y \times\!
...
```

Another example

```plaintext
x := ...
```
Another example

- Often used as a clean-up pass

\[
x := y^zz
\]

\[
\ldots
\]

\[
x := \ldots
\]

\[
x := y
\]

\[
z := z + x
\]

\[
DAE \quad x := y
\]

\[
z := z + y
\]

Another example

if (false) {
    
    \[
    \text{dead assignment elim (unnamed assignment elem)}
    \]

    \[
    \text{Another common clean up opt}
    \]

    \[
    a = new int [10];
    
    \text{for (index = 0; index < 10; index ++)} {
        \text{if (index < 0 || index >= a.length())} {
            \text{throw OutOfBoundsException;}
        }
        a[index] = 0;
    }
}

Another example

- In "lowered" Java:

```java
a = new int[10];
for (index = 0; index < 10; index++) {
    if (index < 0 || index >= a.length()) {
        throw OutOfBoundsException;
    }
    a[index] = 0;
}
```
Another example

```plaintext
p := &x;
*p := 5
y := x + 1;
```

Another example

```plaintext
x := 5;
*p := 3
y := x + 1;
```

Another example

```plaintext
for j := 1 to N
  for i := 1 to M
    a[i] := a[i] + b[i][j]
```

Another example

```plaintext
area(h, w) { return h * w }
```

Another example

```plaintext
h := ...
w := 4;
a := area(h, w)
L * x * 4
L <= 2
```

Another example

```plaintext
area(h, w) { return h * w }
```

Another example

```plaintext
h := ...
w := 4;
a := area(h, w)
```

```plaintext
Many "ills" often become
impotent after inuring
h <= 2
```
Optimization themes

• Don’t compute if you don’t have to
  – unused assignment elimination
• Compute at compile-time if possible
  – constant folding, loop unrolling, inlining
• Compute it as few times as possible
  – CSE, PRE, PDE, loop invariant code motion
• Compute it as cheaply as possible
  – strength reduction
• Enable other optimizations
  – constant and copy prop, pointer analysis
• Compute it with as little code space as possible
  – unreachable code elimination