A Transformational Approach to Resource Analysis with Typed-Norms

Elvira Albert[1], Samir Genaim[1], and Raúl Gutiérrez[2,3]

[1]Universidad Complutense de Madrid

[2]Universidad Politécnica de Madrid

[3]Universitat Politècnica de València

Universidad Complutense de Madrid
June 14, 2016
Madrid, Spain

Motivation

- Automated resource analysis needs to infer how the sizes of data are modified along program's execution.
- Norms define how the size of term are computed (*list-length*, *tree-depth*, *term-size*, ...).
- Type-norms allow defining norms based on type information.
- The choice of the right norm is crucial to obtain a proof for termination and get accurate upper and lower bounds.
- Allowing multiple norms simultaneously give us more chances to success, but the efficiency of the analysis can be degraded considerably.

Main Contribution

1. We propose a transformational approach to use multiple type-norms in resource analysis:
   1. transform the program into an intermediate abstract program in which the relation among data and control flow are preserved,
   2. from such intermediate abstract program, together with a the size relations, we obtain a cost relation system, in which each variable is abstracted with respect to all considered norms valid for its type,
   3. if a solution if obtained from the cost relation system constraints (using a constraint solver), then the solution is transformed into upper and lower resource bounds automatically.
2. we outline an algorithm for the inference of typed-norms which can detect which norms are useful to later infer the resource consumption, and discard norms that are useless for this purpose.

Outline

Example: A Simple Functional Language

```
1 def Int fibonacci(Int n)
2   = case (n <= 0) {
3       true => case n {
4                 0 => 0 ;
5                 1 => 1 ;
6                 m => fibonacci(m - 1) + fibonacci(m - 2) ;
7       } ;
8       false => -1 ;
9     };
10
11 def List<Int> fibonacci_list(List<Int> l)
12   = case l {
13       Nil => Nil ;
14       Cons(n,rest_l) => Cons(fibonacci(n),fibonacci_list(rest_l)) ;
15   };
```

Intermediate Form: An Abstract Syntax

$$
\begin{array}{lcl}
r & ::= & m(\bar{x}, y) \mapsto g, b_1, \ldots, b_n. \\
b & ::= & x{:=}t \mid \ m(\bar{x}, y) \\
g & ::= & \mathit{true} \mid g \wedge g \mid e \ \mathit{op} \ e \mid \mathit{match}(x, t) \mid \mathit{nonmatch}(x, t) \\
t & ::= & e \mid \mathit{Co}(\bar{t}) \\
e & ::= & x \mid n \mid e{+}e \mid e{-}e
\end{array}
$$

where $\mathit{op} \in \{>, =, \geq\}$, $m(\bar{x}, \bar{y})$ is the *head* of the rule, g specifies the conditions for the rule to be applicable and $b_1, \ldots, b_n$ is the rule's *body*.

Example: Intermediate Representation

fibonacci_list(xs,ys)
  $\mapsto$ case$_0$(xs,ys).
case$_0$(xs,ys)
  $\mapsto$ match(xs,Nil),
    ys := Nil.
case$_0$(xs,ys)
  $\mapsto$ nonmatch(xs,Nil),
    match(xs,Cons(n,zs)),
    fibonacci(n,x),
    fibonacci_list(zs,ws),
    ys := Cons(x,ws).
fibonacci(n,m)
  $\mapsto$ x := 0,
    case$_1$(x,n,m).
case$_1$(x,n,m)
  $\mapsto$ $n \geq x$,
    case$_2$(n,m).
case$_1$(x,n,m)
  $\mapsto$ $n < x$,
    m := -1.

case$_2$(n,m)
  $\mapsto$ match(n,0),
    m := 0.
case$_2$(n,m)
  $\mapsto$ nonmatch(n,0),
    match(n,1),
    m := 1.
case$_2$(n,m)
  $\mapsto$ nonmatch(n,0),
    nonmatch(n,1),
    match(n,x),
    y := x - 1,
    fibonacci(y,z),
    v := x - 2,
    fibonacci(v,w),
    m := $z + w$.

Outline

Term Size Norm

$$\|t\|_{ts} = \begin{cases} 1 + \sum_{i=1}^{n} \|t_i\|_{ts} & \text{if } t = Co(t_1, \ldots, t_n) \\ 1 & \text{otherwise} \end{cases}$$

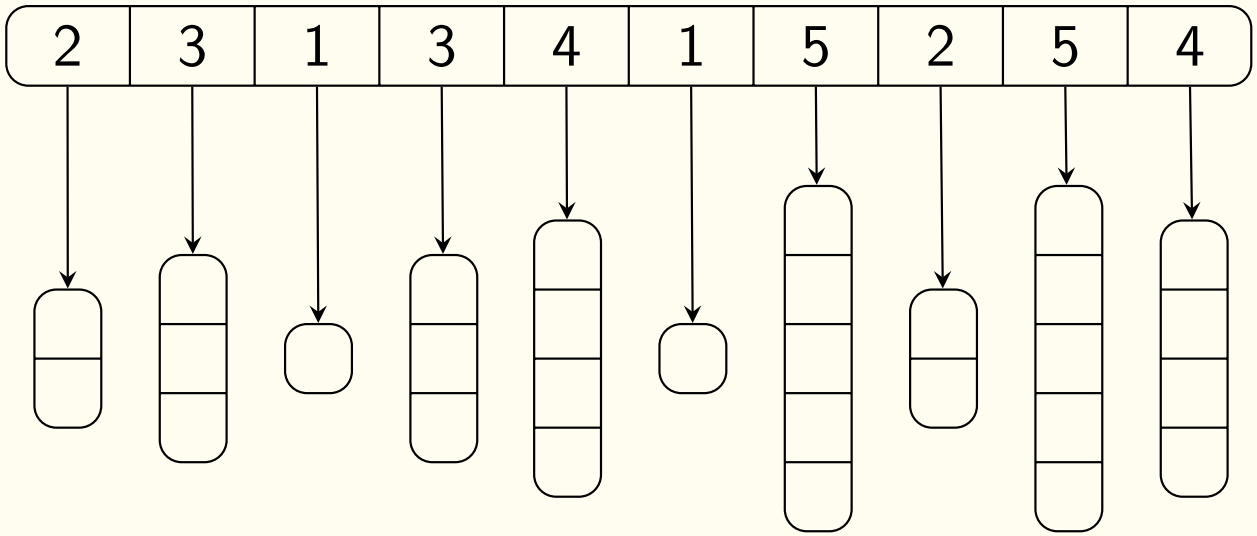
Example: Fibonacci List

2	3	1	3	4	1	5	2	5	4

Term Size Norm

$$\|t\|_{ts} = \begin{cases} 1 + \sum_{i=1}^{n} \|t_i\|_{ts} & \text{if } t = Co(t_1, \ldots, t_n) \\ 1 & \text{otherwise} \end{cases}$$

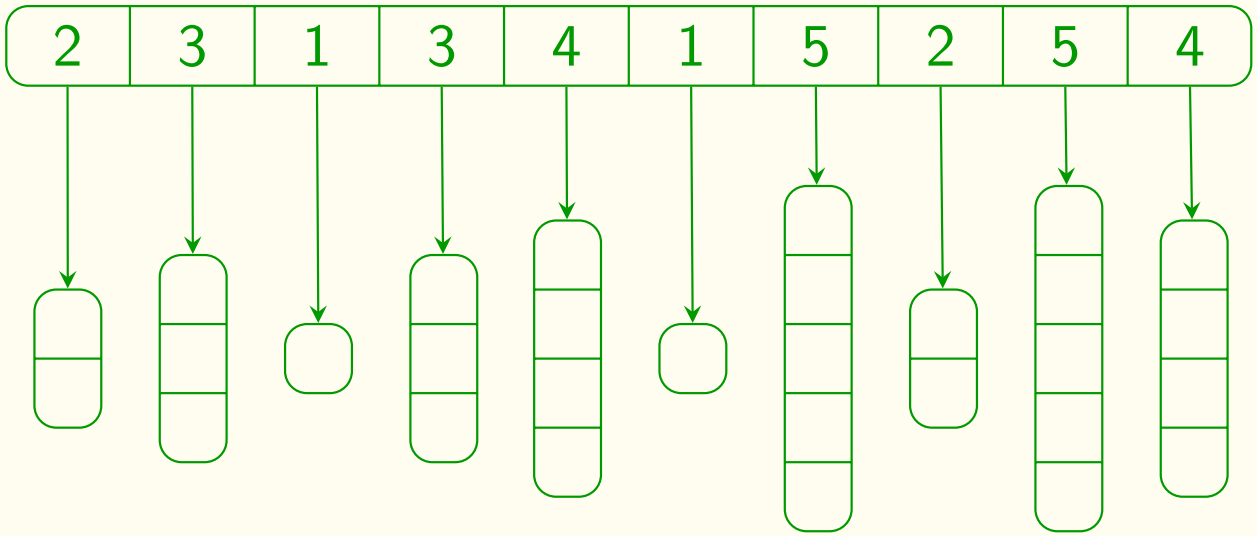
Example: Fibonacci List



Term Size Norm

$$\|t\|_{ts} = \begin{cases} 1 + \sum_{i=1}^{n} \|t_i\|_{ts} & \text{if } t = Co(t_1, \ldots, t_n) \\ 1 & \text{otherwise} \end{cases}$$

Example: Fibonacci List



Term Size Norm

$$\|t\|_{ts} = \begin{cases} 1 + \sum_{i=1}^{n} \|t_i\|_{ts} & \text{if } t = Co(t_1, \ldots, t_n) \\ 1 & \text{otherwise} \end{cases}$$

Example: Fibonacci List

2	3	1	3	4	1	5	2	5	4

Complexity (Upper Bound): Fibonacci List

$$O(l_{ts} \times 2^{l_{ts}})$$

Typed-Norms

$$\|t\|_\sigma = \begin{cases} t & \sigma = \texttt{Int} \text{ and } t \text{ is an integer} \\ \mathit{length}(t) & \sigma = \texttt{String} \text{ and } t \text{ is an string} \\ 1 + \sum_{i=1}^{n} \|t_i\|_\sigma & \text{if } t = \mathit{Co}(t_1, \ldots, t_n) \text{ and } \mathit{type}(t) = \sigma \\ \sum_{i=1}^{n} \|t_i\|_\sigma & \text{if } t = \mathit{Co}(t_1, \ldots, t_n) \text{ and } \mathit{type}(t) \neq \sigma \end{cases}$$

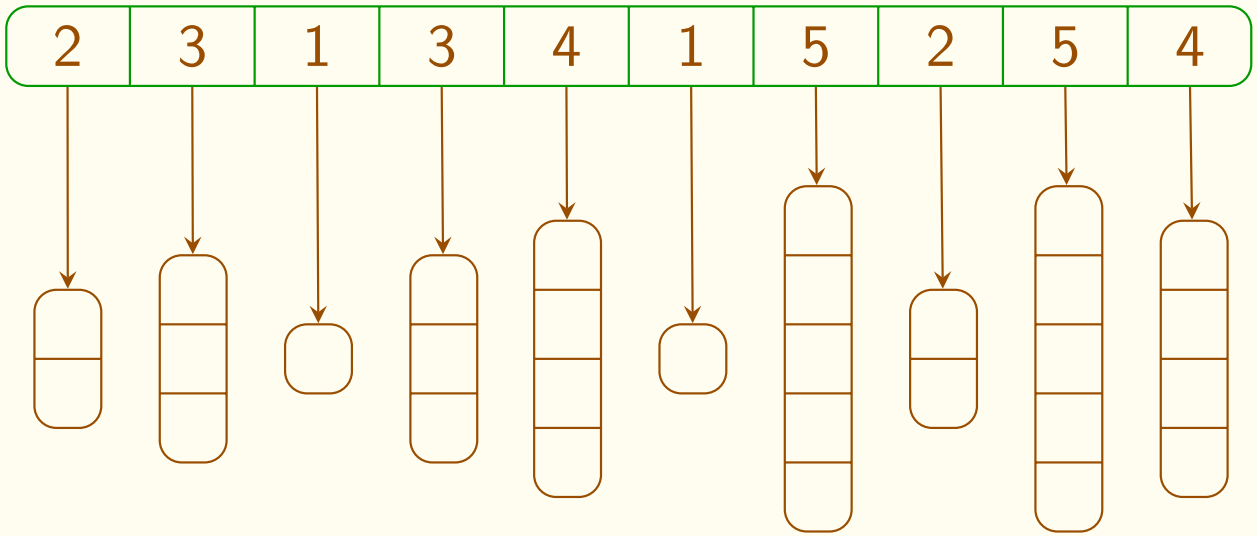
Example: Fibonacci List

2	3	1	3	4	1	5	2	5	4

Typed-Norms

$$\|t\|_\sigma = \begin{cases} t & \sigma = \texttt{Int} \text{ and } t \text{ is an integer} \\ \mathit{length}(t) & \sigma = \texttt{String} \text{ and } t \text{ is an string} \\ 1 + \sum_{i=1}^{n} \|t_i\|_\sigma & \text{if } t = \mathit{Co}(t_1, \ldots, t_n) \text{ and } \mathit{type}(t) = \sigma \\ \sum_{i=1}^{n} \|t_i\|_\sigma & \text{if } t = \mathit{Co}(t_1, \ldots, t_n) \text{ and } \mathit{type}(t) \neq \sigma \end{cases}$$

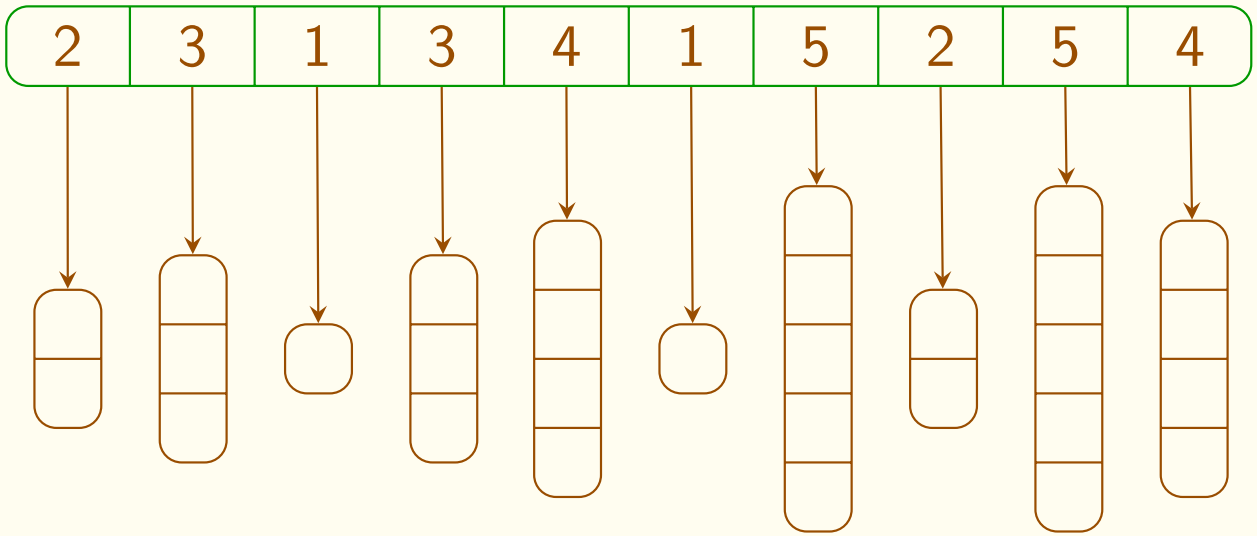
Example: Fibonacci List



Typed-Norms

$$\|t\|_\sigma = \begin{cases} t & \sigma = \texttt{Int} \text{ and } t \text{ is an integer} \\ \mathit{length}(t) & \sigma = \texttt{String} \text{ and } t \text{ is an string} \\ 1 + \sum_{i=1}^{n} \|t_i\|_\sigma & \text{if } t = \mathit{Co}(t_1, \ldots, t_n) \text{ and } \mathit{type}(t) = \sigma \\ \sum_{i=1}^{n} \|t_i\|_\sigma & \text{if } t = \mathit{Co}(t_1, \ldots, t_n) \text{ and } \mathit{type}(t) \neq \sigma \end{cases}$$

Example: Fibonacci List



Typed-Norms

$$\|t\|_\sigma = \begin{cases} t & \sigma = \texttt{Int} \text{ and } t \text{ is an integer} \\ \mathit{length}(t) & \sigma = \texttt{String} \text{ and } t \text{ is an string} \\ 1 + \sum_{i=1}^{n} \|t_i\|_\sigma & \text{if } t = \mathit{Co}(t_1, \ldots, t_n) \text{ and } \mathit{typet} = \sigma \\ \max_{i=1}^{n} \|t_i\|_\sigma & \text{if } t = \mathit{Co}(t_1, \ldots, t_n) \text{ and } \mathit{typet} \neq \sigma \end{cases}$$

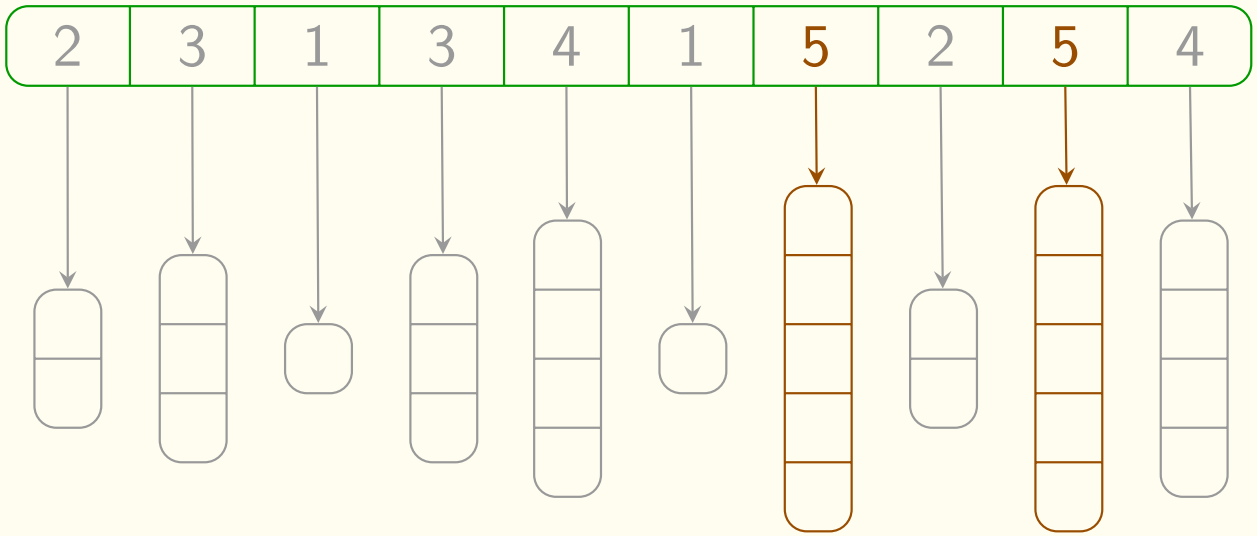
Example: Fibonacci List

2	3	1	3	4	1	5	2	5	4

Typed-Norms

$$\|t\|_\sigma = \begin{cases} t & \sigma = \texttt{Int} \text{ and } t \text{ is an integer} \\ \mathit{length}(t) & \sigma = \texttt{String} \text{ and } t \text{ is an string} \\ 1 + \sum_{i=1}^{n} \|t_i\|_\sigma & \text{if } t = \mathit{Co}(t_1, \ldots, t_n) \text{ and } \mathit{typet} = \sigma \\ \max_{i=1}^{n} \|t_i\|_\sigma & \text{if } t = \mathit{Co}(t_1, \ldots, t_n) \text{ and } \mathit{typet} \neq \sigma \end{cases}$$

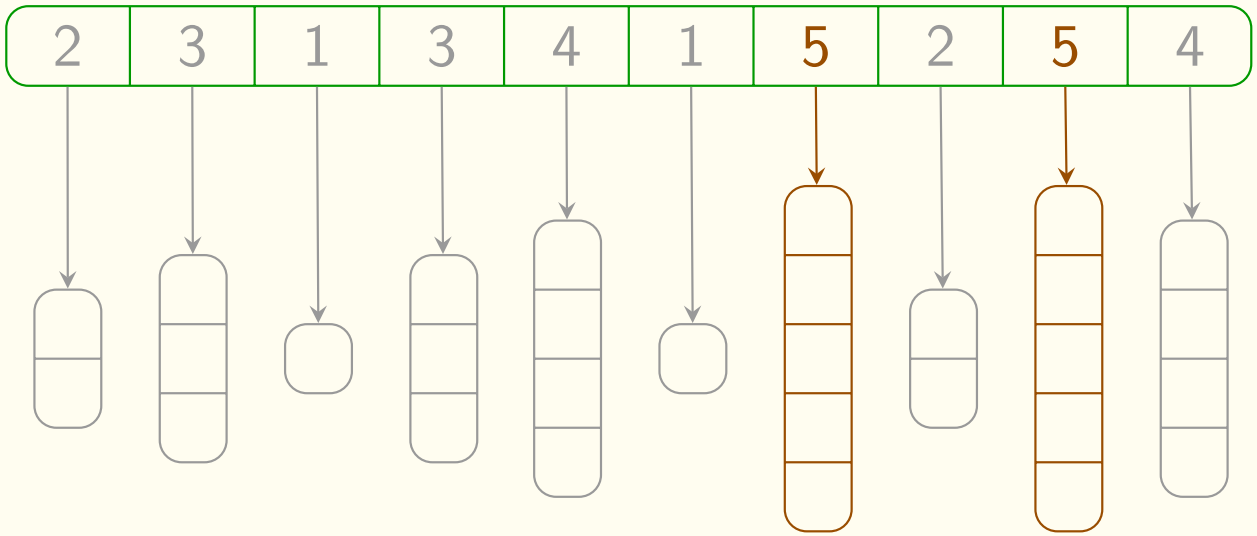
Example: Fibonacci List



Typed-Norms

$$\|t\|_\sigma = \begin{cases} t & \sigma = \texttt{Int} \text{ and } t \text{ is an integer} \\ \mathit{length}(t) & \sigma = \texttt{String} \text{ and } t \text{ is an string} \\ 1 + \sum_{i=1}^{n} \|t_i\|_\sigma & \text{if } t = \mathit{Co}(t_1, \ldots, t_n) \text{ and } \mathit{typet} = \sigma \\ \max_{i=1}^{n} \|t_i\|_\sigma & \text{if } t = \mathit{Co}(t_1, \ldots, t_n) \text{ and } \mathit{typet} \neq \sigma \end{cases}$$

Example: Fibonacci List



Size Abstraction Transformation

- Given a program P, its size abstraction P^α is a program obtained by replacing each rule $p(\overline{x}, \overline{y}) \mapsto g, b_1, \ldots, b_n \in P$ by $p(\overline{X}, \overline{Y}) \mapsto g^\alpha, b_1^\alpha, \ldots, b_n^\alpha$ where:
  - $g_1 \wedge g_2$ is transformed into $g_1^\alpha \wedge g_2^\alpha$,
  - $nonmatch(x, t)$ is transformed into $true$,
  - $p(\bar{x}, \bar{y})$ is transformed into $p(\bar{X}, \bar{Y})$,
  - $match(x, t)$ and $x := t$ is transformed into $\wedge\{X_\sigma = \|t\|_\sigma \mid \sigma \in \texttt{typed\_norms}(x)\}$, where $\texttt{typed\_norms}(x)$ corresponds to the set of types we want x to be measured,
  - $e_1\ op\ e_2$ is transformed into $(e_1\ op\ e_2)[y/Y_{\texttt{Int}}]$ if $\texttt{Int} \in \texttt{typed\_norms}(x)$ and $true$ otherwise.

Example: Size Abstraction

$\mathrm{fibonacci\_list}(xs,ys)$
$\quad \mapsto \quad \mathrm{case}_0(xs,ys).$
$\mathrm{case}_0(xs,ys)$
$\quad \mapsto \quad \mathrm{match}(xs,\mathrm{Nil}),$
$\qquad ys := \mathrm{Nil}.$
$\mathrm{case}_0(xs,ys)$
$\quad \mapsto \quad \mathrm{nonmatch}(xs,\mathrm{Nil}),$
$\qquad \mathrm{match}(xs,\mathrm{Cons}(n,zs))$
$\qquad \mathrm{fibonacci}(n,x),$
$\qquad \mathrm{fibonacci\_list}(zs,ws),$
$\qquad ys := \mathrm{Cons}(x,ws)$
$\mathrm{fibonacci}(n,m)$
$\quad \mapsto \quad x := 0,$
$\qquad \mathrm{case}_1(x,n,m).$
$\mathrm{case}_1(x,n,m)$
$\quad \mapsto \quad n \geq x,$
$\qquad \mathrm{case}_2(n,m).$

$\mathrm{case}_1(x,n,m)$
$\quad \mapsto \quad n < x,$
$\qquad m := \text{-}1.$
$\mathrm{case}_2(n,m)$
$\quad \mapsto \quad \mathrm{match}(n,0),$
$\qquad m := 0.$
$\mathrm{case}_2(n,m)$
$\quad \mapsto \quad \mathrm{nonmatch}(n,0),$
$\qquad \mathrm{match}(n,1),$
$\qquad m := 1.$
$\mathrm{case}_2(n,m)$
$\quad \mapsto \quad \mathrm{nonmatch}(n,0),$
$\qquad \mathrm{nonmatch}(n,1),$
$\qquad \mathrm{match}(n,x),$
$\qquad y := x \text{ - } 1,$
$\qquad \mathrm{fibonacci}(y,z),$
$\qquad v := x \text{ - } 2,$
$\qquad \mathrm{fibonacci}(v,w),$
$\qquad m := z + w.$

Example: Size Abstraction

fibonacci_list(xs,ys)
  $\mapsto$ case$_0$(xs,ys).
case$_0$(xs,ys)
  $\mapsto$ match(xs,Nil),
    ys := Nil.
case$_0$(xs,ys)
  $\mapsto$ nonmatch(xs,Nil),
    match(xs,Cons(n,zs))
    fibonacci(n,x),
    fibonacci_list(zs,ws),
    ys := Cons(x,ws)
fibonacci(n,m)
  $\mapsto$ x := 0,
    case$_1$(x,n,m).
case$_1$(x,n,m)
  $\mapsto$ $n \geq x$,
    case$_2$(n,m).

fibonacci_list(XS_I,XS_L,YS_I,YS_L)
  $\mapsto$ case$_0$(XS_I,XS_L,YS_I,YS_L).

case$_1$(x,n,m)
  $\mapsto$ $n < x$,
    m := -1.
case$_2$(n,m)
  $\mapsto$ nonmatch(n,0),
    match(n,1),
    m := 1.
case$_2$(n,m)
  $\mapsto$ nonmatch(n,0),
    nonmatch(n,1),
    match(n,x),
    y := x - 1,
    fibonacci(y,z),
    v := x - 2,
    fibonacci(v,w),
    m := $z + w$.

Example: Size Abstraction

$\text{fibonacci\_list}(XS_I, XS_L, YS_I, YS_L)$
$\quad \mapsto \quad \text{case}_0(XS_I, XS_L, YS_I, YS_L).$

$\text{case}_0(xs, ys)$
$\quad \mapsto \quad \text{match}(xs, \text{Nil}),$
$\quad\quad\quad ys := \text{Nil}.$

$\text{case}_0(xs, ys)$
$\quad \mapsto \quad \text{nonmatch}(xs, \text{Nil}),$
$\quad\quad\quad \text{match}(xs, \text{Cons}(n, zs))$
$\quad\quad\quad \text{fibonacci}(n, x),$
$\quad\quad\quad \text{fibonacci\_list}(zs, ws),$
$\quad\quad\quad ys := \text{Cons}(x, ws)$

$\text{fibonacci}(n, m)$
$\quad \mapsto \quad x := 0,$
$\quad\quad\quad \text{case}_1(x, n, m).$

$\text{case}_1(x, n, m)$
$\quad \mapsto \quad n \geq x,$
$\quad\quad\quad \text{case}_2(n, m).$

$\text{case}_1(x, n, m)$
$\quad \mapsto \quad n < x,$
$\quad\quad\quad m := \text{-}1.$

$\text{case}_0(XS_I, XS_L, YS_I, YS_L)$
$\quad \mapsto \quad \{XS_L = 1\},$

$\text{case}_2(n, m)$
$\quad \mapsto \quad \text{nonmatch}(n, 0),$
$\quad\quad\quad \text{match}(n, 1),$
$\quad\quad\quad m := 1.$

$\text{case}_2(n, m)$
$\quad \mapsto \quad \text{nonmatch}(n, 0),$
$\quad\quad\quad \text{nonmatch}(n, 1),$
$\quad\quad\quad \text{match}(n, x),$
$\quad\quad\quad y := x \text{ - } 1,$
$\quad\quad\quad \text{fibonacci}(y, z),$
$\quad\quad\quad v := x \text{ - } 2,$
$\quad\quad\quad \text{fibonacci}(v, w),$
$\quad\quad\quad m := z + w.$

Example: Size Abstraction

$\text{fibonacci\_list}(XS_I, XS_L, YS_I, YS_L)$
$\quad \mapsto \quad \text{case}_0(XS_I, XS_L, YS_I, YS_L).$
$\text{case}_0(XS_I, XS_L, YS_I, YS_L)$
$\quad \mapsto \quad \{XS_L = 1\},$
$\qquad ys := \text{Nil}.$
$\text{case}_0(xs, ys)$
$\quad \mapsto \quad \text{nonmatch}(xs, \text{Nil}),$
$\qquad \text{match}(xs, \text{Cons}(n, zs))$
$\qquad \text{fibonacci}(n, x),$
$\qquad \text{fibonacci\_list}(zs, ws),$
$\qquad ys := \text{Cons}(x, ws)$
$\text{fibonacci}(n, m)$
$\quad \mapsto \quad x := 0,$
$\qquad \text{case}_1(x, n, m).$
$\text{case}_1(x, n, m)$
$\quad \mapsto \quad n \geq x,$
$\qquad \text{case}_2(n, m).$

$\text{case}_1(x, n, m)$
$\quad \mapsto \quad n < x,$
$\qquad m := \text{-}1.$
$\text{case}_2(n, m)$
$\quad \{YS_L = 1\},$
$\qquad m := 0.$
$\text{case}_2(n, m)$
$\quad \mapsto \quad \text{nonmatch}(n, 0),$
$\qquad \text{match}(n, 1),$
$\qquad m := 1.$
$\text{case}_2(n, m)$
$\quad \mapsto \quad \text{nonmatch}(n, 0),$
$\qquad \text{nonmatch}(n, 1),$
$\qquad \text{match}(n, x),$
$\qquad y := x \text{ - } 1,$
$\qquad \text{fibonacci}(y, z),$
$\qquad v := x \text{ - } 2,$
$\qquad \text{fibonacci}(v, w),$
$\qquad m := z + w.$

Example: Size Abstraction

$\text{fibonacci\_list}(XS_I, XS_L, YS_I, YS_L)$
$\quad \mapsto \quad \text{case}_0(XS_I, XS_L, YS_I, YS_L).$

$\text{case}_0(XS_I, XS_L, YS_I, YS_L)$
$\quad \mapsto \quad \{XS_L = 1\},$
$\quad\quad\quad\quad \{YS_L = 1\}.$

$\text{case}_0(xs, ys)$
$\quad \mapsto \quad \text{nonmatch}(xs, \text{Nil}),$
$\quad\quad\quad\quad \text{match}(xs, \text{Cons}(n, zs))$
$\quad\quad\quad\quad \text{fibonacci}(n, x),$
$\quad\quad\quad\quad \text{fibonacci\_list}(zs, ws),$
$\quad\quad\quad\quad ys := \text{Cons}(x, ws)$

$\text{fibonacci}(n, m)$
$\quad \mapsto \quad x := 0,$
$\quad\quad\quad\quad \text{case}_1(x, n, m).$

$\text{case}_1(x, n, m)$
$\quad \mapsto \quad n \geq x,$
$\quad\quad\quad\quad \text{case}_2(n, m).$

$\text{case}_1(x, n, m)$
$\quad \mapsto \quad n < x,$
$\quad\quad\quad\quad m := \text{-}1.$

$\text{case}_0(XS_I, XS_L, YS_I, YS_L)$
$\quad \mapsto \quad \text{true},$

$\text{case}_2(n, m)$
$\quad \mapsto \quad \text{nonmatch}(n, 0),$
$\quad\quad\quad\quad \text{match}(n, 1),$
$\quad\quad\quad\quad m := 1.$

$\text{case}_2(n, m)$
$\quad \mapsto \quad \text{nonmatch}(n, 0),$
$\quad\quad\quad\quad \text{nonmatch}(n, 1),$
$\quad\quad\quad\quad \text{match}(n, x),$
$\quad\quad\quad\quad y := x \text{ - } 1,$
$\quad\quad\quad\quad \text{fibonacci}(y, z),$
$\quad\quad\quad\quad v := x \text{ - } 2,$
$\quad\quad\quad\quad \text{fibonacci}(v, w),$
$\quad\quad\quad\quad m := z + w.$

Example: Size Abstraction

$\text{fibonacci\_list}(XS_I, XS_L, YS_I, YS_L)$
$\quad \mapsto \quad \text{case}_0(XS_I, XS_L, YS_I, YS_L).$

$\text{case}_0(XS_I, XS_L, YS_I, YS_L)$
$\quad \mapsto \quad \{XS_L = 1\},$
$\quad\quad\quad\quad \{YS_L = 1\}.$

$\text{case}_0(XS_I, XS_L, YS_I, YS_L)$
$\quad \mapsto \quad \text{match}(xs, \text{Cons}(n, zs))$
$\quad\quad\quad\quad \text{fibonacci}(n, x),$
$\quad\quad\quad\quad \text{fibonacci\_list}(zs, ws),$
$\quad\quad\quad\quad ys := \text{Cons}(x, ws)$

$\text{fibonacci}(n, m)$
$\quad \mapsto \quad x := 0,$
$\quad\quad\quad\quad \text{case}_1(x, n, m).$

$\text{case}_1(x, n, m)$
$\quad \mapsto \quad n \geq x,$
$\quad\quad\quad\quad \text{case}_2(n, m).$

$\text{case}_1(x, n, m)$
$\quad \mapsto \quad n < x,$
$\quad\quad\quad\quad m := \text{-}1.$

$\{XS_I \geq N_I,\ XS_I \geq ZS_I,$
$XS_L = ZS_L + 1,\ ZS_L \geq 1\},$

$\text{case}_2(n, m)$
$\quad \mapsto \quad \text{nonmatch}(n, 0),$
$\quad\quad\quad\quad \text{match}(n, 1),$
$\quad\quad\quad\quad m := 1.$

$\text{case}_2(n, m)$
$\quad \mapsto \quad \text{nonmatch}(n, 0),$
$\quad\quad\quad\quad \text{nonmatch}(n, 1),$
$\quad\quad\quad\quad \text{match}(n, x),$
$\quad\quad\quad\quad y := x \text{ - } 1,$
$\quad\quad\quad\quad \text{fibonacci}(y, z),$
$\quad\quad\quad\quad v := x \text{ - } 2,$
$\quad\quad\quad\quad \text{fibonacci}(v, w),$
$\quad\quad\quad\quad m := z + w.$

Example: Size Abstraction

$$
\begin{array}{l}
\text{fibonacci\_list}(XS_I, XS_L, YS_I, YS_L) \\
\quad \mapsto \quad \text{case}_0(XS_I, XS_L, YS_I, YS_L). \\
\text{case}_0(XS_I, XS_L, YS_I, YS_L) \\
\quad \mapsto \quad \{XS_L = 1\}, \\
\qquad\quad \{YS_L = 1\}. \\
\text{case}_0(XS_I, XS_L, YS_I, YS_L) \\
\quad \mapsto \quad \{XS_I \geq N_I,\ XS_I \geq ZS_I, \\
\qquad\quad XS_L = ZS_L + 1,\ ZS_L \geq 1\}, \\
\qquad\quad \text{fibonacci}(n, x), \\
\qquad\quad \text{fibonacci\_list}(zs, ws), \\
\qquad\quad ys := \text{Cons}(x, ws) \\
\text{fibonacci}(n, m) \\
\quad \mapsto \quad x := 0, \\
\qquad\quad \text{case}_1(x, n, m). \\
\text{case}_1(x, n, m) \\
\quad \mapsto \quad n \geq x, \\
\qquad\quad \text{case}_2(n, m).
\end{array}
$$

$$
\begin{array}{l}
\text{case}_1(x, n, m) \\
\quad \mapsto \quad n < x, \\
\qquad\quad m := \text{-}1. \\
\text{case}_2(n, m) \\
\qquad\quad \text{fibonacci}(N_I, X_I), \\
\qquad\quad m := 0. \\
\text{case}_2(n, m) \\
\quad \mapsto \quad \text{nonmatch}(n, 0), \\
\qquad\quad \text{match}(n, 1), \\
\qquad\quad m := 1. \\
\text{case}_2(n, m) \\
\quad \mapsto \quad \text{nonmatch}(n, 0), \\
\qquad\quad \text{nonmatch}(n, 1), \\
\qquad\quad \text{match}(n, x), \\
\qquad\quad y := x \text{ - } 1, \\
\qquad\quad \text{fibonacci}(y, z), \\
\qquad\quad v := x \text{ - } 2, \\
\qquad\quad \text{fibonacci}(v, w), \\
\qquad\quad m := z + w.
\end{array}
$$

Example: Size Abstraction

$\text{fibonacci\_list}(XS_I, XS_L, YS_I, YS_L)$
$\quad \mapsto \quad \text{case}_0(XS_I, XS_L, YS_I, YS_L).$

$\text{case}_0(XS_I, XS_L, YS_I, YS_L)$
$\quad \mapsto \quad \{XS_L = 1\},$
$\quad\quad\quad\quad \{YS_L = 1\}.$

$\text{case}_0(XS_I, XS_L, YS_I, YS_L)$
$\quad \mapsto \quad \{XS_I \geq N_I,\ XS_I \geq ZS_I,$
$\quad\quad\quad\quad XS_L = ZS_L + 1,\ ZS_L \geq 1\},$
$\quad\quad\quad\quad \text{fibonacci}(N_I, X_I),$
$\quad\quad\quad\quad \text{fibonacci\_list}(zs, ws),$
$\quad\quad\quad\quad ys := \text{Cons}(x, ws)$

$\text{fibonacci}(n, m)$
$\quad \mapsto \quad x := 0,$
$\quad\quad\quad\quad \text{case}_1(x, n, m).$

$\text{case}_1(x, n, m)$
$\quad \mapsto \quad n \geq x,$
$\quad\quad\quad\quad \text{case}_2(n, m).$

$\text{case}_1(x, n, m)$
$\quad \mapsto \quad n < x,$
$\quad\quad\quad\quad m := \text{-}1.$

$\text{case}_2(n, m)$
$\quad\quad\quad\quad \text{fibonacci\_list}(ZS_I, ZS_L, WS_I, WS_L),$
$\quad\quad\quad\quad m := 0.$

$\text{case}_2(n, m)$
$\quad \mapsto \quad \text{nonmatch}(n, 0),$
$\quad\quad\quad\quad \text{match}(n, 1),$
$\quad\quad\quad\quad m := 1.$

$\text{case}_2(n, m)$
$\quad \mapsto \quad \text{nonmatch}(n, 0),$
$\quad\quad\quad\quad \text{nonmatch}(n, 1),$
$\quad\quad\quad\quad \text{match}(n, x),$
$\quad\quad\quad\quad y := x \text{ - } 1,$
$\quad\quad\quad\quad \text{fibonacci}(y, z),$
$\quad\quad\quad\quad v := x \text{ - } 2,$
$\quad\quad\quad\quad \text{fibonacci}(v, w),$
$\quad\quad\quad\quad m := z + w.$

Example: Size Abstraction

$\text{fibonacci\_list}(XS_I, XS_L, YS_I, YS_L)$
$\quad \mapsto \quad \text{case}_0(XS_I, XS_L, YS_I, YS_L).$

$\text{case}_0(XS_I, XS_L, YS_I, YS_L)$
$\quad \mapsto \quad \{XS_L = 1\},$
$\quad\quad\quad\quad \{YS_L = 1\}.$

$\text{case}_0(XS_I, XS_L, YS_I, YS_L)$
$\quad \mapsto \quad \{XS_I \geq N_I,\ XS_I \geq ZS_I,$
$\quad\quad\quad\quad XS_L = ZS_L + 1,\ ZS_L \geq 1\},$
$\quad\quad\quad\quad \text{fibonacci}(N_I, X_I),$
$\quad\quad\quad\quad \text{fibonacci\_list}(ZS_I, ZS_L, WS_I, WS_L),$
$\quad\quad\quad\quad ys := \text{Cons}(x, ws)$

$\text{fibonacci}(n, m)$
$\quad \mapsto \quad x := 0,$
$\quad\quad\quad\quad \text{case}_1(x, n, m).$

$\text{case}_1(x, n, m)$
$\quad \mapsto \quad n \geq x,$
$\quad\quad\quad\quad \text{case}_2(n, m).$

$\text{case}_1(x, n, m)$
$\quad \mapsto \quad n < x,$
$\quad\quad\quad\quad m := \text{-}1.$

$\{YS_I \geq X_I,\ YS_I \geq WS_I,$
$YS_L = WS_L + 1,\ WS_L \geq 1\},$

$\text{case}_2(n, m)$
$\quad \mapsto \quad \text{nonmatch}(n, 0),$
$\quad\quad\quad\quad \text{match}(n, 1),$
$\quad\quad\quad\quad m := 1.$

$\text{case}_2(n, m)$
$\quad \mapsto \quad \text{nonmatch}(n, 0),$
$\quad\quad\quad\quad \text{nonmatch}(n, 1),$
$\quad\quad\quad\quad \text{match}(n, x),$
$\quad\quad\quad\quad y := x \text{ - } 1,$
$\quad\quad\quad\quad \text{fibonacci}(y, z),$
$\quad\quad\quad\quad v := x \text{ - } 2,$
$\quad\quad\quad\quad \text{fibonacci}(v, w),$
$\quad\quad\quad\quad m := z + w.$

Example: Size Abstraction

$\text{fibonacci\_list}(XS_I, XS_L, YS_I, YS_L)$
$\quad \mapsto \quad \text{case}_0(XS_I, XS_L, YS_I, YS_L).$

$\text{case}_0(XS_I, XS_L, YS_I, YS_L)$
$\quad \mapsto \quad \{XS_L = 1\},$
$\quad\quad\quad\quad \{YS_L = 1\}.$

$\text{case}_0(XS_I, XS_L, YS_I, YS_L)$
$\quad \mapsto \quad \{XS_I \geq N_I,\ XS_I \geq ZS_I,$
$\quad\quad\quad\quad XS_L = ZS_L + 1,\ ZS_L \geq 1\},$
$\quad\quad\quad\quad \text{fibonacci}(N_I, X_I),$
$\quad\quad\quad\quad \text{fibonacci\_list}(ZS_I, ZS_L, WS_I, WS_L),$
$\quad\quad\quad\quad \{YS_I \geq X_I,\ YS_I \geq WS_I,$
$\quad\quad\quad\quad YS_L = WS_L + 1,\ WS_L \geq 1\},$

$\text{fibonacci}(n, m)$
$\quad \mapsto \quad x := 0,$
$\quad\quad\quad\quad \text{case}_1(x, n, m).$

$\text{case}_1(x, n, m)$
$\quad \mapsto \quad n \geq x,$
$\quad\quad\quad\quad \text{case}_2(n, m).$

$\text{case}_1(x, n, m)$
$\quad \mapsto \quad n < x,$
$\quad\quad\quad\quad m := \text{-}1.$

$\text{fibonacci}(N_I, M_I)$
$\quad \mapsto \quad \{X_I = 0\},$

$\text{case}_2(n, m)$
$\quad \mapsto \quad \text{nonmatch}(n, 0),$
$\quad\quad\quad\quad \text{match}(n, 1),$
$\quad\quad\quad\quad m := 1.$

$\text{case}_2(n, m)$
$\quad \mapsto \quad \text{nonmatch}(n, 0),$
$\quad\quad\quad\quad \text{nonmatch}(n, 1),$
$\quad\quad\quad\quad \text{match}(n, x),$
$\quad\quad\quad\quad y := x \text{ - } 1,$
$\quad\quad\quad\quad \text{fibonacci}(y, z),$
$\quad\quad\quad\quad v := x \text{ - } 2,$
$\quad\quad\quad\quad \text{fibonacci}(v, w),$
$\quad\quad\quad\quad m := z + w.$

Example: Size Abstraction

$\text{fibonacci\_list}(XS_I, XS_L, YS_I, YS_L)$
$\quad \mapsto \quad \text{case}_0(XS_I, XS_L, YS_I, YS_L).$

$\text{case}_0(XS_I, XS_L, YS_I, YS_L)$
$\quad \mapsto \quad \{XS_L = 1\},$
$\quad\quad\quad\quad \{YS_L = 1\}.$

$\text{case}_0(XS_I, XS_L, YS_I, YS_L)$
$\quad \mapsto \quad \{XS_I \geq N_I,\ XS_I \geq ZS_I,$
$\quad\quad\quad\quad XS_L = ZS_L + 1,\ ZS_L \geq 1\},$
$\quad\quad\quad\quad \text{fibonacci}(N_I, X_I),$
$\quad\quad\quad\quad \text{fibonacci\_list}(ZS_I, ZS_L, WS_I, WS_L),$
$\quad\quad\quad\quad \{YS_I \geq X_I,\ YS_I \geq WS_I,$
$\quad\quad\quad\quad YS_L = WS_L + 1,\ WS_L \geq 1\},$

$\text{fibonacci}(N_I, M_I)$
$\quad \mapsto \quad \{X_I = 0\},$
$\quad\quad\quad\quad \text{case}_1(x, n, m).$

$\text{case}_1(x, n, m)$
$\quad \mapsto \quad n \geq x,$
$\quad\quad\quad\quad \text{case}_2(n, m).$

$\text{case}_1(x, n, m)$
$\quad \mapsto \quad n < x,$
$\quad\quad\quad\quad m := \text{-}1.$

$\text{case}_2(n, m)$
$\quad\quad\quad\quad \text{case}_1(X_I, N_I, M_I).$
$\quad\quad\quad\quad m := 0.$

$\text{case}_2(n, m)$
$\quad \mapsto \quad \text{nonmatch}(n, 0),$
$\quad\quad\quad\quad \text{match}(n, 1),$
$\quad\quad\quad\quad m := 1.$

$\text{case}_2(n, m)$
$\quad \mapsto \quad \text{nonmatch}(n, 0),$
$\quad\quad\quad\quad \text{nonmatch}(n, 1),$
$\quad\quad\quad\quad \text{match}(n, x),$
$\quad\quad\quad\quad y := x - 1,$
$\quad\quad\quad\quad \text{fibonacci}(y, z),$
$\quad\quad\quad\quad v := x - 2,$
$\quad\quad\quad\quad \text{fibonacci}(v, w),$
$\quad\quad\quad\quad m := z + w.$

Example: Size Abstraction

$$
\begin{array}{l}
\text{fibonacci\_list}(XS_I,XS_L,YS_I,YS_L) \\
\quad \mapsto \quad \text{case}_0(XS_I,XS_L,YS_I,YS_L). \\
\text{case}_0(XS_I,XS_L,YS_I,YS_L) \\
\quad \mapsto \quad \{XS_L = 1\}, \\
\qquad\quad \{YS_L = 1\}. \\
\text{case}_0(XS_I,XS_L,YS_I,YS_L) \\
\quad \mapsto \quad \{XS_I \geq N_I,\ XS_I \geq ZS_I, \\
\qquad\quad XS_L = ZS_L + 1,\ ZS_L \geq 1\}, \\
\qquad\quad \text{fibonacci}(N_I,X_I), \\
\qquad\quad \text{fibonacci\_list}(ZS_I,ZS_L,WS_I,WS_L), \\
\qquad\quad \{YS_I \geq X_I,\ YS_I \geq WS_I, \\
\qquad\quad YS_L = WS_L + 1,\ WS_L \geq 1\}, \\
\text{fibonacci}(N_I,M_I) \\
\quad \mapsto \quad \{X_I = 0\}, \\
\qquad\quad \text{case}_1(X_I,N_I,M_I). \\
\text{case}_1(x,n,m) \\
\quad \mapsto \quad n \geq x, \\
\qquad\quad \text{case}_2(n,m).
\end{array}
$$

$$
\begin{array}{l}
\text{case}_1(x,n,m) \\
\quad \mapsto \quad n < x, \\
\qquad\quad m := \text{-}1. \\
\text{case}_1(X_I,N_I,M_I) \\
\quad \mapsto \quad \{N_I \geq X_I\}, \\
\text{case}_2(n,m) \\
\quad \mapsto \quad \text{nonmatch}(n,0), \\
\qquad\quad \text{match}(n,1), \\
\qquad\quad m := 1. \\
\text{case}_2(n,m) \\
\quad \mapsto \quad \text{nonmatch}(n,0), \\
\qquad\quad \text{nonmatch}(n,1), \\
\qquad\quad \text{match}(n,x), \\
\qquad\quad y := x \text{ - } 1, \\
\qquad\quad \text{fibonacci}(y,z), \\
\qquad\quad v := x \text{ - } 2, \\
\qquad\quad \text{fibonacci}(v,w), \\
\qquad\quad m := z + w.
\end{array}
$$

Example: Size Abstraction

$\text{fibonacci\_list}(XS_I, XS_L, YS_I, YS_L)$
$\quad \mapsto \quad \text{case}_0(XS_I, XS_L, YS_I, YS_L).$

$\text{case}_0(XS_I, XS_L, YS_I, YS_L)$
$\quad \mapsto \quad \{XS_L = 1\},$
$\quad\quad\quad\quad \{YS_L = 1\}.$

$\text{case}_0(XS_I, XS_L, YS_I, YS_L)$
$\quad \mapsto \quad \{XS_I \geq N_I,\ XS_I \geq ZS_I,$
$\quad\quad\quad\quad XS_L = ZS_L + 1,\ ZS_L \geq 1\},$
$\quad\quad\quad\quad \text{fibonacci}(N_I, X_I),$
$\quad\quad\quad\quad \text{fibonacci\_list}(ZS_I, ZS_L, WS_I, WS_L),$
$\quad\quad\quad\quad \{YS_I \geq X_I,\ YS_I \geq WS_I,$
$\quad\quad\quad\quad YS_L = WS_L + 1,\ WS_L \geq 1\},$

$\text{fibonacci}(N_I, M_I)$
$\quad \mapsto \quad \{X_I = 0\},$
$\quad\quad\quad\quad \text{case}_1(X_I, N_I, M_I).$

$\text{case}_1(X_I, N_I, M_I)$
$\quad \mapsto \quad \{N_I \geq X_I\},$
$\quad\quad\quad\quad \text{case}_2(n, m).$

$\text{case}_1(x, n, m)$
$\quad \mapsto \quad n < x,$
$\quad\quad\quad\quad m := \text{-}1.$

$\text{case}_2(n, m)$
$\quad\quad\quad\quad \text{case}_2(N_I, M_I).$
$\quad\quad\quad\quad m := 0.$

$\text{case}_2(n, m)$
$\quad \mapsto \quad \text{nonmatch}(n, 0),$
$\quad\quad\quad\quad \text{match}(n, 1),$
$\quad\quad\quad\quad m := 1.$

$\text{case}_2(n, m)$
$\quad \mapsto \quad \text{nonmatch}(n, 0),$
$\quad\quad\quad\quad \text{nonmatch}(n, 1),$
$\quad\quad\quad\quad \text{match}(n, x),$
$\quad\quad\quad\quad y := x \text{ - } 1,$
$\quad\quad\quad\quad \text{fibonacci}(y, z),$
$\quad\quad\quad\quad v := x \text{ - } 2,$
$\quad\quad\quad\quad \text{fibonacci}(v, w),$
$\quad\quad\quad\quad m := z + w.$

Example: Size Abstraction

$\mathrm{fibonacci\_list}(XS_I, XS_L, YS_I, YS_L)$
$\quad \mapsto \quad \mathrm{case}_0(XS_I, XS_L, YS_I, YS_L).$

$\mathrm{case}_0(XS_I, XS_L, YS_I, YS_L)$
$\quad \mapsto \quad \{XS_I \geq N_I,\ XS_I \geq ZS_I,$
$\quad\quad XS_L = ZS_L + 1,\ ZS_L \geq 1\},$
$\quad\quad \mathrm{fibonacci}(N_I, X_I),$
$\quad\quad \mathrm{fibonacci\_list}(ZS_I, ZS_L, WS_I, WS_L),$
$\quad\quad \{YS_I \geq X_I,\ YS_I \geq WS_I,$
$\quad\quad YS_L = WS_L + 1,\ WS_L \geq 1\},$

$\mathrm{fibonacci}(N_I, M_I)$
$\quad \mapsto \quad \{X_I = 0\},$
$\quad\quad \mathrm{case}_1(X_I, N_I, M_I).$

$\mathrm{case}_1(X_I, N_I, M_I)$
$\quad \mapsto \quad \{N_I \geq X_I\},$
$\quad\quad \mathrm{case}_2(N_I, M_I).$

$\mathrm{case}_1(X_I, N_I, M_I)$
$\quad \mapsto \quad \{X_I \geq N_I + 1\},$

$\mathrm{case}_1(x, n, m)$
$\quad \mapsto \quad n < x,$
$\quad\quad m := \text{-}1.$

$\mathrm{case}_2(n, m)$
$\quad \mapsto \quad \mathrm{match}(n, 0),$
$\quad\quad m := 0.$

$\mathrm{case}_2(n, m)$
$\quad \mapsto \quad \mathrm{nonmatch}(n, 0),$
$\quad\quad \mathrm{match}(n, 1),$
$\quad\quad m := 1.$

$\mathrm{case}_2(n, m)$
$\quad \mapsto \quad \mathrm{nonmatch}(n, 0),$
$\quad\quad \mathrm{nonmatch}(n, 1),$
$\quad\quad \mathrm{match}(n, x),$
$\quad\quad y := x \text{ - } 1,$
$\quad\quad \mathrm{fibonacci}(y, z),$
$\quad\quad v := x \text{ - } 2,$
$\quad\quad \mathrm{fibonacci}(v, w),$
$\quad\quad m := z + w.$

Example: Size Abstraction

$\text{fibonacci\_list}(XS_I, XS_L, YS_I, YS_L)$
$\quad \mapsto \quad \text{case}_0(XS_I, XS_L, YS_I, YS_L).$

$\text{case}_0(XS_I, XS_L, YS_I, YS_L)$
$\quad \mapsto \quad \{XS_I = 1\},$
$\qquad \{M_I = -1\}.$

$\text{case}_0(XS_I, XS_L, YS_I, YS_L)$
$\quad \mapsto \quad \{XS_I \geq N_I,\ XS_I \geq ZS_I,$
$\qquad XS_L = ZS_L + 1,\ ZS_L \geq 1\},$
$\qquad \text{fibonacci}(N_I, X_I),$
$\qquad \text{fibonacci\_list}(ZS_I, ZS_L, WS_I, WS_L),$
$\qquad \{YS_I \geq X_I,\ YS_I \geq WS_I,$
$\qquad YS_L = WS_L + 1,\ WS_L \geq 1\},$

$\text{fibonacci}(N_I, M_I)$
$\quad \mapsto \quad \{X_I = 0\},$
$\qquad \text{case}_1(X_I, N_I, M_I).$

$\text{case}_1(X_I, N_I, M_I)$
$\quad \mapsto \quad \{N_I \geq X_I\},$
$\qquad \text{case}_2(N_I, M_I).$

$\text{case}_1(X_I, N_I, M_I)$
$\quad \mapsto \quad \{X_I \geq N_I + 1\},$
$\qquad m := -1.$

$\text{case}_2(n, m)$
$\quad \mapsto \quad \text{match}(n, 0),$
$\qquad m := 0.$

$\text{case}_2(n, m)$
$\quad \mapsto \quad \text{nonmatch}(n, 0),$
$\qquad \text{match}(n, 1),$
$\qquad m := 1.$

$\text{case}_2(n, m)$
$\quad \mapsto \quad \text{nonmatch}(n, 0),$
$\qquad \text{nonmatch}(n, 1),$
$\qquad \text{match}(n, x),$
$\qquad y := x - 1,$
$\qquad \text{fibonacci}(y, z),$
$\qquad v := x - 2,$
$\qquad \text{fibonacci}(v, w),$
$\qquad m := z + w.$

Example: Size Abstraction

$\text{fibonacci\_list}(XS_I, XS_L, YS_I, YS_L)$
$\quad \mapsto \quad \text{case}_0(XS_I, XS_L, YS_I, YS_L).$

$\text{case}_0(XS_I, XS_L, YS_I, YS_L)$
$\quad \mapsto \quad \{XS_I \geq N_I,\ XS_I \geq ZS_I,$
$\quad\quad XS_L = ZS_L + 1,\ ZS_L \geq 1\},$
$\quad\quad \text{fibonacci}(N_I, X_I),$
$\quad\quad \text{fibonacci\_list}(ZS_I, ZS_L, WS_I, WS_L),$
$\quad\quad \{YS_I \geq X_I,\ YS_I \geq WS_I,$
$\quad\quad YS_L = WS_L + 1,\ WS_L \geq 1\},$

$\text{fibonacci}(N_I, M_I)$
$\quad \mapsto \quad \{X_I = 0\},$
$\quad\quad \text{case}_1(X_I, N_I, M_I).$

$\text{case}_1(X_I, N_I, M_I)$
$\quad \mapsto \quad \{N_I \geq X_I\},$
$\quad\quad \text{case}_2(N_I, M_I).$

$\text{case}_1(X_I, N_I, M_I)$
$\quad \mapsto \quad \{X_I \geq N_I + 1\},$
$\quad\quad \{M_I = -1\}.$

$\text{case}_2(n, m)$
$\quad \mapsto \quad \text{match}(n, 0),$
$\quad\quad m := 0.$

$\text{case}_2(n, m)$
$\quad \mapsto \quad \text{nonmatch}(n, 0),$
$\quad\quad \text{match}(n, 1),$
$\quad\quad m := 1.$

$\text{case}_2(n, m)$
$\quad \mapsto \quad \text{nonmatch}(n, 0),$
$\quad\quad \text{nonmatch}(n, 1),$
$\quad\quad \text{match}(n, x),$
$\quad\quad y := x - 1,$
$\quad\quad \text{fibonacci}(y, z),$
$\quad\quad v := x - 2,$
$\quad\quad \text{fibonacci}(v, w),$
$\quad\quad m := z + w.$

Example: Size Abstraction

$\text{fibonacci\_list}(XS_I, XS_L, YS_I, YS_L)$
$\quad \mapsto \quad \text{case}_0(XS_I, XS_L, YS_I, YS_L).$

$\text{case}_0(XS_I, XS_L, YS_I, YS_L)$
$\quad \mapsto \quad \{XS_I = 1\},$
$\qquad \{M_I = 0\}.$

$\text{case}_0(XS_I, XS_L, YS_I, YS_L)$
$\quad \mapsto \quad \{XS_I \geq N_I,\ XS_I \geq ZS_I,$
$\qquad XS_L = ZS_L + 1,\ ZS_L \geq 1\},$
$\qquad \text{fibonacci}(N_I, X_I),$
$\qquad \text{fibonacci\_list}(ZS_I, ZS_L, WS_I, WS_L),$
$\qquad \{YS_I \geq X_I,\ YS_I \geq WS_I,$
$\qquad YS_L = WS_L + 1,\ WS_L \geq 1\},$

$\text{fibonacci}(N_I, M_I)$
$\quad \mapsto \quad \{X_I = 0\},$
$\qquad \text{case}_1(X_I, N_I, M_I).$

$\text{case}_1(X_I, N_I, M_I)$
$\quad \mapsto \quad \{N_I \geq X_I\},$
$\qquad \text{case}_2(N_I, M_I).$

$\text{case}_1(X_I, N_I, M_I)$
$\quad \mapsto \quad \{X_I \geq N_I + 1\},$
$\qquad \{M_I = -1\}.$

$\text{case}_2(N_I, M_I)$
$\quad \mapsto \quad \{N_I = 0\},$
$\qquad m := 0.$

$\text{case}_2(n, m)$
$\quad \mapsto \quad \text{nonmatch}(n, 0),$
$\qquad \text{match}(n, 1),$
$\qquad m := 1.$

$\text{case}_2(n, m)$
$\quad \mapsto \quad \text{nonmatch}(n, 0),$
$\qquad \text{nonmatch}(n, 1),$
$\qquad \text{match}(n, x),$
$\qquad y := x - 1,$
$\qquad \text{fibonacci}(y, z),$
$\qquad v := x - 2,$
$\qquad \text{fibonacci}(v, w),$
$\qquad m := z + w.$

Example: Size Abstraction

$\text{fibonacci\_list}(XS_I, XS_L, YS_I, YS_L)$
$\quad \mapsto \quad \text{case}_0(XS_I, XS_L, YS_I, YS_L).$

$\text{case}_0(XS_I, XS_L, YS_I, YS_L)$
$\quad \mapsto \quad \{XS_I \geq N_I,\ XS_I \geq ZS_I,$
$\quad\quad XS_L = ZS_L + 1,\ ZS_L \geq 1\},$
$\quad\quad \text{fibonacci}(N_I, X_I),$
$\quad\quad \text{fibonacci\_list}(ZS_I, ZS_L, WS_I, WS_L),$
$\quad\quad \{YS_I \geq X_I,\ YS_I \geq WS_I,$
$\quad\quad YS_L = WS_L + 1,\ WS_L \geq 1\},$

$\text{fibonacci}(N_I, M_I)$
$\quad \mapsto \quad \{X_I = 0\},$
$\quad\quad \text{case}_1(X_I, N_I, M_I).$

$\text{case}_1(X_I, N_I, M_I)$
$\quad \mapsto \quad \{N_I \geq X_I\},$
$\quad\quad \text{case}_2(N_I, M_I).$

$\text{case}_2(N_I, M_I)$
$\quad \mapsto \quad \text{true},$

$\text{case}_1(X_I, N_I, M_I)$
$\quad \mapsto \quad \{X_I \geq N_I + 1\},$
$\quad\quad \{M_I = -1\}.$

$\text{case}_2(N_I, M_I)$
$\quad \mapsto \quad \{N_I = 0\},$
$\quad\quad \{M_I = 0\}.$

$\text{case}_2(n, m)$
$\quad \mapsto \quad \text{nonmatch}(n, 0),$
$\quad\quad \text{match}(n, 1),$
$\quad\quad m := 1.$

$\text{case}_2(n, m)$
$\quad \mapsto \quad \text{nonmatch}(n, 0),$
$\quad\quad \text{nonmatch}(n, 1),$
$\quad\quad \text{match}(n, x),$
$\quad\quad y := x - 1,$
$\quad\quad \text{fibonacci}(y, z),$
$\quad\quad v := x - 2,$
$\quad\quad \text{fibonacci}(v, w),$
$\quad\quad m := z + w.$

Example: Size Abstraction

$\text{fibonacci\_list}(XS_I,XS_L,YS_I,YS_L)$
$\quad\mapsto\quad \text{case}_0(XS_I,XS_L,YS_I,YS_L).$

$\text{case}_0(XS_I,XS_L,YS_I,YS_L)$
$\quad\mapsto\quad \{XS_I = 1\},$
$\qquad\qquad \{N_I = 1\},$
$\text{case}_0(XS_I,XS_L,YS_I,YS_L)$
$\quad\mapsto\quad \{XS_I \geq N_I,\ XS_I \geq ZS_I,$
$\qquad\qquad XS_L = ZS_L + 1,\ ZS_L \geq 1\},$
$\qquad\qquad \text{fibonacci}(N_I,X_I),$
$\qquad\qquad \text{fibonacci\_list}(ZS_I,ZS_L,WS_I,WS_L),$
$\qquad\qquad \{YS_I \geq X_I,\ YS_I \geq WS_I,$
$\qquad\qquad YS_L = WS_L + 1,\ WS_L \geq 1\},$

$\text{fibonacci}(N_I,M_I)$
$\quad\mapsto\quad \{X_I = 0\},$
$\qquad\qquad \text{case}_1(X_I,N_I,M_I).$

$\text{case}_1(X_I,N_I,M_I)$
$\quad\mapsto\quad \{N_I \geq X_I\},$
$\qquad\qquad \text{case}_2(N_I,M_I).$

$\text{case}_1(X_I,N_I,M_I)$
$\quad\mapsto\quad \{X_I \geq N_I + 1\},$
$\qquad\qquad \{M_I = -1\}.$

$\text{case}_2(N_I,M_I)$
$\quad\mapsto\quad \{N_I = 0\},$
$\qquad\qquad \{M_I = 0\}.$

$\text{case}_2(N_I,M_I)$
$\qquad\qquad \text{match}(n,1),$
$\qquad\qquad m := 1.$

$\text{case}_2(n,m)$
$\quad\mapsto\quad \text{nonmatch}(n,0),$
$\qquad\qquad \text{nonmatch}(n,1),$
$\qquad\qquad \text{match}(n,x),$
$\qquad\qquad y := x - 1,$
$\qquad\qquad \text{fibonacci}(y,z),$
$\qquad\qquad v := x - 2,$
$\qquad\qquad \text{fibonacci}(v,w),$
$\qquad\qquad m := z + w.$

Example: Size Abstraction

$\text{fibonacci\_list}(XS_I,XS_L,YS_I,YS_L)$
$\quad\mapsto\quad \text{case}_0(XS_I,XS_L,YS_I,YS_L).$

$\text{case}_0(XS_I,XS_L,YS_I,YS_L)$
$\quad\mapsto\quad \{XS_I = 1\},$
$\qquad\qquad \{M_I = 1\}.$

$\text{case}_0(XS_I,XS_L,YS_I,YS_L)$
$\quad\mapsto\quad \{XS_I \geq N_I,\ XS_I \geq ZS_I,$
$\qquad\qquad XS_L = ZS_L + 1,\ ZS_L \geq 1\},$
$\qquad\qquad \text{fibonacci}(N_I,X_I),$
$\qquad\qquad \text{fibonacci\_list}(ZS_I,ZS_L,WS_I,WS_L),$
$\qquad\qquad \{YS_I \geq X_I,\ YS_I \geq WS_I,$
$\qquad\qquad YS_L = WS_L + 1,\ WS_L \geq 1\},$

$\text{fibonacci}(N_I,M_I)$
$\quad\mapsto\quad \{X_I = 0\},$
$\qquad\qquad \text{case}_1(X_I,N_I,M_I).$

$\text{case}_1(X_I,N_I,M_I)$
$\quad\mapsto\quad \{N_I \geq X_I\},$
$\qquad\qquad \text{case}_2(N_I,M_I).$

$\text{case}_1(X_I,N_I,M_I)$
$\quad\mapsto\quad \{X_I \geq N_I + 1\},$
$\qquad\qquad \{M_I = -1\}.$

$\text{case}_2(N_I,M_I)$
$\quad\mapsto\quad \{N_I = 0\},$
$\qquad\qquad \{M_I = 0\}.$

$\text{case}_2(N_I,M_I)$
$\qquad\qquad \{N_I = 1\},$
$\qquad\qquad m := 1.$

$\text{case}_2(n,m)$
$\quad\mapsto\quad \text{nonmatch}(n,0),$
$\qquad\qquad \text{nonmatch}(n,1),$
$\qquad\qquad \text{match}(n,x),$
$\qquad\qquad y := x - 1,$
$\qquad\qquad \text{fibonacci}(y,z),$
$\qquad\qquad v := x - 2,$
$\qquad\qquad \text{fibonacci}(v,w),$
$\qquad\qquad m := z + w.$

Example: Size Abstraction

$\mathsf{fibonacci\_list}(XS_I,XS_L,YS_I,YS_L)$
$\quad\mapsto\quad \mathsf{case}_0(XS_I,XS_L,YS_I,YS_L).$

$\mathsf{case}_0(XS_I,XS_L,YS_I,YS_L)$
$\quad\mapsto\quad \{XS_I \geq NS_I,\ XS_I \geq ZS_I,$
$\qquad XS_L = ZS_L + 1,\ ZS_L \geq 1\},$
$\qquad \mathsf{fibonacci}(N_I,X_I),$
$\qquad \mathsf{fibonacci\_list}(ZS_I,ZS_L,WS_I,WS_L),$
$\qquad \{YS_I \geq X_I,\ YS_I \geq WS_I,$
$\qquad YS_L = WS_L + 1,\ WS_L \geq 1\},$

$\mathsf{fibonacci}(N_I,M_I)$
$\quad\mapsto\quad \{X_I = 0\},$
$\qquad \mathsf{case}_1(X_I,N_I,M_I).$

$\mathsf{case}_1(X_I,N_I,M_I)$
$\quad\mapsto\quad \{N_I \geq X_I\},$
$\qquad \mathsf{case}_2(N_I,M_I).$

$\mathsf{case}_2(N_I,M_I)$
$\quad\mapsto\quad \mathsf{true},$
$\qquad \mathsf{true},$

$\mathsf{case}_1(X_I,N_I,M_I)$
$\quad\mapsto\quad \{X_I \geq N_I + 1\},$
$\qquad \{M_I = -1\}.$

$\mathsf{case}_2(N_I,M_I)$
$\quad\mapsto\quad \{N_I = 0\},$
$\qquad \{M_I = 0\}.$

$\mathsf{case}_2(N_I,M_I)$
$\qquad \{N_I = 1\},$
$\qquad \{M_I = 1\}.$

$\mathsf{case}_2(n,m)$
$\quad\mapsto\quad \mathsf{nonmatch}(n,0),$
$\qquad \mathsf{nonmatch}(n,1),$
$\qquad \mathsf{match}(n,x),$
$\qquad y := x - 1,$
$\qquad \mathsf{fibonacci}(y,z),$
$\qquad v := x - 2,$
$\qquad \mathsf{fibonacci}(v,w),$
$\qquad m := z + w.$

Example: Size Abstraction

$\text{fibonacci\_list}(XS_I,XS_L,YS_I,YS_L)$
$\quad\mapsto\quad \text{case}_0(XS_I,XS_L,YS_I,YS_L).$

$\text{case}_0(XS_I,XS_L,YS_I,YS_L)$
$\quad\mapsto\quad \{XS_I = 1\},$
$\qquad\qquad \{N_I = X_I\},$
$\text{case}_0(XS_I,XS_L,YS_I,YS_L)$
$\quad\mapsto\quad \{XS_I \geq N_I,\ XS_I \geq ZS_I,$
$\qquad\qquad XS_L = ZS_L + 1,\ ZS_L \geq 1\},$
$\qquad\qquad \text{fibonacci}(N_I,X_I),$
$\qquad\qquad \text{fibonacci\_list}(ZS_I,ZS_L,WS_I,WS_L),$
$\qquad\qquad \{YS_I \geq X_I,\ YS_I \geq WS_I,$
$\qquad\qquad YS_L = WS_L + 1,\ WS_L \geq 1\},$

$\text{fibonacci}(N_I,M_I)$
$\quad\mapsto\quad \{X_I = 0\},$
$\qquad\qquad \text{case}_1(X_I,N_I,M_I).$

$\text{case}_1(X_I,N_I,M_I)$
$\quad\mapsto\quad \{N_I \geq X_I\},$
$\qquad\qquad \text{case}_2(N_I,M_I).$

$\text{case}_1(X_I,N_I,M_I)$
$\quad\mapsto\quad \{X_I \geq N_I + 1\},$
$\qquad\qquad \{M_I = -1\}.$

$\text{case}_2(N_I,M_I)$
$\quad\mapsto\quad \{N_I = 0\},$
$\qquad\qquad \{M_I = 0\}.$

$\text{case}_2(N_I,M_I)$
$\qquad\qquad \{N_I = 1\},$
$\qquad\qquad \{M_I = 1\}.$

$\text{case}_2(N_I,M_I)$
$\quad\mapsto\quad \text{match}(n,x),$
$\qquad\qquad y := x - 1,$
$\qquad\qquad \text{fibonacci}(y,z),$
$\qquad\qquad v := x - 2,$
$\qquad\qquad \text{fibonacci}(v,w),$
$\qquad\qquad m := z + w.$

Example: Size Abstraction

$\text{fibonacci\_list}(XS_I, XS_L, YS_I, YS_L)$
$\quad \mapsto \quad \text{case}_0(XS_I, XS_L, YS_I, YS_L).$

$\text{case}_0(XS_I, XS_L, YS_I, YS_L)$
$\quad \mapsto \quad \{XS_I = 1\},$
$\qquad \{Y_I = X_I - 1\},$
$\text{case}_0(XS_I, XS_L, YS_I, YS_L)$
$\quad \mapsto \quad \{XS_I \geq N_I,\ XS_I \geq ZS_I,$
$\qquad XS_L = ZS_L + 1,\ ZS_L \geq 1\},$
$\qquad \text{fibonacci}(N_I, X_I),$
$\qquad \text{fibonacci\_list}(ZS_I, ZS_L, WS_I, WS_L),$
$\qquad \{YS_I \geq X_I,\ YS_I \geq WS_I,$
$\qquad YS_L = WS_L + 1,\ WS_L \geq 1\},$

$\text{fibonacci}(N_I, M_I)$
$\quad \mapsto \quad \{X_I = 0\},$
$\qquad \text{case}_1(X_I, N_I, M_I).$

$\text{case}_1(X_I, N_I, M_I)$
$\quad \mapsto \quad \{N_I \geq X_I\},$
$\qquad \text{case}_2(N_I, M_I).$

$\text{case}_1(X_I, N_I, M_I)$
$\quad \mapsto \quad \{X_I \geq N_I + 1\},$
$\qquad \{M_I = -1\}.$

$\text{case}_2(N_I, M_I)$
$\quad \mapsto \quad \{N_I = 0\},$
$\qquad \{M_I = 0\}.$

$\text{case}_2(N_I, M_I)$
$\qquad \{N_I = 1\},$
$\qquad \{M_I = 1\}.$

$\text{case}_2(N_I, M_I)$
$\quad \mapsto \quad \{N_I = X_I\},$
$\qquad y := x - 1,$
$\qquad \text{fibonacci}(y, z),$
$\qquad v := x - 2,$
$\qquad \text{fibonacci}(v, w),$
$\qquad m := z + w.$

Example: Size Abstraction

$\mathsf{fibonacci\_list}(XS_I, XS_L, YS_I, YS_L)$
$\quad \mapsto \quad \mathsf{case}_0(XS_I, XS_L, YS_I, YS_L).$

$\mathsf{case}_0(XS_I, XS_L, YS_I, YS_L)$
$\quad \mapsto \quad \{XS_I = 1\},$
$\qquad\qquad \mathsf{fibonacci}(Y_I, Z_I),$
$\mathsf{case}_0(XS_I, XS_L, YS_I, YS_L)$
$\quad \mapsto \quad \{XS_I \geq N_I,\ XS_I \geq ZS_I,$
$\qquad\qquad XS_L = ZS_L + 1,\ ZS_L \geq 1\},$
$\qquad\qquad \mathsf{fibonacci}(N_I, X_I),$
$\qquad\qquad \mathsf{fibonacci\_list}(ZS_I, ZS_L, WS_I, WS_L),$
$\qquad\qquad \{YS_I \geq X_I,\ YS_I \geq WS_I,$
$\qquad\qquad YS_L = WS_L + 1,\ WS_L \geq 1\},$

$\mathsf{fibonacci}(N_I, M_I)$
$\quad \mapsto \quad \{X_I = 0\},$
$\qquad\qquad \mathsf{case}_1(X_I, N_I, M_I).$

$\mathsf{case}_1(X_I, N_I, M_I)$
$\quad \mapsto \quad \{N_I \geq X_I\},$
$\qquad\qquad \mathsf{case}_2(N_I, M_I).$

$\mathsf{case}_1(X_I, N_I, M_I)$
$\quad \mapsto \quad \{X_I \geq N_I + 1\},$
$\qquad\qquad \{M_I = -1\}.$

$\mathsf{case}_2(N_I, M_I)$
$\quad \mapsto \quad \{N_I = 0\},$
$\qquad\qquad \{M_I = 0\}.$

$\mathsf{case}_2(N_I, M_I)$
$\qquad\qquad \{N_I = 1\},$
$\qquad\qquad \{M_I = 1\}.$

$\mathsf{case}_2(N_I, M_I)$
$\quad \mapsto \quad \{N_I = X_I\},$
$\qquad\qquad \{Y_I = X_I - 1\},$
$\qquad\qquad \mathsf{fibonacci}(y, z),$
$\qquad\qquad v := x - 2,$
$\qquad\qquad \mathsf{fibonacci}(v, w),$
$\qquad\qquad m := z + w.$

Example: Size Abstraction

$\text{fibonacci\_list}(XS_I, XS_L, YS_I, YS_L)$
$\quad \mapsto \quad \text{case}_0(XS_I, XS_L, YS_I, YS_L).$

$\text{case}_0(XS_I, XS_L, YS_I, YS_L)$
$\quad \mapsto \quad \{XS_I = 1\},$
$\qquad \{V_I = X_I - 2\},$
$\text{case}_0(XS_I, XS_L, YS_I, YS_L)$
$\quad \mapsto \quad \{XS_I \geq N_I,\ XS_I \geq ZS_I,$
$\qquad XS_L = ZS_L + 1,\ ZS_L \geq 1\},$
$\qquad \text{fibonacci}(N_I, X_I),$
$\qquad \text{fibonacci\_list}(ZS_I, ZS_L, WS_I, WS_L),$
$\qquad \{YS_I \geq X_I,\ YS_I \geq WS_I,$
$\qquad YS_L = WS_L + 1,\ WS_L \geq 1\},$

$\text{fibonacci}(N_I, M_I)$
$\quad \mapsto \quad \{X_I = 0\},$
$\qquad \text{case}_1(X_I, N_I, M_I).$

$\text{case}_1(X_I, N_I, M_I)$
$\quad \mapsto \quad \{N_I \geq X_I\},$
$\qquad \text{case}_2(N_I, M_I).$

$\text{case}_1(X_I, N_I, M_I)$
$\quad \mapsto \quad \{X_I \geq N_I + 1\},$
$\qquad \{M_I = -1\}.$

$\text{case}_2(N_I, M_I)$
$\quad \mapsto \quad \{N_I = 0\},$
$\qquad \{M_I = 0\}.$

$\text{case}_2(N_I, M_I)$
$\qquad \{N_I = 1\},$
$\qquad \{M_I = 1\}.$

$\text{case}_2(N_I, M_I)$
$\quad \mapsto \quad \{N_I = X_I\},$
$\qquad \{Y_I = X_I - 1\},$
$\qquad \text{fibonacci}(Y_I, Z_I),$
$\qquad v := x - 2,$
$\qquad \text{fibonacci}(v, w),$
$\qquad m := z + w.$

Example: Size Abstraction

$\text{fibonacci\_list}(XS_I, XS_L, YS_I, YS_L)$
$\quad \mapsto \quad \text{case}_0(XS_I, XS_L, YS_I, YS_L).$

$\text{case}_0(XS_I, XS_L, YS_I, YS_L)$
$\quad \mapsto \quad \{XS_I = 1\},$
$\quad\quad\quad \text{fibonacci}(V_I, W_I),$
$\quad\quad\quad \{YS_I = V_I + W_I\}.$

$\text{case}_0(XS_I, XS_L, YS_I, YS_L)$
$\quad \mapsto \quad \{XS_I \geq N_I,\ XS_I \geq ZS_I,$
$\quad\quad\quad XS_L = ZS_L + 1,\ ZS_L \geq 1\},$
$\quad\quad\quad \text{fibonacci}(N_I, X_I),$
$\quad\quad\quad \text{fibonacci\_list}(ZS_I, ZS_L, WS_I, WS_L),$
$\quad\quad\quad \{YS_I \geq X_I,\ YS_I \geq WS_I,$
$\quad\quad\quad YS_L = WS_L + 1,\ WS_L \geq 1\},$

$\text{fibonacci}(N_I, M_I)$
$\quad \mapsto \quad \{X_I = 0\},$
$\quad\quad\quad \text{case}_1(X_I, N_I, M_I).$

$\text{case}_1(X_I, N_I, M_I)$
$\quad \mapsto \quad \{N_I \geq X_I\},$
$\quad\quad\quad \text{case}_2(N_I, M_I).$

$\text{case}_1(X_I, N_I, M_I)$
$\quad \mapsto \quad \{X_I \geq N_I + 1\},$
$\quad\quad\quad \{M_I = -1\}.$

$\text{case}_2(N_I, M_I)$
$\quad \mapsto \quad \{N_I = 0\},$
$\quad\quad\quad \{M_I = 0\}.$

$\text{case}_2(N_I, M_I)$
$\quad\quad\quad \{N_I = 1\},$
$\quad\quad\quad \{M_I = 1\}.$

$\text{case}_2(N_I, M_I)$
$\quad \mapsto \quad \{N_I = X_I\},$
$\quad\quad\quad \{Y_I = X_I - 1\},$
$\quad\quad\quad \text{fibonacci}(Y_I, Z_I),$
$\quad\quad\quad \{V_I = X_I - 2\},$
$\quad\quad\quad \text{fibonacci}(v, w),$
$\quad\quad\quad m := z + w.$

Example: Size Abstraction

$\text{fibonacci\_list}(XS_I, XS_L, YS_I, YS_L)$
$\quad \mapsto \quad \text{case}_0(XS_I, XS_L, YS_I, YS_L).$

$\text{case}_0(XS_I, XS_L, YS_I, YS_L)$
$\quad \mapsto \quad \{XS_I = 1\},$
$\quad\quad\quad \{M_I = Z_I + W_I\},$
$\quad\quad\quad \{XS_I \geq N_I,\ XS_I \geq ZS_I,$
$\quad\quad\quad XS_L = ZS_L + 1,\ ZS_L \geq 1\},$
$\quad\quad\quad \text{fibonacci}(N_I, X_I),$
$\quad\quad\quad \text{fibonacci\_list}(ZS_I, ZS_L, WS_I, WS_L),$
$\quad\quad\quad \{YS_I \geq X_I,\ YS_I \geq WS_I,$
$\quad\quad\quad YS_L = WS_L + 1,\ WS_L \geq 1\},$

$\text{fibonacci}(N_I, M_I)$
$\quad \mapsto \quad \{X_I = 0\},$
$\quad\quad\quad \text{case}_1(X_I, N_I, M_I).$

$\text{case}_1(X_I, N_I, M_I)$
$\quad \mapsto \quad \{N_I \geq X_I\},$
$\quad\quad\quad \text{case}_2(N_I, M_I).$

$\text{case}_1(X_I, N_I, M_I)$
$\quad \mapsto \quad \{X_I \geq N_I + 1\},$
$\quad\quad\quad \{M_I = -1\}.$

$\text{case}_2(N_I, M_I)$
$\quad \mapsto \quad \{N_I = 0\},$
$\quad\quad\quad \{M_I = 0\}.$

$\text{case}_2(N_I, M_I)$
$\quad\quad\quad \{N_I = 1\},$
$\quad\quad\quad \{M_I = 1\}.$

$\text{case}_2(N_I, M_I)$
$\quad \mapsto \quad \{N_I = X_I\},$
$\quad\quad\quad \{Y_I = X_I - 1\},$
$\quad\quad\quad \text{fibonacci}(Y_I, Z_I),$
$\quad\quad\quad \{V_I = X_I - 2\},$
$\quad\quad\quad \text{fibonacci}(V_I, W_I),$
$\quad\quad\quad m := z + w.$

Example: Size Abstraction

$$
\begin{array}{l}
\text{fibonacci\_list}(XS_I,XS_L,YS_I,YS_L) \\
\quad \mapsto \quad \text{case}_0(XS_I,XS_L,YS_I,YS_L). \\
\text{case}_0(XS_I,XS_L,YS_I,YS_L) \\
\quad \mapsto \quad \{XS_L = 1\}, \\
\qquad\quad \{YS_L = 1\}. \\
\text{case}_0(XS_I,XS_L,YS_I,YS_L) \\
\quad \mapsto \quad \{XS_I \geq N_I,\ XS_I \geq ZS_I, \\
\qquad\quad XS_L = ZS_L + 1,\ ZS_L \geq 1\}, \\
\qquad\quad \text{fibonacci}(N_I,X_I), \\
\qquad\quad \text{fibonacci\_list}(ZS_I,ZS_L,WS_I,WS_L), \\
\qquad\quad \{YS_I \geq X_I,\ YS_I \geq WS_I, \\
\qquad\quad YS_L = WS_L + 1,\ WS_L \geq 1\}, \\
\text{fibonacci}(N_I,M_I) \\
\quad \mapsto \quad \{X_I = 0\}, \\
\qquad\quad \text{case}_1(X_I,N_I,M_I). \\
\text{case}_1(X_I,N_I,M_I) \\
\quad \mapsto \quad \{N_I \geq X_I\}, \\
\qquad\quad \text{case}_2(N_I,M_I).
\end{array}
$$

$$
\begin{array}{l}
\text{case}_1(X_I,N_I,M_I) \\
\quad \mapsto \quad \{X_I \geq N_I + 1\}, \\
\qquad\quad \{M_I = -1\}. \\
\text{case}_2(N_I,M_I) \\
\quad \mapsto \quad \{N_I = 0\}, \\
\qquad\quad \{M_I = 0\}. \\
\text{case}_2(N_I,M_I) \\
\qquad\quad \{N_I = 1\}, \\
\qquad\quad \{M_I = 1\}. \\
\text{case}_2(N_I,M_I) \\
\quad \mapsto \quad \{N_I = X_I\}, \\
\qquad\quad \{Y_I = X_I - 1\}, \\
\qquad\quad \text{fibonacci}(Y_I,Z_I), \\
\qquad\quad \{V_I = X_I - 2\}, \\
\qquad\quad \text{fibonacci}(V_I,W_I), \\
\qquad\quad \{M_I = Z_I + W_I\}.
\end{array}
$$

Example: Complexity

$$\|t\|_\sigma = \begin{cases} t & \sigma = \texttt{Int} \text{ and } t \text{ is an integer} \\ \mathit{length}(t) & \sigma = \texttt{String} \text{ and } t \text{ is an string} \\ 1 + \sum_{i=1}^{n} \|t_i\|_\sigma & \text{if } t = \mathit{Co}(t_1, \ldots, t_n) \text{ and } \mathit{typet} = \sigma \\ \max_{i=1}^{n} \|t_i\|_\sigma & \text{if } t = \mathit{Co}(t_1, \ldots, t_n) \text{ and } \mathit{typet} \neq \sigma \end{cases}$$

Example: Fibonacci List

2	3	1	3	4	1	5	2	5	4

Complexity (Upper Bound): Fibonacci List

$$O(l_{List<Int>} \times 2^{l_{Int}})$$

Outline

Inference of Typed-Norms

- Automated resource analysis needs to infer how the sizes of data are modified along program's execution.
- Norms define how the size of term are computed (*list-length*, *tree-depth*, *term-size*, ...).
- Type-norms allow defining norms based on type information.
- The choice of the right norm is crucial to obtain a proof for termination and get accurate upper and lower bounds.
- Allowing multiple norms simultaneously give us more chances to success, but the efficiency of the analysis can be degraded considerably.

Inference of Typed-Norms

- Allowing multiple norms simultaneously give us more chances to success, but the efficiency of the analysis can be degraded considerably.
  - Goal: develop an analysis that eliminates useless abstractions:
    - Remove variables that do not affect the cost.
    - Remove useless (typed) size information.

Inference of Typed-Norms: Algorithm

- Initialization. This step starts by setting typed norms(x) to $\emptyset$ for each variable x in the program. Then, it identifies the set of cost-significant guards, and uses each such guard to modify related typed_norms(x) as follows:
  - If the guard is match(x, t), variable x has a type T, and T is a recursive type, then T is added to typed_norms(x).
  - If the guard is of the form e1 op e2, and variable x appears in e1 or e2, then Int is added to typed_norms(x).

Inference of Typed-Norms: Algorithm

$\text{fibonacci\_list}(xs_\emptyset, ys_\emptyset)$
$\quad \mapsto \quad \text{case}_0(xs_\emptyset, ys_\emptyset).$

$\text{case}_0(xs_\emptyset, ys_\emptyset)$
$\quad \mapsto \quad \text{match}(xs_{\{L\}}, \text{Nil}),$
$\qquad\quad ys_\emptyset := \text{Nil}.$

$\text{case}_0(xs_\emptyset, ys_\emptyset)$
$\quad \mapsto \quad \text{nonmatch}(xs_\emptyset, \text{Nil}),$
$\qquad\quad \text{match}(xs_{\{L\}}, \text{Cons}(n_\emptyset, zs_\emptyset)),$
$\qquad\quad \text{fibonacci}(n_\emptyset, x_\emptyset),$
$\qquad\quad \text{fibonacci\_list}(zs_\emptyset, ws_\emptyset),$
$\qquad\quad ys_\emptyset := \text{Cons}(x_\emptyset, ws_\emptyset).$

$\text{fibonacci}(n_\emptyset, m_\emptyset)$
$\quad \mapsto \quad x_\emptyset := 0,$
$\qquad\quad \text{case}_1(x_\emptyset, n_\emptyset, m_\emptyset).$

$\text{case}_1(x_\emptyset, n_\emptyset, m_\emptyset)$
$\quad \mapsto \quad n_{\{I\}} \geq x_{\{I\}},$
$\qquad\quad \text{case}_2(n_\emptyset, m_\emptyset).$

$\text{case}_1(x_\emptyset, n_\emptyset, m_\emptyset)$
$\quad \mapsto \quad n_{\{I\}} < x_{\{I\}},$
$\qquad\quad m_\emptyset := -1.$

$\text{case}_2(n_\emptyset, m_\emptyset)$
$\quad \mapsto \quad \text{match}(n_{\{I\}}, 0),$
$\qquad\quad m_\emptyset := 0.$

$\text{case}_2(n_\emptyset, m_\emptyset)$
$\quad \mapsto \quad \text{nonmatch}(n_\emptyset, 0),$
$\qquad\quad \text{match}(n_{\{I\}}, 1),$
$\qquad\quad m_\emptyset := 1.$

$\text{case}_2(n_\emptyset, m_\emptyset)$
$\quad \mapsto \quad \text{nonmatch}(n_\emptyset, 0),$
$\qquad\quad \text{nonmatch}(n_\emptyset, 1),$
$\qquad\quad \text{match}(n_{\{I\}}, x_\emptyset),$
$\qquad\quad y := x_\emptyset - 1,$
$\qquad\quad \text{fibonacci}(y_\emptyset, z_\emptyset),$
$\qquad\quad v_\emptyset := x_\emptyset - 2,$
$\qquad\quad \text{fibonacci}(v_\emptyset, w_\emptyset),$
$\qquad\quad m := z_\emptyset + w_\emptyset.$

Inference of Typed-Norms: Algorithm

- Propagation. The initial information computed in the first step must be propagated backwards to other variables in the program:
  - For match(x,t) and nonmatch(x,t), if y in vars(t), and T in typed_norms(y), then we add T to typed_norms(x).
  - For $x := t$, if T in typed_norms(x) we add T to typed_norms(y) for each variable y in vars(t) as far as type(y) $\leq$ T.
  - Built-in functions propagate norms to its arguments.
  - All other instructions do not modify any information.

Inference of Typed-Norms: Algorithm

$\mathrm{fibonacci\_list}(xs_\emptyset, ys_\emptyset)$
$\quad \mapsto \quad \mathrm{case}_0(xs_\emptyset, ys_\emptyset).$
$\mathrm{case}_0(xs_\emptyset, ys_\emptyset)$
$\quad \mapsto \quad \mathrm{match}(xs_{\{L\}}, \mathrm{Nil}),$
$\qquad ys_\emptyset := \mathrm{Nil}.$
$\mathrm{case}_0(xs_\emptyset, ys_\emptyset)$
$\quad \mapsto \quad \mathrm{nonmatch}(xs_\emptyset, \mathrm{Nil}),$
$\qquad \mathrm{match}(xs_{\{L\}}, \mathrm{Cons}(n_\emptyset, zs_\emptyset)),$
$\qquad \mathrm{fibonacci}(n_\emptyset, x_\emptyset),$
$\qquad \mathrm{fibonacci\_list}(zs_\emptyset, ws_\emptyset),$
$\qquad ys_\emptyset := \mathrm{Cons}(x_\emptyset, ws_\emptyset).$
$\mathrm{fibonacci}(n_\emptyset, m_\emptyset)$
$\quad \mapsto \quad x_\emptyset := 0,$
$\qquad \mathrm{case}_1(x_\emptyset, n_\emptyset, m_\emptyset).$
$\mathrm{case}_1(x_\emptyset, n_\emptyset, m_\emptyset)$
$\quad \mapsto \quad n_{\{I\}} \geq x_{\{I\}},$
$\qquad \mathrm{case}_2(n_\emptyset, m_\emptyset).$
$\mathrm{case}_1(x_\emptyset, n_\emptyset, m_\emptyset)$
$\quad \mapsto \quad n_{\{I\}} < x_{\{I\}},$
$\qquad m_\emptyset := -1.$

$\mathrm{case}_2(n_\emptyset, m_\emptyset)$
$\quad \mapsto \quad \mathrm{match}(n_{\{I\}}, 0),$
$\qquad m_\emptyset := 0.$
$\mathrm{case}_2(n_\emptyset, m_\emptyset)$
$\quad \mapsto \quad \mathrm{nonmatch}(n_\emptyset, 0),$
$\qquad \mathrm{match}(n_{\{I\}}, 1),$
$\qquad m_\emptyset := 1.$
$\mathrm{case}_2(n_\emptyset, m_\emptyset)$
$\quad \mapsto \quad \mathrm{nonmatch}(n_\emptyset, 0),$
$\qquad \mathrm{nonmatch}(n_\emptyset, 1),$
$\qquad \mathrm{match}(n_{\{I\}}, x_\emptyset),$
$\qquad y_\emptyset := x_\emptyset - 1,$
$\qquad \mathrm{fibonacci}(y_\emptyset, z_\emptyset),$
$\qquad v_\emptyset := x_\emptyset - 2,$
$\qquad \mathrm{fibonacci}(v_\emptyset, w_\emptyset),$
$\qquad m_\emptyset := z_\emptyset + w_\emptyset.$

Inference of Typed-Norms: Algorithm

$\text{fibonacci\_list}(xs_\emptyset, ys_\emptyset)$
$\quad \mapsto \quad \text{case}_0(xs_\emptyset, ys_\emptyset).$

$\text{case}_0(xs_{\{L\}}, ys_\emptyset)$
$\quad \mapsto \quad \text{match}(xs_{\{L\}}, \text{Nil}),$
$\quad\quad\quad ys_\emptyset := \text{Nil}.$

$\text{case}_0(xs_{\{L\}}, ys_\emptyset)$
$\quad \mapsto \quad \text{nonmatch}(xs_{\{L\}}, \text{Nil}),$
$\quad\quad\quad \text{match}(xs_{\{L\}}, \text{Cons}(n_\emptyset, zs_\emptyset)),$
$\quad\quad\quad \text{fibonacci}(n_\emptyset, x_\emptyset),$
$\quad\quad\quad \text{fibonacci\_list}(zs_\emptyset, ws_\emptyset),$
$\quad\quad\quad ys_\emptyset := \text{Cons}(x_\emptyset, ws_\emptyset).$

$\text{fibonacci}(n_\emptyset, m_\emptyset)$
$\quad \mapsto \quad x_\emptyset := 0,$
$\quad\quad\quad \text{case}_1(x_\emptyset, n_\emptyset, m_\emptyset).$

$\text{case}_1(x_{\{I\}}, n_{\{I\}}, m_\emptyset)$
$\quad \mapsto \quad n_{\{I\}} \geq x_{\{I\}},$
$\quad\quad\quad \text{case}_2(n_{\{I\}}, m_\emptyset).$

$\text{case}_1(x_{\{I\}}, n_{\{I\}}, m_\emptyset)$
$\quad \mapsto \quad n_{\{I\}} < x_{\{I\}},$
$\quad\quad\quad m_\emptyset := \text{-}1.$

$\text{case}_2(n_{\{I\}}, m_\emptyset)$
$\quad \mapsto \quad \text{match}(n_{\{I\}}, 0),$
$\quad\quad\quad m_\emptyset := 0.$

$\text{case}_2(n_{\{I\}}, m_\emptyset)$
$\quad \mapsto \quad \text{nonmatch}(n_{\{I\}}, 0),$
$\quad\quad\quad \text{match}(n_{\{I\}}, 1),$
$\quad\quad\quad m_\emptyset := 1.$

$\text{case}_2(n_{\{I\}}, m_\emptyset)$
$\quad \mapsto \quad \text{nonmatch}(n_{\{I\}}, 0),$
$\quad\quad\quad \text{nonmatch}(n_{\{I\}}, 1),$
$\quad\quad\quad \text{match}(n_{\{I\}}, x_\emptyset),$
$\quad\quad\quad y_\emptyset := x_\emptyset \text{ - } 1,$
$\quad\quad\quad \text{fibonacci}(y_\emptyset, z_\emptyset),$
$\quad\quad\quad v_\emptyset := x_\emptyset \text{ - } 2,$
$\quad\quad\quad \text{fibonacci}(v_\emptyset, w_\emptyset),$
$\quad\quad\quad m_\emptyset := z_\emptyset + w_\emptyset.$

Inference of Typed-Norms: Algorithm

$\text{fibonacci\_list}(xs_{\{L\}}, ys_{\emptyset})$
$\quad \mapsto \quad \text{case}_0(xs_{\{L\}}, ys_{\emptyset}).$

$\text{case}_0(xs_{\{L\}}, ys_{\emptyset})$
$\quad \mapsto \quad \text{match}(xs_{\{L\}}, \text{Nil}),$
$\qquad\quad ys_{\emptyset} := \text{Nil}.$

$\text{case}_0(xs_{\{L\}}, ys_{\emptyset})$
$\quad \mapsto \quad \text{nonmatch}(xs_{\{L\}}, \text{Nil}),$
$\qquad\quad \text{match}(xs_{\{L\}}, \text{Cons}(n_{\emptyset}, zs_{\emptyset})),$
$\qquad\quad \text{fibonacci}(n_{\emptyset}, x_{\emptyset}),$
$\qquad\quad \text{fibonacci\_list}(zs_{\emptyset}, ws_{\emptyset}),$
$\qquad\quad ys_{\emptyset} := \text{Cons}(x_{\emptyset}, ws_{\emptyset}).$

$\text{fibonacci}(n_{\{I\}}, m_{\emptyset})$
$\quad \mapsto \quad x_{\{I\}} := 0,$
$\qquad\quad \text{case}_1(x_{\{I\}}, n_{\{I\}}, m_{\emptyset}).$

$\text{case}_1(x_{\{I\}}, n_{\{I\}}, m_{\emptyset})$
$\quad \mapsto \quad n_{\{I\}} \geq x_{\{I\}},$
$\qquad\quad \text{case}_2(n_{\{I\}}, m_{\emptyset}).$

$\text{case}_1(x_{\{I\}}, n_{\{I\}}, m_{\emptyset})$
$\quad \mapsto \quad n_{\{I\}} < x_{\{I\}},$
$\qquad\quad m_{\emptyset} := \text{-}1.$

$\text{case}_2(n_{\{I\}}, m_{\emptyset})$
$\quad \mapsto \quad \text{match}(n_{\{I\}}, 0),$
$\qquad\quad m_{\emptyset} := 0.$

$\text{case}_2(n_{\{I\}}, m_{\emptyset})$
$\quad \mapsto \quad \text{nonmatch}(n_{\{I\}}, 0),$
$\qquad\quad \text{match}(n_{\{I\}}, 1),$
$\qquad\quad m_{\emptyset} := 1.$

$\text{case}_2(n_{\{I\}}, m_{\emptyset})$
$\quad \mapsto \quad \text{nonmatch}(n_{\{I\}}, 0),$
$\qquad\quad \text{nonmatch}(n_{\{I\}}, 1),$
$\qquad\quad \text{match}(n_{\{I\}}, x_{\emptyset}),$
$\qquad\quad y_{\emptyset} := x_{\emptyset} \text{ - } 1,$
$\qquad\quad \text{fibonacci}(y_{\emptyset}, z_{\emptyset}),$
$\qquad\quad v_{\emptyset} := x_{\emptyset} \text{ - } 2,$
$\qquad\quad \text{fibonacci}(v_{\emptyset}, w_{\emptyset}),$
$\qquad\quad m_{\emptyset} := z_{\emptyset} + w_{\emptyset}.$

Inference of Typed-Norms: Algorithm

$\text{fibonacci\_list}(xs_{\{L\}}, ys_{\emptyset})$
$\quad \mapsto \quad \text{case}_0(xs_{\{L\}}, ys_{\emptyset}).$

$\text{case}_0(xs_{\{L\}}, ys_{\emptyset})$
$\quad \mapsto \quad \text{match}(xs_{\{L\}}, \text{Nil}),$
$\qquad\quad ys_{\emptyset} := \text{Nil}.$

$\text{case}_0(xs_{\{L\}}, ys_{\emptyset})$
$\quad \mapsto \quad \text{nonmatch}(xs_{\{L\}}, \text{Nil}),$
$\qquad\quad \text{match}(xs_{\{L\}}, \text{Cons}(n_{\{I\}}, zs_{\{L\}})),$
$\qquad\quad \text{fibonacci}(n_{\{I\}}, x_{\emptyset}),$
$\qquad\quad \text{fibonacci\_list}(zs_{\{L\}}, ws_{\emptyset}),$
$\qquad\quad ys_{\emptyset} := \text{Cons}(x_{\emptyset}, ws_{\emptyset}).$

$\text{fibonacci}(n_{\{I\}}, m_{\emptyset})$
$\quad \mapsto \quad x_{\{I\}} := 0,$
$\qquad\quad \text{case}_1(x_{\{I\}}, n_{\{I\}}, m_{\emptyset}).$

$\text{case}_1(x_{\{I\}}, n_{\{I\}}, m_{\emptyset})$
$\quad \mapsto \quad n_{\{I\}} \geq x_{\{I\}},$
$\qquad\quad \text{case}_2(n_{\{I\}}, m_{\emptyset}).$

$\text{case}_1(x_{\{I\}}, n_{\{I\}}, m_{\emptyset})$
$\quad \mapsto \quad n_{\{I\}} < x_{\{I\}},$
$\qquad\quad m_{\emptyset} := \text{-}1.$

$\text{case}_2(n_{\{I\}}, m_{\emptyset})$
$\quad \mapsto \quad \text{match}(n_{\{I\}}, 0),$
$\qquad\quad m_{\emptyset} := 0.$

$\text{case}_2(n_{\{I\}}, m_{\emptyset})$
$\quad \mapsto \quad \text{nonmatch}(n_{\{I\}}, 0),$
$\qquad\quad \text{match}(n_{\{I\}}, 1),$
$\qquad\quad m_{\emptyset} := 1.$

$\text{case}_2(n_{\{I\}}, m_{\emptyset})$
$\quad \mapsto \quad \text{nonmatch}(n_{\{I\}}, 0),$
$\qquad\quad \text{nonmatch}(n_{\{I\}}, 1),$
$\qquad\quad \text{match}(n_{\{I\}}, x_{\emptyset}),$
$\qquad\quad y_{\{I\}} := x_{\emptyset} \text{ - } 1,$
$\qquad\quad \text{fibonacci}(y_{\{I\}}, z_{\emptyset}),$
$\qquad\quad v_{\{I\}} := x_{\emptyset} \text{ - } 2,$
$\qquad\quad \text{fibonacci}(v_{\{I\}}, w_{\emptyset}),$
$\qquad\quad m_{\emptyset} := z_{\emptyset} + w_{\emptyset}.$

Inference of Typed-Norms: Algorithm

$\text{fibonacci\_list}(xs_{\{L\}}, ys_{\emptyset})$
$\quad \mapsto \quad \text{case}_0(xs_{\{L\}}, ys_{\emptyset}).$

$\text{case}_0(xs_{\{I,L\}}, ys_{\emptyset})$
$\quad \mapsto \quad \text{match}(xs_{\{I,L\}}, \text{Nil}),$
$\qquad\quad ys_{\emptyset} := \text{Nil}.$

$\text{case}_0(xs_{\{I,L\}}, ys_{\emptyset})$
$\quad \mapsto \quad \text{nonmatch}(xs_{\{I,L\}}, \text{Nil}),$
$\qquad\quad \text{match}(xs_{\{I,L\}}, \text{Cons}(n_{\{I\}}, zs_{\{L\}})),$
$\qquad\quad \text{fibonacci}(n_{\{I\}}, x_{\emptyset}),$
$\qquad\quad \text{fibonacci\_list}(zs_{\{L\}}, ws_{\emptyset}),$
$\qquad\quad ys_{\emptyset} := \text{Cons}(x_{\emptyset}, ws_{\emptyset}).$

$\text{fibonacci}(n_{\{I\}}, m_{\emptyset})$
$\quad \mapsto \quad x_{\{I\}} := 0,$
$\qquad\quad \text{case}_1(x_{\{I\}}, n_{\{I\}}, m_{\emptyset}).$

$\text{case}_1(x_{\{I\}}, n_{\{I\}}, m_{\emptyset})$
$\quad \mapsto \quad n_{\{I\}} \geq x_{\{I\}},$
$\qquad\quad \text{case}_2(n_{\{I\}}, m_{\emptyset}).$

$\text{case}_1(x_{\{I\}}, n_{\{I\}}, m_{\emptyset})$
$\quad \mapsto \quad n_{\{I\}} < x_{\{I\}},$
$\qquad\quad m_{\emptyset} := \text{-}1.$

$\text{case}_2(n_{\{I\}}, m_{\emptyset})$
$\quad \mapsto \quad \text{match}(n_{\{I\}}, 0),$
$\qquad\quad m_{\emptyset} := 0.$

$\text{case}_2(n_{\{I\}}, m_{\emptyset})$
$\quad \mapsto \quad \text{nonmatch}(n_{\{I\}}, 0),$
$\qquad\quad \text{match}(n_{\{I\}}, 1),$
$\qquad\quad m_{\emptyset} := 1.$

$\text{case}_2(n_{\{I\}}, m_{\emptyset})$
$\quad \mapsto \quad \text{nonmatch}(n_{\{I\}}, 0),$
$\qquad\quad \text{nonmatch}(n_{\{I\}}, 1),$
$\qquad\quad \text{match}(n_{\{I\}}, x_{\{I\}}),$
$\qquad\quad y_{\{I\}} := x_{\{I\}} \text{ - } 1,$
$\qquad\quad \text{fibonacci}(y_{\{I\}}, z_{\emptyset}),$
$\qquad\quad v_{\{I\}} := x_{\{I\}} \text{ - } 2,$
$\qquad\quad \text{fibonacci}(v_{\{I\}}, w_{\emptyset}),$
$\qquad\quad m_{\emptyset} := z_{\emptyset} + w_{\emptyset}.$

Inference of Typed-Norms: Algorithm

$\text{fibonacci\_list}(xs_{\{I,L\}}, ys_{\emptyset})$
$\quad \mapsto \quad \text{case}_0(xs_{\{I,L\}}, ys_{\emptyset}).$

$\text{case}_0(xs_{\{I,L\}}, ys_{\emptyset})$
$\quad \mapsto \quad \text{match}(xs_{\{I,L\}}, \text{Nil}),$
$\qquad\quad ys_{\emptyset} := \text{Nil}.$

$\text{case}_0(xs_{\{I,L\}}, ys_{\emptyset})$
$\quad \mapsto \quad \text{nonmatch}(xs_{\{I,L\}}, \text{Nil}),$
$\qquad\quad \text{match}(xs_{\{I,L\}}, \text{Cons}(n_{\{I\}}, zs_{\{L\}})),$
$\qquad\quad \text{fibonacci}(n_{\{I\}}, x_{\emptyset}),$
$\qquad\quad \text{fibonacci\_list}(zs_{\{L\}}, ws_{\emptyset}),$
$\qquad\quad ys_{\emptyset} := \text{Cons}(x_{\emptyset}, ws_{\emptyset}).$

$\text{fibonacci}(n_{\{I\}}, m_{\emptyset})$
$\quad \mapsto \quad x_{\{I\}} := 0,$
$\qquad\quad \text{case}_1(x_{\{I\}}, n_{\{I\}}, m_{\emptyset}).$

$\text{case}_1(x_{\{I\}}, n_{\{I\}}, m_{\emptyset})$
$\quad \mapsto \quad n_{\{I\}} \geq x_{\{I\}},$
$\qquad\quad \text{case}_2(n_{\{I\}}, m_{\emptyset}).$

$\text{case}_1(x_{\{I\}}, n_{\{I\}}, m_{\emptyset})$
$\quad \mapsto \quad n_{\{I\}} < x_{\{I\}},$
$\qquad\quad m_{\emptyset} := \text{-1}.$

$\text{case}_2(n_{\{I\}}, m_{\emptyset})$
$\quad \mapsto \quad \text{match}(n_{\{I\}}, 0),$
$\qquad\quad m_{\emptyset} := 0.$

$\text{case}_2(n_{\{I\}}, m_{\emptyset})$
$\quad \mapsto \quad \text{nonmatch}(n_{\{I\}}, 0),$
$\qquad\quad \text{match}(n_{\{I\}}, 1),$
$\qquad\quad m_{\emptyset} := 1.$

$\text{case}_2(n_{\{I\}}, m_{\emptyset})$
$\quad \mapsto \quad \text{nonmatch}(n_{\{I\}}, 0),$
$\qquad\quad \text{nonmatch}(n_{\{I\}}, 1),$
$\qquad\quad \text{match}(n_{\{I\}}, x_{\{I\}}),$
$\qquad\quad y_{\{I\}} := x_{\{I\}} \text{ - } 1,$
$\qquad\quad \text{fibonacci}(y_{\{I\}}, z_{\emptyset}),$
$\qquad\quad v_{\{I\}} := x_{\{I\}} \text{ - } 2,$
$\qquad\quad \text{fibonacci}(v_{\{I\}}, w_{\emptyset}),$
$\qquad\quad m_{\emptyset} := z_{\emptyset} + w_{\emptyset}.$

Inference of Typed-Norms: Algorithm

$\text{fibonacci\_list}(xs_{\{I,L\}}, ys_{\emptyset})$
$\quad \mapsto \quad \text{case}_0(xs_{\{I,L\}}, ys_{\emptyset}).$

$\text{case}_0(xs_{\{I,L\}}, ys_{\emptyset})$
$\quad \mapsto \quad \text{match}(xs_{\{I,L\}}, \text{Nil}),$
$\qquad\quad ys_{\emptyset} := \text{Nil}.$

$\text{case}_0(xs_{\{I,L\}}, ys_{\emptyset})$
$\quad \mapsto \quad \text{nonmatch}(xs_{\{I,L\}}, \text{Nil}),$
$\qquad\quad \text{match}(xs_{\{I,L\}}, \text{Cons}(n_{\{I\}}, zs_{\{I,L\}})),$
$\qquad\quad \text{fibonacci}(n_{\{I\}}, x_{\emptyset}),$
$\qquad\quad \text{fibonacci\_list}(zs_{\{I,L\}}, ws_{\emptyset}),$
$\qquad\quad ys_{\emptyset} := \text{Cons}(x_{\emptyset}, ws_{\emptyset}).$

$\text{fibonacci}(n_{\{I\}}, m_{\emptyset})$
$\quad \mapsto \quad x_{\{I\}} := 0,$
$\qquad\quad \text{case}_1(x_{\{I\}}, n_{\{I\}}, m_{\emptyset}).$

$\text{case}_1(x_{\{I\}}, n_{\{I\}}, m_{\emptyset})$
$\quad \mapsto \quad n_{\{I\}} \geq x_{\{I\}},$
$\qquad\quad \text{case}_2(n_{\{I\}}, m_{\emptyset}).$

$\text{case}_1(x_{\{I\}}, n_{\{I\}}, m_{\emptyset})$
$\quad \mapsto \quad n_{\{I\}} < x_{\{I\}},$
$\qquad\quad m_{\emptyset} := \text{-1}.$

$\text{case}_2(n_{\{I\}}, m_{\emptyset})$
$\quad \mapsto \quad \text{match}(n_{\{I\}}, 0),$
$\qquad\quad m_{\emptyset} := 0.$

$\text{case}_2(n_{\{I\}}, m_{\emptyset})$
$\quad \mapsto \quad \text{nonmatch}(n_{\{I\}}, 0),$
$\qquad\quad \text{match}(n_{\{I\}}, 1),$
$\qquad\quad m_{\emptyset} := 1.$

$\text{case}_2(n_{\{I\}}, m_{\emptyset})$
$\quad \mapsto \quad \text{nonmatch}(n_{\{I\}}, 0),$
$\qquad\quad \text{nonmatch}(n_{\{I\}}, 1),$
$\qquad\quad \text{match}(n_{\{I\}}, x_{\{I\}}),$
$\qquad\quad y_{\{I\}} := x_{\{I\}} \text{ - } 1,$
$\qquad\quad \text{fibonacci}(y_{\{I\}}, z_{\emptyset}),$
$\qquad\quad v_{\{I\}} := x_{\{I\}} \text{ - } 2,$
$\qquad\quad \text{fibonacci}(v_{\{I\}}, w_{\emptyset}),$
$\qquad\quad m_{\emptyset} := z_{\emptyset} + w_{\emptyset}.$

Inference of Typed-Norms: Algorithm

$\text{fibonacci\_list}(xs_{\{I,L\}}, ys_{\emptyset})$
$\quad \mapsto \quad \text{case}_0(xs_{\{I,L\}}, ys_{\emptyset}).$

$\text{case}_0(xs_{\{I,L\}}, ys_{\emptyset})$
$\quad \mapsto \quad \text{match}(xs_{\{I,L\}}, \text{Nil}),$
$\qquad\quad ys_{\emptyset} := \text{Nil}.$

$\text{case}_0(xs_{\{I,L\}}, ys_{\emptyset})$
$\quad \mapsto \quad \text{nonmatch}(xs_{\{I,L\}}, \text{Nil}),$
$\qquad\quad \text{match}(xs_{\{I,L\}}, \text{Cons}(n_{\{I\}}, zs_{\{I,L\}})),$
$\qquad\quad \text{fibonacci}(n_{\{I\}}, x_{\emptyset}),$
$\qquad\quad \text{fibonacci\_list}(zs_{\{I,L\}}, ws_{\emptyset}),$
$\qquad\quad ys_{\emptyset} := \text{Cons}(x_{\emptyset}, ws_{\emptyset}).$

$\text{fibonacci}(n_{\{I\}}, m_{\emptyset})$
$\quad \mapsto \quad x_{\{I\}} := 0,$
$\qquad\quad \text{case}_1(x_{\{I\}}, n_{\{I\}}, m_{\emptyset}).$

$\text{case}_1(x_{\{I\}}, n_{\{I\}}, m_{\emptyset})$
$\quad \mapsto \quad n_{\{I\}} \geq x_{\{I\}},$
$\qquad\quad \text{case}_2(n_{\{I\}}, m_{\emptyset}).$

$\text{case}_1(x_{\{I\}}, n_{\{I\}}, m_{\emptyset})$
$\quad \mapsto \quad n_{\{I\}} < x_{\{I\}},$
$\qquad\quad m_{\emptyset} := \text{-}1.$

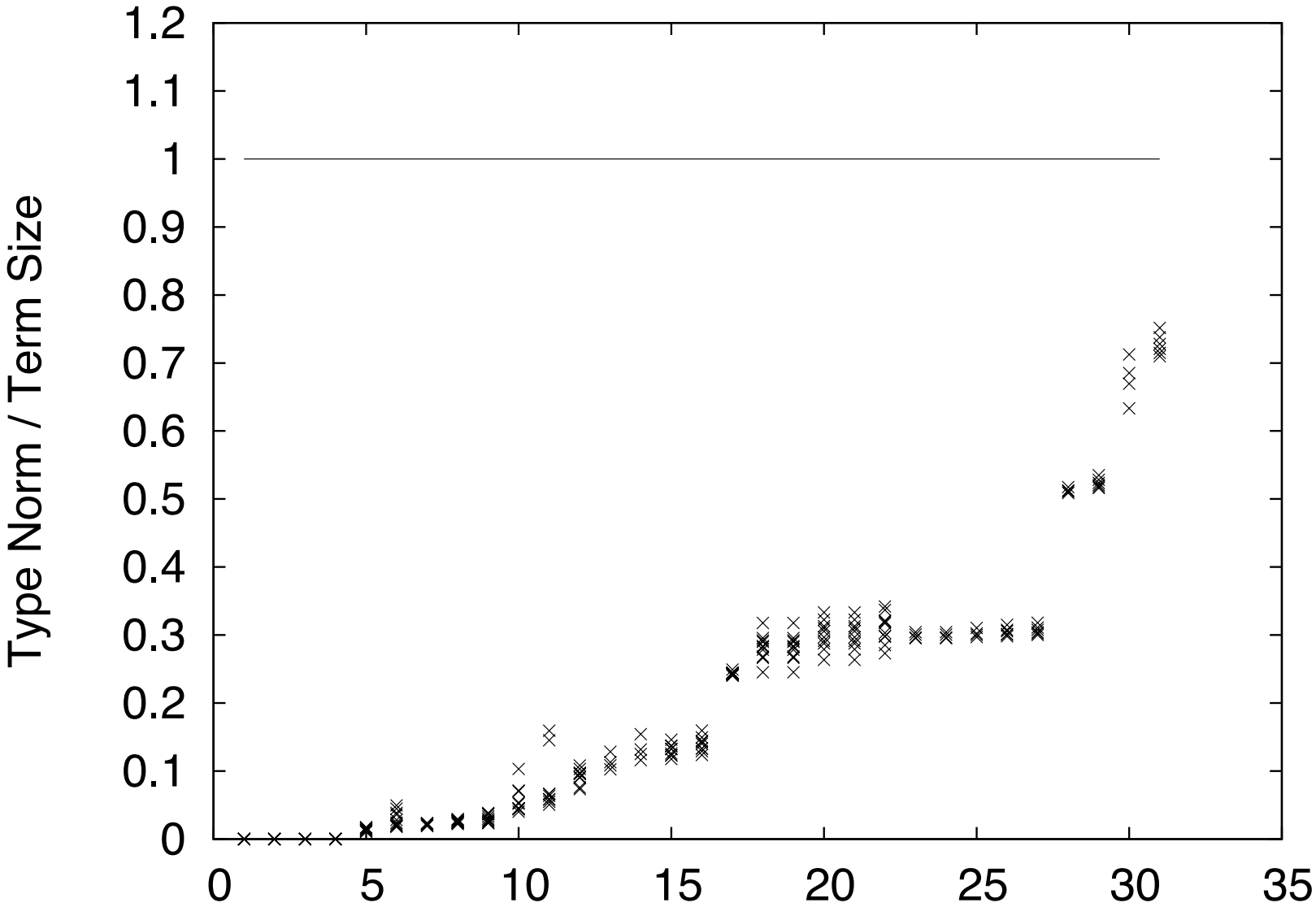
$\text{case}_2(n_{\{I\}}, m_{\emptyset})$
$\quad \mapsto \quad \text{match}(n_{\{I\}}, 0),$
$\qquad\quad m_{\emptyset} := 0.$

$\text{case}_2(n_{\{I\}}, m_{\emptyset})$
$\quad \mapsto \quad \text{nonmatch}(n_{\{I\}}, 0),$
$\qquad\quad \text{match}(n_{\{I\}}, 1),$
$\qquad\quad m_{\emptyset} := 1.$

$\text{case}_2(n_{\{I\}}, m_{\emptyset})$
$\quad \mapsto \quad \text{nonmatch}(n_{\{I\}}, 0),$
$\qquad\quad \text{nonmatch}(n_{\{I\}}, 1),$
$\qquad\quad \text{match}(n_{\{I\}}, x_{\{I\}}),$
$\qquad\quad y_{\{I\}} := x_{\{I\}} \text{ - } 1,$
$\qquad\quad \text{fibonacci}(y_{\{I\}}, z_{\emptyset}),$
$\qquad\quad v_{\{I\}} := x_{\{I\}} \text{ - } 2,$
$\qquad\quad \text{fibonacci}(v_{\{I\}}, w_{\emptyset}),$
$\qquad\quad m_{\emptyset} := z_{\emptyset} + w_{\emptyset}.$

Outline

Experiments



Experiments

The experiments have been performed on an Intel Core 2 Duo at 2.4GHz with 8GB of RAM, running OS X 10.9.

Table: Run-Time statistics (in ms.) on 61 functions: (1) using term-size; (2) using all type-norms; and (3) using only significant typed-norms. Total is $T_{sa}+T_{ac}+T_{ub}$.

Configuration	T_{sa}	Av. T_{sa}	T_{ac}	Av. T_{ac}	T_{ub}	Av. T_{ub}	**Total**
(1)	0	0	120	2	3911	65	4031
(2)	1633	27	631	11	14230	234	16494
(3)	2161	36	255	5	4488	74	6904

Outline

Conclusions

- We have presented a novel transformational approach to resource analysis with typed-norms.
- This transformation has the advantage that its formalization can be done by only adapting the first phase of cost analysis in which the program is transformed into an intermediate abstract program.
- We also outlined an algorithm for the inference of typed-norms which can detect automatically which norms are useful to later infer the resource consumption.