Tour of common optimizations

Simple example

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

Another example

```plaintext
x := a + b;
...
if (...) {
  a := read();
  x' := a + b; x = x'
  print(x);
}
```

Another example

```plaintext
x := a + b;
...
y := a + b; x'
```
Another example

```plaintext
if (...) {
    a := read();
    x := a + b;
    print(x);
} else {
    t := a + b;
    ... Partial Redundancy
    Eliminate PKE
    ...
    y := a + b;
```

Another example

```plaintext
x := y
...
What if we run CSE now?
z := z + x
```

Another example

```plaintext
x := y
...
What if we run CSE now?
z := z + y \times x
```

Another example

```plaintext
x := y
...
z := z + y \times x
```

Another example

```plaintext
x := y**z
...
x := ...
```
Another example

• Often used as a clean-up pass

\[
x := y \quad \text{Copy prop} \quad x := y \quad \text{DAE} \quad x := y
\]

\[
z := z + x \quad \rightarrow \quad z := z + y \quad \rightarrow \quad z := z + y
\]

Another example

\[
x := y^{x^2}
\]

... [if \(x\) is not used]

\[
x := ...
\]

Another example

• In Java:

\[
a = \text{new int}[10];
\]

\[
\text{for} \ (\text{index} = 0; \ \text{index} < 10; \ \text{index}++) \ \{
\]

\[
a[\text{index}] = 100;
\]

\[
\}
\]

Another example

• In “lowered” Java:

\[
a = \text{new int}[10]; \quad a.length = 10
\]

\[
\text{for} \ (\text{index} = 0; \ \text{index} < 10; \ \text{index}++) \ \{
\]

\[
\text{if} \ \{ \ \text{index} < 0 \ \text{||} \ \text{index} >= a.length() \ \}\ \{
\]

\[
\text{throw OutOfBoundsException;}
\]

\[
\}
\]

\[
[0..9] \ a[\text{index}] = 0;
\]

\[
\}
\]
```plaintext
Another example

\[ p := &x; \]
\[ *p := 5 \]
\[ y := x + 1; \]

\[ x := 5; \]
\[ *p := 3 \]
\[ y := x + 1; \rightarrow \]

Another example

\[ \text{for } j := 1 \text{ to } N \]
\[ \text{for } i := 1 \text{ to } M \]
\[ a[i] := a[i] + b[j]; \]

Another example

\[ \text{area}(h,w) \{ \text{ return } h * w \} \]
\[ h := \ldots; \]
\[ w := 4; \]
\[ a := \text{area}(h,w) \]
\[ h * w \]
\[ h * 4 \]
\[ h << 2 \]

Another example

\[ \text{area}(h,w) \{ \text{ return } h * w \} \]
\[ h := \ldots; \]
\[ w := 4; \]
\[ a := \text{area}(h,w) \]
\[ h * w \]
\[ h << 2 \]

Another example

point/alias analysis
\[ p := &x; \]
\[ *p := 5 \]
\[ y := x + 1; \]

Another example

loop invariant code motion
\[ \text{for } j := 1 \text{ to } N \]
\[ \text{for } i := 1 \text{ to } M \]
\[ a[i] := a[i] + b[j]; \]
```

```markdown
Another example

\[ p := &x; \]
\[ *p := 5 \]
\[ y := x + 1; \]

Another example

\[ \text{for } j := 1 \text{ to } N \]
\[ \text{for } i := 1 \text{ to } M \]
\[ a[i] := a[i] + b[j]; \]

Another example

\[ \text{area}(h,w) \{ \text{ return } h * w \} \]
\[ h := \ldots; \]
\[ w := 4; \]
\[ a := \text{area}(h,w) \]
\[ h * w \]
\[ h * 4 \]
\[ h << 2 \]

Another example

\[ \text{area}(h,w) \{ \text{ return } h * w \} \]
\[ h := \ldots; \]
\[ w := 4; \]
\[ a := \text{area}(h,w) \]
\[ h * w \]
\[ h * 4 \]
\[ h << 2 \]
```
## 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