15-411: Induction Variables

Jan Hoffmann

Example

Check if an array is sorted

```
bool is_sorted(int[] A, int n)
//@requires 0 <= n && n <= \length(A);
{
  for (int i = 0; i < n-1; i++)
    //@loop_invariant 0 <= i;
    if (A[i] > A[i+1]) return false;
  return true;
}
```

Translation to SSA Form

(without array bound checks)

```
is_sorted(A, n) :
        i_0 ← 0
        goto loop(i_0)
loop(i_1) :
        t_0 ← n − 1
        if (i_1 ≥ t_0) goto rtrue
        t_1 ← 4 ∗ i_1
        t_2 ← A + t_1
        t_3 ← M[t_2]
        t_4 ← i_1 + 1
        t_5 ← 4 ∗ t_4
        t_6 ← A + t_5
        t_7 ← M[t_6]
        if (t_3 > t_7) goto rfalse
        i_2 ← i_1 + 1
        goto loop(i_2)
rtrue :
        return 1
rfalse :
        return 0
```

Basic induction variable

```
is_sorted(A, n) :
        i0 ← 0
        goto loop(i0)
loop(i1) :
        t0 ← n − 1
        if (i1 ≥ t0) goto rtrue
        t1 ← 4 ∗ i1
        t2 ← A + t1
        t3 ← M[t2]
        t4 ← i1 + 1
        t5 ← 4 ∗ t4
        t6 ← A + t5
        t7 ← M[t6]
        if (t3 > t7) goto rfalse
        i2 ← i1 + 1
        goto loop(i2)
rtrue :
        return 1
rfalse :
        return 0
```

Translation to SSA Form

(without array bound checks)

Basic induction variable

Derived induction variable

Translation to SSA Form

(without array bound checks)

```
is_sorted(A, n) :
        i0 ← 0
        goto loop(i0)
loop(i1) :
        t0 ← n − 1
        if (i1 ≥ t0) goto rtrue
        t1 ← 4 ∗ i1
        t2 ← A + t1
        t3 ← M[t2]
        t4 ← i1 + 1
        t5 ← 4 ∗ t4
        t6 ← A + t5
        t7 ← M[t6]
        if (t3 > t7) goto rfalse
        i2 ← i1 + 1
        goto loop(i2)
rtrue :
        return 1
rfalse :
        return 0
```

A first optimization

is_sorted(A, n) :

$i_0 \leftarrow 0$

goto loop(i_0)

loop(i_1) :

$t_0 \leftarrow n - 1$

if $(i_1 \geq t_0)$ goto rtrue

$t_1 \leftarrow 4 * i_1$

$t_2 \leftarrow A + t_1$

$t_3 \leftarrow M[t_2]$

$t_4 \leftarrow i_1 + 1$

$t_5 \leftarrow 4 * t_4$

$t_6 \leftarrow A + t_5$

$t_7 \leftarrow M[t_6]$

if $(t_3 > t_7)$ goto rfalse

$i_2 \leftarrow i_1 + 1$

goto loop(i_2)

A first optimization

Common subexpression elimination applies

is_sorted(A, n) :
$i_0 \leftarrow 0$
goto loop(i_0)
loop(i_1) :
$t_0 \leftarrow n - 1$
if $(i_1 \geq t_0)$ goto rtrue
$t_1 \leftarrow 4 * i_1$
$t_2 \leftarrow A + t_1$
$t_3 \leftarrow M[t_2]$
$t_4 \leftarrow i_1 + 1$
$t_5 \leftarrow 4 * t_4$
$t_6 \leftarrow A + t_5$
$t_7 \leftarrow M[t_6]$
if $(t_3 > t_7)$ goto rfalse
$i_2 \leftarrow i_1 + 1$
goto loop(i_2)

A first optimization

Common subexpression elimination applies

```
is_sorted(A, n) :
    i_0 ← 0
    goto loop(i_0)
loop(i_1) :
    t_0 ← n − 1
    if (i_1 ≥ t_0) goto rtrue
    t_1 ← 4 ∗ i_1
    t_2 ← A + t_1
    t_3 ← M[t_2]
    t_4 ← i_1 + 1
    t_5 ← 4 ∗ t_4
    t_6 ← A + t_5
    t_7 ← M[t_6]
    if (t_3 > t_7) goto rfalse
    i_2 ← i_1 + 1
    goto loop(i_2)
```

A first optimization

is_sorted(A, n) :

$i_0 \leftarrow 0$

goto loop(i_0)

loop(i_1) :

$t_0 \leftarrow n - 1$

if $(i_1 \geq t_0)$ goto rtrue

$t_1 \leftarrow 4 * i_1$

$t_2 \leftarrow A + t_1$

$t_3 \leftarrow M[t_2]$

$t_4 \leftarrow i_1 + 1$

$t_5 \leftarrow 4 * t_4$

$t_6 \leftarrow A + t_5$

$t_7 \leftarrow M[t_6]$

if $(t_3 > t_7)$ goto rfalse

$i_2 \leftarrow i_1 + 1$

goto loop(i_2)

Common subexpression elimination applies

Preserve induction variable i

is_sorted(A, n) :

$i_0 \leftarrow 0$

goto loop(i_0)

loop(i_1) :

$t_0 \leftarrow n - 1$

if $(i_1 \geq t_0)$ goto rtrue

$t_1 \leftarrow 4 * i_1$

$t_2 \leftarrow A + t_1$

$t_3 \leftarrow M[t_2]$

$t_4 \leftarrow i_1 + 1$

$t_5 \leftarrow 4 * t_4$

$t_6 \leftarrow A + t_5$

$t_7 \leftarrow M[t_6]$

if $(t_3 > t_7)$ goto rfalse

$i_2 \leftarrow t_4$

goto loop(i_2)

is_sorted(A, n) :
$i_0 \leftarrow 0$
goto loop(i_0)
loop(i_1) :
$t_0 \leftarrow n - 1$
if $(i_1 \geq t_0)$ goto rtrue
$t_1 \leftarrow 4 * i_1$
$t_2 \leftarrow A + t_1$
$t_3 \leftarrow M[t_2]$
$t_4 \leftarrow i_1 + 1$
$t_5 \leftarrow 4 * t_4$
$t_6 \leftarrow A + t_5$
$t_7 \leftarrow M[t_6]$
if $(t_3 > t_7)$ goto rfalse
$i_2 \leftarrow t_4$
goto loop(i_2)

is_sorted(A, n) :
$i_0 \leftarrow 0$
goto loop(i_0)
loop(i_1) :
$t_0 \leftarrow n - 1$
if $(i_1 \geq t_0)$ goto rtrue
$t_1 \leftarrow 4 * i_1$
$t_2 \leftarrow A + t_1$
$t_3 \leftarrow M[t_2]$
$i_2 \leftarrow i_1 + 1$
$t_5 \leftarrow 4 * i_2$
$t_6 \leftarrow A + t_5$
$t_7 \leftarrow M[t_6]$
if $(t_3 > t_7)$ goto rfalse

goto loop(i_2)

Consider induction variable t1

$$t_1 \leftarrow 4 * i_1$$

Idea: compute t1 from a previous iteration of t1

Introduce new variable j = 4*i

is_sorted(A, n) :

$i_0 \leftarrow 0$

goto loop(i_0)

loop(i_1) :

$t_0 \leftarrow n - 1$

if $(i_1 \geq t_0)$ goto rtrue

$t_1 \leftarrow 4 * i_1$

$t_2 \leftarrow A + t_1$

$t_3 \leftarrow M[t_2]$

$i_2 \leftarrow i_1 + 1$

$t_5 \leftarrow 4 * i_2$

$t_6 \leftarrow A + t_5$

$t_7 \leftarrow M[t_6]$

if $(t_3 > t_7)$ goto rfalse

goto loop(i_2)

Consider induction variable t1

$$t_1 \leftarrow 4 * i_1$$

Idea: compute t1 from a previous iteration of t1

Introduce new variable j = 4*i

is_sorted(A, n) :

$i_0 \leftarrow 0$

$j_0 \leftarrow 4 * i_0$ @ensures $j_0 = 4 * i_0$

goto loop(i_0, j_0)

loop(i_1, j_1) : @requires $j_1 = 4 * i_1$

$t_0 \leftarrow n - 1$

if $(i_1 \geq t_0)$ goto rtrue

$t_1 \leftarrow j_1$ @assert $j_1 = 4 * i_1$

$t_2 \leftarrow A + t_1$

$t_3 \leftarrow M[t_2]$

$i_2 \leftarrow i_1 + 1$

$j_2 \leftarrow 4 * i_2$ @ensures $j_2 = 4 * i_2$

$t_4 \leftarrow i_2$

$t_5 \leftarrow 4 * t_4$

$t_6 \leftarrow A + t_5$

$t_7 \leftarrow M[t_6]$

if $(t_3 > t_7)$ goto rfalse

goto loop(i_2, j_2)

```
is_sorted(A, n) :
    i_0 ← 0
    j_0 ← 4 * i_0
    goto loop(i_0, j_0)
loop(i_1, j_1) :
    t_0 ← n − 1
    if (i_1 ≥ t_0) goto rtrue
    t_1 ← j_1
    t_2 ← A + t_1
    t_3 ← M[t_2]
    i_2 ← i_1 + 1
    j_2 ← 4 * i_2
    t_4 ← i_2
    t_5 ← 4 * t_4
    t_6 ← A + t_5
    t_7 ← M[t_6]
    if (t_3 > t_7) goto rfalse
    goto loop(i_2, j_2)
```

```
is_sorted(A, n) :
    i_0 ← 0
    j_0 ← 0                          @ensures j_0 = 4 * i_0
    goto loop(i_0, j_0)
loop(i_1, j_1) :                     @requires j_1 = 4 * i_1
    t_0 ← n − 1
    if (i_1 ≥ t_0) goto rtrue
    t_1 ← j_1                        @assert j_1 = 4 * i_1
    t_2 ← A + t_1
    t_3 ← M[t_2]
    i_2 ← i_1 + 1
    j_2 ← j_1 + 4                    @ensures j_2 = 4 * i_2
    t_4 ← i_2
    t_5 ← 4 * t_4
    t_6 ← A + t_5
    t_7 ← M[t_6]
    if (t_3 > t_7) goto rfalse
    goto loop(i_2, j_2)
```

is_sorted(A, n) :
$i_0 \leftarrow 0$
$j_0 \leftarrow 4 * i_0$
goto loop(i_0, j_0)
loop(i_1, j_1) :
$t_0 \leftarrow n - 1$
if $(i_1 \geq t_0)$ goto rtrue
$t_1 \leftarrow j_1$
$t_2 \leftarrow A + t_1$
$t_3 \leftarrow M[t_2]$
$i_2 \leftarrow i_1 + 1$
$j_2 \leftarrow 4 * i_2$
$t_4 \leftarrow i_2$
$t_5 \leftarrow 4 * t_4$
$t_6 \leftarrow A + t_5$
$t_7 \leftarrow M[t_6]$
if $(t_3 > t_7)$ goto rfalse
goto loop(i_2, j_2)

$j_0 = 4 * i_0 = 0$

is_sorted(A, n) :
$i_0 \leftarrow 0$
$j_0 \leftarrow 0$
goto loop(i_0, j_0)
loop(i_1, j_1) :
$t_0 \leftarrow n - 1$
if $(i_1 \geq t_0)$ goto rtrue
$t_1 \leftarrow j_1$
$t_2 \leftarrow A + t_1$
$t_3 \leftarrow M[t_2]$
$i_2 \leftarrow i_1 + 1$
$j_2 \leftarrow j_1 + 4$
$t_4 \leftarrow i_2$
$t_5 \leftarrow 4 * t_4$
$t_6 \leftarrow A + t_5$
$t_7 \leftarrow M[t_6]$
if $(t_3 > t_7)$ goto rfalse
goto loop(i_2, j_2)

@ensures $j_0 = 4 * i_0$
@requires $j_1 = 4 * i_1$
@assert $j_1 = 4 * i_1$
@ensures $j_2 = 4 * i_2$

is_sorted(A, n) :
$i_0 \leftarrow 0$
$j_0 \leftarrow 4 * i_0$
goto loop(i_0, j_0)
loop(i_1, j_1) :
$t_0 \leftarrow n - 1$
if $(i_1 \geq t_0)$ goto rtrue
$t_1 \leftarrow j_1$
$t_2 \leftarrow A + t_1$
$t_3 \leftarrow M[t_2]$
$i_2 \leftarrow i_1 + 1$
$j_2 \leftarrow 4 * i_2$
$t_4 \leftarrow i_2$
$t_5 \leftarrow 4 * t_4$
$t_6 \leftarrow A + t_5$
$t_7 \leftarrow M[t_6]$
if $(t_3 > t_7)$ goto rfalse
goto loop(i_2, j_2)

$j_0 = 4 * i_0 = 0$

is_sorted(A, n) :
$i_0 \leftarrow 0$
$j_0 \leftarrow 0$
goto loop(i_0, j_0)
loop(i_1, j_1) :
$t_0 \leftarrow n - 1$
if $(i_1 \geq t_0)$ goto rtrue
$t_1 \leftarrow j_1$
$t_2 \leftarrow A + t_1$
$t_3 \leftarrow M[t_2]$
$i_2 \leftarrow i_1 + 1$
$j_2 \leftarrow j_1 + 4$
$t_4 \leftarrow i_2$
$t_5 \leftarrow 4 * t_4$
$t_6 \leftarrow A + t_5$
$t_7 \leftarrow M[t_6]$
if $(t_3 > t_7)$ goto rfalse
goto loop(i_2, j_2)

@ensures $j_0 = 4 * i_0$
@requires $j_1 = 4 * i_1$
@assert $j_1 = 4 * i_1$
@ensures $j_2 = 4 * i_2$

$j_2 = 4 * i_2 = 4 * (i_1 + 1) = 4 * i_1 + 4 = j_1 + 4$

is_sorted(A, n) :

$i_0 \leftarrow 0$

$j_0 \leftarrow 0$

goto loop(i_0, j_0)

loop(i_1, j_1) :

$t_0 \leftarrow n - 1$

if $(i_1 \geq t_0)$ goto rtrue

$t_1 \leftarrow j_1$

$t_2 \leftarrow A + t_1$

$t_3 \leftarrow M[t_2]$

$i_2 \leftarrow i_1 + 1$

$j_2 \leftarrow j_1 + 4$

$t_4 \leftarrow i_2$

$t_5 \leftarrow 4 * t_4$

$t_6 \leftarrow A + t_5$

$t_7 \leftarrow M[t_6]$

if $(t_3 > t_7)$ goto rfalse

goto loop(i_2, j_2)

is_sorted(A, n) :

$i_0 \leftarrow 0$

$j_0 \leftarrow 0$

goto loop(i_0, j_0)

loop(i_1, j_1) :

$t_0 \leftarrow n - 1$

if $(i_1 \geq t_0)$ goto rtrue

$t_1 \leftarrow j_1$

$t_2 \leftarrow A + t_1$

$t_3 \leftarrow M[t_2]$

$i_2 \leftarrow i_1 + 1$

$j_2 \leftarrow j_1 + 4$

$t_4 \leftarrow i_2$

$t_5 \leftarrow 4 * t_4$

$t_6 \leftarrow A + t_5$

$t_7 \leftarrow M[t_6]$

if $(t_3 > t_7)$ goto rfalse

goto loop(i_2, j_2)

Loop hoisting

is_sorted(A, n) :

$i_0 \leftarrow 0$

$j_0 \leftarrow 0$

goto loop(i_0, j_0)

loop(i_1, j_1) :

$t_0 \leftarrow n - 1$

if $(i_1 \geq t_0)$ goto rtrue

$t_1 \leftarrow j_1$

$t_2 \leftarrow A + t_1$

$t_3 \leftarrow M[t_2]$

$i_2 \leftarrow i_1 + 1$

$j_2 \leftarrow j_1 + 4$

$t_4 \leftarrow i_2$

$t_5 \leftarrow 4 * t_4$

$t_6 \leftarrow A + t_5$

$t_7 \leftarrow M[t_6]$

if $(t_3 > t_7)$ goto rfalse

goto loop(i_2, j_2)

Loop hoisting

Copy propagation

is_sorted(A, n) :
    $i_0 \leftarrow 0$
    $j_0 \leftarrow 0$
    goto loop(i_0, j_0)
loop(i_1, j_1) :
    $t_0 \leftarrow n - 1$
    if $(i_1 \geq t_0)$ goto rtrue
    $t_1 \leftarrow j_1$
    $t_2 \leftarrow A + t_1$
    $t_3 \leftarrow M[t_2]$
    $i_2 \leftarrow i_1 + 1$
    $j_2 \leftarrow j_1 + 4$
    $t_4 \leftarrow i_2$
    $t_5 \leftarrow 4 * t_4$
    $t_6 \leftarrow A + t_5$
    $t_7 \leftarrow M[t_6]$
    if $(t_3 > t_7)$ goto rfalse
    goto loop(i_2, j_2)

is_sorted(A, n) :
    $i_0 \leftarrow 0$
    $j_0 \leftarrow 0$ @ensures $j_0 = 4 * i_0$
    $t_0 \leftarrow n - 1$
    goto loop(i_0, j_0)
loop(i_1, j_1) : @requires $j_1 = 4 * i_1$
    if $(i_1 \geq t_0)$ goto rtrue
    $t_2 \leftarrow A + j_1$
    $t_3 \leftarrow M[t_2]$
    $i_2 \leftarrow i_1 + 1$
    $j_2 \leftarrow j_1 + 4$ @ensures $j_2 = 4 * i_2$
    $t_5 \leftarrow 4 * i_2$
    $t_6 \leftarrow A + t_5$
    $t_7 \leftarrow M[t_6]$
    if $(t_3 > t_7)$ goto rfalse
    goto loop(i_2, j_2)

is_sorted(A, n) :
$i_0 \leftarrow 0$
$j_0 \leftarrow 0$
$t_0 \leftarrow n - 1$
goto loop(i_0, j_0)
loop(i_1, j_1) :
if $(i_1 \geq t_0)$ goto rtrue
$t_2 \leftarrow A + j_1$
$t_3 \leftarrow M[t_2]$
$i_2 \leftarrow i_1 + 1$
$j_2 \leftarrow j_1 + 4$
$t_5 \leftarrow 4 * i_2$
$t_6 \leftarrow A + t_5$
$t_7 \leftarrow M[t_6]$
if $(t_3 > t_7)$ goto rfalse
goto loop(i_2, j_2)

is_sorted(A, n) :
$i_0 \leftarrow 0$
$j_0 \leftarrow 0$ @ensures $j_0 = 4 * i_0$
$t_0 \leftarrow n - 1$
goto loop(i_0, j_0)
loop(i_1, j_1) : @requires $j_1 = 4 * i_1$
if $(i_1 \geq t_0)$ goto rtrue
$t_2 \leftarrow A + j_1$
$t_3 \leftarrow M[t_2]$
$i_2 \leftarrow i_1 + 1$
$j_2 \leftarrow j_1 + 4$ @ensures $j_2 = 4 * i_2$
$t_6 \leftarrow A + j_2$
$t_7 \leftarrow M[t_6]$
if $(t_3 > t_7)$ goto rfalse
goto loop(i_2, j_2)

is_sorted(A, n) :

$i_0 \leftarrow 0$

$j_0 \leftarrow 0$

$t_0 \leftarrow n - 1$

goto loop(i_0, j_0)

loop(i_1, j_1) :

if $(i_1 \geq t_0)$ goto rtrue

$t_2 \leftarrow A + j_1$

$t_3 \leftarrow M[t_2]$

$i_2 \leftarrow i_1 + 1$

$j_2 \leftarrow j_1 + 4$

$t_6 \leftarrow A + j_2$

$t_7 \leftarrow M[t_6]$

if $(t_3 > t_7)$ goto rfalse

goto loop(i_2, j_2)

t2 is another induction variable.

is_sorted(A, n) :

$i_0 \leftarrow 0$

$j_0 \leftarrow 0$

$t_0 \leftarrow n - 1$

goto loop(i_0, j_0)

loop(i_1, j_1) :

if $(i_1 \geq t_0)$ goto rtrue

$t_2 \leftarrow A + j_1$

$t_3 \leftarrow M[t_2]$

$i_2 \leftarrow i_1 + 1$

$j_2 \leftarrow j_1 + 4$

$t_6 \leftarrow A + j_2$

$t_7 \leftarrow M[t_6]$

if $(t_3 > t_7)$ goto rfalse

goto loop(i_2, j_2)

t2 is another induction variable.

We introduce k to keep track of k = A + j

is_sorted(A, n) :

$i_0 \leftarrow 0$

$j_0 \leftarrow 0$ @ensures $j_0 = 4 * i_0$

$k_0 \leftarrow A + j_0$ @ensures $k_0 = A + j_0$

$t_0 \leftarrow n - 1$

goto loop(i_0, j_0, k_0)

loop(i_1, j_1, k_1) : @requires $j_1 = 4 * i_1 \wedge k_1 = A + j_1$

if $(i_1 \geq t_0)$ goto rtrue

$t_2 \leftarrow k_1$

$t_3 \leftarrow M[t_2]$

$i_2 \leftarrow i_1 + 1$

$j_2 \leftarrow j_1 + 4$ @ensures $j_2 = 4 * i_2$

$k_2 \leftarrow k_1 + 4$ @ensures $k_2 = A + j_2$

$t_6 \leftarrow A + j_2$

$t_7 \leftarrow M[t_6]$

if $(t_3 > t_7)$ goto rfalse

goto loop(i_2, j_2, k_2)

$$k_2 = A + j_2 = A + j_1 + 4 = k_1 + 4$$

```
is_sorted(A, n) :
    i0 ← 0
    j0 ← 0
    k0 ← A + j0
    t0 ← n − 1
    goto loop(i0, j0, k0)
loop(i1, j1, k1) :
    if (i1 ≥ t0) goto rtrue
    t2 ← k1
    t3 ← M[t2]
    i2 ← i1 + 1
    j2 ← j1 + 4
    k2 ← k1 + 4
    t6 ← A + j2
    t7 ← M[t6]
    if (t3 > t7) goto rfalse
    goto loop(i2, j2, k2)
```

```
is_sorted(A, n) :
    i0 ← 0
    j0 ← 0                  @ensures j0 = 4 * i0
    k0 ← A                  @ensures k0 = A + j0
    t0 ← n − 1
    goto loop(i0, j0, k0)
loop(i1, j1, k1) :          @requires j1 = 4 * i1 ∧
    if (i1 ≥ t0) goto rtrue
    t3 ← M[k1]
    i2 ← i1 + 1
    j2 ← j1 + 4             @ensures j2 = 4 * i2
    k2 ← k1 + 4             @ensures k2 = A + j2
    t7 ← M[k2]
    if (t3 > t7) goto rfalse
    goto loop(i2, j2, k2)
```

```
is_sorted(A, n) :
    i0 ← 0
    j0 ← 0
    k0 ← A + j0
    t0 ← n − 1
    goto loop(i0, j0, k0)
loop(i1, j1, k1) :
    if (i1 ≥ t0) goto rtrue
    t2 ← k1
    t3 ← M[t2]
    i2 ← i1 + 1
    j2 ← j1 + 4
    k2 ← k1 + 4
    t6 ← A + j2
    t7 ← M[t6]
    if (t3 > t7) goto rfalse
    goto loop(i2, j2, k2)
```

```
is_sorted(A, n) :
    i0 ← 0
    j0 ← 0                    @ensures j0 = 4 * i0
    k0 ← A                    @ensures k0 = A + j0
    t0 ← n − 1
    goto loop(i0, j0, k0)
loop(i1, j1, k1) :            @requires j1 = 4 * i1 ∧
    if (i1 ≥ t0) goto rtrue
    t3 ← M[k1]
    i2 ← i1 + 1
    j2 ← j1 + 4               @ensures j2 = 4 * i2
    k2 ← k1 + 4               @ensures k2 = A + j2
    t7 ← M[k2]
    if (t3 > t7) goto rfalse
    goto loop(i2, j2, k2)
```

One more round of constant propagation, dead code elim., and CSE

is_sorted(A, n) :
$i_0 \leftarrow 0$
$j_0 \leftarrow 0$
$k_0 \leftarrow A + j_0$
$t_0 \leftarrow n - 1$
goto loop(i_0, j_0, k_0)
loop(i_1, j_1, k_1) :
if $(i_1 \geq t_0)$ goto rtrue
$t_2 \leftarrow k_1$
$t_3 \leftarrow M[t_2]$
$i_2 \leftarrow i_1 + 1$
$j_2 \leftarrow j_1 + 4$
$k_2 \leftarrow k_1 + 4$
$t_6 \leftarrow A + j_2$
$t_7 \leftarrow M[t_6]$
if $(t_3 > t_7)$ goto rfalse
goto loop(i_2, j_2, k_2)

is_sorted(A, n) :
$i_0 \leftarrow 0$
$j_0 \leftarrow 0$ @ensures $j_0 = 4 * i_0$
$k_0 \leftarrow A$ @ensures $k_0 = A + j_0$
$t_0 \leftarrow n - 1$
goto loop(i_0, j_0, k_0)
loop(i_1, j_1, k_1) : @requires $j_1 = 4 * i_1 \wedge$
if $(i_1 \geq t_0)$ goto rtrue
$t_3 \leftarrow M[k_1]$
$i_2 \leftarrow i_1 + 1$
$j_2 \leftarrow j_1 + 4$ @ensures $j_2 = 4 * i_2$
$k_2 \leftarrow k_1 + 4$ @ensures $k_2 = A + j_2$
$t_7 \leftarrow M[k_2]$
if $(t_3 > t_7)$ goto rfalse
goto loop(i_2, j_2, k_2)

One more round of constant propagation, dead code elim., and CSE

is_sorted(A, n) :
$i_0 \leftarrow 0$
$j_0 \leftarrow 0$
$k_0 \leftarrow A + j_0$
$t_0 \leftarrow n - 1$
goto loop(i_0, j_0, k_0)
loop(i_1, j_1, k_1) :
if $(i_1 \geq t_0)$ goto rtrue
$t_2 \leftarrow k_1$
$t_3 \leftarrow M[t_2]$
$i_2 \leftarrow i_1 + 1$
$j_2 \leftarrow j_1 + 4$
$k_2 \leftarrow k_1 + 4$
$t_6 \leftarrow A + j_2$
$t_7 \leftarrow M[t_6]$
if $(t_3 > t_7)$ goto rfalse
goto loop(i_2, j_2, k_2)

is_sorted(A, n) :
$i_0 \leftarrow 0$
$j_0 \leftarrow 0$ @ensures $j_0 = 4 * i_0$
$k_0 \leftarrow A$ @ensures $k_0 = A + j_0$
$t_0 \leftarrow n - 1$
goto loop(i_0, j_0, k_0)
loop(i_1, j_1, k_1) : @requires $j_1 = 4 * i_1 \wedge$
if $(i_1 \geq t_0)$ goto rtrue
$t_3 \leftarrow M[k_1]$
$i_2 \leftarrow i_1 + 1$
$j_2 \leftarrow j_1 + 4$ @ensures $j_2 = 4 * i_2$
$k_2 \leftarrow k_1 + 4$ @ensures $k_2 = A + j_2$
$t_7 \leftarrow M[k_2]$
if $(t_3 > t_7)$ goto rfalse
goto loop(i_2, j_2, k_2)

One more round of constant propagation, dead code elim., and CSE

is_sorted(A, n) :
$i_0 \leftarrow 0$
$j_0 \leftarrow 0$
$k_0 \leftarrow A + j_0$
$t_0 \leftarrow n - 1$
goto loop(i_0, j_0, k_0)
loop(i_1, j_1, k_1) :
if $(i_1 \geq t_0)$ goto rtrue
$t_2 \leftarrow k_1$
$t_3 \leftarrow M[t_2]$
$i_2 \leftarrow i_1 + 1$
$j_2 \leftarrow j_1 + 4$
$k_2 \leftarrow k_1 + 4$
$t_6 \leftarrow A + j_2$
$t_7 \leftarrow M[t_6]$
if $(t_3 > t_7)$ goto rfalse
goto loop(i_2, j_2, k_2)

is_sorted(A, n) :
$i_0 \leftarrow 0$
$j_0 \leftarrow 0$ @ensures $j_0 = 4 * i_0$
$k_0 \leftarrow A$ @ensures $k_0 = A + j_0$
$t_0 \leftarrow n - 1$
goto loop(i_0, j_0, k_0)
loop(i_1, j_1, k_1) : @requires $j_1 = 4 * i_1 \wedge$
if $(i_1 \geq t_0)$ goto rtrue
$t_3 \leftarrow M[k_1]$
$i_2 \leftarrow i_1 + 1$
$j_2 \leftarrow j_1 + 4$ @ensures $j_2 = 4 * i_2$
$k_2 \leftarrow k_1 + 4$ @ensures $k_2 = A + j_2$
$t_7 \leftarrow M[k_2]$
if $(t_3 > t_7)$ goto rfalse
goto loop(i_2, j_2, k_2)

One more round of constant propagation, dead code elim., and CSE

is_sorted(A, n) :

$i_0 \leftarrow 0$

$j_0 \leftarrow 0$

$k_0 \leftarrow A$

$t_0 \leftarrow n - 1$

goto loop(i_0, j_0, k_0)

loop(i_1, j_1, k_1) :

if $(i_1 \geq t_0)$ goto rtrue

$t_3 \leftarrow M[k_1]$

$i_2 \leftarrow i_1 + 1$

$j_2 \leftarrow j_1 + 4$

$k_2 \leftarrow k_1 + 4$

$t_7 \leftarrow M[k_2]$

if $(t_3 > t_7)$ goto rfalse

goto loop(i_2, j_2, k_2)

is_sorted(A, n) :

$i_0 \leftarrow 0$

$j_0 \leftarrow 0$

$k_0 \leftarrow A$

$t_0 \leftarrow n - 1$

goto loop(i_0, j_0, k_0)

loop(i_1, j_1, k_1) :

if $(i_1 \geq t_0)$ goto rtrue

$t_3 \leftarrow M[k_1]$

$i_2 \leftarrow i_1 + 1$

$j_2 \leftarrow j_1 + 4$

$k_2 \leftarrow k_1 + 4$

$t_7 \leftarrow M[k_2]$

if $(t_3 > t_7)$ goto rfalse

goto loop(i_2, j_2, k_2)

j1, j2, and j3 are no longer needed.

is_sorted(A, n) :

$i_0 \leftarrow 0$

$j_0 \leftarrow 0$

$k_0 \leftarrow A$

$t_0 \leftarrow n - 1$

goto loop(i_0, j_0, k_0)

loop(i_1, j_1, k_1) :

if $(i_1 \geq t_0)$ goto rtrue

$t_3 \leftarrow M[k_1]$

$i_2 \leftarrow i_1 + 1$

$j_2 \leftarrow j_1 + 4$

$k_2 \leftarrow k_1 + 4$

$t_7 \leftarrow M[k_2]$

if $(t_3 > t_7)$ goto rfalse

goto loop(i_2, j_2, k_2)

j1, j2, and j3 are no longer needed.

is_sorted(A, n) :
$i_0 \leftarrow 0$
$j_0 \leftarrow 0$
$k_0 \leftarrow A$
$t_0 \leftarrow n - 1$
goto loop(i_0, j_0, k_0)
loop(i_1, j_1, k_1) :
if $(i_1 \geq t_0)$ goto rtrue
$t_3 \leftarrow M[k_1]$
$i_2 \leftarrow i_1 + 1$
$j_2 \leftarrow j_1 + 4$
$k_2 \leftarrow k_1 + 4$
$t_7 \leftarrow M[k_2]$
if $(t_3 > t_7)$ goto rfalse
goto loop(i_2, j_2, k_2)

j1, j2, and j3 are no longer needed.

is_sorted(A, n) :
$i_0 \leftarrow 0$
$k_0 \leftarrow A$ @ensures $k_0 = A + 4 * i_0$
$t_0 \leftarrow n - 1$
goto loop(i_0, k_0)
loop(i_1, k_1) : @requires $k_1 = A + 4 * i_1$
if $(i_1 \geq t_0)$ goto rtrue
$t_3 \leftarrow M[k_1]$
$i_2 \leftarrow i_1 + 1$
$k_2 \leftarrow k_1 + 4$ @ensures $k_2 = A + 4 * i_2$
$t_7 \leftarrow M[k_2]$
if $(t_3 > t_7)$ goto rfalse
goto loop(i_2, k_2)

is_sorted(A, n) :

$i_0 \leftarrow 0$

$k_0 \leftarrow A$

$t_0 \leftarrow n - 1$

goto loop(i_0, k_0)

loop(i_1, k_1) :

if $(i_1 \geq t_0)$ goto rtrue

$t_3 \leftarrow M[k_1]$

$i_2 \leftarrow i_1 + 1$

$k_2 \leftarrow k_1 + 4$

$t_7 \leftarrow M[k_2]$

if $(t_3 > t_7)$ goto rfalse

goto loop(i_2, k_2)

is_sorted(A, n) :

$i_0 \leftarrow 0$

$k_0 \leftarrow A$

$t_0 \leftarrow n - 1$

goto loop(i_0, k_0)

loop(i_1, k_1) :

if $(i_1 \geq t_0)$ goto rtrue

$t_3 \leftarrow M[k_1]$

$i_2 \leftarrow i_1 + 1$

$k_2 \leftarrow k_1 + 4$

$t_7 \leftarrow M[k_2]$

if $(t_3 > t_7)$ goto rfalse

goto loop(i_2, k_2)

$i_1 \geq t_0$ iff $A + 4 * i_1 \geq A + 4 * t_0$

is_sorted(A, n) :
$i_0 \leftarrow 0$
$k_0 \leftarrow A$
$t_0 \leftarrow n - 1$
goto loop(i_0, k_0)
loop(i_1, k_1) :
if $(i_1 \geq t_0)$ goto rtrue
$t_3 \leftarrow M[k_1]$
$i_2 \leftarrow i_1 + 1$
$k_2 \leftarrow k_1 + 4$
$t_7 \leftarrow M[k_2]$
if $(t_3 > t_7)$ goto rfalse
goto loop(i_2, k_2)

is_sorted(A, n) :
$i_0 \leftarrow 0$
$k_0 \leftarrow A$
$t_0 \leftarrow n - 1$
goto loop(i_0, k_0)
loop(i_1, k_1) :
if $(k_1 \geq A + 4 * t_0)$ goto rtrue
$t_3 \leftarrow M[k_1]$
$i_2 \leftarrow i_1 + 1$
$k_2 \leftarrow k_1 + 4$
$t_7 \leftarrow M[k_2]$
if $(t_3 > t_7)$ goto rfalse
goto loop(i_2, k_2)

$i_1 \geq t_0$ iff $A + 4 * i_1 \geq A + 4 * t_0$

is_sorted(A, n) :

$i_0 \leftarrow 0$

$k_0 \leftarrow A$

$t_0 \leftarrow n - 1$

goto loop(i_0, k_0)

loop(i_1, k_1) :

if $(k_1 \geq A + 4 * t_0)$ goto rtrue

$t_3 \leftarrow M[k_1]$

$i_2 \leftarrow i_1 + 1$

$k_2 \leftarrow k_1 + 4$

$t_7 \leftarrow M[k_2]$

if $(t_3 > t_7)$ goto rfalse

goto loop(i_2, k_2)

is_sorted(A, n) :

$i_0 \leftarrow 0$

$k_0 \leftarrow A$

$t_0 \leftarrow n - 1$

goto loop(i_0, k_0)

loop(i_1, k_1) :

if $(k_1 \geq A + 4 * t_0)$ goto rtrue

$t_3 \leftarrow M[k_1]$

$i_2 \leftarrow i_1 + 1$

$k_2 \leftarrow k_1 + 4$

$t_7 \leftarrow M[k_2]$

if $(t_3 > t_7)$ goto rfalse

goto loop(i_2, k_2)

i0, i1, and i2 are no longer needed.

is_sorted(A, n) :
$i_0 \leftarrow 0$
$k_0 \leftarrow A$
$t_0 \leftarrow n - 1$
goto loop(i_0, k_0)
loop(i_1, k_1) :
if $(k_1 \geq A + 4 * t_0)$ goto rtrue
$t_3 \leftarrow M[k_1]$
$i_2 \leftarrow i_1 + 1$
$k_2 \leftarrow k_1 + 4$
$t_7 \leftarrow M[k_2]$
if $(t_3 > t_7)$ goto rfalse
goto loop(i_2, k_2)

i0, i1, and i2 are no longer needed.

is_sorted(A, n) :
$k_0 \leftarrow A$
$t_0 \leftarrow n - 1$
$t_8 \leftarrow 4 * t_0$
$t_9 \leftarrow A + t_8$
goto loop(k_0)
loop(k_1) :
if $(k_1 \geq t_9)$ goto rtrue
$t_3 \leftarrow M[k_1]$
$k_2 \leftarrow k_1 + 4$
$t_7 \leftarrow M[k_2]$
if $(t_3 > t_7)$ goto rfalse
goto loop(k_2)

is_sorted(A, n) :
$i_0 \leftarrow 0$
$k_0 \leftarrow A$
$t_0 \leftarrow n - 1$
goto loop(i_0, k_0)
loop(i_1, k_1) :
if $(k_1 \geq A + 4 * t_0)$ goto rtrue
$t_3 \leftarrow M[k_1]$
$i_2 \leftarrow i_1 + 1$
$k_2 \leftarrow k_1 + 4$
$t_7 \leftarrow M[k_2]$
if $(t_3 > t_7)$ goto rfalse
goto loop(i_2, k_2)

i0, i1, and i2 are no longer needed.

is_sorted(A, n) :
$k_0 \leftarrow A$
$t_0 \leftarrow n - 1$
$t_8 \leftarrow 4 * t_0$
$t_9 \leftarrow A + t_8$
goto loop(k_0)
loop(k_1) :
if $(k_1 \geq t_9)$ goto rtrue
$t_3 \leftarrow M[k_1]$
$k_2 \leftarrow k_1 + 4$
$t_7 \leftarrow M[k_2]$
if $(t_3 > t_7)$ goto rfalse
goto loop(k_2)

Unrolling the loop once can remove one memory access per iteration.