Easy Counting and Ranking for Simple Loops

Alain KETTERLIN

IMPACT 2024: January, 17, 2024
Problem Statement

Integer Polynomials

Algorithms

Conclusion
Quantitative aspects of loops

for $i = 0$ to $n$
  S1
  for $j = 0$ to $i$
    S2
  S3
  for $j = i+1$ to $n$
    S4
    for $k = 0$ to $i$
      S5
    S6

▷ **Counting:** how many instruction executions overall?
  \[
  \left(\frac{5n+6n^2+n^3}{6}\right)
  \]

▷ **Ranking:** how many instructions before $S5(i,j,k)$?
  \[
  \left(\frac{6+(9n-7)i+(3n-9)i^2-2i^3+6(i+2)j+6k}{6}\right)
  \]

▷ **Unranking:** what is the instruction with rank $r$?

(cholesky from PolyBench v3)
Existing solutions in the polyhedral framework

- Clauss/Ehrhart: polynomials with periodic coefficients
- Verdoolaege/Barvinok: step-polynomials (in integer parts)

Pros

- extremely general/powerful solutions
- work at the polyhedral level

Cons

- unconventional form of the result
- work at the polyhedral level
Loops to sums to polynomials

for $i = 0$ to $n$ \[ \Rightarrow \sum_{i=0}^{n-1} (1 + \sum_{j=0}^{i-1} (1 + 1 + \sum_{k=0}^{i-1} (1 + 1)) = \left(\frac{5n+6n^2+n^3}{6}\right) \]

S1
- for $j = 0$ to $i$
  - S2
S3
- for $j = i+1$ to $n$
  - S4
- for $k = 0$ to $i$
  - S5
S6

Mechanical translation + algebra:
→ when is this possible/correct?
→ how to implement this?
Loop syntax

- parameters, with affine inequalities
- arbitrary nesting of loops with affine bounds
- no min/max, no mod or integer parts

Loop semantics

\[
\text{for } i=l \text{ to } u \ldots \Rightarrow \sum_{i=l}^{u-1} \ldots
\]

This only makes sense for *simple* loops, with:

1. *unit step*: loop counter incremented by 1
2. *bounds coherence*: \( l \leq u \) for every instance of the loop

The latter needs explicit verification
Problem Statement

Integer Polynomials

Algorithms

Conclusion
Representing integer polynomials

\[ \sum_{i=0}^{n} a_i x^i \]  

a ring for the coefficients + a monomial basis

Correctness: all polynomials are integer-valued
Completeness: all integer sequences can be represented

The problem with integer polynomials

<table>
<thead>
<tr>
<th></th>
<th>Correct</th>
<th>Complete</th>
<th>In practice</th>
</tr>
</thead>
<tbody>
<tr>
<td>( a_i \in \mathbb{Z} )</td>
<td>( \checkmark )</td>
<td>( \frac{x(x-1)}{2} )</td>
<td>essentially unusable</td>
</tr>
<tr>
<td>( a_i \in \mathbb{Q} )</td>
<td>( \frac{x(x-2)}{2} )</td>
<td>( \checkmark )</td>
<td>correctness proofs</td>
</tr>
</tbody>
</table>
\[ x^k \triangleq \binom{x}{k} = \frac{x \cdot (x - 1) \cdots (x - k + 1)}{k!} = \frac{x!}{k!(x-k)!} = \frac{x^k}{k!} \]

- \( x^k \) appears in Pascal’s triangle, with
  \[ (x + 1)^{k+1} = x^k + x^{k+1} \]
- defined for all \( x \in \mathbb{Z}, k \in \mathbb{Z}_{\geq 0} \)
  \[ (-x)^{k} = (-1)^k (x + k - 1)^k \]
- relation between the various powers:
  \[ x^n = \frac{1}{n!} \sum_{k=0}^{n} (-1)^{n-k} \binom{n}{k} x^k \]
  \[ x^n = \sum_{k=0}^{n} k! \{n\} x^k \]
  with \( \binom{n}{k} \) and \( \{n\} \) the unsigned Stirling numbers
triangles are to $x^{k\mid}$ what squares are to $x^k$

\[ x^0| = \emptyset \quad x^1| = \quad x^2| = \quad x^3| = \quad \ldots \]

triangular sums and loops

\[
\sum_{i_1=0}^{n-1} \sum_{i_2=0}^{i_1-1} \cdots \sum_{i_{d-1}=0}^{i_{d-2}-1} 1 = n^{d\mid}
\]

for $i=0$ to $n$

\[
\sum_{j=0}^{i-1} j^{d\mid} = \sum_{k=0}^{j-1} 1
\]

for $j=0$ to $i$

for $k=0$ to $j$

\[ S \]

for $i = 0$ to $n$

\[ S_1 \]

for $j = 0$ to $i$

\[ S_2 \]

for $j = i+1$ to $n$

\[ S_3 \]

for $k = 0$ to $i$

\[ S_4 \]

\[ S_5 \]

\[ S_6 \]
Polynomials = integer coefficients and binomial powers

\[ \sum_{i=0}^{n} a_i x^i \] with \( a_i \in \mathbb{Z} \)

this representation is correct and complete

Completeness via interpolation

For any sequence of integers \( v_0, \ldots, v_n \),

there is a unique interpolating polynomial

\[ p(x) = \sum_{i=0}^{n} a_i \cdot x^i \]

\begin{align*}
  v_0 &= p(0) = a_0 + a_1 \cdot 0^1 + a_2 \cdot 0^2 + a_3 \cdot 0^3 \ldots \\
  v_1 &= p(1) = a_0 + a_1 \cdot 1^1 + a_2 \cdot 1^2 + a_3 \cdot 1^3 \ldots \\
  v_2 &= p(2) = a_0 + a_1 \cdot 2^1 + a_2 \cdot 2^2 + a_3 \cdot 2^3 \ldots \\
  \vdots
\end{align*}

\[ a_i = \sum_{j=0}^{i} (-1)^{i-j} \cdot i^j \cdot v_j \]

Implementation note: no need for rational numbers
Variation and Summation

\[ \Delta x^{k+1} = (x + 1)^{k+1} - x^{k+1} = x^k \]

\[ \sum_{x=a}^{b-1} x^k = x^{k+1} \bigg|_a^b = b^{k+1} - a^{k+1} \]

Discrete calculus terminology

finite difference \( \Delta f(x) = f(x + 1) - f(x) \)
anti-difference \( \Delta^{-1} f(x) = \sum f(x) \)

Discrete analogs to: derivative, anti-derivative (or indefinite integral)
\[ \sum_{x=a}^{b-1} p(x) = \Delta^{-1} p(x) \bigg|_a^b = \Delta^{-1} p(b) - \Delta^{-1} p(a) \]

- given a loop with known per-iteration count: e.g.,
  
  for \( i=5 \) to \( N \)
  
  \( \ldots \) (executes \( 3i^1 + 7i^2 \) instructions)

- the total count of the loop is:

\[
\sum_{i=5}^{N-1} 3 \cdot i^1 + 7 \cdot i^2 \\
\downarrow \Delta^{-1}
\]

\[
= 3 \cdot i^2 + 7 \cdot i^3 \bigg|_5^N = (3 \cdot N^2 + 7 \cdot N^3) - (3 \cdot 5^2 + 7 \cdot 5^3) \\
=100
\]
\[ \sum_{x=a}^{b-1} p(x) = \Delta^{-1} p(x) \bigg|_{a}^{b} = \Delta^{-1} p(b) - \Delta^{-1} p(a) \]

- given a loop with known per-iteration count: e.g.,
  
  `for i=5 to N
  . . . (executes 3i^{1\text{st}} + 7i^{2\text{nd}} instructions)`

- the count before (the start of) an iteration \(i\) (the rank of . . .)

\[ \sum_{i=5}^{i-1} p(i) \quad \text{(hmm . . . )} \]

\[ = \Delta^{-1} p(i) \bigg|_{5}^{i} = (3 \cdot i^{2\text{nd}} + 7 \cdot i^{3\text{rd}}) - (3 \cdot 5^{2\text{nd}} + 7 \cdot 5^{3\text{rd}}) \]

\[ \Delta^{-1} p(5) \]
Abstract Syntax Trees

- strict alternation between loops and sequences of statements
  
  Loop := for id = expr to expr Seq
  
  Seq := do (Loop|name)+ done

- every statement and sequence is decorated with polynomials
▶ bottom-up traversal, post-order addition/summation

▶ on a basic instruction:

\[
S \rightarrow 1
\]

▶ on a sequence:

\[
\text{do} \quad \rightarrow c_0 + c_1 + \cdots \\
\quad \text{\hspace{0.5cm}} s_0 \quad \text{(with count } c_0) \\
\quad \text{\hspace{0.5cm}} s_1 \quad \text{(with count } c_1) \\
\quad \ldots \\
\text{done}
\]

▶ on a loop:

\[
\text{for } i=l \text{ to } u \quad \rightarrow \Delta^{-1}c(u) - \Delta^{-1}c(l) \\
\quad \text{\hspace{0.5cm}} \text{do (with count } c) \\
\quad \quad \ldots \\
\text{done}
\]
with n when n >= 0 (count)
do
  for i = 0 to n
do
    S1
    for j = 0 to i
do S2 done
    S3
    for j = 1+i to n
do
      S4
      for k = 0 to i
do S5 done
      S6
done
done
with n when n >= 0 (count)
do
  for i = 0 to n
do
    S1
    for j = 0 to i
do
      S2
    done
  S3
  for j = 1+i to n
do
    S4
    for k = 0 to i
do
      S5
    done
  S6
done
done
with n when n >= 0 do
  for i = 0 to n do
    for j = 0 to i do
      do S2 done
    done
  done
S1
for j = 1+i to n do
  for k = 0 to i do
    S4
  S5 done
S6
done
S3
done
S4
done
S5
done
with n when n >= 0 do
  for i = 0 to n do
    S1
    for j = 0 to i do S2 done
    S3
    for j = 1+i to n do
      S4
      for k = 0 to i do S5 done
    S6
  done
done

(count)

\[
\sum_{j=0}^{i-1} 1 = \sum_{j=0}^{i-1} \quad \text{(count)}
\]
with n when n >= 0

do

for i = 0 to n
do

<table>
<thead>
<tr>
<th>S1</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>for j = 0 to i</td>
<td>i</td>
</tr>
<tr>
<td>do S2 done</td>
<td>1</td>
</tr>
</tbody>
</table>

S3

for j = 1+i to n
do

<table>
<thead>
<tr>
<th>S4</th>
</tr>
</thead>
<tbody>
<tr>
<td>for k = 0 to i</td>
</tr>
<tr>
<td>do S5 done</td>
</tr>
</tbody>
</table>

S6
done
done
done
with n when n >= 0 (count)
do
    for i = 0 to n
do
        for j = 0 to i
do S1
        for j = 1+i to n
do S2
            for k = 0 to i
do S3
done
    done
        for k = 0 to i
do S4
done
    done
        for k = 0 to i
do S5
done
    done
        for k = 0 to i
do S6
done
    done
        for k = 0 to i
do S7
done
with n when n >= 0 (count)
do
  for i = 0 to n
do
    S1 1
    for j = 0 to i
      do S2 done
    S3 1
    for j = 1+i to n
      do
        S4 1
        for k = 0 to i
          do S5 done
      S6 done
done
    done
done
with n when n \geq 0 
\begin{align*}
\text{do} & \\
\text{for i = 0 to n} & \\
\text{do} & \\
\text{for j = 0 to i} & \\
\text{do S2 done} & \\
\text{S3} & 1 \\
\text{for j = 1+i to n} & \\
\text{do} & \\
\text{S4} & 1 \\
\text{for k = 0 to i} & \\
\text{do S5 done} & \\
\text{S6} & \\
\text{done} & \\
\text{done} & \\
\end{align*}
with n when n >= 0 (count)
do
  for i = 0 to n
    do
      S1
      for j = 0 to i
        do S2 done
      S3
      for j = 1+i to n
        do
          S4
          for k = 0 to i
            do S5 done
          S6
        done
    done
  done
done
with \( n \) when \( n \geq 0 \) (count) do
  for \( i = 0 \) to \( n \) do
    for \( j = 0 \) to \( i \) do S2 done
    done
  for \( j = 1+i \) to \( n \) do S4 done
  done
done

\[
\begin{array}{c|c}
S1 & 1 \\
\hline
\text{for } j = 0 \text{ to } i & i \\
\hline
\text{do S2 done} & 1 \\
\hline
S3 & 1 \\
\hline
\text{for } j = 1+i \text{ to } n & 2+i \\
\hline
\text{do S4 done} & 1 \\
\hline
S6 & 1 \\
\hline
\end{array}
\]
with \( n \) when \( n \geq 0 \)
do
  for \( i = 0 \) to \( n \)
do
    S1
    for \( j = 0 \) to \( i \)
do S2 done
    S3
    for \( j = 1+i \) to \( n \)
do S4
    for \( k = 0 \) to \( i \)
do S5 done
    S6
done
done

(\text{count})

\[
\begin{align*}
\text{S1} & \quad 1 \\
\text{for } j = 0 \text{ to } i & \quad i \\
& \quad \text{do } S2 \text{ done } \quad 1 \\
\text{S3} & \quad 1 \\
\text{for } j = 1+i \text{ to } n & \quad -2 + 2n - 4i + in - 2i^{2+1} \\
& \quad \text{do } S4 \text{ done } \quad 2 + i \\
\text{S4} & \quad 1 \\
\text{for } k = 0 \text{ to } i & \quad i \\
& \quad \text{do } S5 \text{ done } \quad 1 \\
\text{S5} & \quad 1 \\
\end{align*}
\]

\[
\sum_{j=1+i}^{n-1} (2 + i) = (2 + i) \cdot j^{\lfloor 1+i \rfloor} \left\lfloor \frac{n}{1+i} \right\rfloor
\]
with n when n >= 0
   do
      for i = 0 to n
         do
            S1
            for j = 0 to i
               do S2 done
            S3
            for j = 1 + i to n
               do S4 done
            S6
         done
      done
   done

\[
\begin{align*}
\text{(count)} = 1 + i + 1 + \cdots \\
2n - 3i + in - 2i^{2i} = 1 + i + 1 + \cdots
\end{align*}
\]
with \( n \) when \( n \geq 0 \)
do

\[
\text{for } i = 0 \text{ to } n \\
\text{do} \\
\quad \text{S1} \\
\quad \text{for } j = 0 \text{ to } i \\
\qquad \text{do S2 done} \\
\quad \text{S3} \\
\quad \text{for } j = 1+i \text{ to } n \\
\qquad \text{do S4} \\
\qquad \quad \text{for } k = 0 \text{ to } i \\
\qquad \quad \quad \text{do S5 done} \\
\qquad \text{S6} \\
\text{done}
\]

do

\[
\sum_{i=0}^{n-1} = 2n + 3n^2 + n^3 = 2n - 3i + in - 2i^2 + 2 + i
\]
with n when n >= 0 (count)
\[
\begin{align*}
\text{do} & \quad \text{for } i = 0 \text{ to } n \\
& \quad \text{do} \\
& \quad \quad \text{S1} \\
& \quad \quad \quad \text{for } j = 0 \text{ to } i \\
& \quad \quad \quad \quad \text{do } S2 \text{ done} \\
& \quad \quad \text{S3} \\
& \quad \quad \text{for } j = 1+i \text{ to } n \\
& \quad \quad \quad \text{do} \\
& \quad \quad \quad \quad S4 \\
& \quad \quad \quad \quad \quad \text{for } k = 0 \text{ to } i \\
& \quad \quad \quad \quad \quad \quad \text{do } S5 \text{ done} \\
& \quad \quad \quad \text{S6} \\
& \quad \quad \text{done} \\
& \quad \text{done} \\
& \quad \text{done}
\end{align*}
\]
\[
= \ldots
\]
\[
2n + 3n^2 + n^3
\]
with n when n >= 0 
do 
 for i = 0 to n 
do 
 S1 
 for j = 0 to i 
do S2 done 
 S3 
 for j = 1+i to n 
do 
 S4 
 for k = 0 to i 
do S5 done 
 S6 
done 
done 

(count) 
$2n + 3n^2 + n^3$

2 lines 
3 triangles 
1 pyramid
▶ top-down traversal, propagating ranks and using counts
→ when processing a node, assign ranks to all its children

▶ on a sequence:

\[
\begin{align*}
do & \quad (\text{with rank } r) \\
s_0 & \quad (\text{with count } c_0) \rightarrow r \\
s_1 & \quad (\text{with count } c_1) \rightarrow r + c_0 \\
s_2 & \quad (\text{with count } c_2) \rightarrow r + c_0 + c_1 \\
\ldots & \\
done
\end{align*}
\]

▶ on a loop

\[
\begin{align*}
\text{for } i=l \ldots & \quad (\text{with rank } r) \\
do & \quad (\text{with count } c) \rightarrow r + \Delta_i^{-1} c - \Delta_i^{-1} c(l) \\
\ldots & \\
done
\end{align*}
\]

▶ on a basic instruction: its rank is already set
with n when n \geq 0
\text{do}
\text{for } i = 0 \text{ to } n
\text{do}
S1
\text{for } j = 0 \text{ to } i
\text{do } S2 \text{ done}
S3
\text{for } j = 1+i \text{ to } n
\text{do}
S4
\text{for } k = 0 \text{ to } i
\text{do } S5 \text{ done}
S6
\text{done}
done
done

\text{the root of the AST is primed with rank 0}

\begin{align*}
\text{(rank)} &= 0 \\
\text{(count)} &= 2n - 3i + in - 2i^2
\end{align*}
with \( n \) when \( n \geq 0 \)

\[
\Rightarrow \text{do}
\]

\[
\text{for } i = 0 \text{ to } n
\]

\[
\text{do}
\]

\[
S1
\]

\[
\text{for } j = 0 \text{ to } i
\]

\[
\text{do } S2 \text{ done}
\]

\[
S3
\]

\[
\text{for } j = 1+i \text{ to } n
\]

\[
\text{do}
\]

\[
S4
\]

\[
\text{for } k = 0 \text{ to } i
\]

\[
\text{do } S5 \text{ done}
\]

\[
S6
\]

\[
\text{done}
\]

\[
\text{done}
\]

\[
\text{the } 1^{\text{st}} \text{ statement in a sequence inherits the rank of the sequence}
\]

\[
\begin{align*}
\text{(rank)} & : (count) \\
0 & : 2n - 3i + in - 2\lfloor i \rfloor \\
0 & : 1 \\
i & : i \\
1 & : 1 \\
i & : 1 \\
2 + i & : 1 \\
i & : 1 \\
1 & : 1
\end{align*}
\]
with n when n >= 0
do
⇒ for i = 0 to n
do
  S1
  for j = 0 to i
do S2 done
S3
for j = 1+i to n
do
  S4
  for k = 0 to i
do S5 done
S6
done
done

\[
\begin{align*}
\text{(rank)} & 0 \\
\text{(count)} & \Delta
\end{align*}
\]
\[
\begin{align*}
2in - 3i^2 + i^2n - 2i^3 & \quad + \Delta^{-1} \quad 2n - 3i + in - 2i^2 \\
\text{= rank}_{\text{for}} & \quad (=0 \text{ here}) \\
+ \Delta^{-1}_{i} \text{count}_{\text{do}} & \quad (\text{wrt } i) \\
- \Delta^{-1} \text{count}_{\text{do}(0)} & \quad (=0 \text{ here}) \\
\end{align*}
\]
\[
\begin{align*}
\text{2 + i} & 1 \\
\text{i} & 1 \\
\end{align*}
\]
with \( n \) when \( n \geq 0 \)
do  
  for \( i = 0 \) to \( n \)  
    \( \leadsto \) do
      S1
      for \( j = 0 \) to \( i \)  
        do S2 done
      S3
      for \( j = 1 + i \) to \( n \)  
        do S4
          for \( k = 0 \) to \( i \)  
            do S5 done
          S6
        done
    done
  done

\begin{align*}
\text{(rank)} & \quad \text{(count)} \\
0 & \quad 0 \\
2 in - 3 i^2 & + i^2 n - 2 i^3 & \quad 2 n - 3 i + in - 2 i^2 \\
2 in - 3 i^2 & + i^2 n - 2 i^3 & + \\
1 + 2 in - 3 i^2 & + i^2 n - 2 i^3 & + \\
1 & + 2 in - 3 i^2 & + i^2 n - 2 i^3 & + \\
2 + i & + 2 in - 3 i^2 & + i^2 n - 2 i^3 & + \\
S1 \text{ inherits the rank of the sequence} & \quad \text{others accumulate counts} \\
1 & \quad i \\
1 & \quad 1
\end{align*}
with n when n >= 0
do
  for i = 0 to n
do
    S1
  ➞ for j = 0 to i
do S2 done
S3
for j = 1+i to n
do
  S4
  for k = 0 to i
do S5 done
S6
done
done

(rank)
  0
  0
  2in - 3i^2| + i^2|n - 2i^3|
  2n - 3i + in - 2i^2|
  2in - 3i^2| + i^2|n - 2i^3|
  1 + 2in - 3i^2| + i^2|n - 2i^3| + j
  1 + i + 2in - 3i^2| + i^2|n - 2i^3|
  2 + i + 2in - 3i^2| + i^2|n - 2i^3|

(count)
  1
  1
  1

= \text{rank}_\text{for} + \Delta^{-1}_j\text{count}_\text{do} \quad \text{(wrt } j, \text{ gives } j) \\
- \Delta^{-1}_0\text{count}_\text{do}(0) \quad \text{(=0 here)}
with \( n \) when \( n \geq 0 \)
do  
  for \( i = 0 \) to \( n \)
do  
    S1  
    for \( j = 0 \) to \( i \)
do S2 done  
S3  
\[ \Rightarrow \]  
  for \( j = 1+i \) to \( n \)
do  
    S4  
    for \( k = 0 \) to \( i \)
do S5 done  
S6  
done  
  done

\[(\text{rank}) \quad \text{(count)}\]
\[
0 
0 
2in - 3i^2 + i^2n - 2i^3 
2n - 3i + in - 2i^2 
1 
1 
1 
1 
\]
\[
\Delta^{-1}2 + i \]
\[
= \text{rank_for} 
+ \Delta^{-1}\text{count_do} \quad \text{(wrt } j, \text{ gives } 2j + ji) \]
\[
- \Delta^{-1}\text{count_do}(1+i) \quad (= (2(1+i) + (1+i)i) \]
with $n$ when $n \geq 0$

do

for $i = 0$ to $n$
do

S1

for $j = 0$ to $i$
do S2 done

S3

for $j = 1+i$ to $n$
do  

S4

for $k = 0$ to $i$
do S5 done

S6
done
done

(rank)  

$0$

(count)

$0$

$2in - 3i^2 + i^2n - 2i^3$

$2in - 3i^2 + i^2n - 2i^3$

$2n - 3i + in - 2i^2$

$1$

$i$

$1 + 2in - 3i^2 + i^2n - 2i^3 + j$

$1 + i + 2in - 3i^2 + i^2n - 2i^3$

$1$

$2 + i + 2in - 3i^2 + i^2n - 2i^3$

$-3i + 2in - 5i^2 + i^2n - 2i^3 + 2j + ji$

$-3i + 2in - 5i^2 + i^2n - 2i^3 + 2j + ji$

$2 + i$

$1$

$1 - 3i + 2in - 5i^2 + i^2n - 2i^3 + 2j + ji$

$1 - 2i + 2in - 5i^2 + i^2n - 2i^3 + 2j + ji$

inheritance, plus a simple accumulation of counts
with \( n \) when \( n \geq 0 \)
do
  for \( i = 0 \) to \( n \)
done
S1
  for \( j = 0 \) to \( i \)
do
  S2
  done
S3
  for \( j = 1+i \) to \( n \)
do
  S4
  \( \Rightarrow \) for \( k = 0 \) to \( i \)
do
  S5
  done
S6
done
done

\[(\text{rank})\]
\begin{align*}
0 \\
0 \\
2in - 3i^2 + i^2n - 2i^3 \\
2n - 3i + in - 2i^2 \\
1 \\
i \\
1 \\
1 \\
1 \\
1 \\
2 + i \\
2 + i + j \\
1 \\
i \\
1 \\
\Rightarrow \Delta^{-1} - 1 \\
1 - 2i + 2in - 5i^2 + i^2n - 2i^3 + 2j + ji
\end{align*}

\[(\text{count})\]
\begin{align*}
= \text{rank}_{\text{for}} \\
+ \Delta_k^{-1} \text{count}_{\text{do}} \quad \text{(wrt } k, \text{ gives } k) \\
- \Delta^{-1} \text{count}_{\text{do}}(0) \quad (=0 \text{ here})
\end{align*}
with \( n \) when \( n \geq 0 \)
do
  for \( i = 0 \) to \( n \)
do
    S1
    for \( j = 0 \) to \( i \)
do S2 done
  S3
  for \( j = 1+i \) to \( n \)
do
    S4
    for \( k = 0 \) to \( i \)
do S5 done
  S6
done
done

Note: many loop-body ranks are \textbf{linear} in the counter of the loop
Polyhedral arrays of structure of... (in extended Pascal syntax): arrays as loops, records as sequences

data :
array [a:=0..n-1] of
    record
      s1: real;
      j1: array [b:=0 .. a-1] of real;
      s3: real;
      j2: array [b:=a+1 .. n-1] of
          record
            s4: real;
            k1: array [c:=0 .. a-1] of real;
            s6: real;
          end;
    end;
end;
Polyhedral arrays of structure of... (in extended Pascal syntax): arrays as loops, records as sequences, count as size

```pascal
data :
array [a:=0..n-1] of
  record
    s1: real;
    j1: array [b:=0 .. a-1] of real;
    s3: real;
  j2: array [b:=a+1 .. n-1] of
    record
      s4: real;
      k1: array [c:=0 .. a-1] of real;
      s6: real;
    end
  end
end;
```

position in this array

size of data[a].j2

\[ = -2 + 2n - 4a + an - 2a^2 \]
Polyhedral arrays of structure of... (in extended Pascal syntax): arrays as loops, records as sequences, count as size, rank as offset

data :
array [a:=0..n-1] of
record
  s1: real;
  j1: array [b:=0 .. a-1] of
      real;
  s3: real;
  j2: array [b:=a+1 .. n-1] of
      record
        s4: real;
        k1: array [c:=0 .. a-1] of
            real;
        s6: real;
      end;
end;

position in this array

size of data[a].j2 = $-2 + 2n - 4a + an - 2a^2$

offset of data[a].j2[b].k1[c] = $1 - 3a + 2an - 5a^2 + a^2n - 2a^3 + 2b + ba + c$
Polyhedral arrays of structure of... (in extended Pascal syntax): arrays as loops, records as sequences, count as size, rank as offset

data :
array [a:=0..n-1] of
  record
    s1: real;
    j1: array [b:=0 .. a-1] of real;
    s3: real;
  j2: array [b:=a+1 .. n-1] of
    record
      s4: real;
      k1: array [c:=0 .. a-1] of real;
      s6: real;
    end;
  end;
end;

position in this array

size of data[a].j2
= -2 + 2n - 4a + an - 2a^2

offset of data[a].j2[b].k1[c]
= 1 - 3a + 2an - 5a^2 + a^2n - 2a^3 + 2b + ba + c

This array has the shape of the Cholesky kernel iteration domain...
Polyhedral arrays of structure of... (in extended Pascal syntax): arrays as loops, records as sequences, count as size, rank as offset

```pascal
data :
array [a:=0..n-1] of record
  s1: real;
  j1: array [b:=0 .. a-1] of real;
  s3: real;
  j2: array [b:=a+1 .. n-1] of record
    s4: real;
    k1: array [c:=0 .. a-1] of real;
    s6: real;
  end;
end;
```

position in this array

size of data[a].j2

\[ = -2 + 2n - 4a + an - 2a^2 \]

offset of data[a].j2[b].k1[c]

\[ = 1 - 3a + 2an - 5a^2 + a^2n - 2a^3 + 2b + ba + c \]

This array has the shape of the Cholesky kernel iteration domain...

(my humble tribute to Niklaus Wirth)
Given a *valid* rank $R$ (a number)
Find a path down the AST to determine:

- the location of the instruction with rank $R$
- the values of the enclosing loop counters

1. On a sequence of statements:

   $$\text{do } s_0 \text{ (with rank } r_0) \text{ with } \max\{p \mid r_p(\vec{v}) \leq R\} \rightarrow \text{max } \{p \mid r_p(\vec{v}) \leq R\}$$
   $$s_1 \text{ (with rank } r_1)$$
   $$\ldots$$
   $$\text{done}$$

2. On a loop

   $$\text{for } i=l \text{ to } u \rightarrow \max\{i \mid r(\vec{v}, i) \leq R\}$$
   $$\text{do (with rank } r) \rightarrow \max\{i \mid r(\vec{v}, i) \leq R\}$$
   $$\ldots$$
   $$\text{done}$$

≡ a root-finding problem...
Given a valid rank $R$ (a number) and the values of the parameters
Find a path down the AST to determine:

- the location of the instruction with rank $R$
- the values of the enclosing loop counters

1. On a sequence of statements:
   
   ```
   do 
   $s_0$ (with rank $r_0$) 
   $s_1$ (with rank $r_1$) 
   . . . 
   done
   ```

   → $\max\{p \mid r_p(\vec{v}) \leq R\}$

   all variables in scope have known values (in $\vec{v}$)

   → simple scan

2. On a loop
   
   ```
   for i=l to u 
   do (with rank $r$) 
   . . . 
   done
   ```

   → $\max\{i \mid r(\vec{v}, i) \leq R\}$

   a root-finding problem requires numerical resolution
   ($r(\vec{v}, i)$ is univariate in $i$)
generate code computing the result, to be used at runtime

use a solver: \texttt{unisolve} \((p, l, u, R)\)
returns \(\max\{x \mid l \leq x < u \land p(x) \leq R\}\)

\begin{verbatim}
def dyn_unrank (n, RANK):
    ⇒ i = unisolve ([0, 2n, -3+n, -2], 0, n, RANK)
    if RANK < 1 + 2in - 3i^2 + i^2n - 2i^3:
        return ([i], [0, 0])
    elif RANK < 1 + i + 2in - 3i^2 + i^2n - 2i^3:
        ⇒ j = unisolve ([1 + 2in - 3i^2 + i^2n - 2i^3], 1, 0, i, RANK)
        return ([i, j], [0, 1, 0])
    elif RANK < 2 + i + 2in - 3i^2 + i^2n - 2i^3:
        return ([i], [0, 2])
    else:
        ⇒ j = unisolve ([-3i + 2in - 5i^2 + i^2n - 2i^3, 2 + i], 1 + i, n, RANK)
        if RANK < 1 - 3i + 2in - 5i^2 + i^2n - 2i^3 + 2j + ji:
            return ([i, j], [0, 3, 0])
        elif RANK < 1 - 2i + 2in - 5i^2 + i^2n - 2i^3 + 2j + ji:
            ⇒ k = unisolve ([1 - 3i + 2in - 5i^2 + i^2n - 2i^3 + 2j + ji, 1], 0, i, RANK)
            return ([i, j, k], [0, 3, 1, 0])
        else:
            return ([i, j], [0, 3, 2])
\end{verbatim}
generate code computing the result, to be used at runtime

use a solver: \texttt{unisolve}\((p, l, u, R)\)

returns\(\max\{x \mid l \leq x < u \land p(x) \leq R\}\)

```python
def dyn_unrank\((n, RANK)\):
    \(i = \text{unisolve}\([0, 2n, -3+n, -2], 0, n, RANK)\)
    if \(RANK < 1 + 2in - 3i^2 + i^2n - 2i^3:\)
        return ([\(i\], [\(0, 0\)])
    elif \(RANK < 1 + i + 2in - 3i^2 + i^2n - 2i^3:\)
        \(j = \text{unisolve}\([1 + 2in - 3i^2 + i^2n - 2i^3], 1], 0, i, RANK)\)
        return ([\(i, j\], [\(0, 1, 0\)])
    elif \(RANK < 2 + i + 2in - 3i^2 + i^2n - 2i^3:\)
        return ([\(i\], [\(0, 2\)])
    else:
        \(j = \text{unisolve}\([-3i + 2in - 5i^2 + i^2n - 2i^3], 2 + i], 1 + i, n, RANK)\)
        if \(RANK < 1 - 3i + 2in - 5i^2 + i^2n - 2i^3 + 2j + ji:\)
            return ([\(i, j\], [\(0, 3, 0\)])
        elif \(RANK < 1 - 2i + 2in - 5i^2 + i^2n - 2i^3 + 2j + ji:\)
            \(k = \text{unisolve}\([1 - 3i + 2in - 5i^2 + i^2n - 2i^3 + 2j + ji], 1], 0, i, RANK)\)
            return ([\(i, j, k\], [\(0, 3, 1, 0\)])
        else:
            return ([\(i, j\], [\(0, 3, 2\)])
```
Uniform random sampling: 10% with $n = 10 \rightarrow 27$ out of 275
Uniform random sampling: 10% with $n = 10 \rightarrow 27$ out of 275

Slicing: 4-way with $n = 10 \rightarrow 275 = 3 \times 69 + 68$
Problem Statement

Integer Polynomials

Algorithms

Conclusion
Conclusion

- **Pros**
  - simple mathematical foundations
  - efficient algorithms
  - lightweight implementation

- **Cons: strict restrictions on loops**
  1. unit step
  2. bounds coherence

- **Topics not covered in this talk**
  - multivariate integer polynomials
  - symbolic algebraic operations

- **Some trivial extensions:**
  - polynomial bounds
  - weighted instructions
Problem Statement
  Basic Strategy
  Simple Loops
Integer Polynomials
  Representation Mismatch
  Binomial powers
  Sums of Polynomials

Algorithms
  Counting
  Ranking
  Rank Inversion

Conclusion