Locus

```
for i
  for j
    for k
```

Locus program with steps
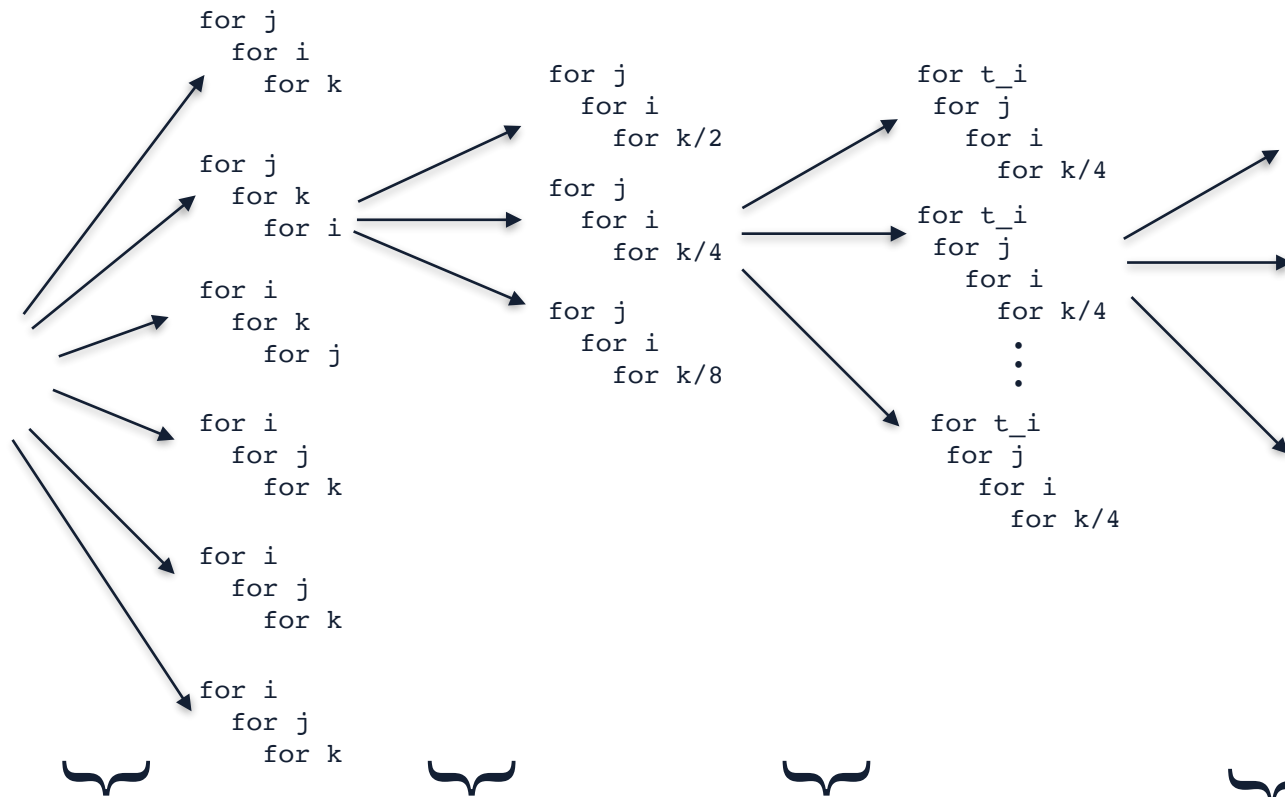to the best variant found

Locus program

+

Locus system

$\Rightarrow$

1. Selects variants (avoid explosion)
2. Runs
3. Determine the best variant

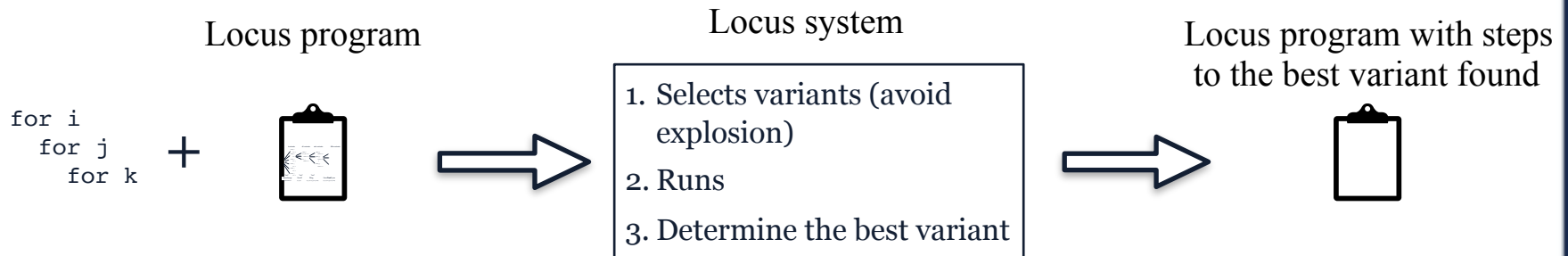6 variants   18 variants   126 variants   882 variants

```
for j
  for i
    for k

for j
  for k
    for i

for i
  for k
    for j

for i
  for j
    for k

for i
  for j
    for k

for i
  for j
    for k
```

```
for j
  for i
    for k/2

for j
  for i
    for k/4

for j
  for i
    for k/8
```

```
for t_i
  for j
    for i
      for k/4

for t_i
  for j
    for i
      for k/4

        ⋮

for t_i
  for j
    for i
      for k/4
```

Interchange
all permutations

Unroll
(2, 4, 8)

Tiling
(2, 4, 8, 16, 32, 64, 128)

Unroll-and-jam
(2, 4, 8, 16, 32, 64, 128)

Locus

Locus program

```
for i
  for j
    for k
```
+

Locus system

1. Selects variants (avoid explosion)
2. Runs
3. Determine the best variant

Locus program with steps to the best variant found

Locus

- Semi-automatic approach to assist performance experts and code developers in the performance optimization of programs in C, C++, and Fortran

- Orchestrates the application of transformations to a baseline version of the code

- Specially for optimizing complex, long-lived applications running on different environments