On the limits of the Classical Approach to Cost Analysis

Diego E. Alonso-Blas, Samir Genaim

Complutense University of Madrid (UCM)

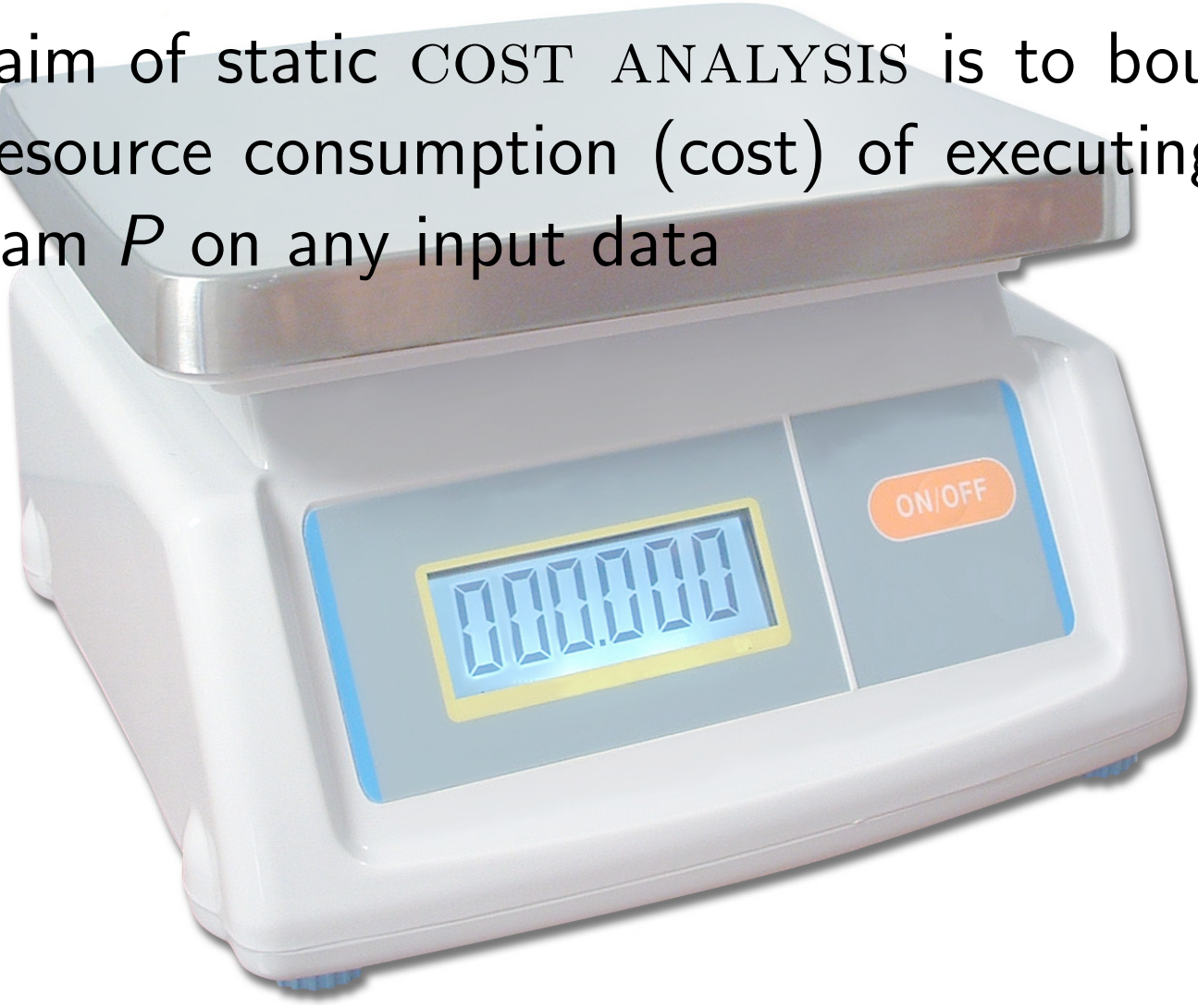
October 16^{th} 2012

What is Cost Analysis?

What is Cost Analysis?

What is Cost Analysis?

The aim of static COST ANALYSIS is to bound the resource consumption (cost) of executing a program P on any input data



What is Cost Analysis?

The aim of static COST ANALYSIS is to bound the resource consumption (cost) of executing a program P on any input data

What is Cost Analysis?

The aim of static COST ANALYSIS is to bound the resource consumption (cost) of executing a program P on any input data

Resources

- Loop iterations
- Execution steps
- Visits to program point
- Stack and Heap space

What is Cost Analysis?

The aim of static COST ANALYSIS is to bound the resource consumption (cost) of executing a program P on any input data

Resources

- Loop iterations
- Execution steps
- Visits to program point
- Stack and Heap space

Bound functions

- Upper Bounds (*worst case*)
- Lower Bounds (*best case*)
- Non-Asymptotic: $P(x) = 2x^2 + 3$
- Asymptotic: $P^u(x) \in \mathcal{O}(x^2)$

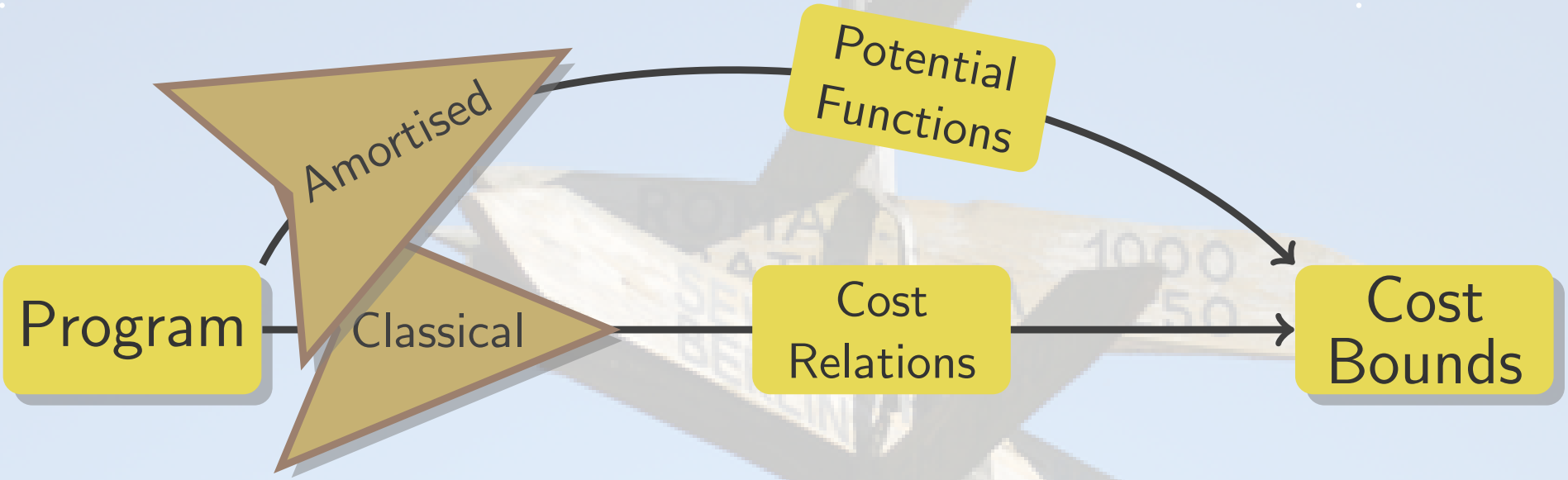
What is Cost Analysis?

What is the right way to Cost Analysis?

What is the right way to Cost Analysis?

'75: Classical Approach: abstract program into Cost Relations, solve these into close-form bounds

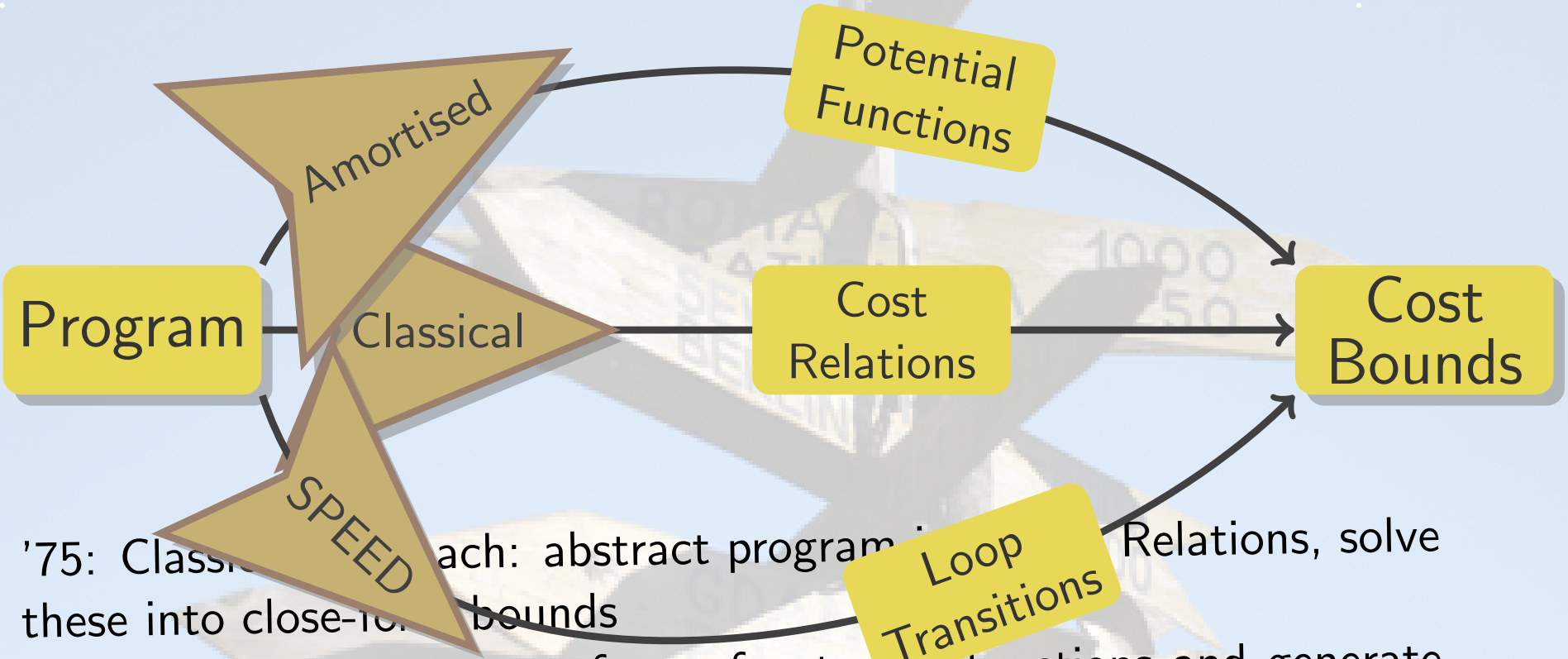
What is the right way to Cost Analysis?



'75: Classical Approach: abstract program into Cost Relations, solve these into close-form bounds

'03: Amortised: assume a form of potential functions and generate constraints using a typed-based system

What is the right way to Cost Analysis?

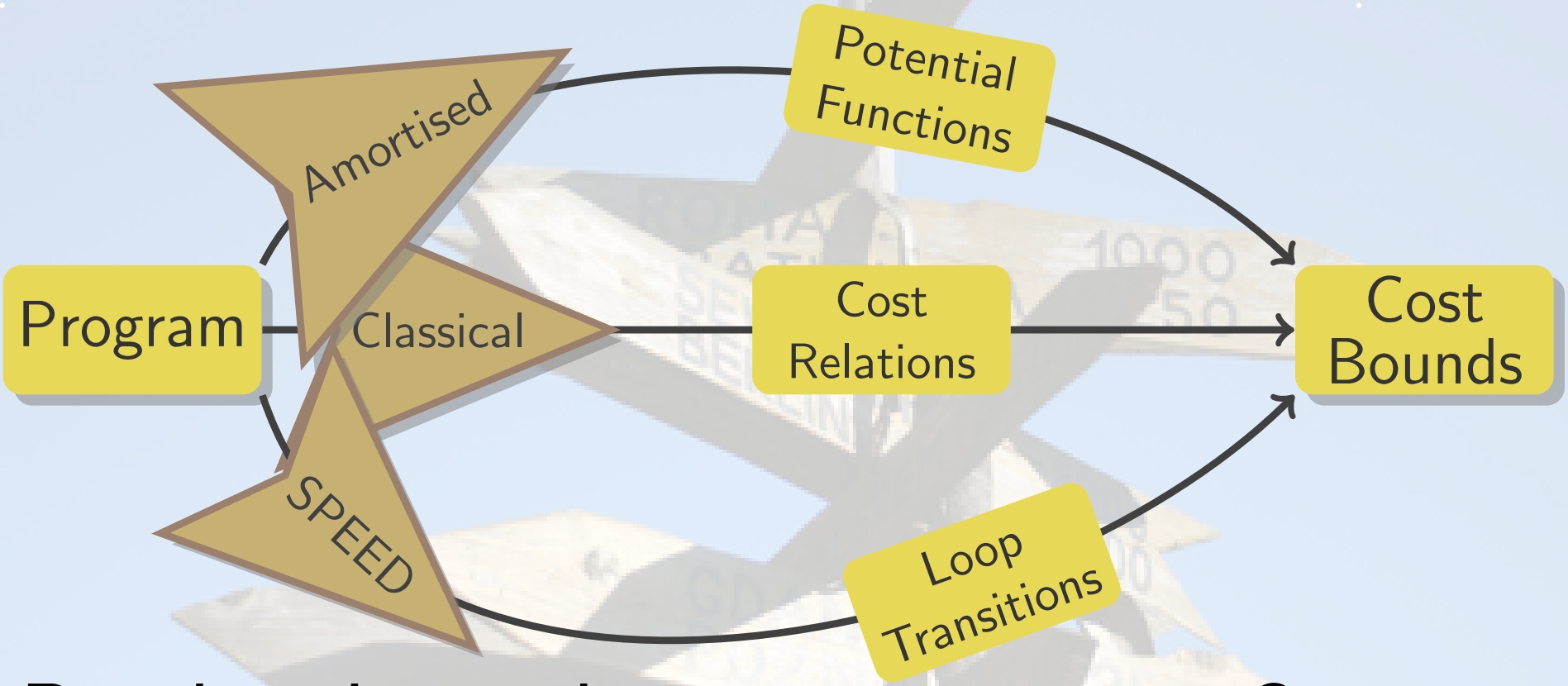


'75: Classi ach: abstract program i Relations, solve these into close-fo bounds

'03: Amortised: assume a form of poten ial functions and generate constraints using a typed-based system

'08: SPEED: abstract individual loop transitions, generate bounds for each transition, combine bounds

What is the right way to Cost Analysis?



Do they have the same precission?

Can they handle same programs?

Cost Analysis: A working example

```
class Stack {
  Node top;

  //@requires m >= 1
  void rpop(int m){
    if (top != null && *){
      top=top.next; //pop
      //@acquire(m)
    }
  }

  //@requires m >= 0
  void main(int m) {
    while( m > 0 ) {
      rpop(m)  ;
      // push('a');
      top = new Node(*,top);
      //@acquire(1)
      m = m-1 ;
    }}}
```

Cost Analysis: A working example

```
class Stack {
  Node top;

  //@requires m >= 1
  void rpop(int m){
    if (top != null && *){
      top=top.next; //pop
      //@acquire(m)
    }
  }

  //@requires m >= 0
  void main(int m) {
    while( m > 0 ) {
      rpop(m)  ;
      // push('a');
      top = new Node(*,top);
      //@acquire(1)
      m = m-1 ;
    }}}
```

Upper Bounds (Worst Case)

$$rpop^u(t, m) = m$$
$$main^u(t, m) = \frac{m^2+m}{2} + m$$

Meaning of an Upper Bound

Executing `main[this.top, m]`, takes at most $m^2 + m$ resources

Cost Analysis: A working example

```
class Stack {
  Node top;

  //@requires m >= 1
  void rpop(int m){
    if (top != null && *){
      top=top.next; //pop
      //@acquire(m)
    }
  }

  //@requires m >= 0
  void main(int m) {
    while( m > 0 ) {
      rpop(m)  ;
      // push('a');
      top = new Node(*,top);
      //@acquire(1)
      m = m-1 ;
    }}}
```

Upper Bounds (Worst Case)

$$rpop^u(t, m) = m$$
$$main^u(t, m) = \frac{m^2+m}{2} + m$$

Cost Analysis: A working example

```
class Stack {
  Node top;

  //@requires m >= 1
  void rpop(int m){
    if (top != null && *){
while  top=top.next; //pop
      //@acquire(m)
    }
  }

  //@requires m >= 0
  void main(int m) {
    while( m > 0 ) {
      rpop(m)  ;
      // push('a');
      top = new Node(*,top);
      //@acquire(1)
      m = m-1 ;
    }}}
```

Upper Bounds (Worst Case)

$$rpop^u(t, m) = m$$
$$main^u(t, m) = \frac{m^2+m}{2} + m$$

Cost Analysis: A working example

```
class Stack {
  Node top;

  //@requires m >= 1
  void rpop(int m){
    if (top != null && *){
      top=top.next; //pop
      //@acquire(m)
    }
  }

  //@requires m >= 0
  void main(int m) {
    while( m > 0 ) {
      rpop(m)  ;
      // push('a');
      top = new Node(*,top);
      //@acquire(1)
      m = m-1 ;
    }}}
```

while

Upper Bounds (Worst Case)

$$rpop^u(t, m) = m{*}\mathbf{t}$$

$$main^u(t, m) = \frac{m^2+m}{2} + m{*}\mathbf{t}$$

Cost Analysis: A working example

```
class Stack {
  Node top;

  //@requires m >= 1
  void rpop(int m){
    if (top != null && *){
while  top=top.next; //pop
      //@acquire(m)
    }
  }

  //@requires m >= 0
  void main(int m) {
    while( m > 0 ) {
      rpop(m)  ;
      // push('a');
      top = new Node(*,top);
      //@acquire(1)
      m = m-1 ;
    }}}
```

Upper Bounds (Worst Case)

$$rpop^u(t,m) = m{*}\mathbf{t}$$
$$main^u(t,m) = \frac{m^2+m}{2} + m{*}\mathbf{t}$$

Inferred upper bounds (UB)

$$rpop^u(t,m) = m * t$$
$$main^u(t,m) = \frac{m^3+3t(m^2+m)+5m}{6}$$

Cost Analysis: A working example

```
class Stack {
  Node top;

  //@requires m >= 1
  void rpop(int m){
    if (top != null && *){
while top=top.next; //pop
      //@acquire(m)
    }
  }

  //@requires m >= 0
  void main(int m) {
    while( m > 0 ) {
      rpop(m)  ;
      // push('a');
      top = new Node(*,top);
      //@acquire(1)
      m = m-1 ;
    }}}
```

Upper Bounds (Worst Case)

$$rpop^u(t,m) = m{*}\mathbf{t}$$

$$main^u(t,m) = \frac{m^2+m}{2} + m{*}\mathbf{t}$$

$$\mathcal{O}(m^2 + tm)$$

Inferred

$$\mathcal{O}(m^3 + tm^2)$$

$$rpop^u(t,m) = m * t$$

$$main^u(t,m) = \frac{m^3+3t(m^2+m)+5m}{6}$$

Cost Analysis: A working example

```
class Stack {
  Node top;

  //@requires m >= 1
  void rpop(int m){
    if (top != null && *){
      top=top.next; //pop
      //@acquire(1)
    }
  }

  //@requires m >
  void main(int m
    while( m > 0 )
      rpop(m)  ;
      // push('a');
      top = new Node(*,top);
      //@acquire(1)
      m = m-1 ;
    }}}
```

while

Why?

Upper Bounds (Worst Case)

$$rpop^u(t,m) = m{*}\mathbf{t}$$

$$main^u(t,m) = \frac{m^2+m}{2} + m{*}\mathbf{t}$$

$\mathcal{O}(m^2 + tm)$

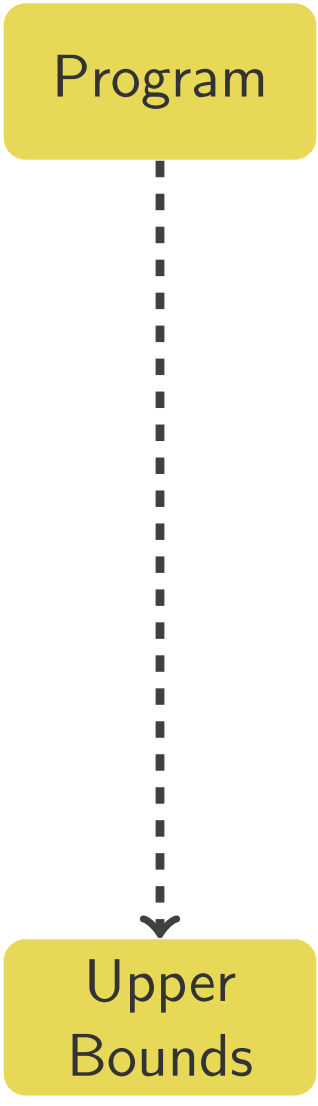
red

$\mathcal{O}(m^3 + tm^2)$

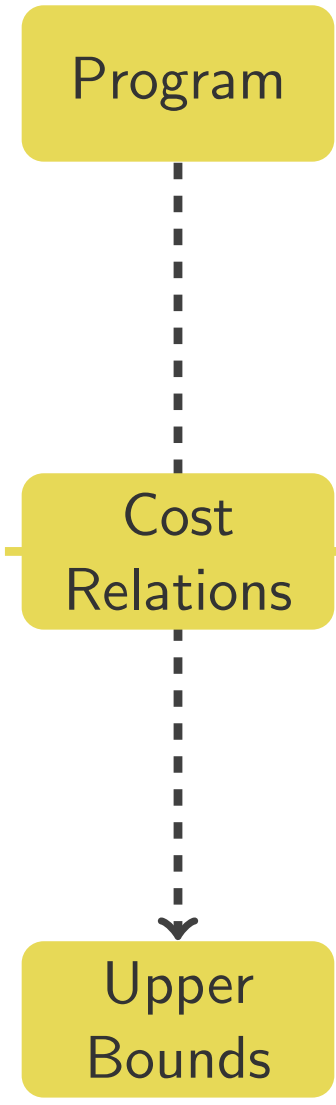
$$rpop^u(t,m) = m * t$$

$$main^u(t,m) = \frac{m^3+3t(m^2+m)+5m}{6}$$

Classical Approach to Cost Analysis



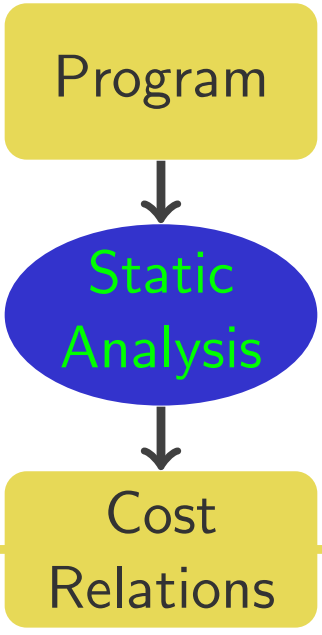
Classical Approach to Cost Analysis



The Classical approach uses Cost Relations to split Cost Analysis in two independent phases

Classical Approach to Cost Analysis

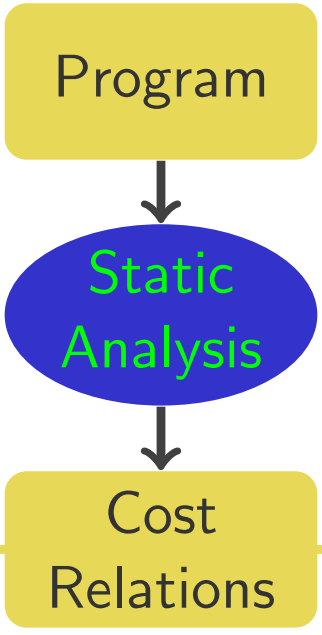
Classical Approach to Cost Analysis



An **analysis** abstracts the program semantics into a system of recursive **Cost Relations**

Upper Bounds

Classical Approach to Cost Analysis

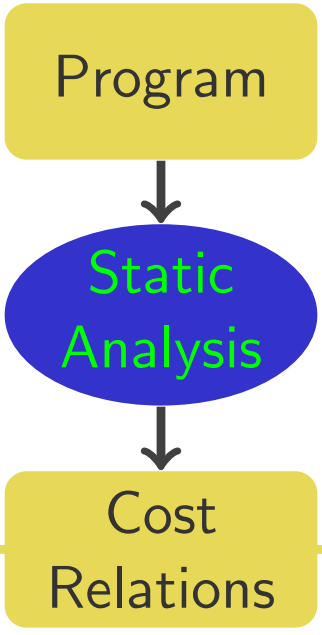


An **analysis** abstracts the program semantics into a system of recursive **Cost Relations**

Cost Relations, model the relation between the cost of each program fragment and its input sizes

Upper Bounds

Classical Approach to Cost Analysis



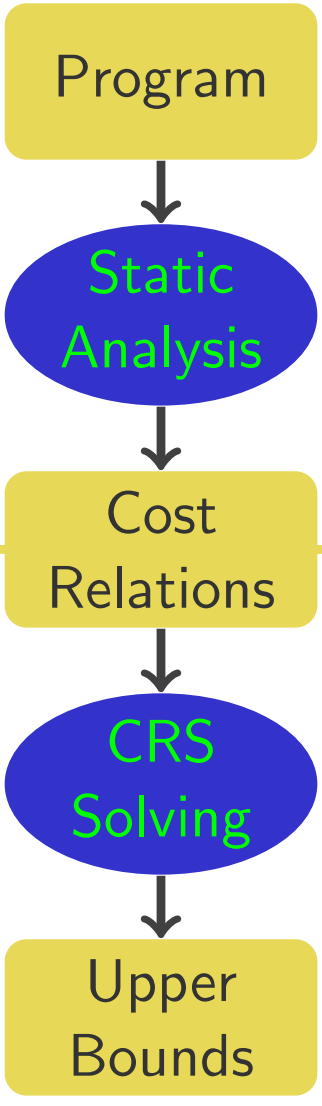
An **analysis** abstracts the program semantics into a system of recursive **Cost Relations**

Cost Relations, model the relation between the cost of each program fragment and its input sizes

Cost Relations have a recursive syntax, and a purely numerical yet undecidable semantics as $\mathbb{Z}^* \times \mathbb{R}^+$

Upper Bounds

Classical Approach to Cost Analysis



An **analysis** abstracts the program semantics into a system of recursive **Cost Relations**

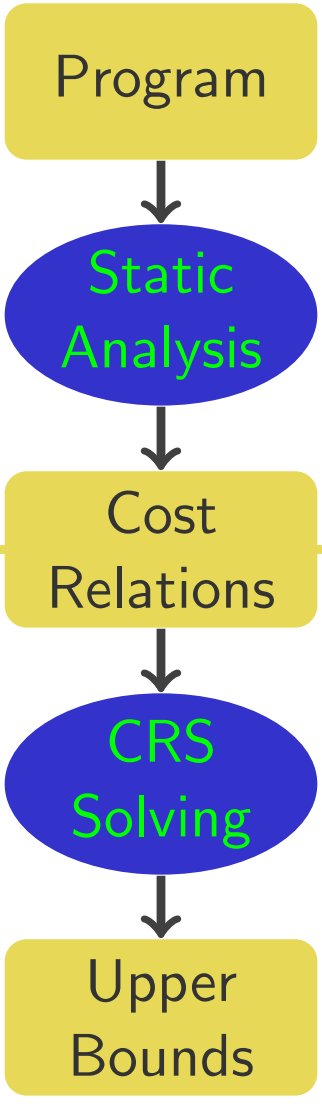
Cost Relations, model the relation between the cost of each program fragment and its input sizes

---

Cost Relations have a recursive syntax, and a purely numerical yet undecidable semantics as $\mathbb{Z}^* \times \mathbb{R}^+$

A dedicated **CRS solver** (e.g. PUBS, CAS) solves CRs into closed-form (non-recursive) bounds

Classical Approach to Cost Analysis



An **analysis** abstracts
a system of recursive

Cost Relations, moc
cost of each program

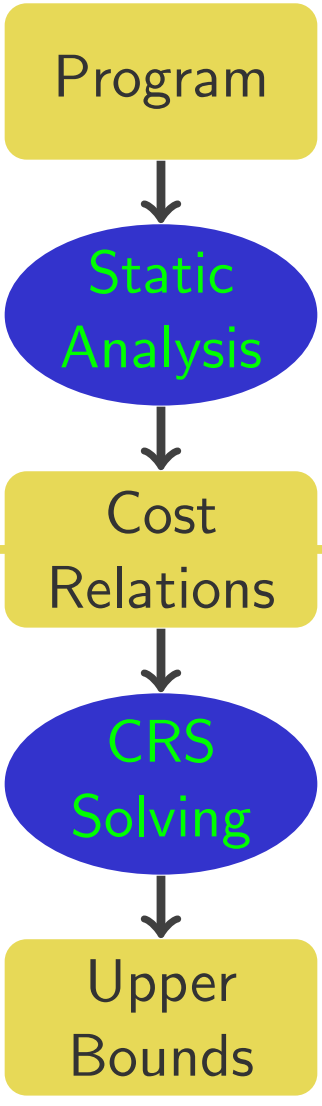
Resources

- Loop iterations
- Execution steps
- Visits to a program point
- Stack and Heap memory

Cost Relations have a recursive syntax, and a purely numerical yet undecidable semantics as $\mathbb{Z}^* \times \mathbb{R}^+$

A dedicated **CRS solver** (e.g. PUBS, CAS) solves CRs into closed-form (non-recursive) bounds

Classical Approach to Cost Analysis



An **analysis** abstracts
a system of recursive

Cost Relations, moc
cost of each program

Resources

- Loop iterations
- Execution steps
- Visits to a program point
- Stack and Heap memory

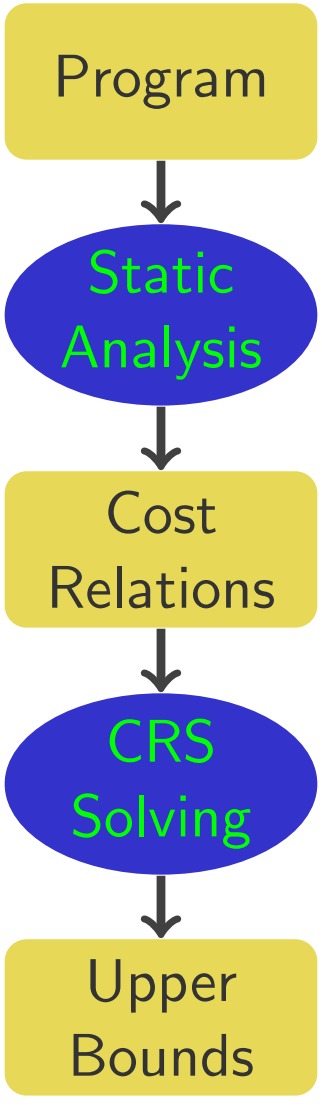
Cost Relations have
numerical yet unde

A dedicated **CRS s**
CRs into closed-for

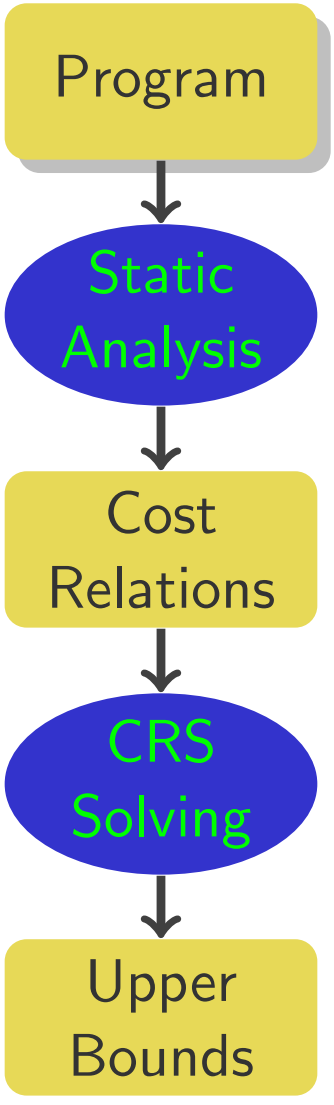
Bound functions

- Upper Bounds (***worst case***)
- Lower Bounds (***best case***)
- Non-Asymptotic: $2x^2 + 3x + 2$
- Asymptotic: $P^u(x) \in \mathcal{O}(x^2)$

Classical Approach to Cost Analysis: working example



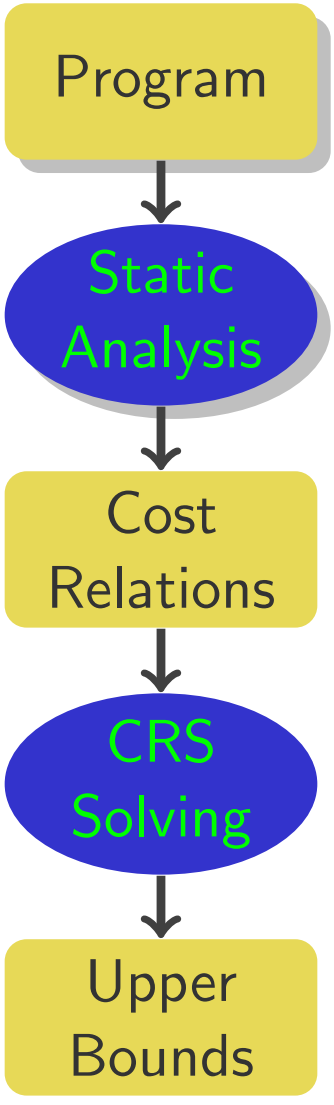
Classical Approach to Cost Analysis: working example




```
//@requires m >= 1
void rpop(int m){
  if (top != null && *){
    //pop
    top=top.next ;
    //@acquire(m)
}}
```

```
//@requires m >= 0
void main(int m) {
  while( m > 0 ) {
    rpop(m)  ;
    // push(*);
    top = new
     Node(*,top);
    //@acquire(1)
    m = m-1     ;
}}}
```

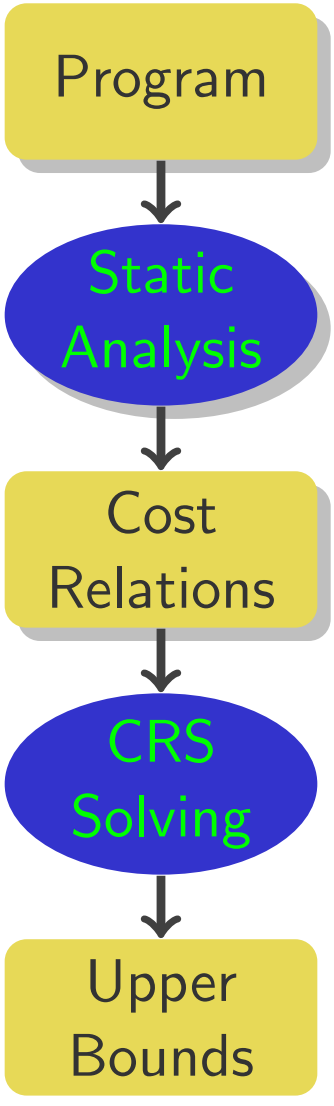
Classical Approach to Cost Analysis: working example




```
//@requires m >= 1
void rpop(int m){
  if (top != null && *){
    //pop
    top=top.next ;
    //@acquire(m)
}}
```

```
//@requires m >= 0
void main(int m) {
  while( m > 0 ) {
    rpop(m)  ;
    // push(*);
    top = new
     Node(*,top);
    //@acquire(1)
    m = m-1      ;
}}}
```

Classical Approach to Cost Analysis: working example




```
//@requires m >= 1
void rpop(int m){
  if (top != null && *){
    //pop
    top=top.next ;
    //@acquire(m)
}}
```

```
//@requires m >= 0
void main(int m) {
  while( m > 0 ) {
    rpop(m)  ;
    // push(*);
    top = new
     Node(*,top);
    //@acquire(1)
    m = m-1      ;
}}}
```

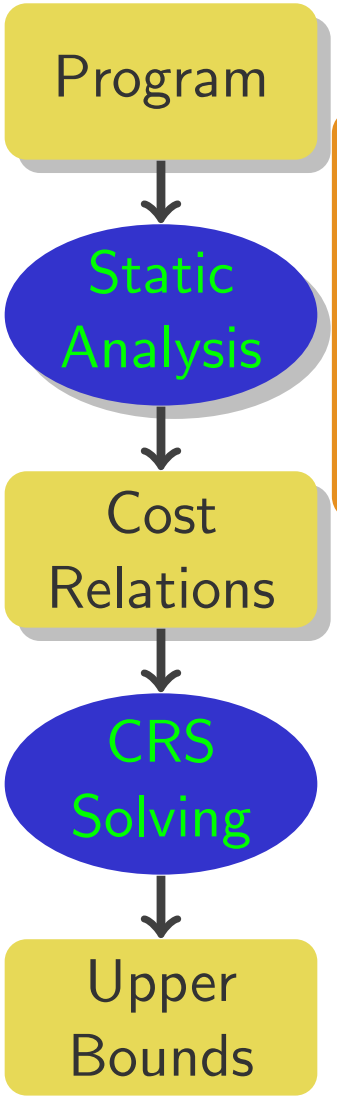
$$main(t,m) = 0 \quad \{m = 0, t \geq 0\}$$

$$main(t,m) = \begin{matrix} rpop(t,m) + 1 + \\ main(t_2, m_1) \end{matrix} \quad \left\{ \begin{matrix} m \geq 1, t \geq 0, m_1 = m - 1 \\ t_2 = t' + 1, t \geq t' \geq t - 1 \end{matrix} \right\}$$

$$rpop(t,m) = 0 \quad \{m \geq 1\}$$

$$rpop(t,m) = m \quad \{m \geq 1, t \geq 1\}$$

Classical Approach to Cost Analysis: working example



main : $\mathbb{Z}^2 \mapsto \mathcal{P}(\mathbb{R})$

$main(t, m)$ contains possible costs of executing `main(this.top,m)` if `(this.top,m)` have sizes (t, m)

```
//@requires m >= 1
                {
```

```
//@requires m >= 0
void main(int m) {
  while( m > 0 ) {
    rpop(m)   ;
    // push(*);
    top = new
     Node(*,top);
    //@acquire(1)
    m = m-1      ;
}}}
```

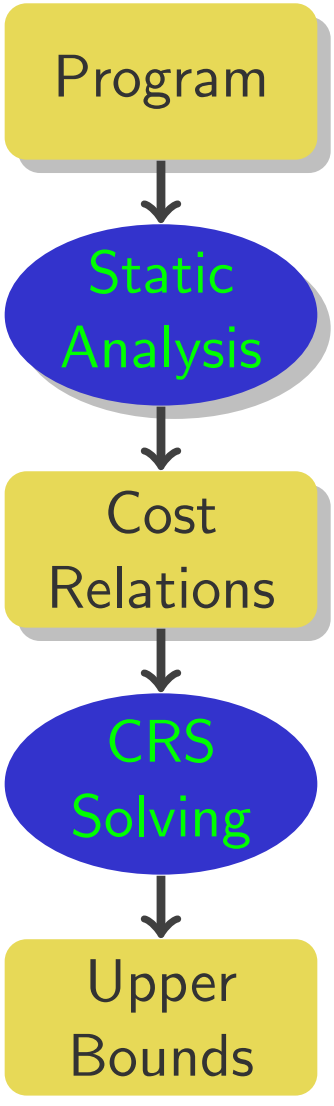
$$main(t, m) = 0 \quad \{m = 0, t \geq 0\}$$

$$main(t, m) = \begin{matrix} rpop(t, m) + 1 \\ main(t_2, m_1) \end{matrix} \quad \left\{ \begin{matrix} m \geq 1, t \geq 0, m_1 = m - 1 \\ t_2 = t' + 1, t \geq t' \geq t - 1 \end{matrix} \right\}$$

$$rpop(t, m) = 0 \quad \{m \geq 1\}$$

$$rpop(t, m) = m \quad \{m \geq 1, t \geq 1\}$$

Classical Approach to Cost Analysis: working example




```
//@requires m >= 1
void rpop(int m){
  if (top != null && *){
    //pop
    top=top.next ;
    //@acquire(m)
}}
```

```
//@requires m >= 0
void main(int m) {
  while( m > 0 ) {
    rpop(m)  ;
    // push(*);
    top = new
     Node(*,top);
    //@acquire(1)
    m = m-1      ;
}}}
```

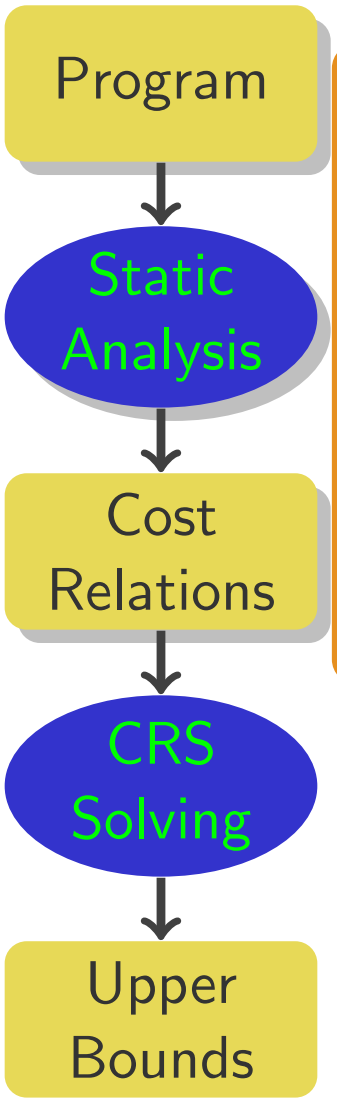
$$main(t,m) = 0 \quad \{m = 0, t \geq 0\}$$

$$main(t,m) = \begin{matrix} rpop(t,m) + 1 + \\ main(t_2, m_1) \end{matrix} \quad \left\{ \begin{matrix} m \geq 1, t \geq 0, m_1 = m - 1 \\ t_2 = t' + 1, t \geq t' \geq t - 1 \end{matrix} \right\}$$

$$rpop(t,m) = 0 \quad \{m \geq 1\}$$

$$rpop(t,m) = m \quad \{m \geq 1, t \geq 1\}$$

Classical Approach to Cost Analysis: working example



Modular definition

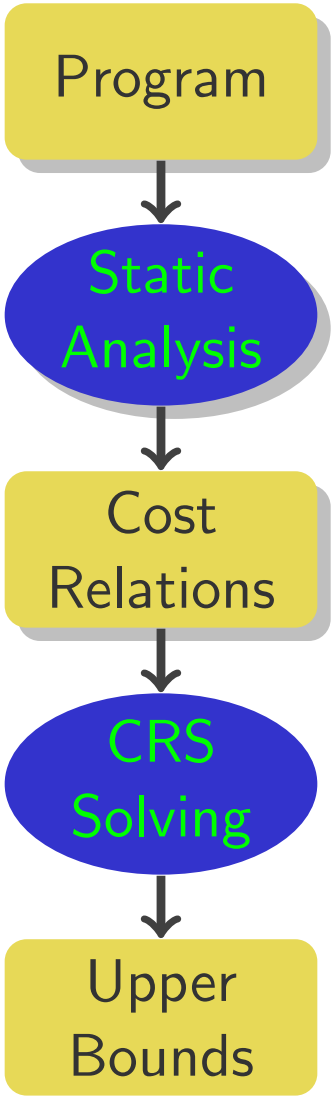
Cost Relations define the cost of executing a program fragment in terms of other fragments cost

- Control flow $\rightarrow$ **Calls**
- Guards $\rightarrow$ **Constraints**
- Iteration $\rightarrow$ **Recursion**

```
//@requires m >= 0
void main(int m) {
  while( m > 0 ) {
    rpop(m)   ;
    // push(*);
    top = new
     Node(*,top);
    //@acquire(1)
    m = m-1      ;
}}}
```

$$
\begin{aligned}
main(t,m) &= 0 && \{m = 0, t \geq 0\} \\
main(t,m) &= rpop(t,m) + 1 + main(t_2, m_1) && \left\{ \begin{array}{l} m \geq 1, t \geq 0, m_1 = m - 1 \\ t_2 = t' + 1, t \geq t' \geq t - 1 \end{array} \right\} \\
rpop(t,m) &= 0 && \{m \geq 1\} \\
rpop(t,m) &= m && \{m \geq 1, t \geq 1\}
\end{aligned}
$$

Classical Approach to Cost Analysis: working example




```
//@requires m >= 1
void rpop(int m){
  if (top != null && *){
    //pop
    top=top.next ;
    //@acquire(m)
}}
```

```
//@requires m >= 0
void main(int m) {
  while( m > 0 ) {
    rpop(m)  ;
    // push(*);
    top = new
     Node(*,top);
    //@acquire(1)
    m = m-1      ;
}}}
```

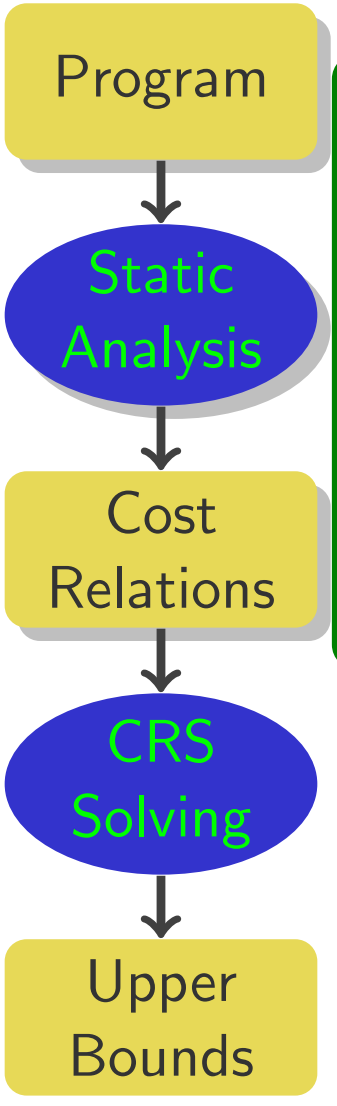
$$main(t,m) = 0 \quad \{m = 0, t \geq 0\}$$

$$main(t,m) = \begin{matrix} rpop(t,m) + 1 + \\ main(t_2, m_1) \end{matrix} \quad \left\{ \begin{matrix} m \geq 1, t \geq 0, m_1 = m - 1 \\ t_2 = t' + 1, t \geq t' \geq t - 1 \end{matrix} \right\}$$

$$rpop(t,m) = 0 \quad \{m \geq 1\}$$

$$rpop(t,m) = m \quad \{m \geq 1, t \geq 1\}$$

Classical Approach to Cost Analysis: working example



Size analysis

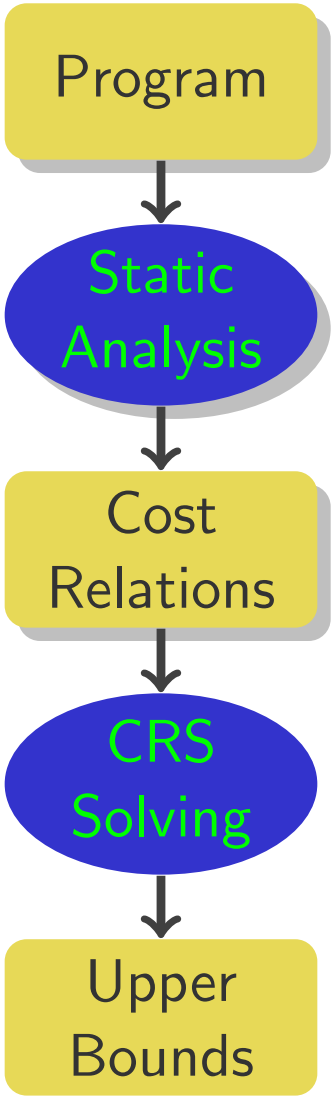
Global size analysis infers how data sizes change along execution.

Data	$\rightarrow$	**Sizes**
States	$\rightarrow$	**SSA form**
Instructions	$\rightarrow$	**Constraints**
Calls	$\rightarrow$	**Postconditions**

```
//@requires m >= 0
void main(int m) {
  while( m > 0 ) {
    rpop(m)   ;
    // push(*);
    top = new
     Node(*,top);
    //@acquire(1)
    m = m-1       ;
}}}
```

$$
\begin{aligned}
main(t,m) &= 0 && \{m = 0, t \geq 0\}\\
main(t,m) &= rpop(t,m) + 1 + main(t_2, m_1) && \left\{\begin{array}{l} m \geq 1, t \geq 0, m_1 = m - 1 \\ t_2 = t' + 1, t \geq t' > t - 1 \end{array}\right\}\\
rpop(t,m) &= 0 && \{m \geq 1\}\\
rpop(t,m) &= m && \{m \geq 1, t \geq 1\}
\end{aligned}
$$

Classical Approach to Cost Analysis: working example




```
//@requires m >= 1
void rpop(int m){
  if (top != null && *){
    //pop
    top=top.next ;
    //@acquire(m)
}}
```

```
//@requires m >= 0
void main(int m) {
  while( m > 0 ) {
    rpop(m)  ;
    // push(*);
    top = new
     Node(*,top);
    //@acquire(1)
    m = m-1      ;
}}}
```

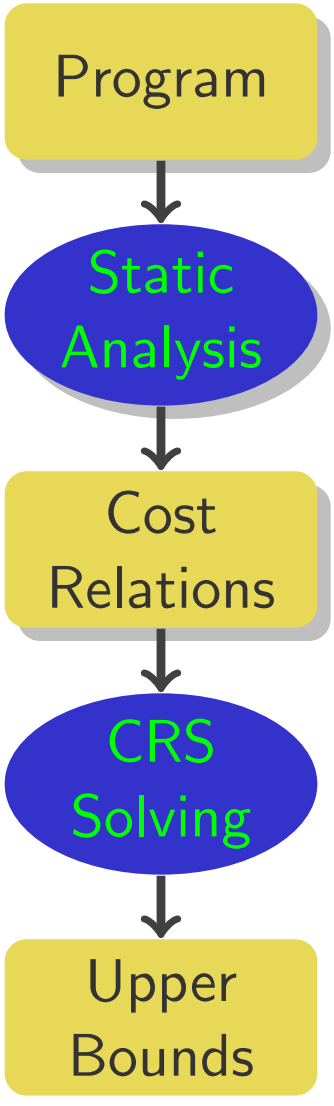
$$main(t,m) = 0 \quad \{m = 0, t \geq 0\}$$

$$main(t,m) = \begin{matrix} rpop(t,m) + 1 + \\ main(t_2, m_1) \end{matrix} \quad \left\{ \begin{matrix} m \geq 1, t \geq 0, m_1 = m - 1 \\ t_2 = t' + 1, t \geq t' \geq t - 1 \end{matrix} \right\}$$

$$rpop(t,m) = 0 \quad \{m \geq 1\}$$

$$rpop(t,m) = m \quad \{m \geq 1, t \geq 1\}$$

Classical Approach to Cost Analysis: working example




```
//@requires m >= 1
void rpop(int m){
  if (top != null && *){
    //pop
    top=top.next ;
    //@acquire(m)
}}
```

```
//@requires m >= 0
void main(int m) {
  while( m > 0 ) {
    rpop(m)  ;
    // push(*);
    top = new
     Node(*,top);
    //@acquire(1)
    m = m-1      ;
}}}
```

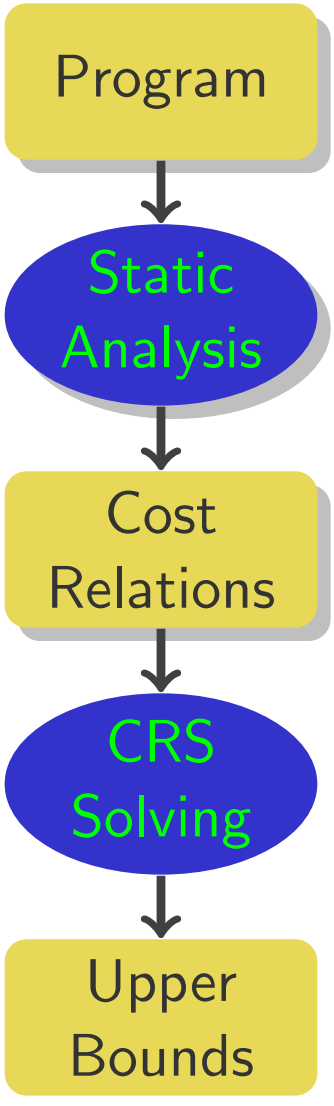
$$main(t,m) = 0 \quad \{m = 0, t \geq 0\}$$

$$main(t,m) = \begin{matrix} rpop(t,m) + 1 + \\ main(t_2, m_1) \end{matrix} \quad \left\{ \begin{matrix} m \geq 1, t \geq 0, m_1 = m - 1 \\ t_2 = t' + 1, t \geq t' \geq t - 1 \end{matrix} \right\}$$

$$rpop(t,m) = 0 \quad \{m \geq 1\}$$

$$rpop(t,m) = m \quad \{m \geq 1, t \geq 1\}$$

Classical Approach to Cost Analysis: working example




```
//@requires m >= 1
void rpop(int m){

//@requires m >= 0
void main(int m) {
```

Soundness & Precission of Analysis

Sound: For every program execution there is a CR evaluation with same cost

Precise: Every CR evaluation matches at least one program execution

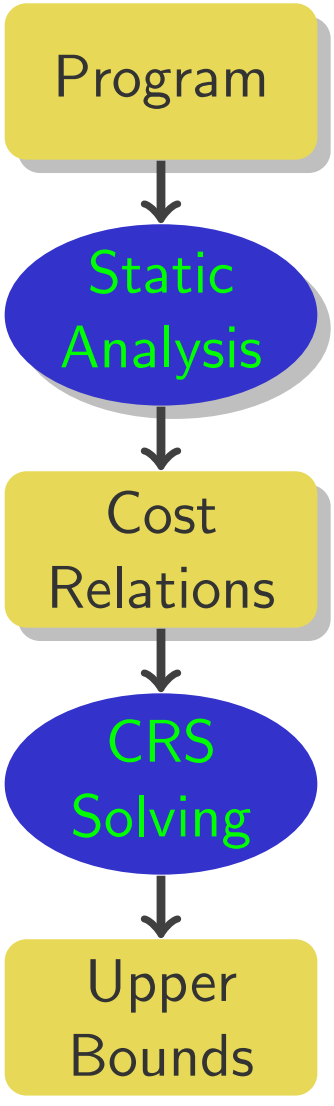
$$main(t,m) = 0 \quad \{m = 0, t \geq 0\}$$

$$main(t,m) = \begin{matrix} rpop(t,m) + 1 + \\ main(t_2, m_1) \end{matrix} \quad \left\{ \begin{matrix} m \geq 1, t \geq 0, m_1 = m - 1 \\ t_2 = t' + 1, t \geq t' \geq t - 1 \end{matrix} \right\}$$

$$rpop(t,m) = 0 \quad \{m \geq 1\}$$

$$rpop(t,m) = m \quad \{m \geq 1, t \geq 1\}$$

Classical Approach to Cost Analysis: working example




```
//@requires m >= 1
void rpop(int m){
  if (top != null && *){
    //pop
    top=top.next ;
    //@acquire(m)
}}
```

```
//@requires m >= 0
void main(int m) {
  while( m > 0 ) {
    rpop(m)  ;
    // push(*);
    top = new
     Node(*,top);
    //@acquire(1)
    m = m-1      ;
}}}
```

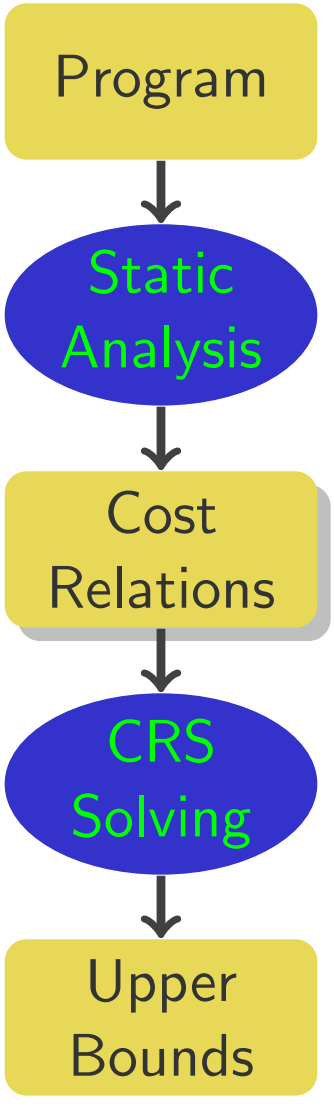
$$main(t,m) = 0 \quad \{m = 0, t \geq 0\}$$

$$main(t,m) = \begin{matrix} rpop(t,m) + 1 + \\ main(t_2, m_1) \end{matrix} \quad \left\{ \begin{matrix} m \geq 1, t \geq 0, m_1 = m - 1 \\ t_2 = t' + 1, t \geq t' \geq t - 1 \end{matrix} \right\}$$

$$rpop(t,m) = 0 \quad \{m \geq 1\}$$

$$rpop(t,m) = m \quad \{m \geq 1, t \geq 1\}$$

Classical Approach to Cost Analysis: working example




```
//@requires m >= 1
void rpop(int m){
  if (top != null && *){
    //pop
    top=top.next ;
    //@acquire(m)
}}
```

```
//@requires m >= 0
void main(int m) {
  while( m > 0 ) {
    rpop(m)  ;
    // push(*);
    top = new
     Node(*,top);
    //@acquire(1)
    m = m-1     ;
}}}
```

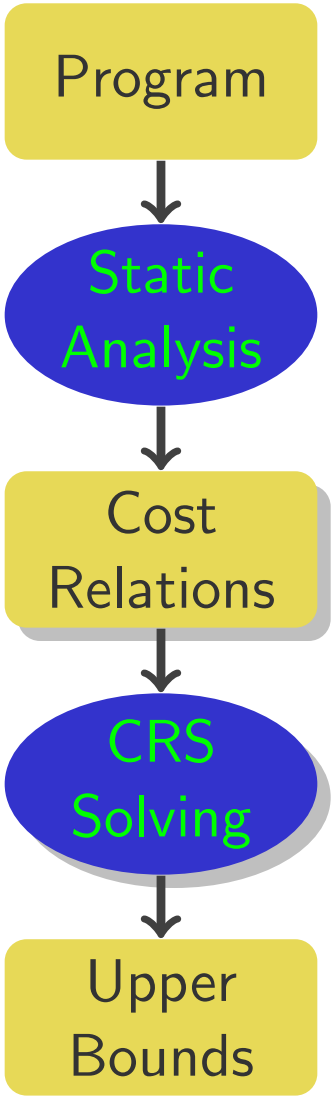
$$main(t,m) = 0 \quad \{m = 0, t \geq 0\}$$

$$main(t,m) = \begin{matrix} rpop(t,m) + 1 + \\ main(t_2, m_1) \end{matrix} \quad \left\{ \begin{matrix} m \geq 1, t \geq 0, m_1 = m - 1 \\ t_2 = t' + 1, t \geq t' \geq t - 1 \end{matrix} \right\}$$

$$rpop(t,m) = 0 \quad \{m \geq 1\}$$

$$rpop(t,m) = m \quad \{m \geq 1, t \geq 1\}$$

Classical Approach to Cost Analysis: working example




```
//@requires m >= 1
void rpop(int m){
  if (top != null && *){
    //pop
    top=top.next ;
    //@acquire(m)
}}
```

```
//@requires m >= 0
void main(int m) {
  while( m > 0 ) {
    rpop(m)  ;
    // push(*);
    top = new
     Node(*,top);
    //@acquire(1)
    m = m-1     ;
}}}
```

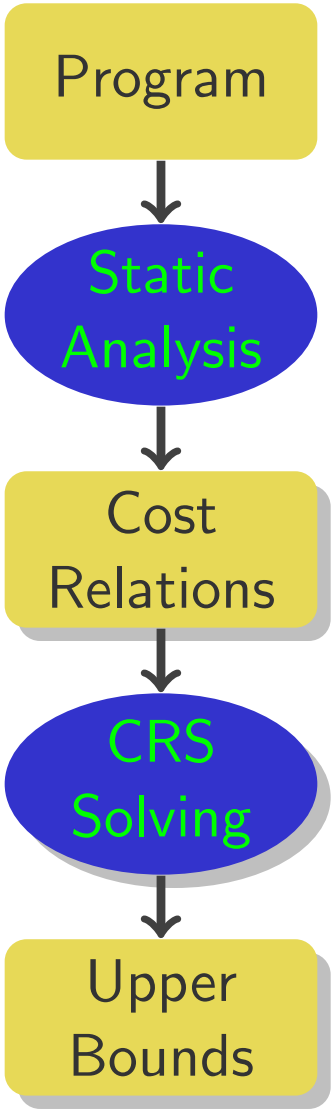
$$main(t,m) = 0 \quad \{m = 0, t \geq 0\}$$

$$main(t,m) = \begin{matrix} rpop(t,m) + 1 + \\ main(t_2, m_1) \end{matrix} \quad \left\{ \begin{matrix} m \geq 1, t \geq 0, m_1 = m - 1 \\ t_2 = t' + 1, t \geq t' \geq t - 1 \end{matrix} \right\}$$

$$rpop(t,m) = 0 \quad \{m \geq 1\}$$

$$rpop(t,m) = m \quad \{m \geq 1, t \geq 1\}$$

Classical Approach to Cost Analysis: working example




```
//@requires m >= 1
void rpop(int m){
  if (top != null && *){
    //pop
    top=top.next ;
    //@acquire(m)
}}
```

```
//@requires m >= 0
void main(int m) {
  while( m > 0 ) {
    rpop(m)  ;
    // push(*);
    top = new
     Node(*,top);
    //@acquire(1)
    m = m-1     ;
}}}
```

$$main^u(t,m) = \frac{m^2+m}{2} + m \qquad rpop^u(t,m) = m$$

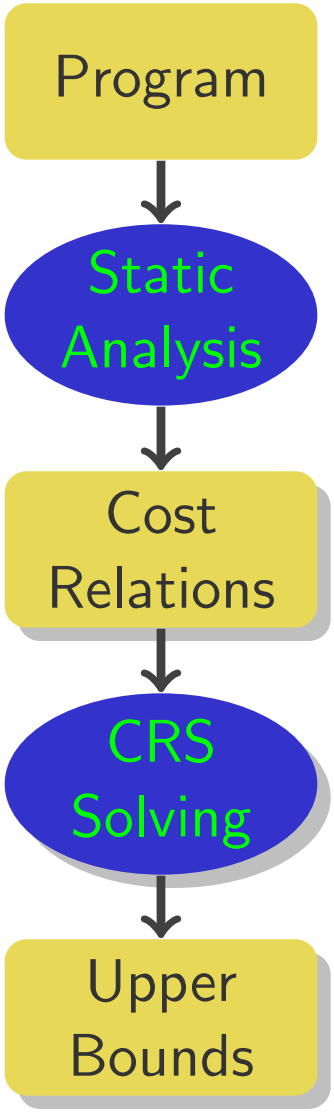
$$main(t,m) = 0 \qquad \{m = 0, t \geq 0\}$$

$$main(t,m) = \begin{matrix} rpop(t,m) + 1 + \\ main(t_2, m_1) \end{matrix} \quad \left\{ \begin{matrix} m \geq 1, t \geq 0, m_1 = m-1 \\ t_2 = t' + 1, t \geq t' \geq t-1 \end{matrix} \right\}$$

$$rpop(t,m) = 0 \qquad \{m \geq 1\}$$

$$rpop(t,m) = m \qquad \{m \geq 1, t \geq 1\}$$

Classical Approach to Cost Analysis: working example




```
//@requires m >= 1
void rpop(int m){
```

```
//@requires m >= 0
void main(int m) {
```

Soundness & Precission of UBs wrt CRs

Sound: **If** $c \in main(t, m)$ **then** $c \leq main^u(t, m)$
Precise: **If** $main^u(t, m) = c$ **then** $c \in main(t, m)$

$$main^u(t, m) = \frac{m^2 + m}{2} + m \quad rpop^u(t, m) = m$$

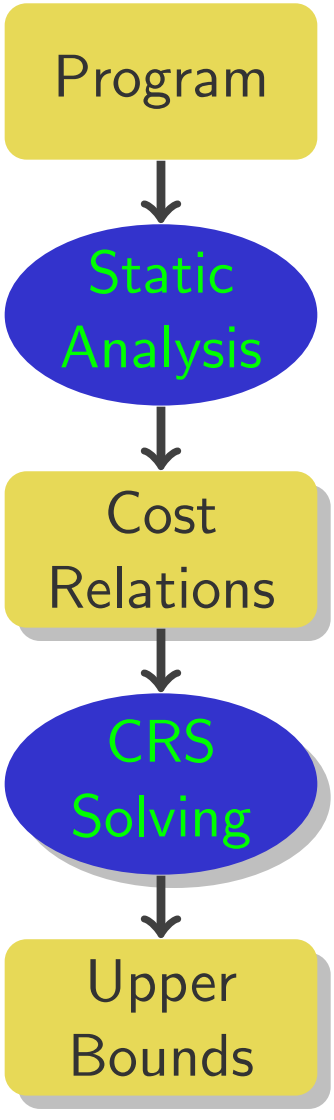
$$main(t, m) = 0 \quad \{m = 0, t \geq 0\}$$

$$main(t, m) = \begin{matrix} rpop(t, m) + 1 + \\ main(t_2, m_1) \end{matrix} \quad \left\{ \begin{matrix} m \geq 1, t \geq 0, m_1 = m - 1 \\ t_2 = t' + 1, t \geq t' \geq t - 1 \end{matrix} \right\}$$

$$rpop(t, m) = 0 \quad \{m \geq 1\}$$

$$rpop(t, m) = m \quad \{m \geq 1, t \geq 1\}$$

Classical Approach to Cost Analysis: working example




```
//@requires m >= 1
void rpop(int m){
  if (top != null && *){
    //pop
    top=top.next ;
    //@acquire(m)
}}
```

```
//@requires m >= 0
void main(int m) {
  while( m > 0 ) {
    rpop(m)  ;
    // push(*);
    top = new
     Node(*,top);
    //@acquire(1)
    m = m-1     ;
}}}
```

$$main^u(t,m) = \frac{m^2+m}{2} + m \qquad rpop^u(t,m) = m$$

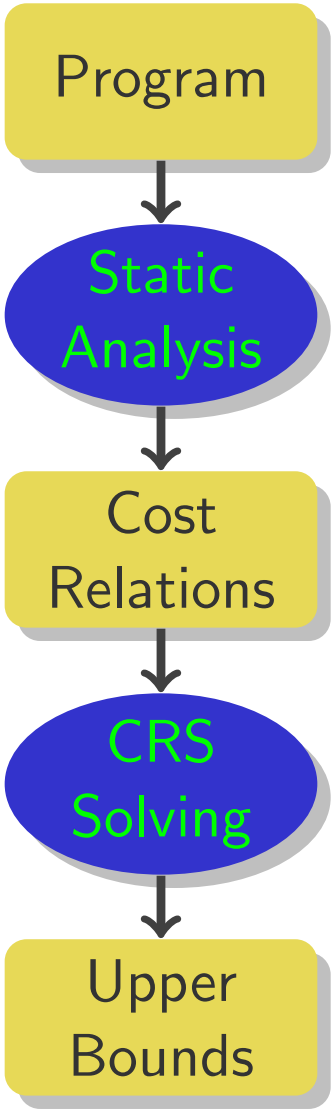
$$main(t,m) = 0 \qquad \{m = 0, t \geq 0\}$$

$$main(t,m) = \begin{matrix} rpop(t,m) + 1 + \\ main(t_2, m_1) \end{matrix} \quad \left\{ \begin{matrix} m \geq 1, t \geq 0, m_1 = m-1 \\ t_2 = t' + 1, t \geq t' \geq t-1 \end{matrix} \right\}$$

$$rpop(t,m) = 0 \qquad \{m \geq 1\}$$

$$rpop(t,m) = m \qquad \{m \geq 1, t \geq 1\}$$

Classical Approach to Cost Analysis: working example




```
//@requires m >= 1
void rpop(int m){
  if (top != null && *){
    //pop
    top=top.next ;
    //@acquire(m)
}}
```

```
//@requires m >= 0
void main(int m) {
  while( m > 0 ) {
    rpop(m)  ;
    // push(*);
    top = new
     Node(*,top);
    //@acquire(1)
    m = m-1     ;
}}}
```

$$main^u(t,m) = \frac{m^2+m}{2} + m \qquad rpop^u(t,m) = m$$

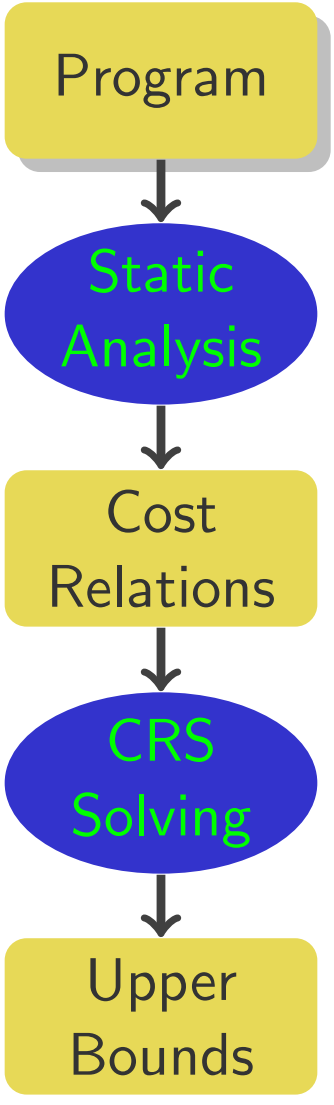
$$main(t,m) = 0 \qquad \{m = 0, t \geq 0\}$$

$$main(t,m) = \begin{matrix} rpop(t,m) + 1 + \\ main(t_2, m_1) \end{matrix} \quad \left\{ \begin{matrix} m \geq 1, t \geq 0, m_1 = m-1 \\ t_2 = t' + 1, t \geq t' \geq t-1 \end{matrix} \right\}$$

$$rpop(t,m) = 0 \qquad \{m \geq 1\}$$

$$rpop(t,m) = m \qquad \{m \geq 1, t \geq 1\}$$

Classical Approach to Cost Analysis: working example




```
//@requires m >= 1
void rpop(int m){
  if (top != null && *){
    //pop
    top=top.next ;
    //@acquire(m)
}}
```

```
//@requires m >= 0
void main(int m) {
  while( m > 0 ) {
    rpop(m)  ;
    // push(*);
    top = new
     Node(*,top);
    //@acquire(1)
    m = m-1     ;
}}}
```

$$main^u(t,m) = \frac{m^2+m}{2} + m \qquad rpop^u(t,m) = m$$

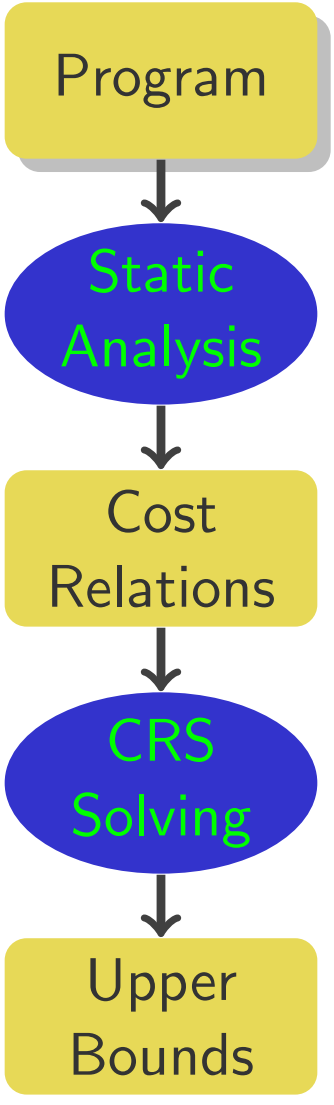
$$main(t,m) = 0 \qquad \{m = 0, t \geq 0\}$$

$$main(t,m) = \begin{matrix} rpop(t,m) + 1 + \\ main(t_2, m_1) \end{matrix} \quad \left\{ \begin{matrix} m \geq 1, t \geq 0, m_1 = m-1 \\ t_2 = t' + 1, t \geq t' \geq t-1 \end{matrix} \right\}$$

$$rpop(t,m) = 0 \qquad \{m \geq 1\}$$

$$rpop(t,m) = m \qquad \{m \geq 1, t \geq 1\}$$

Classical Approach to Cost Analysis: working example




```
//@requires m >= 1
void rpop(int m){
  if(top != null && *){
while//pop
    top=top.next ;
    //@acquire(m)
}}
```

```
//@requires m >= 0
void main(int m) {
  while( m > 0 ) {
    rpop(m)  ;
    // push(*);
    top = new
     Node(*,top);
    //@acquire(1)
    m = m-1     ;
}}}
```

$$main^u(t,m) = \frac{m^2+m}{2} + m \qquad rpop^u(t,m) = m$$

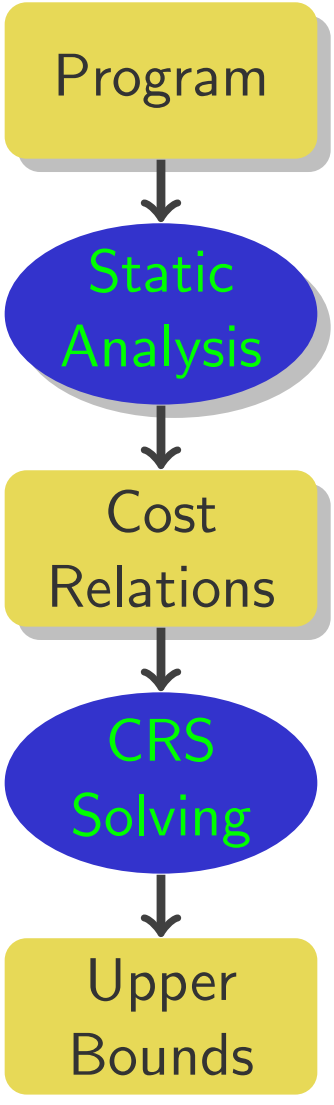
$$main(t,m) = 0 \qquad \{m = 0, t \geq 0\}$$

$$main(t,m) = \begin{matrix} rpop(t,m) + 1 + \\ main(t_2, m_1) \end{matrix} \quad \left\{ \begin{matrix} m \geq 1, t \geq 0, m_1 = m-1 \\ t_2 = t' + 1, t \geq t' \geq t-1 \end{matrix} \right\}$$

$$rpop(t,m) = 0 \qquad \{m \geq 1\}$$

$$rpop(t,m) = m \qquad \{m \geq 1, t \geq 1\}$$

Classical Approach to Cost Analysis: working example




```
//@requires m >= 1
void rpop(int m){
  if(top != null && *){
while//pop
    top=top.next ;
    //@acquire(m)
}}
```

```
//@requires m >= 0
void main(int m) {
  while( m > 0 ) {
    rpop(m)  ;
    // push(*);
    top = new
     Node(*,top);
    //@acquire(1)
    m = m-1      ;
}}}
```

$$main^u(t,m) = \frac{m^2+m}{2} + m \qquad rpop^u(t,m) = m$$

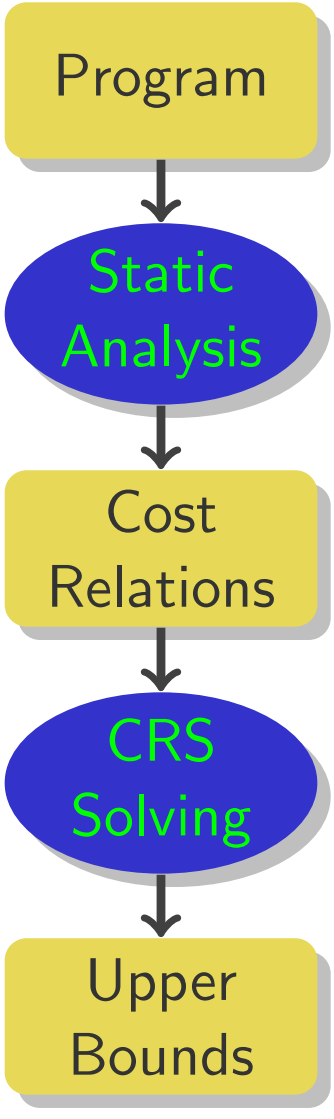
$$main(t,m) = 0 \quad \{m = 0, t \geq 0\}$$

$$main(t,m) = \begin{matrix} rpop(t,m) + 1 + \\ main(t_2, m_1) \end{matrix} \quad \left\{ \begin{matrix} m \geq 1, t \geq 0, m_1 = m-1 \\ t_2 = t'+1, t \geq t' \geq 0 \end{matrix} \right\} + 1$$

$$rpop(t,m) = 0 \quad \{m \geq 1\}$$

$$rpop(t,m) = m + rpop(t_1, m) \quad \{m \geq 1, t \geq 1, t_1 = t-1\}$$

Classical Approach to Cost Analysis: working example




```
//@requires m >= 1
void rpop(int m){
  if(top != null && *){
while//pop
    top=top.next ;
    //@acquire(m)
}}
```

```
//@requires m >= 0
void main(int m) {
  while( m > 0 ) {
    rpop(m)  ;
    // push(*);
    top = new
     Node(*,top);
    //@acquire(1)
    m = m-1     ;
}}}
```

$$main^u(t,m) = \frac{m^3 + 3t(m^2+m) + 5m}{6} \quad rpop^u(t,m) = m * t$$

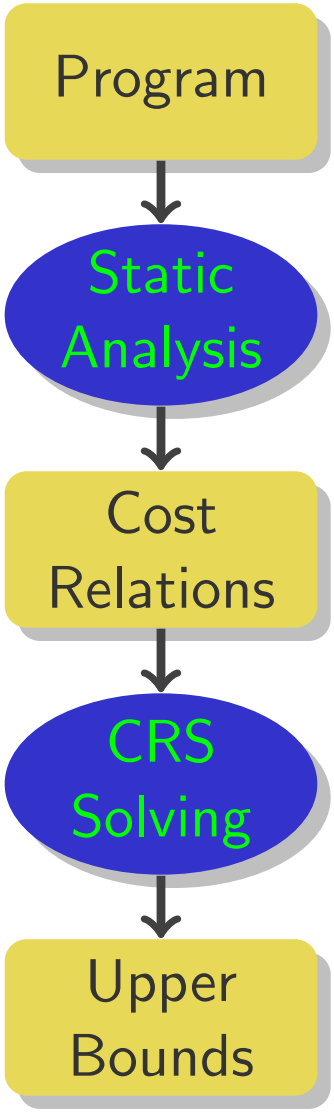
$$main(t,m) = 0 \quad \{m = 0, t \geq 0\}$$

$$main(t,m) = \begin{matrix} rpop(t,m) + 1 + \\ main(t_2, m_1) \end{matrix} \quad \left\{ \begin{matrix} m \geq 1, t \geq 0, m_1 = m - 1 \\ t_2 = t' + 1, t \geq t' \geq 0 \end{matrix} \right\}$$

$$rpop(t,m) = 0 \quad \{m \geq 1\}$$

$$rpop(t,m) = m + rpop(t_1, m) \quad \{m \geq 1, t \geq 1, t_1 = t - 1\}$$

Classical Approach to Cost Analysis: working example




```
//@requires m >= 1
void rpop(int m){
  if(top != null && *){
while//pop
    top=top.next ;
    //@acquire(m)
}}
```

```
//@requires m >= 0
void main(int m) {
  while( m > 0 ) {
    rpop(m)  ;
    // push(*);
    top = new
     Node(*,top);
    //@acquire(1)
    m = m-1      ;
}}}
```

What fails?

$$main^u(t,m) = \frac{m^3 + 3t(m^2+m) + 5m}{6} \qquad rpop^u(t,m) = m * t$$

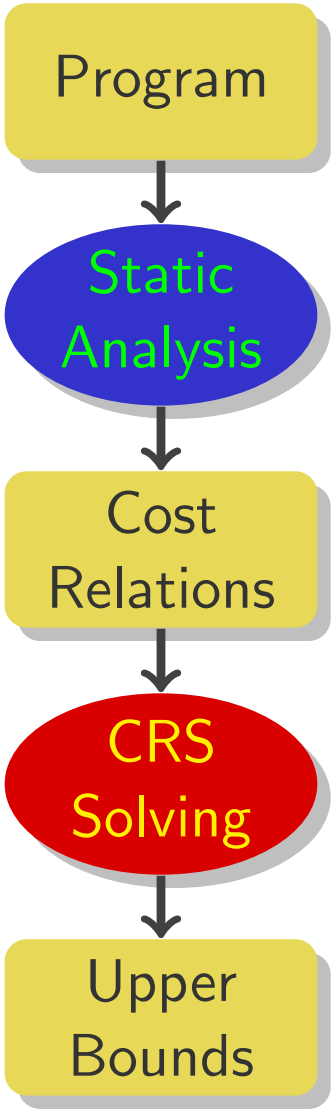
$$main(t,m) = 0 \quad \{m = 0, t \geq 0\}$$

$$main(t,m) = rpop(t,m) + 1 + main(t_2,m_1) \quad \left\{ \begin{array}{l} m \geq 1, t \geq 0, m_1 = m-1 \\ t_2 = t' + 1, t \geq t' \geq 0 \end{array} \right\}$$

$$rpop(t,m) = 0 \quad \{m \geq 1\}$$

$$rpop(t,m) = m + rpop(t_1,m) \quad \{m \geq 1, t \geq 1, t_1 = t-1\}$$

Classical Approach to Cost Analysis: working example



?

```
//@requires m >= 1
void rpop(int m){
  if(top != null && *){
while//pop
    top=top.next ;
    //@acquire(m)
}}
```

```
//@requires m >= 0
void main(int m) {
  while( m > 0 ) {
    rpop(m)  ;
    // push(*);
    top = new
     Node(*,top);
    //@acquire(1)
    m = m-1      ;
}}}
```

What fails?

$$main^u(t,m) = \frac{m^3 + 3t(m^2+m) + 5m}{6} \qquad rpop^u(t,m) = m * t$$

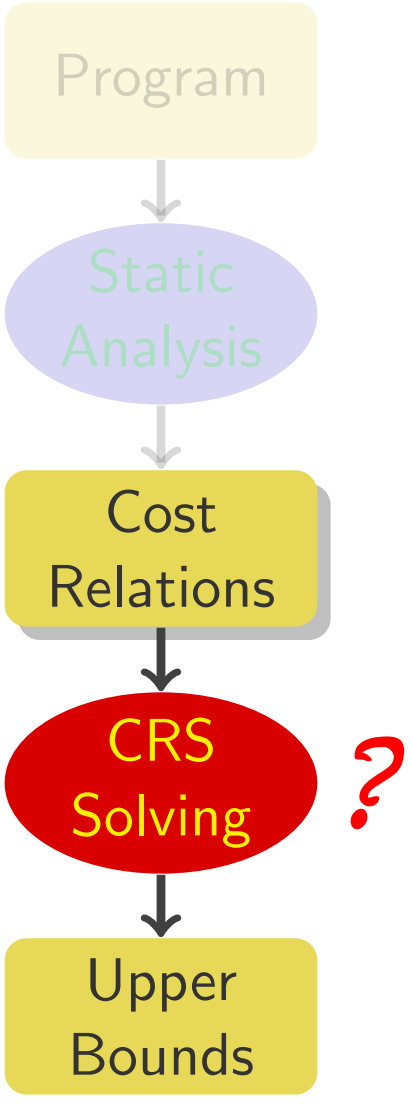
$$main(t,m) = 0 \quad \{m = 0, t \geq 0\}$$

$$main(t,m) = rpop(t,m) + 1 + main(t_2,m_1) \quad \left\{ \begin{array}{l} m \geq 1, t \geq 0, m_1 = m-1 \\ t_2 = t' + 1, t \geq t' \geq 0 \end{array} \right\}$$

$$rpop(t,m) = 0 \quad \{m \geq 1\}$$

$$rpop(t,m) = m + rpop(t_1,m) \quad \{m \geq 1, t \geq 1, t_1 = t-1\}$$

Classical Approach to Cost Analysis: working example



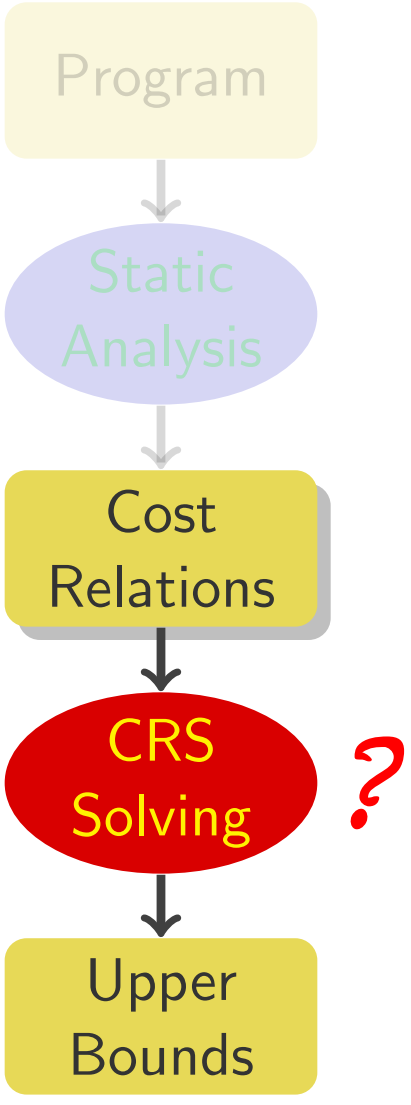
We consider a worst-case evaluation of $main(5, 3)$

$$main^u(t, m) = \frac{m^3 + 3t(m^2 + m) + 5m}{6} \qquad rpop^u(t, m) = m * t$$

$$\begin{array}{lll} main(t, m) = & 0 & \{m = 0, t \geq 0\} \\ main(t, m) = & rpop(t, m) + 1 + main(t_2, m_1) & \left\{ \begin{array}{l} m \geq 1, t \geq 0, m_1 = m - 1 \\ t_2 = t' + 1, t \geq t' \geq 0 \end{array} \right\} \end{array}$$

$$\begin{array}{lll} rpop(t, m) = & 0 & \{m \geq 1\} \\ rpop(t, m) = & m + rpop(t_1, m) & \{m \geq 1, t \geq 1, t_1 = t - 1\} \end{array}$$

Classical Approach to Cost Analysis: working example



?

We consider a worst-case evaluation of $main(5,3)$

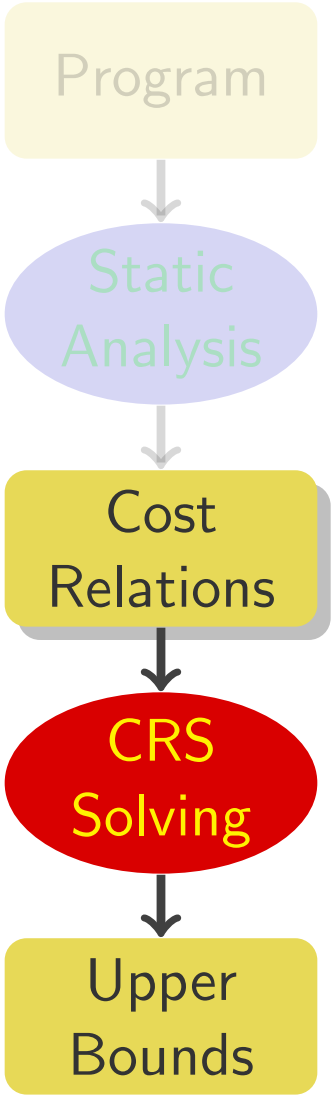
$main(5,3)$

$$main^u(t,m) = \frac{m^3 + 3t(m^2+m) + 5m}{6} \qquad rpop^u(t,m) = m * t$$

$$\begin{array}{lll} main(t,m) = 0 & & \{m=0, t\geq 0\} \\ main(t,m) = rpop(t,m)+1+ & & \left\{\begin{array}{l} m\geq 1, t\geq 0, m_1 = m-1 \\ t_2 = t'+1, t\geq t'\geq 0 \end{array}\right\} \\ \quad\quad main(t_2,m_1) & & \end{array}$$

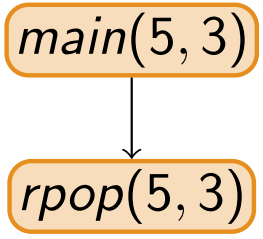
$$\begin{array}{lll} rpop(t,m) = 0 & & \{m\geq 1\} \\ rpop(t,m) = m + rpop(t_1,m) & & \{m\geq 1, t\geq 1, t_1 = t-1\} \end{array}$$

Classical Approach to Cost Analysis: working example



?

We consider a worst-case evaluation of $main(5,3)$

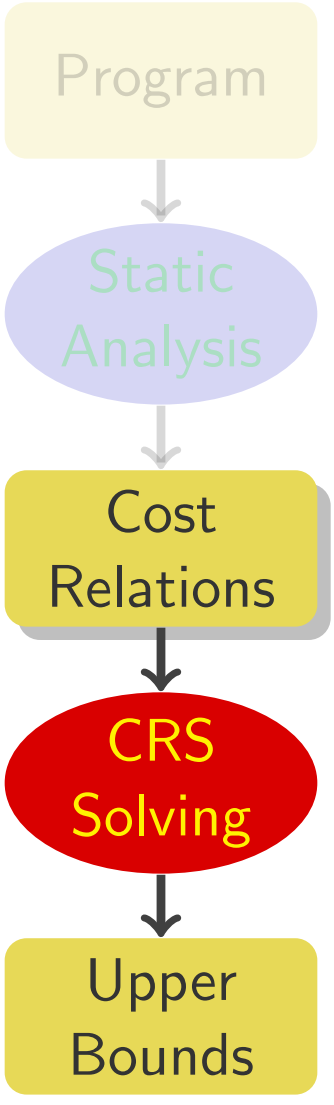


$$main^u(t,m) = \frac{m^3 + 3t(m^2+m) + 5m}{6} \qquad rpop^u(t,m) = m * t$$

$$\begin{aligned}
main(t,m) &= 0 && \{m = 0, t \geq 0\}\\
main(t,m) &= rpop(t,m) + 1 + main(t_2,m_1) && \left\{\begin{array}{l} m \geq 1, t \geq 0, m_1 = m-1 \\ t_2 = t'+1, t \geq t' \geq 0 \end{array}\right\}
\end{aligned}$$

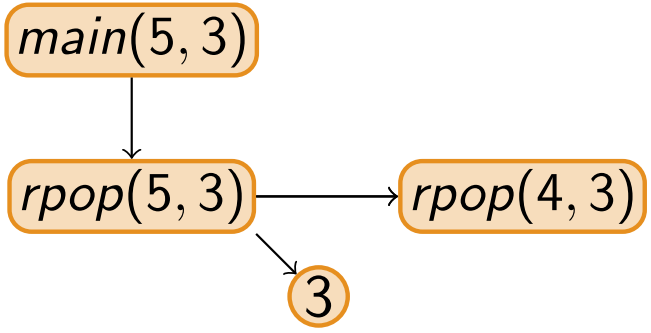
$$\begin{aligned}
rpop(t,m) &= 0 && \{m \geq 1\}\\
rpop(t,m) &= m + rpop(t_1,m) && \{m \geq 1, t \geq 1, t_1 = t-1\}
\end{aligned}$$

Classical Approach to Cost Analysis: working example



?

We consider a worst-case evaluation of $main(5,3)$

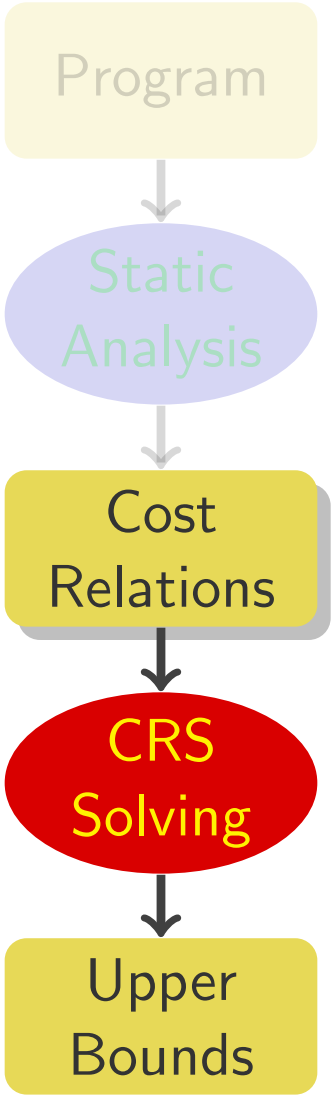


$$main^u(t,m) = \frac{m^3 + 3t(m^2+m) + 5m}{6} \qquad rpop^u(t,m) = m * t$$

$$\begin{aligned} main(t,m) &= 0 && \{m = 0, t \geq 0\} \\ main(t,m) &= rpop(t,m) + 1 + main(t_2, m_1) && \left\{ \begin{array}{l} m \geq 1, t \geq 0, m_1 = m - 1 \\ t_2 = t' + 1, t \geq t' \geq 0 \end{array} \right\} \end{aligned}$$

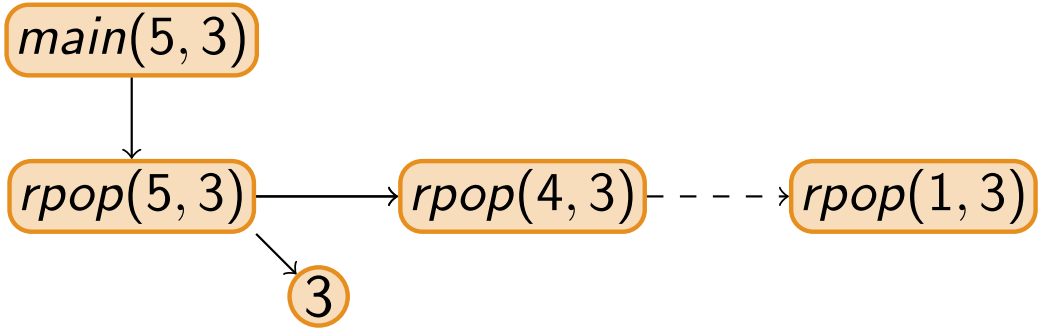
$$\begin{aligned} rpop(t,m) &= 0 && \{m \geq 1\} \\ rpop(t,m) &= m + rpop(t_1, m) && \{m \geq 1, t \geq 1, t_1 = t - 1\} \end{aligned}$$

Classical Approach to Cost Analysis: working example



?

We consider a worst-case evaluation of $main(5,3)$

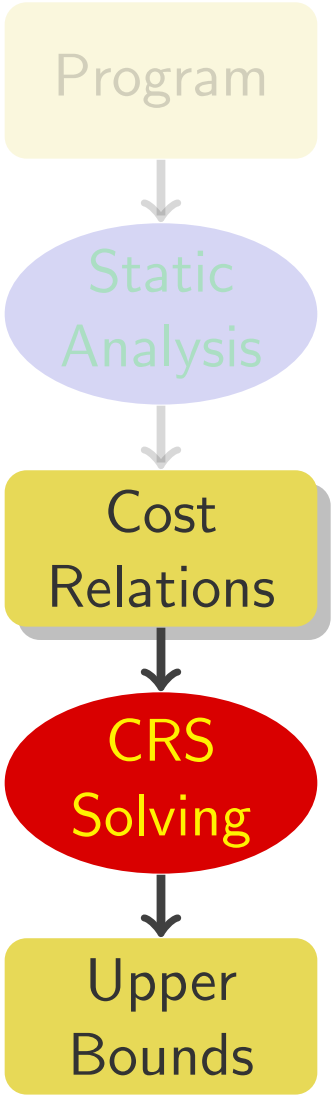


$$main^u(t,m) = \frac{m^3 + 3t(m^2+m) + 5m}{6} \qquad rpop^u(t,m) = m * t$$

$$\begin{aligned} main(t,m) &= 0 && \{m = 0, t \geq 0\} \\ main(t,m) &= rpop(t,m) + 1 + main(t_2, m_1) && \left\{ \begin{array}{l} m \geq 1, t \geq 0, m_1 = m - 1 \\ t_2 = t' + 1, t \geq t' \geq 0 \end{array} \right\} \end{aligned}$$

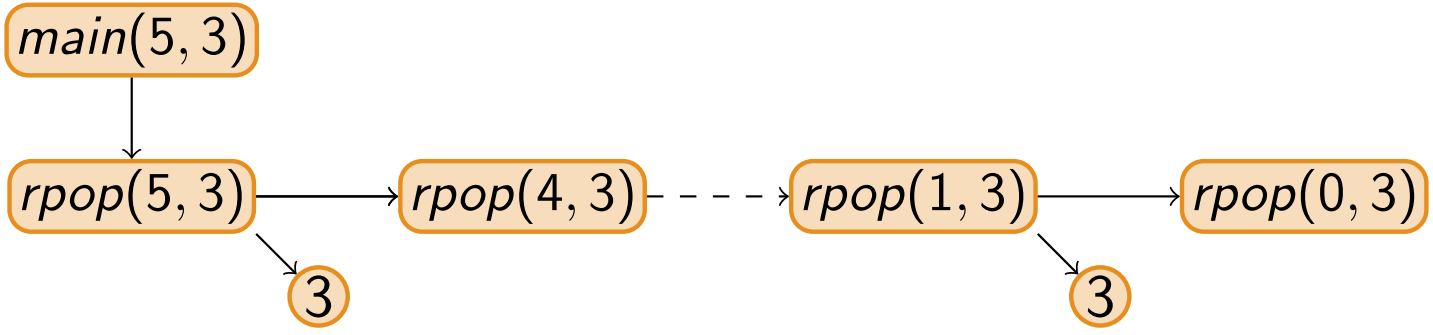
$$\begin{aligned} rpop(t,m) &= 0 && \{m \geq 1\} \\ rpop(t,m) &= m + rpop(t_1, m) && \{m \geq 1, t \geq 1, t_1 = t - 1\} \end{aligned}$$

Classical Approach to Cost Analysis: working example



?

We consider a worst-case evaluation of $main(5,3)$

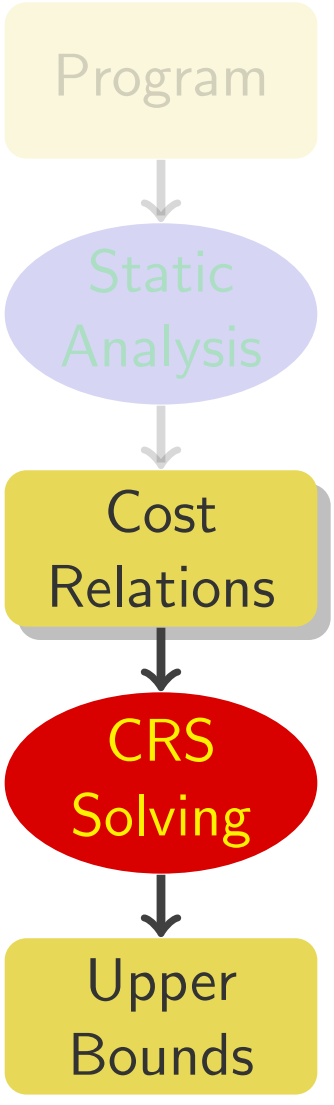


$$main^u(t,m) = \frac{m^3 + 3t(m^2+m) + 5m}{6} \qquad rpop^u(t,m) = m * t$$

$$\begin{aligned} main(t,m) &= 0 && \{m = 0, t \geq 0\} \\ main(t,m) &= rpop(t,m) + 1 + main(t_2, m_1) && \left\{ \begin{array}{l} m \geq 1, t \geq 0, m_1 = m - 1 \\ t_2 = t' + 1, t \geq t' \geq 0 \end{array} \right\} \end{aligned}$$

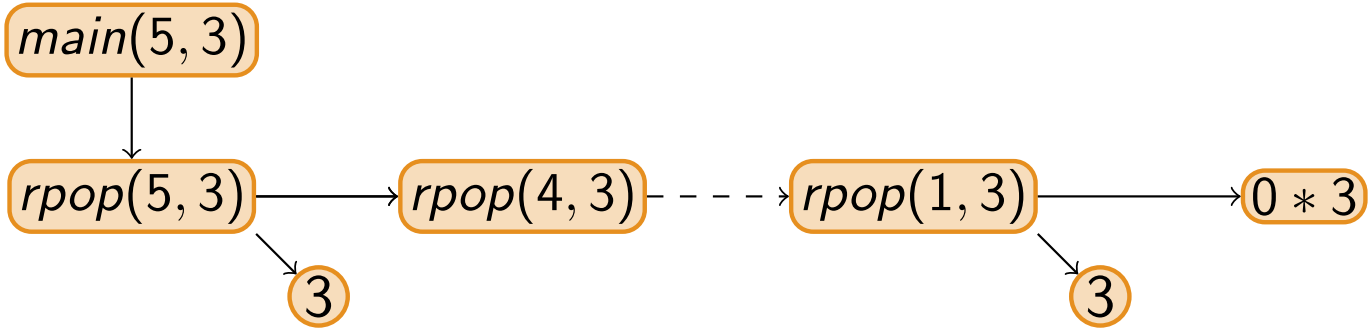
$$\begin{aligned} rpop(t,m) &= 0 && \{m \geq 1\} \\ rpop(t,m) &= m + rpop(t_1, m) && \{m \geq 1, t \geq 1, t_1 = t - 1\} \end{aligned}$$

Classical Approach to Cost Analysis: working example



?

We consider a worst-case evaluation of $main(5,3)$



$$main^u(t,m) = \frac{m^3 + 3t(m^2+m) + 5m}{6} \qquad rpop^u(t,m) = m * t$$

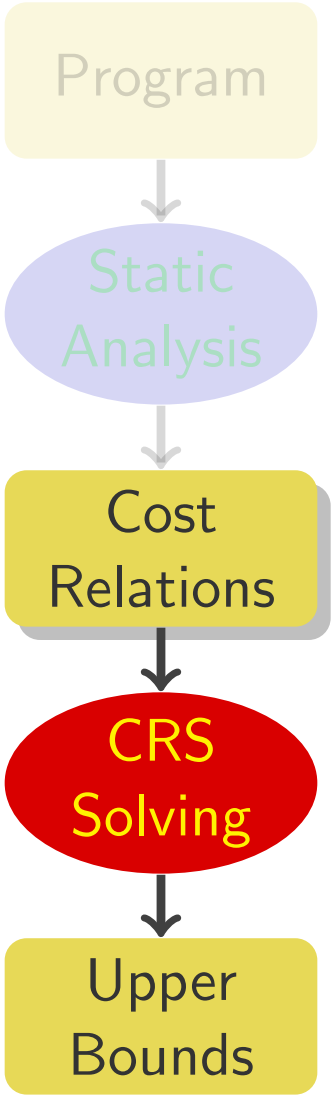
$$main(t,m) = 0 \quad \{m = 0, t \geq 0\}$$

$$main(t,m) = \begin{matrix} rpop(t,m) \\ main(t_2,m_1) \end{matrix} + 1 + \left\{ \begin{matrix} m \geq 1, t \geq 0, m_1 = m-1 \\ t_2 = t'+1, t \geq t' \geq 0 \end{matrix} \right\}$$

$$rpop(t,m) = 0 \quad \{m \geq 1\}$$

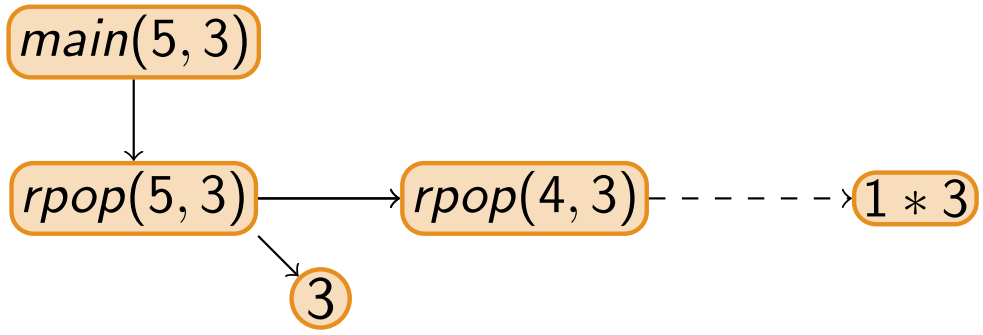
$$rpop(t,m) = m + rpop(t_1,m) \quad \{m \geq 1, t \geq 1, t_1 = t-1\}$$

Classical Approach to Cost Analysis: working example



?

We consider a worst-case evaluation of $main(5,3)$

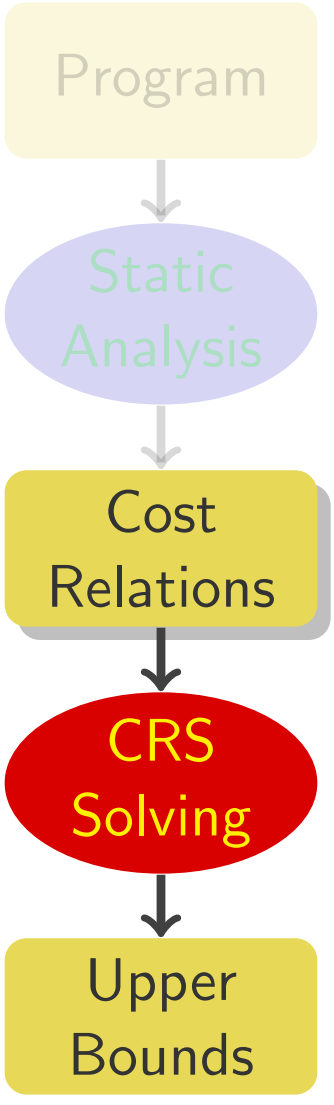


$$main^u(t,m) = \frac{m^3 + 3t(m^2+m) + 5m}{6} \qquad rpop^u(t,m) = m * t$$

$$\begin{aligned} main(t,m) &= 0 && \{m = 0, t \geq 0\} \\ main(t,m) &= rpop(t,m) + 1 + main(t_2, m_1) && \left\{ \begin{array}{l} m \geq 1, t \geq 0, m_1 = m - 1 \\ t_2 = t' + 1, t \geq t' \geq 0 \end{array} \right\} \end{aligned}$$

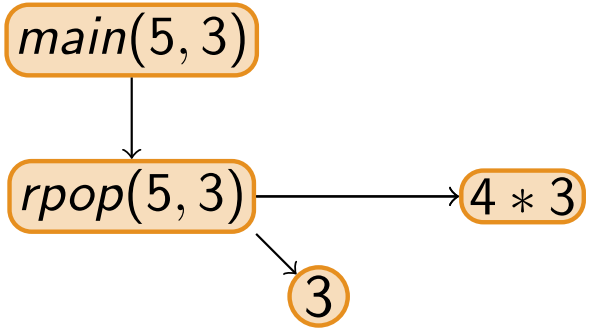
$$\begin{aligned} rpop(t,m) &= 0 && \{m \geq 1\} \\ rpop(t,m) &= m + rpop(t_1, m) && \{m \geq 1, t \geq 1, t_1 = t - 1\} \end{aligned}$$

Classical Approach to Cost Analysis: working example



?

We consider a worst-case evaluation of $main(5,3)$

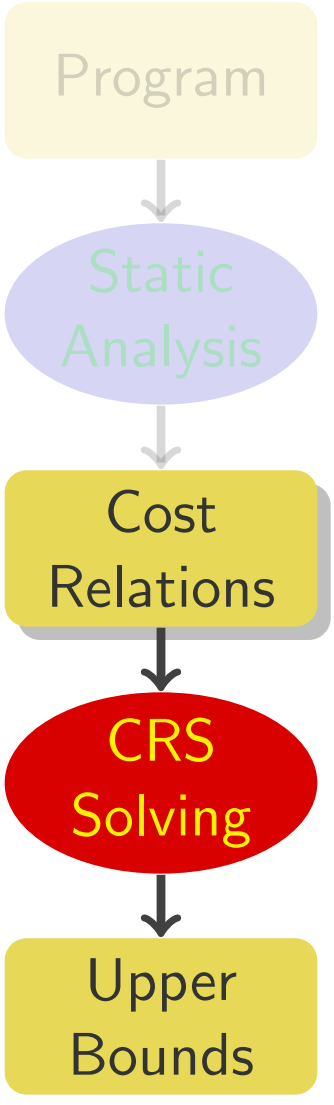


$$main^u(t,m) = \frac{m^3 + 3t(m^2+m) + 5m}{6} \qquad rpop^u(t,m) = m * t$$

$$\begin{aligned} main(t,m) &= 0 && \{m = 0, t \geq 0\} \\ main(t,m) &= rpop(t,m) + 1 + main(t_2, m_1) && \left\{ \begin{array}{l} m \geq 1, t \geq 0, m_1 = m - 1 \\ t_2 = t' + 1, t \geq t' \geq 0 \end{array} \right\} \end{aligned}$$

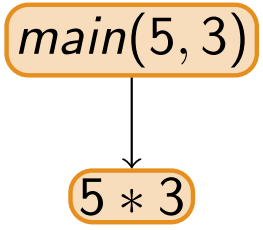
$$\begin{aligned} rpop(t,m) &= 0 && \{m \geq 1\} \\ rpop(t,m) &= m + rpop(t_1, m) && \{m \geq 1, t \geq 1, t_1 = t - 1\} \end{aligned}$$

Classical Approach to Cost Analysis: working example



?

We consider a worst-case evaluation of $main(5,3)$

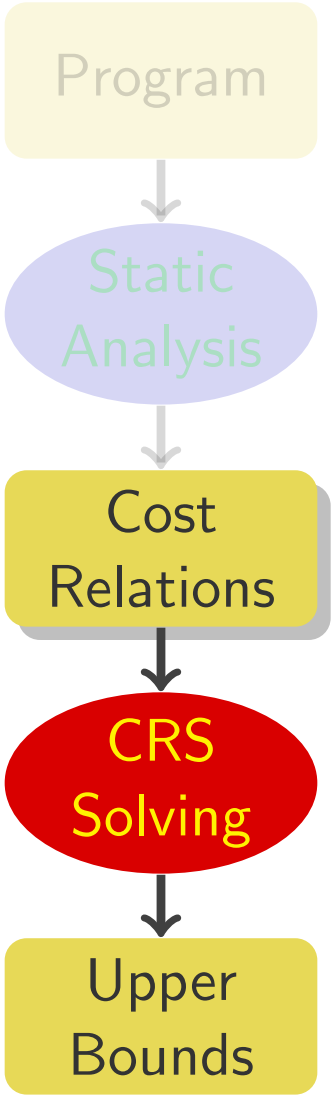


$$main^u(t,m) = \frac{m^3 + 3t(m^2+m) + 5m}{6} \qquad rpop^u(t,m) = m * t$$

$$\begin{aligned} main(t,m) &= 0 && \{m = 0, t \geq 0\} \\ main(t,m) &= rpop(t,m) + 1 + main(t_2, m_1) && \left\{ \begin{array}{l} m \geq 1, t \geq 0, m_1 = m - 1 \\ t_2 = t' + 1, t \geq t' \geq 0 \end{array} \right\} \end{aligned}$$

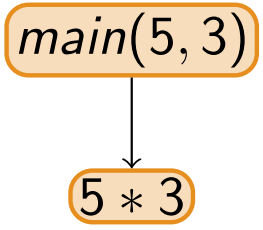
$$\begin{aligned} rpop(t,m) &= 0 && \{m \geq 1\} \\ rpop(t,m) &= m + rpop(t_1, m) && \{m \geq 1, t \geq 1, t_1 = t - 1\} \end{aligned}$$

Classical Approach to Cost Analysis: working example



?

We consider a worst-case evaluation of $main(5,3)$

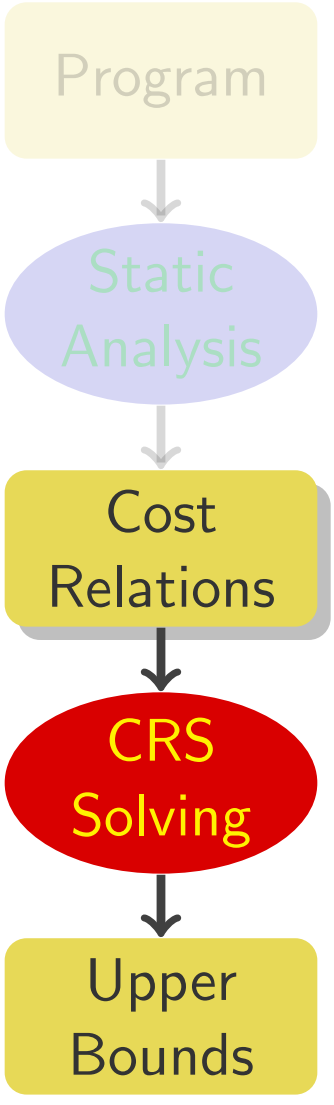


$$main^u(t,m) = \frac{m^3 + 3t(m^2+m) + 5m}{6} \qquad rpop^u(t,m) = m * t$$

$$\begin{aligned} main(t,m) &= 0 && \{m = 0, t \geq 0\} \\ main(t,m) &= rpop(t,m) + 1 + main(t_2, m_1) && \left\{ \begin{array}{l} m \geq 1, t \geq 0, m_1 = m - 1 \\ t_2 = t' + 1, t \geq t' \geq 0 \end{array} \right\} \end{aligned}$$

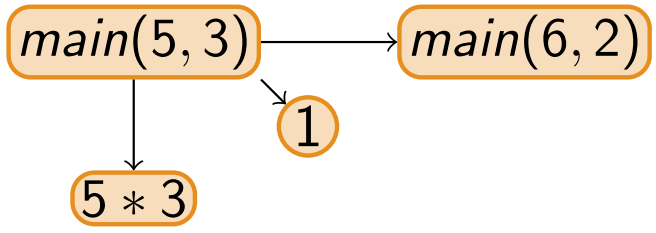
$$\begin{aligned} rpop(t,m) &= 0 && \{m \geq 1\} \\ rpop(t,m) &= m + rpop(t_1, m) && \{m \geq 1, t \geq 1, t_1 = t - 1\} \end{aligned}$$

Classical Approach to Cost Analysis: working example



?

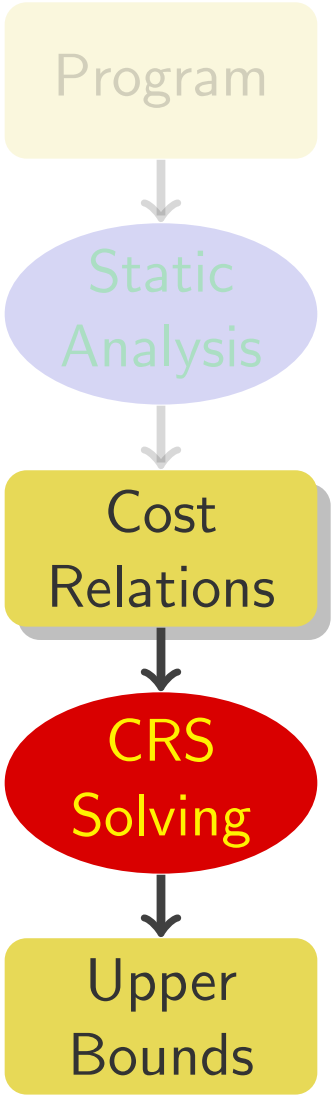
We consider a worst-case evaluation of $main(5,3)$



$$main^u(t,m) = \frac{m^3 + 3t(m^2+m) + 5m}{6} \qquad rpop^u(t,m) = m * t$$

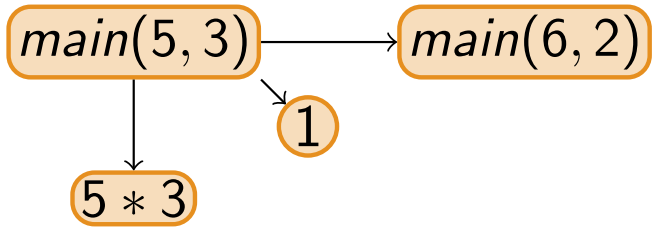
$$\begin{aligned}
main(t,m) &= 0 && \{m = 0, t \geq 0\}\\
main(t,m) &= rpop(t,m) + 1 + main(t_2,m_1) && \left\{\begin{array}{l} m \geq 1, t \geq 0, m_1 = m-1 \\ t_2 = t'+1, t \geq t' \geq 0 \end{array}\right\}\\
rpop(t,m) &= 0 && \{m \geq 1\}\\
rpop(t,m) &= m + rpop(t_1,m) && \{m \geq 1, t \geq 1, t_1 = t-1\}
\end{aligned}$$

Classical Approach to Cost Analysis: working example



?

We consider a worst-case evaluation of $main(5,3)$

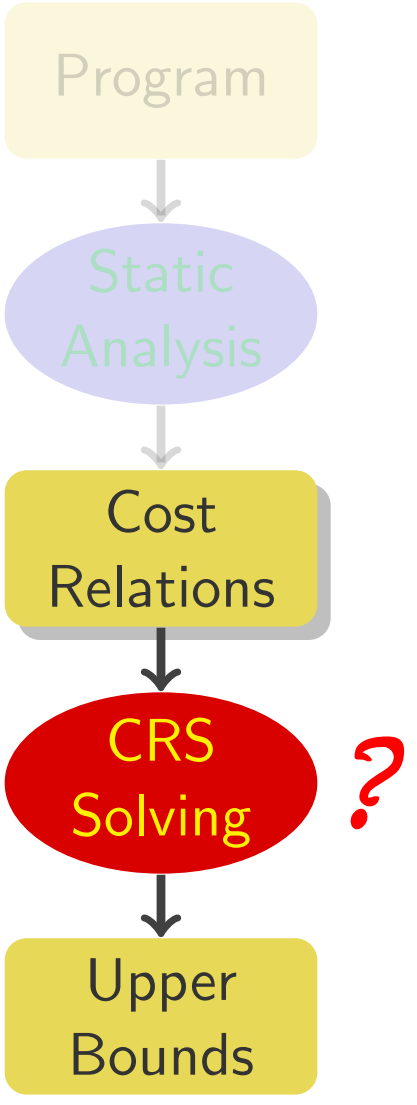


$$main^u(t,m) = \frac{m^3 + 3t(m^2+m) + 5m}{6} \qquad rpop^u(t,m) = m * t$$

$$\begin{aligned}
main(t,m) &= 0 && \{m = 0, t \geq 0\}\\
main(t,m) &= rpop(t,m) + 1 + main(t_2, m_1) && \left\{\begin{array}{l} m \geq 1, t \geq 0, m_1 = m - 1 \\ t_2 = t' + 1, t \geq t' \geq 0 \end{array}\right\}
\end{aligned}$$

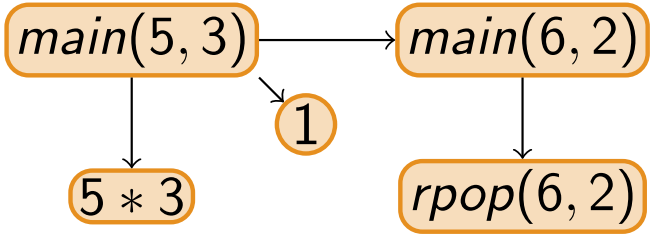
$$\begin{aligned}
rpop(t,m) &= 0 && \{m \geq 1\}\\
rpop(t,m) &= m + rpop(t_1, m) && \{m \geq 1, t \geq 1, t_1 = t - 1\}
\end{aligned}$$

Classical Approach to Cost Analysis: working example



?

We consider a worst-case evaluation of $main(5,3)$

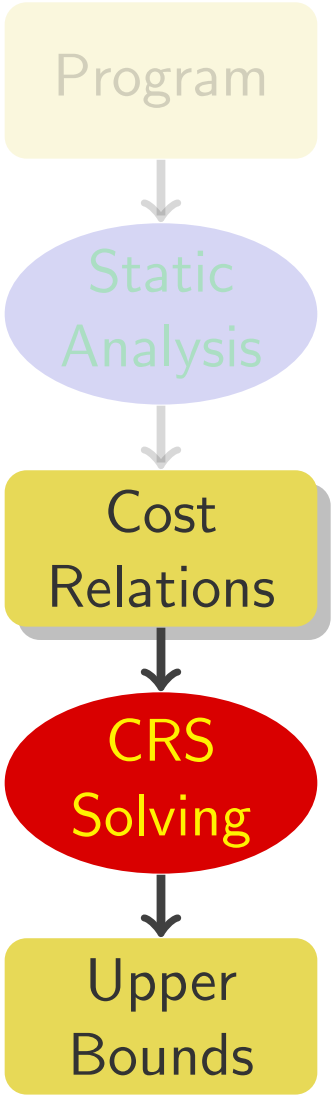


$$main^u(t,m) = \frac{m^3 + 3t(m^2+m) + 5m}{6} \qquad rpop^u(t,m) = m * t$$

$$\begin{aligned} main(t,m) &= 0 && \{m = 0, t \geq 0\} \\ main(t,m) &= rpop(t,m) + 1 + main(t_2, m_1) && \left\{ \begin{array}{l} m \geq 1, t \geq 0, m_1 = m - 1 \\ t_2 = t' + 1, t \geq t' \geq 0 \end{array} \right\} \end{aligned}$$

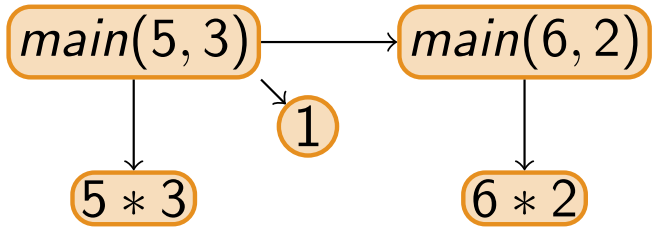
$$\begin{aligned} rpop(t,m) &= 0 && \{m \geq 1\} \\ rpop(t,m) &= m + rpop(t_1, m) && \{m \geq 1, t \geq 1, t_1 = t - 1\} \end{aligned}$$

Classical Approach to Cost Analysis: working example



?

We consider a worst-case evaluation of $main(5,3)$

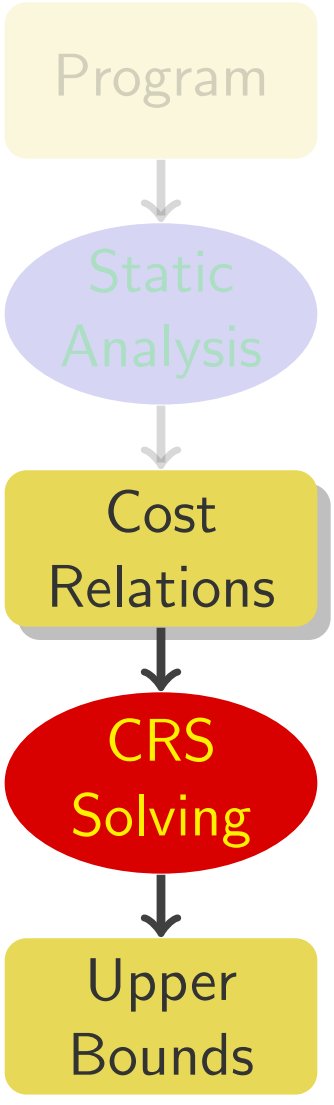


$$main^u(t,m) = \frac{m^3 + 3t(m^2+m) + 5m}{6} \qquad rpop^u(t,m) = m * t$$

$$\begin{aligned} main(t,m) &= 0 && \{m = 0, t \geq 0\} \\ main(t,m) &= rpop(t,m) + 1 + main(t_2, m_1) && \left\{ \begin{array}{l} m \geq 1, t \geq 0, m_1 = m - 1 \\ t_2 = t' + 1, t \geq t' \geq 0 \end{array} \right\} \end{aligned}$$

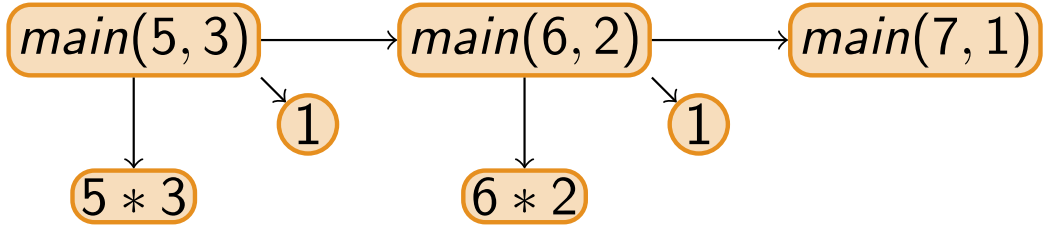
$$\begin{aligned} rpop(t,m) &= 0 && \{m \geq 1\} \\ rpop(t,m) &= m + rpop(t_1, m) && \{m \geq 1, t \geq 1, t_1 = t - 1\} \end{aligned}$$

Classical Approach to Cost Analysis: working example



?

We consider a worst-case evaluation of $main(5,3)$



$$main^u(t,m) = \frac{m^3 + 3t(m^2+m) + 5m}{6} \qquad rpop^u(t,m) = m * t$$

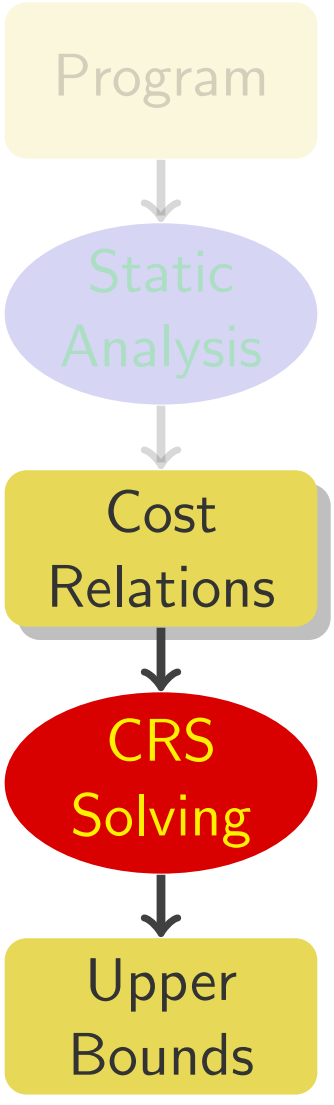
$$main(t,m) = 0 \quad \{m = 0, t \geq 0\}$$

$$main(t,m) = rpop(t,m) + 1 + main(t_2, m_1) \quad \left\{ \begin{array}{l} m \geq 1, t \geq 0, m_1 = m - 1 \\ t_2 = t' + 1, t \geq t' \geq 0 \end{array} \right\}$$

$$rpop(t,m) = 0 \quad \{m \geq 1\}$$

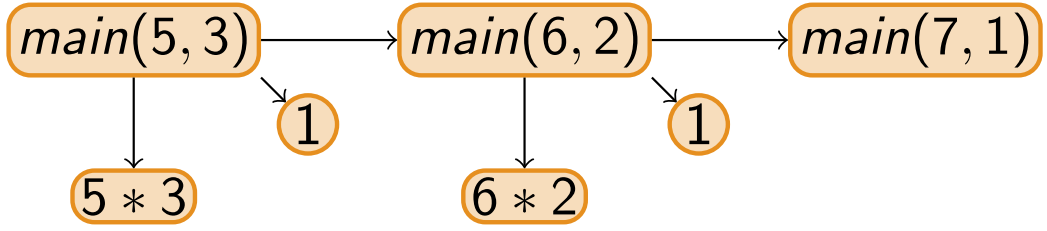
$$rpop(t,m) = m + rpop(t_1, m) \quad \{m \geq 1, t \geq 1, t_1 = t - 1\}$$

Classical Approach to Cost Analysis: working example



?

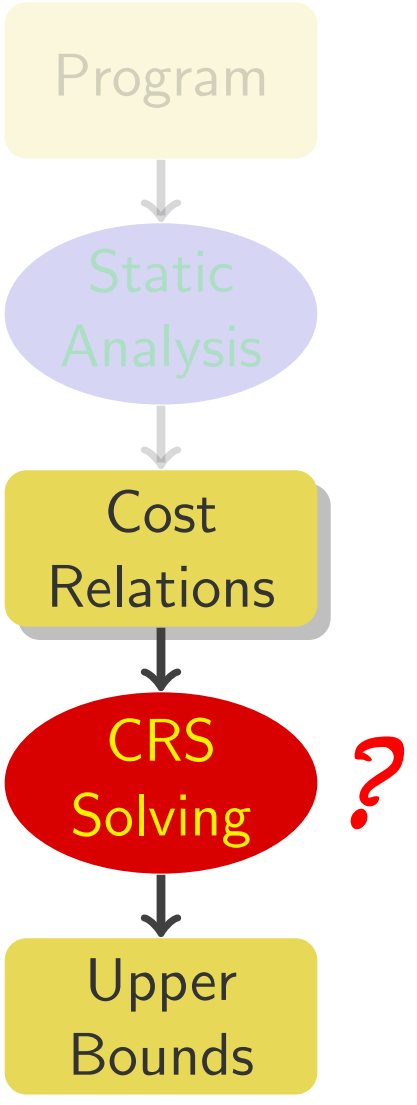
We consider a worst-case evaluation of $main(5,3)$



$$main^u(t,m) = \frac{m^3 + 3t(m^2+m) + 5m}{6} \qquad rpop^u(t,m) = m * t$$

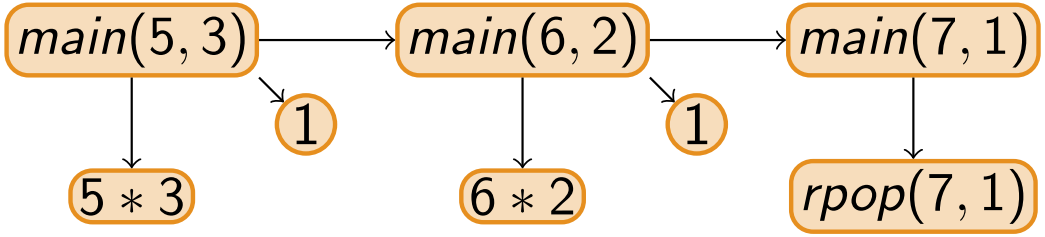
$$\begin{array}{lll} main(t,m) = 0 & & \{m = 0, t \geq 0\} \\ main(t,m) = \begin{array}{l} rpop(t,m) + 1 + \\ main(t_2, m_1) \end{array} & & \left\{ \begin{array}{l} m \geq 1, t \geq 0, m_1 = m - 1 \\ t_2 = t' + 1, t \geq t' \geq 0 \end{array} \right\} \\ \\ rpop(t,m) = 0 & & \{m \geq 1\} \\ rpop(t,m) = m + rpop(t_1, m) & & \{m \geq 1, t \geq 1, t_1 = t - 1\} \end{array}$$

Classical Approach to Cost Analysis: working example



?

We consider a worst-case evaluation of $main(5,3)$

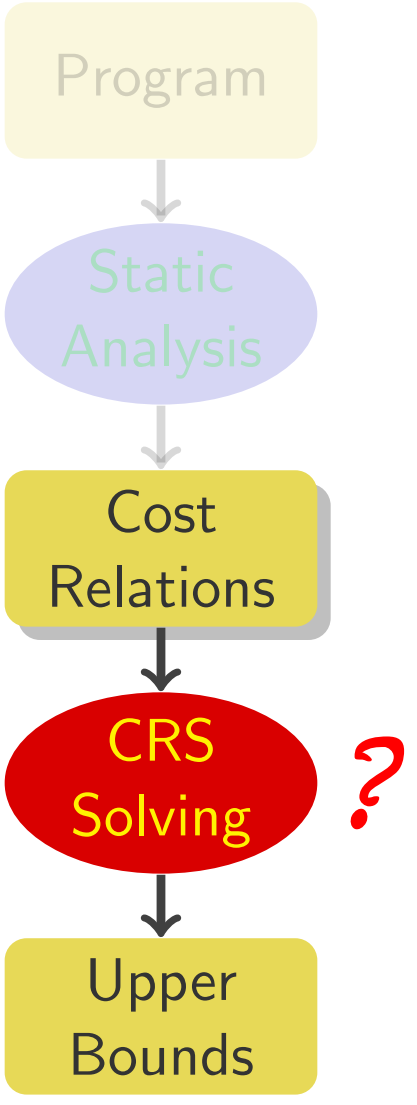


$$main^u(t,m) = \frac{m^3 + 3t(m^2+m) + 5m}{6} \qquad rpop^u(t,m) = m * t$$

$$\begin{aligned}
main(t,m) &= 0 && \{m = 0, t \geq 0\}\\
main(t,m) &= rpop(t,m) + 1 + main(t_2,m_1) && \left\{\begin{array}{l} m \geq 1, t \geq 0, m_1 = m-1 \\ t_2 = t'+1, t \geq t' \geq 0 \end{array}\right\}
\end{aligned}$$

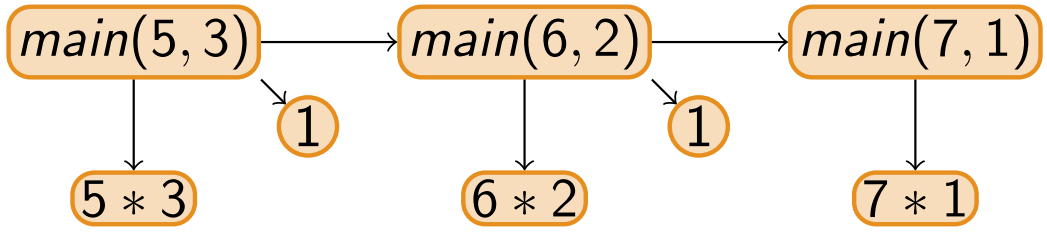
$$\begin{aligned}
rpop(t,m) &= 0 && \{m \geq 1\}\\
rpop(t,m) &= m + rpop(t_1,m) && \{m \geq 1, t \geq 1, t_1 = t-1\}
\end{aligned}$$

Classical Approach to Cost Analysis: working example



?

We consider a worst-case evaluation of $main(5,3)$

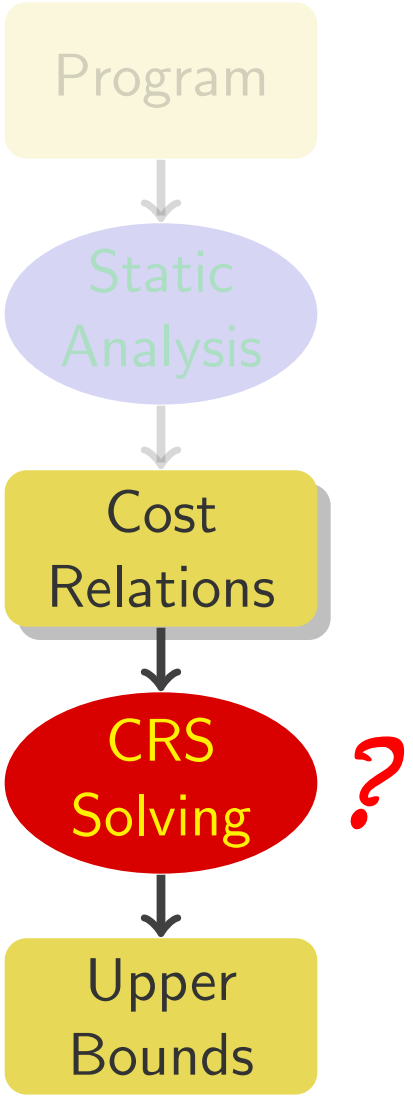


$$main^u(t,m) = \frac{m^3 + 3t(m^2+m) + 5m}{6} \qquad rpop^u(t,m) = m * t$$

$$\begin{aligned} main(t,m) &= 0 && \{m = 0, t \geq 0\} \\ main(t,m) &= rpop(t,m) + 1 + main(t_2, m_1) && \left\{ \begin{array}{l} m \geq 1, t \geq 0, m_1 = m - 1 \\ t_2 = t' + 1, t \geq t' \geq 0 \end{array} \right\} \end{aligned}$$

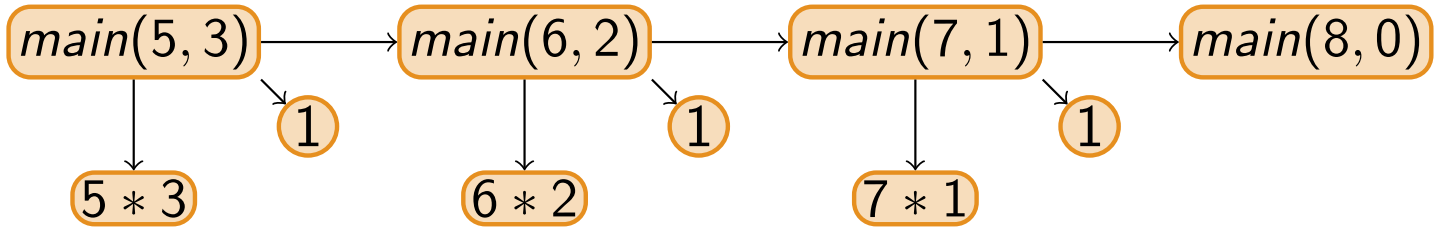
$$\begin{aligned} rpop(t,m) &= 0 && \{m \geq 1\} \\ rpop(t,m) &= m + rpop(t_1, m) && \{m \geq 1, t \geq 1, t_1 = t - 1\} \end{aligned}$$

Classical Approach to Cost Analysis: working example



?

We consider a worst-case evaluation of $main(5,3)$

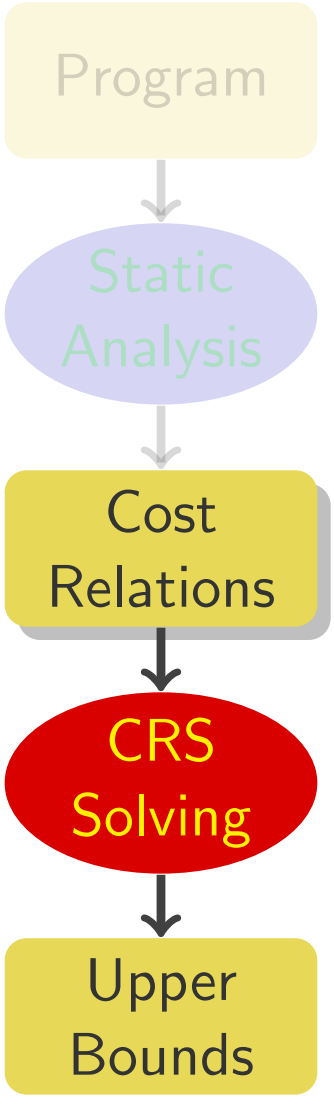


$$main^u(t,m) = \frac{m^3 + 3t(m^2+m) + 5m}{6} \qquad rpop^u(t,m) = m * t$$

$$\begin{aligned}
main(t,m) &= 0 && \{m = 0, t \geq 0\}\\
main(t,m) &= rpop(t,m) + 1 + main(t_2, m_1) && \left\{\begin{array}{l} m \geq 1, t \geq 0, m_1 = m-1 \\ t_2 = t' + 1, t \geq t' \geq 0 \end{array}\right\}
\end{aligned}$$

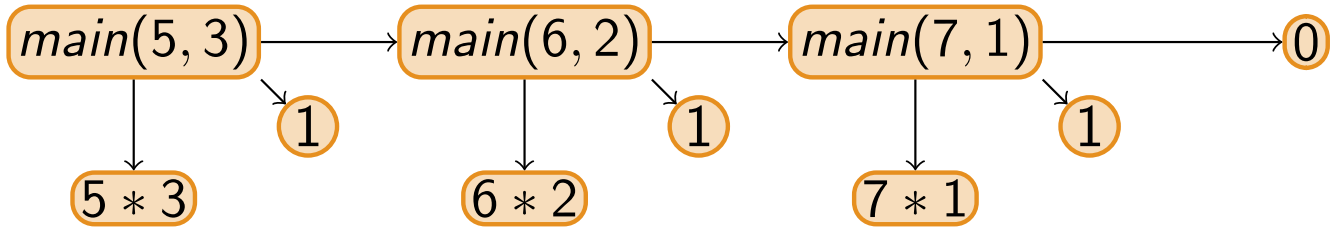
$$\begin{aligned}
rpop(t,m) &= 0 && \{m \geq 1\}\\
rpop(t,m) &= m + rpop(t_1, m) && \{m \geq 1, t \geq 1, t_1 = t-1\}
\end{aligned}$$

Classical Approach to Cost Analysis: working example



?

We consider a worst-case evaluation of $main(5,3)$

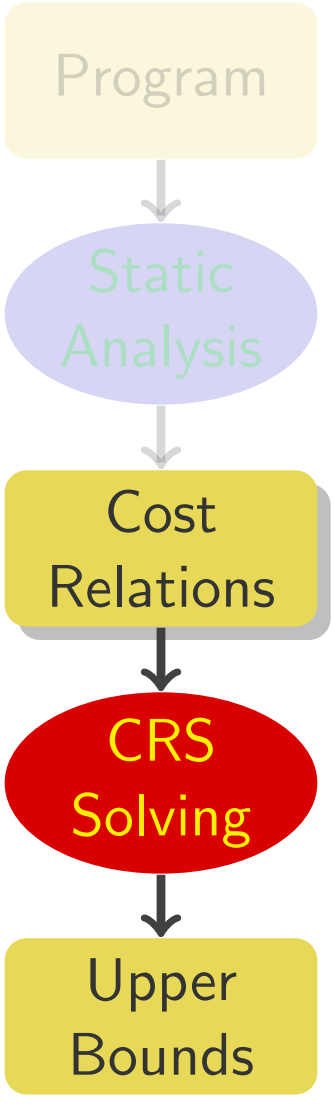


$$main^u(t,m) = \frac{m^3 + 3t(m^2+m) + 5m}{6} \qquad rpop^u(t,m) = m * t$$

$$\begin{aligned}
main(t,m) &= 0 && \{m = 0, t \geq 0\}\\
main(t,m) &= rpop(t,m) + 1 + main(t_2, m_1) && \left\{\begin{array}{l} m \geq 1, t \geq 0, m_1 = m-1 \\ t_2 = t' + 1, t \geq t' \geq 0 \end{array}\right\}
\end{aligned}$$

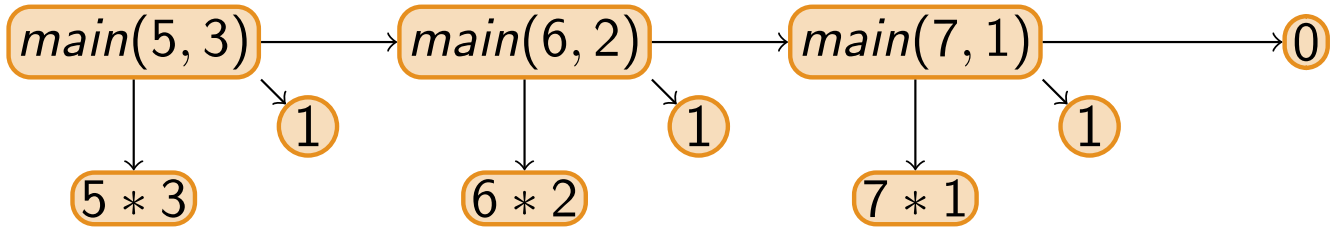
$$\begin{aligned}
rpop(t,m) &= 0 && \{m \geq 1\}\\
rpop(t,m) &= m + rpop(t_1, m) && \{m \geq 1, t \geq 1, t_1 = t-1\}
\end{aligned}$$

Classical Approach to Cost Analysis: working example



?

We consider a worst-case evaluation of $main(5,3)$



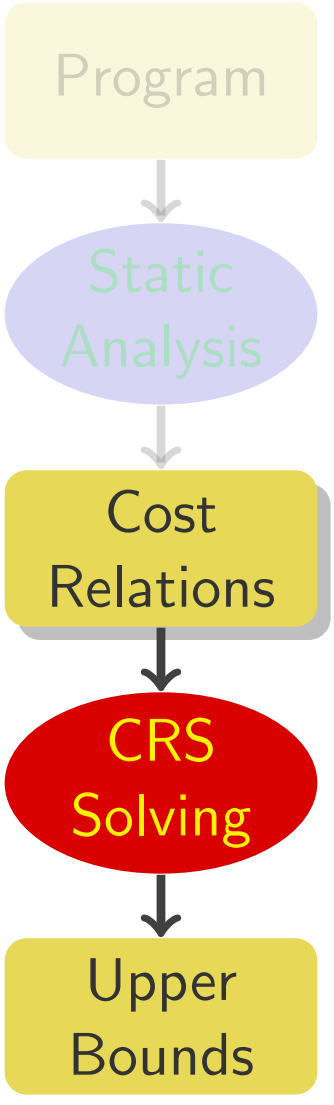
$$main(5,3) = 5*3+1+6*2+1+7*1+1+0 = 37$$

$$main^u(t,m) = \frac{m^3 + 3t(m^2+m) + 5m}{6} \qquad rpop^u(t,m) = m * t$$

$$\begin{aligned} main(t,m) &= 0 && \{m = 0, t \geq 0\} \\ main(t,m) &= rpop(t,m) + 1 + main(t_2, m_1) && \left\{ \begin{array}{l} m \geq 1, t \geq 0, m_1 = m-1 \\ t_2 = t' + 1, t \geq t' \geq 0 \end{array} \right\} \end{aligned}$$

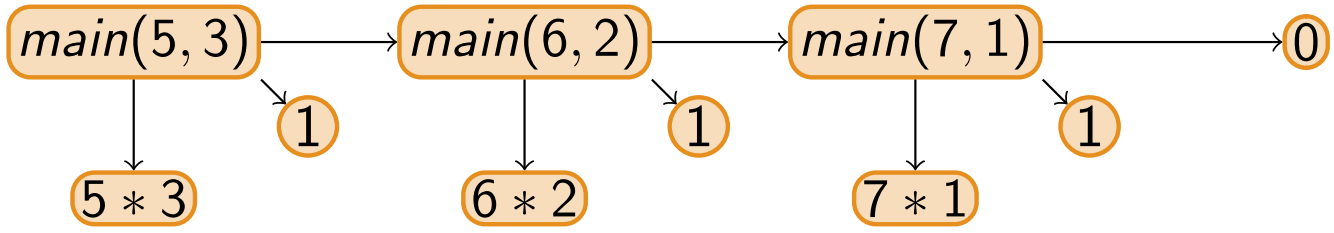
$$\begin{aligned} rpop(t,m) &= 0 && \{m \geq 1\} \\ rpop(t,m) &= m + rpop(t_1, m) && \{m \geq 1, t \geq 1, t_1 = t-1\} \end{aligned}$$

Classical Approach to Cost Analysis: working example



?

We consider a worst-case evaluation of $main(5,3)$



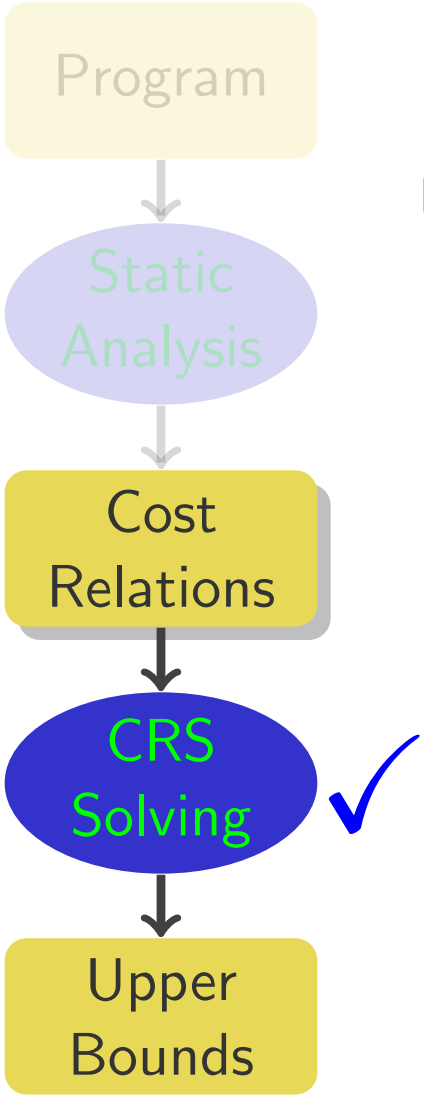
$$main^u(5,3) = \frac{3^3+3*5(3^2+3)+5*3}{6} = 37$$

$$main^u(t,m) = \frac{m^3+3t(m^2+m)+5m}{6} \qquad rpop^u(t,m) = m*t$$

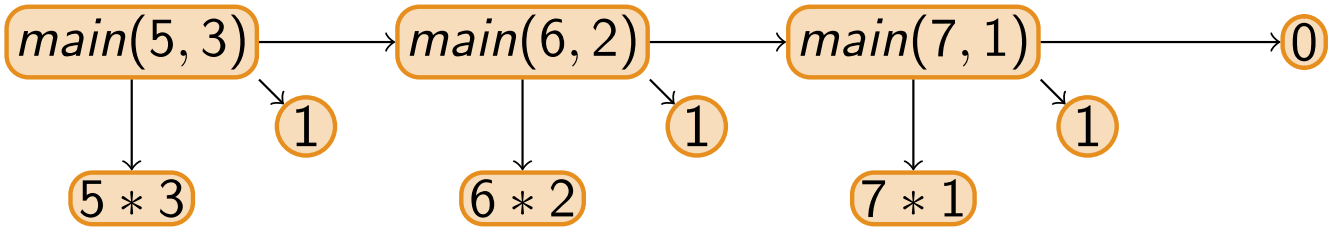
$$\begin{array}{ll} main(t,m) = 0 & \{m=0, t\geq 0\} \\ main(t,m) = rpop(t,m)+1+ main(t_2,m_1) & \left\{\begin{array}{l} m\geq 1, t\geq 0, m_1 = m-1 \\ t_2 = t'+1, t\geq t'\geq 0 \end{array}\right\} \end{array}$$

$$\begin{array}{ll} rpop(t,m) = 0 & \{m\geq 1\} \\ rpop(t,m) = m + rpop(t_1,m) & \{m\geq 1, t\geq 1, t_1 = t-1\} \end{array}$$

Classical Approach to Cost Analysis: working example



We consider a worst-case evaluation of $main(5,3)$



PUBS gives the exact UB

$m) = m * t$

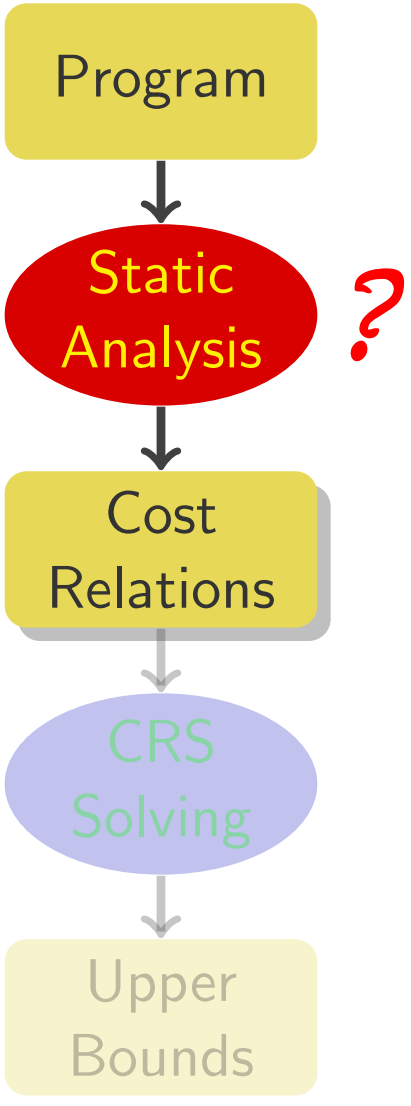
$$main(t,m) = 0 \qquad \{m = 0, t \geq 0\}$$

$$main(t,m) = \begin{matrix} rpop(t,m) + 1 + \\ main(t_2, m_1) \end{matrix} \quad \left\{ \begin{matrix} m \geq 1, t \geq 0, m_1 = m - 1 \\ t_2 = t' + 1, t \geq t' \geq 0 \end{matrix} \right\}$$

$$rpop(t,m) = 0 \qquad \{m \geq 1\}$$

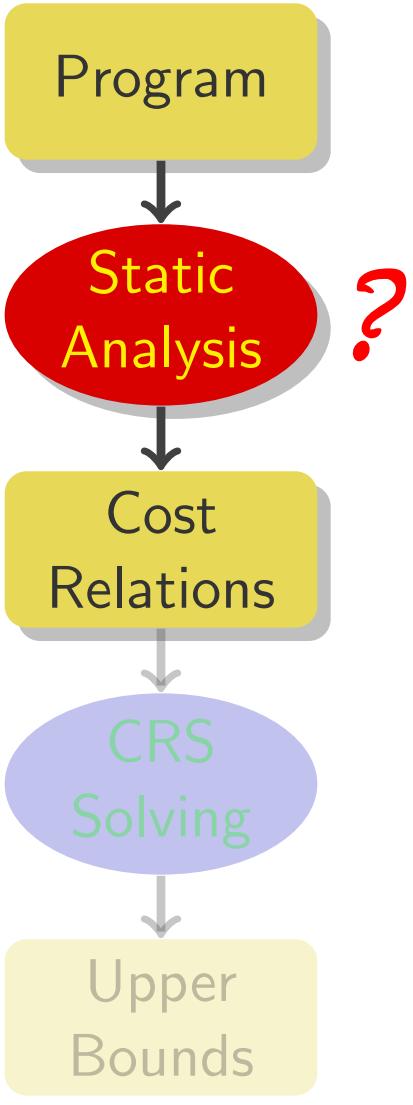
$$rpop(t,m) = m + rpop(t_1, m) \qquad \{m \geq 1, t \geq 1, t_1 = t - 1\}$$

Classical Approach to Cost Analysis: working example



$$
\begin{array}{lll}
main(t,m) = & 0 & \{m = 0, t \geq 0\} \\
main(t,m) = & rpop(t,m) + 1 + main(t_2, m_1) & \left\{ \begin{array}{l} m \geq 1, t \geq 0, m_1 = m - 1 \\ t_2 = t' + 1, t \geq t' \geq 0 \end{array} \right\} \\
 & & \\
rpop(t,m) = & 0 & \{m \geq 1\} \\
rpop(t,m) = & m + rpop(t_1, m) & \{m \geq 1, t \geq 1, t_1 = t - 1\}
\end{array}
$$

Classical Approach to Cost Analysis: working example



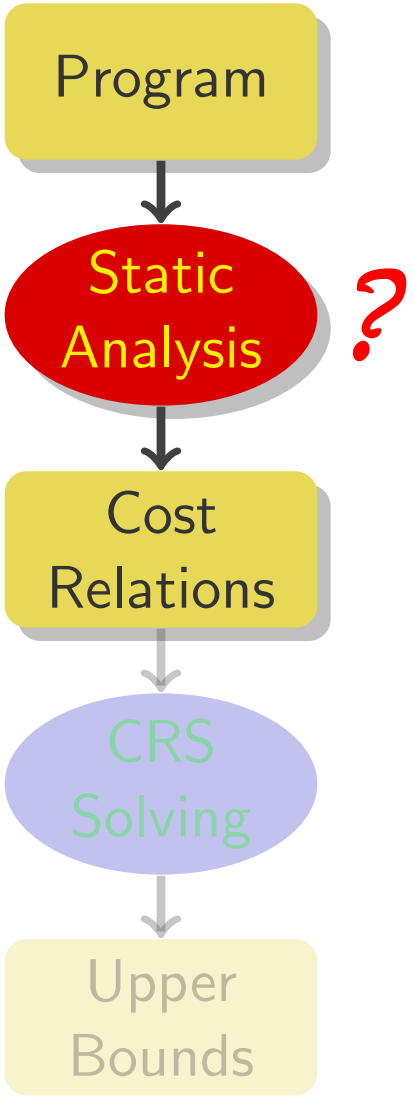

```
//@requires m >= 1
void rpop(int m){
  while(top!=null && *){
    //pop
    top=top.next ;
    //@acquire(m)
}}
```

```
//@requires m >= 0
void main(int m) {
  while( m > 0 ) {
    rpop(m)  ;
    // push(*);
    top = new
     Node(*,top);
    //@acquire(1)
    m = m-1      ;
}}}
```

$$
\begin{array}{ll}
main(t,m) = 0 & \{m = 0, t \geq 0\} \\
main(t,m) = \begin{array}{l} rpop(t,m) + 1 + \\ main(t_2, m_1) \end{array} & \left\{ \begin{array}{l} m \geq 1, t \geq 0, m_1 = m - 1 \\ t_2 = t' + 1, t \geq t' \geq 0 \end{array} \right\}
\end{array}
$$

$$
\begin{array}{ll}
rpop(t,m) = 0 & \{m \geq 1\} \\
rpop(t,m) = m + rpop(t_1, m) & \{m \geq 1, t \geq 1, t_1 = t - 1\}
\end{array}
$$

Classical Approach to Cost Analysis: working example



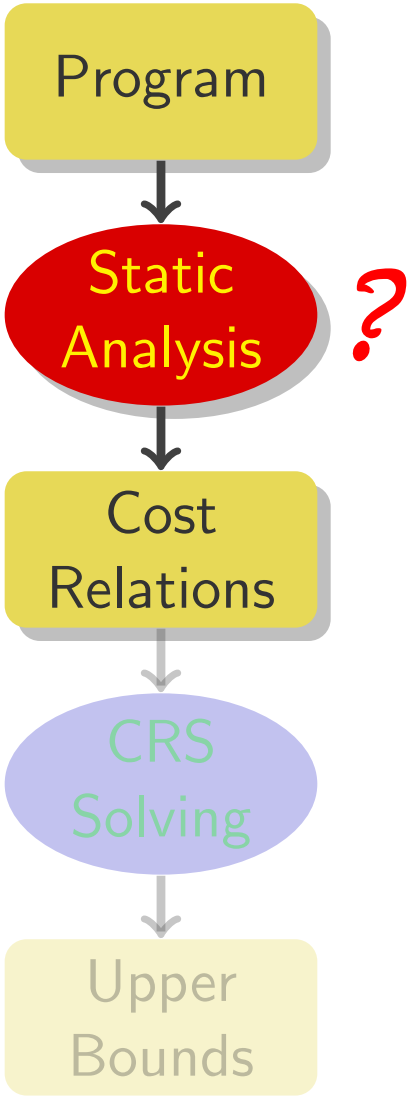

```
//@requires m >= 1
void rpop(int m){
  while(top!=null && *){
    //pop
    top=top.next ;
    //@acquire(m)
}}
```

```
//@requires m >= 0
void main(int m) {
  while( m > 0 ) {
    rpop(m)  ;
    // push(*);
    top = new
     Node(*,top);
    //@acquire(1)
    m = m-1      ;
}}}
```

$$
\begin{array}{ll}
main(t,m) = 0 & \{m = 0, t \geq 0\} \\
main(t,m) = \begin{array}{l} rpop(t,m) + 1 + \\ main(t_2, m_1) \end{array} & \left\{ \begin{array}{l} m \geq 1, t \geq 0, m_1 = m - 1 \\ t_2 = t' + 1, t \geq t' \geq 0 \end{array} \right\}
\end{array}
$$

$$
\begin{array}{ll}
rpop(t,m) = 0 & \{m \geq 1\} \\
rpop(t,m) = m + rpop(t_1, m) & \{m \geq 1, t \geq 1, t_1 = t - 1\}
\end{array}
$$

Classical Approach to Cost Analysis: working example



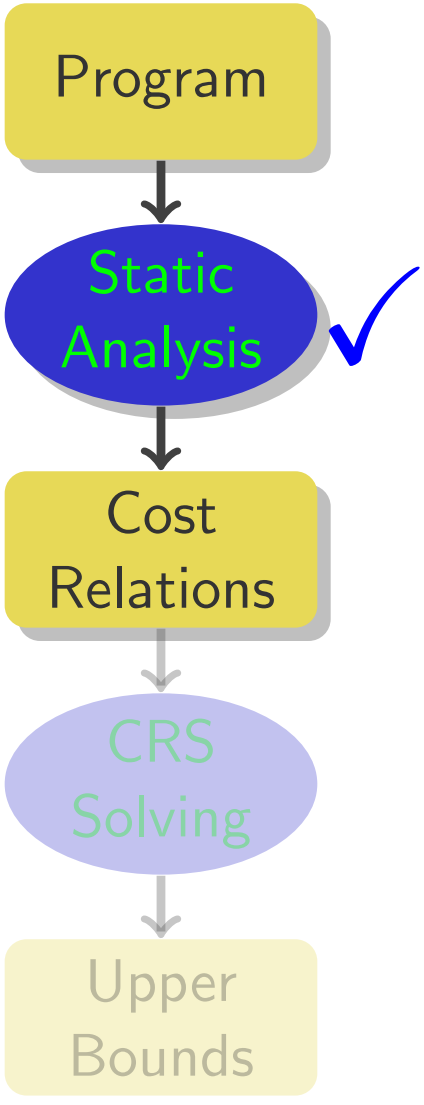

```
//@requires m >= 1
void rpop(int m){
  while(top!=null && *){
    //pop
    top=top.next ;
    //@acquire(m)
}}
```

```
//@requires m >= 0
void main(int m) {
  while( m > 0 ) {
    rpop(m)  ;
    // push(*);
    top = new
     Node(*,top);
    //@acquire(1)
    m = m-1      ;
}}}
```

$s \geq s' \geq 0$ is the exact `rpop` size postcondition

$$
\begin{array}{lll}
main(t,m) = 0 & & \{m = 0, t \geq 0\} \\
main(t,m) = \begin{array}{l} rpop(t,m) + 1 + \\ main(t_2, m_1) \end{array} & & \left\{ \begin{array}{l} m \geq 1, t \geq 0, m_1 = m - 1 \\ t_2 = t' + 1, t \geq t' \geq 0 \end{array} \right\} \\
\\
rpop(t,m) = 0 & & \{m \geq 1\} \\
rpop(t,m) = m + rpop(t_1, m) & & \{m \geq 1, t \geq 1, t_1 = t - 1\}
\end{array}
$$

Classical Approach to Cost Analysis: working example



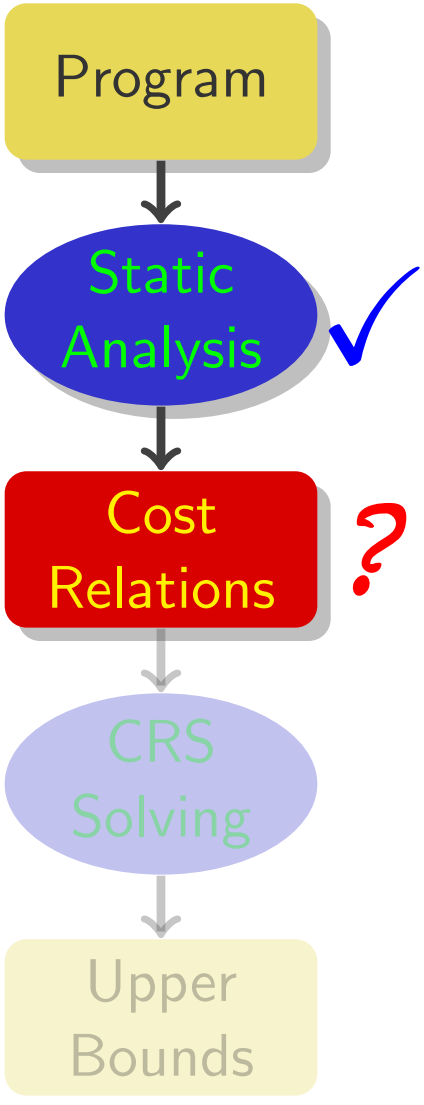

```
//@requires m >= 1
void rpop(int m){
  while(top!=null && *){
    //pop
    top=top.next ;
    //@acquire(m)
}}
```

```
//@requires m >= 0
void main(int m) {
  while( m > 0 ) {
    rpop(m)  ;
    // push(*);
    top = new
     Node(*,top);
    //@acquire(1)
    m = m-1      ;
}}}
```

$s \geq s' \geq 0$ is the exact rpop size postcondition

$$
\begin{array}{lll}
main(t,m) = 0 & & \{m = 0, t \geq 0\} \\
main(t,m) = \begin{array}{l} rpop(t,m) + 1 + \\ main(t_2, m_1) \end{array} & & \left\{ \begin{array}{l} m \geq 1, t \geq 0, m_1 = m - 1 \\ t_2 = t' + 1, t \geq t' \geq 0 \end{array} \right\} \\
\\
rpop(t,m) = 0 & & \{m \geq 1\} \\
rpop(t,m) = m + rpop(t_1, m) & & \{m \geq 1, t \geq 1, t_1 = t - 1\}
\end{array}
$$

Classical Approach to Cost Analysis: working example




```
//@requires m >= 1
void rpop(int m){
  while(top!=null && *){
    //pop
    top=top.next ;
    //@acquire(m)
}}
```

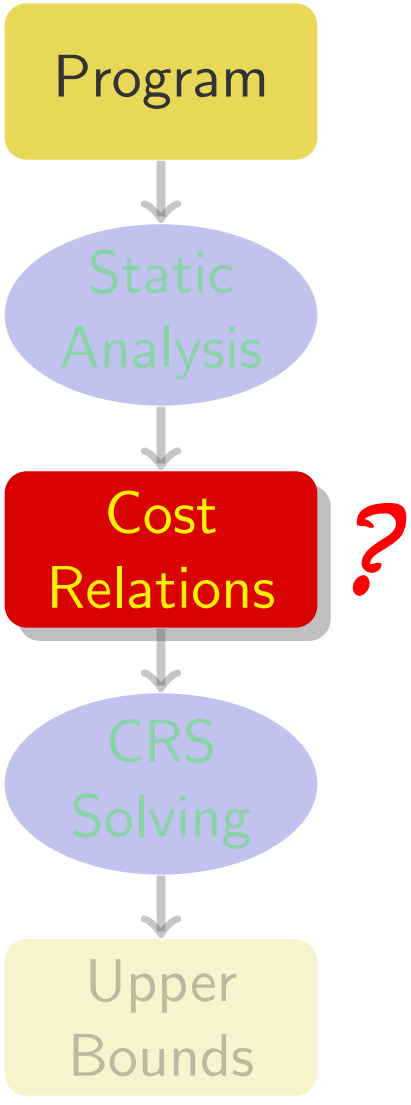
```
//@requires m >= 0
void main(int m) {
  while( m > 0 ) {
    rpop(m)  ;
    // push(*);
    top = new
     Node(*,top);
    //@acquire(1)
    m = m-1      ;
}}}
```

$s \geq s' \geq 0$ is the exact rpop size postcondition

$$
\begin{array}{ll}
main(t,m) = 0 & \{m = 0, t \geq 0\} \\
main(t,m) = \begin{array}{l} rpop(t,m) + 1 + \\ main(t_2, m_1) \end{array} & \left\{ \begin{array}{l} m \geq 1, t \geq 0, m_1 = m - 1 \\ t_2 = t' + 1, t \geq t' \geq 0 \end{array} \right\}
\end{array}
$$

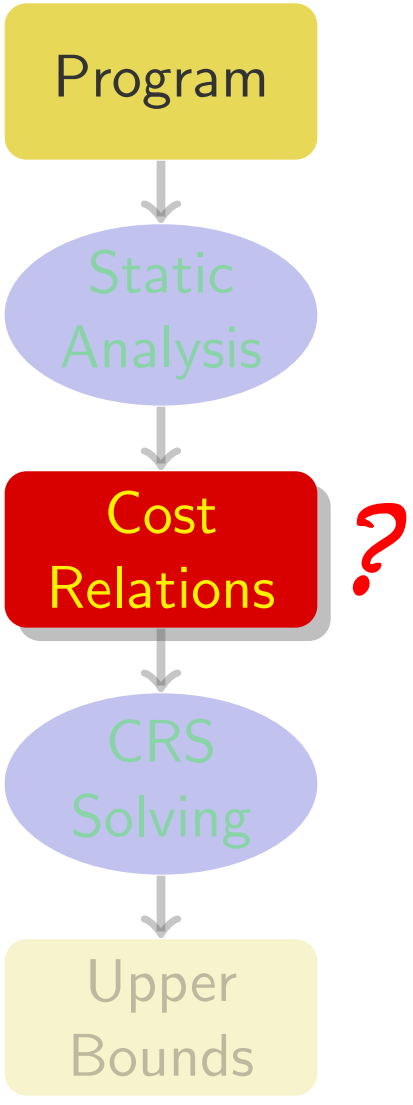
$$
\begin{array}{ll}
rpop(t,m) = 0 & \{m \geq 1\} \\
rpop(t,m) = m + rpop(t_1, m) & \{m \geq 1, t \geq 1, t_1 = t - 1\}
\end{array}
$$

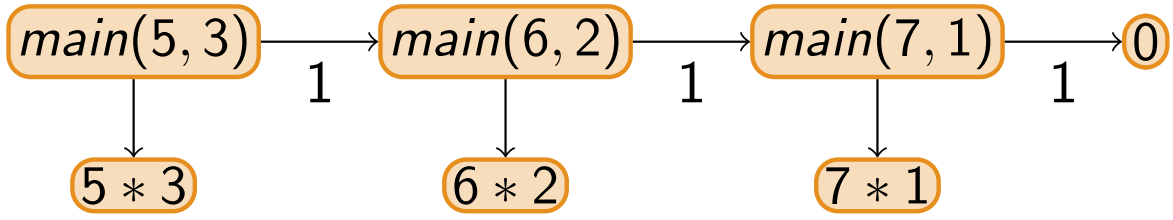
Classical Approach to Cost Analysis: working example



$$
\begin{array}{lll}
main(t,m) = & 0 & \{m = 0, t \geq 0\} \\
main(t,m) = & rpop(t,m) + 1 + main(t_2, m_1) & \left\{ \begin{array}{l} m \geq 1, t \geq 0, m_1 = m - 1 \\ t_2 = t' + 1, t \geq t' \geq 0 \end{array} \right\} \\
 & & \\
rpop(t,m) = & 0 & \{m \geq 1\} \\
rpop(t,m) = & m + rpop(t_1, m) & \{m \geq 1, t \geq 1, t_1 = t - 1\}
\end{array}
$$

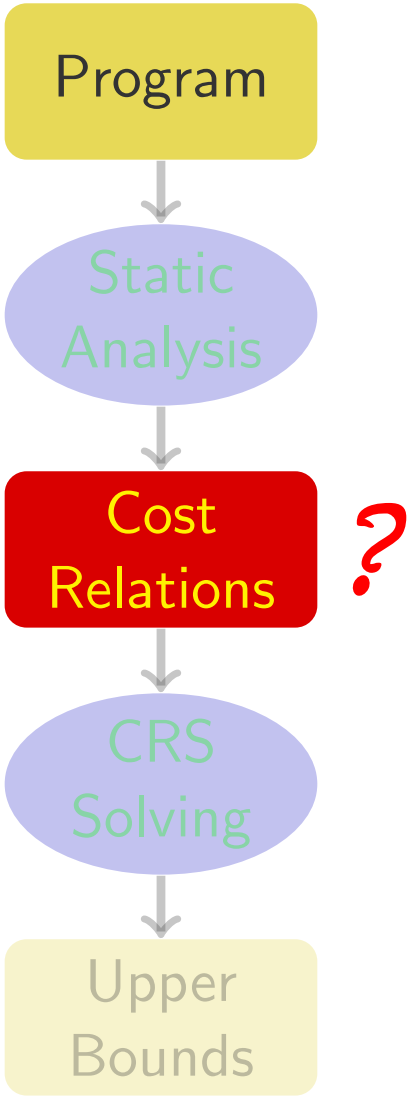
Classical Approach to Cost Analysis: working example



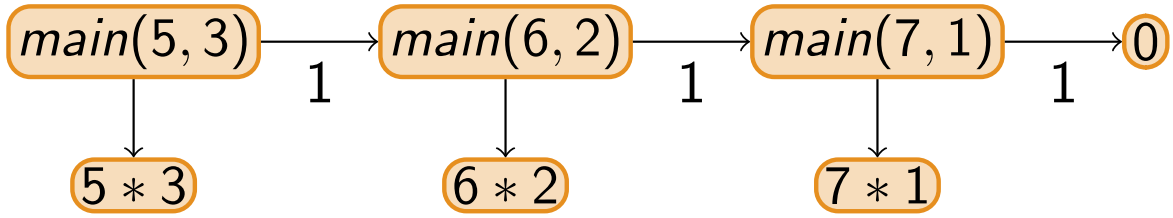


$$
\begin{array}{lll}
main(t,m) = & 0 & \{m = 0, t \geq 0\} \\
main(t,m) = & rpop(t,m) + 1 + \\ & main(t_2, m_1) & \left\{ \begin{array}{l} m \geq 1, t \geq 0, m_1 = m - 1 \\ t_2 = t' + 1, t \geq t' \geq 0 \end{array} \right\} \\
rpop(t,m) = & 0 & \{m \geq 1\} \\
rpop(t,m) = & m + rpop(t_1, m) & \{m \geq 1, t \geq 1, t_1 = t - 1\}
\end{array}
$$

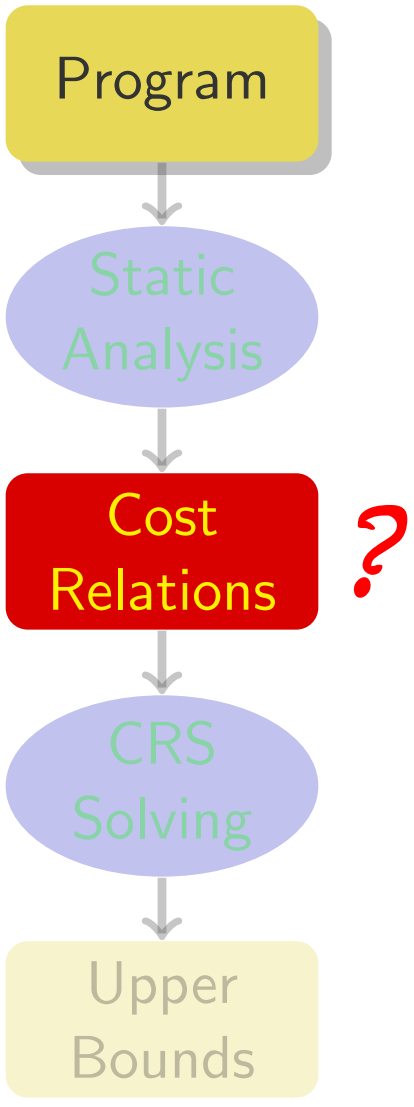
Classical Approach to Cost Analysis: working example



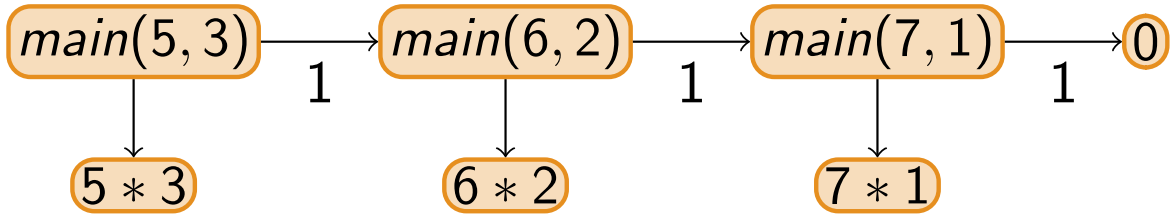
?



Classical Approach to Cost Analysis: working example



?

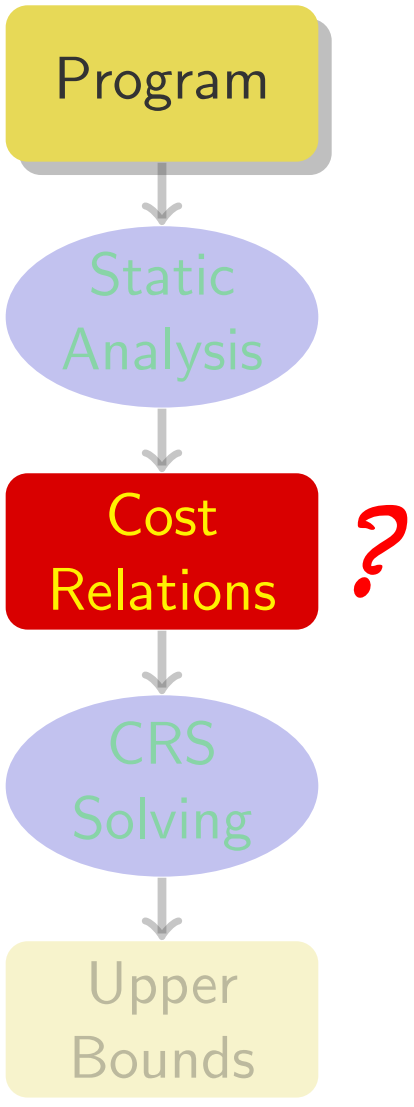


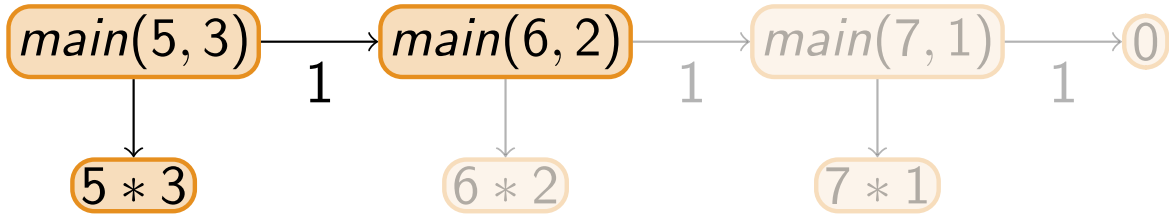

```
//@requires m >= 0
void main(int m) {
  while( m > 0 ) {
    rpop(m) ;
    top=new Node(*,top);
    //@acquire(1)
    m = m-1      ; }}}
```

```
//@requires m >= 1
void rpop(int m){
  if (top!=null && *){
  //pop
      top=top.next ;
      //@acquire(m)
}}
```

while

Classical Approach to Cost Analysis: working example



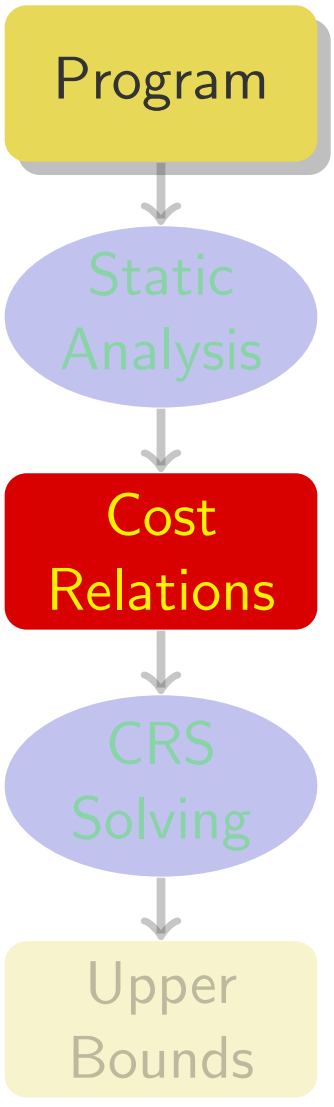


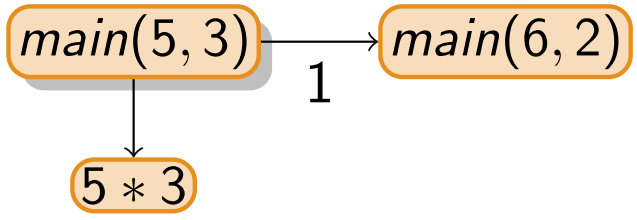

```
//@requires m >= 0
void main(int m) {
  while( m > 0 ) {
    rpop(m) ;
    top=new Node(*,top);
    //@acquire(1)
    m = m-1      ; }}}
```

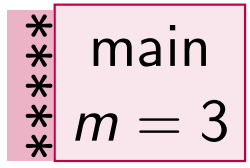
```
//@requires m >= 1
void rpop(int m){
  if (top!=null && *){
  //pop
      top=top.next ;
      //@acquire(m)
}}
```

while

Classical Approach to Cost Analysis: working example







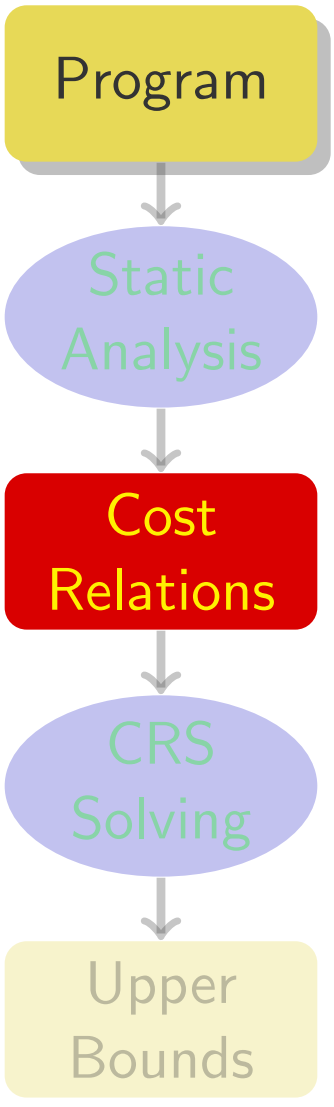
?

```
//@requires m >= 0
void main(int m) {
  while( m > 0 ) {
    rpop(m) ;
    top=new Node(*,top);
    //@acquire(1)
    m = m-1      ; }}}
```

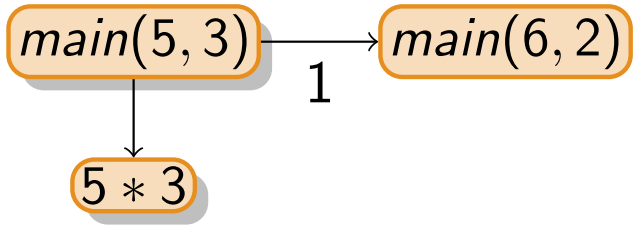
```
//@requires m >= 1
void rpop(int m){
  if (top!=null && *){
    //pop
    top=top.next ;
    //@acquire(m)
}}
```

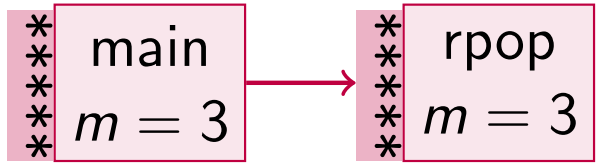
while

Classical Approach to Cost Analysis: working example



?



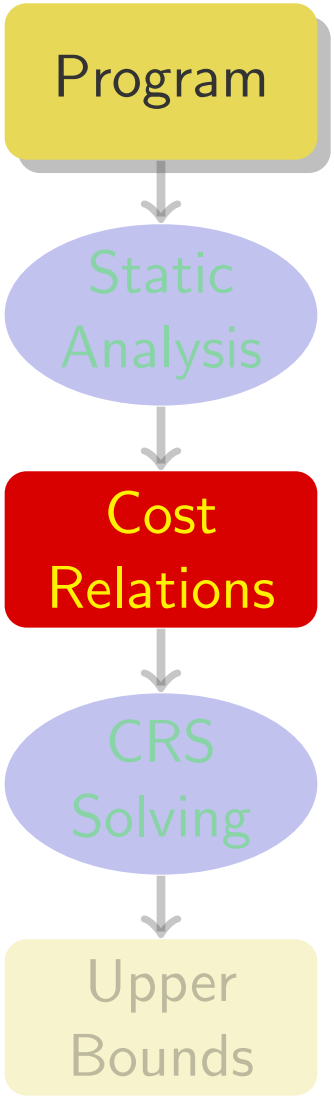



```
//@requires m >= 0
void main(int m) {
  while( m > 0 ) {
    rpop(m) ;
    top=new Node(*,top);
    //@acquire(1)
    m = m-1      ; }}}
```

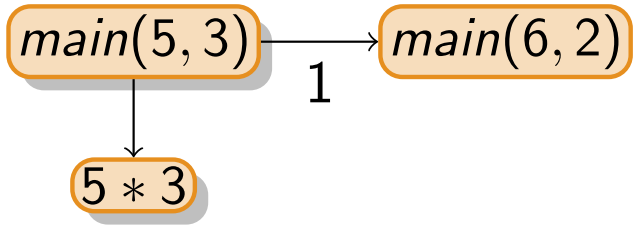
```
//@requires m >= 1
void rpop(int m){
  if (top!=null && *){
  //pop
      top=top.next ;
      //@acquire(m)
}}
```

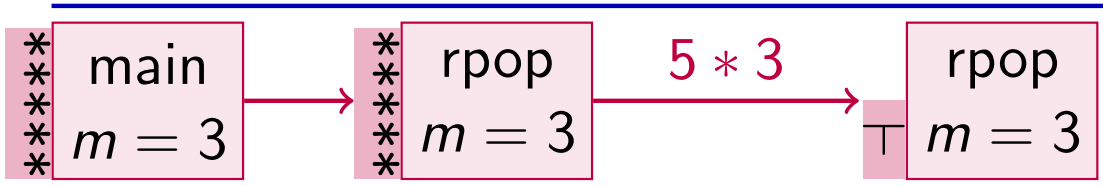
while

Classical Approach to Cost Analysis: working example



?

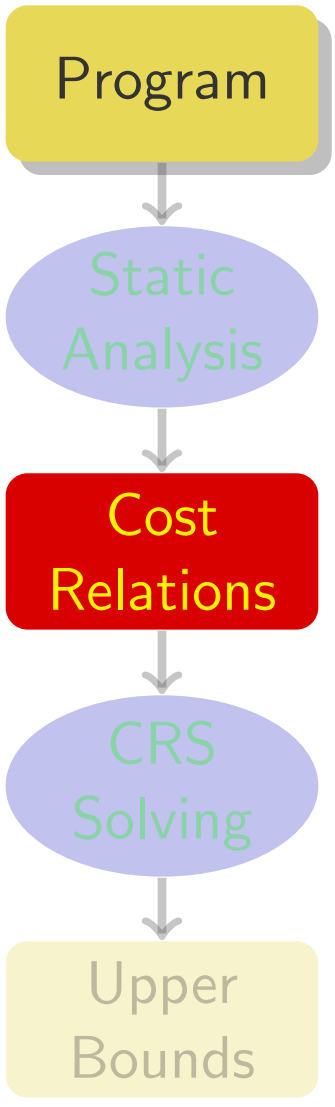




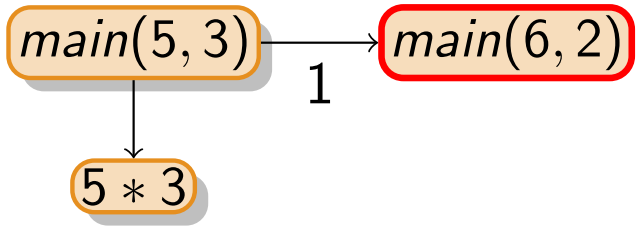

```
//@requires m >= 0
void main(int m) {
  while( m > 0 ) {
    rpop(m) ;
    top=new Node(*,top);
    //@acquire(1)
    m = m-1      ; }}}
```

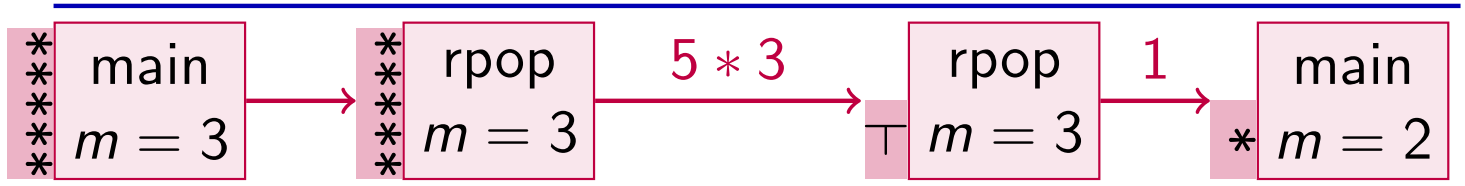
```
//@requires m >= 1
void rpop(int m){
  if (top!=null && *){
  while
    //pop
    top=top.next ;
    //@acquire(m)
}}
```

Classical Approach to Cost Analysis: working example



?



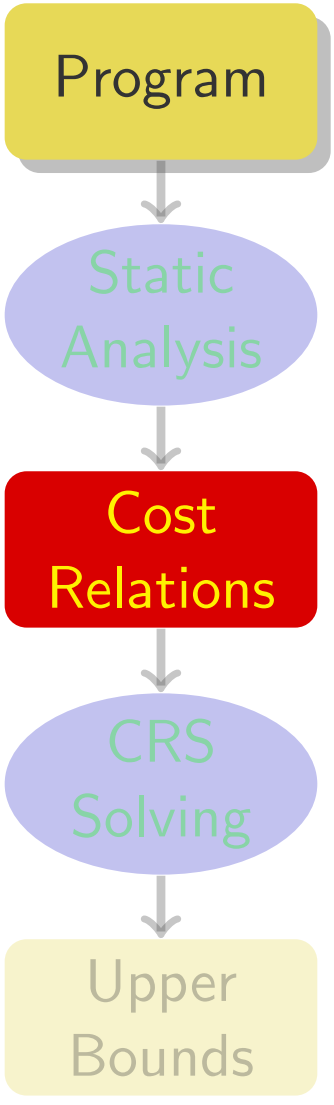



```
//@requires m >= 0
void main(int m) {
  while( m > 0 ) {
    rpop(m) ;
    top=new Node(*,top);
    //@acquire(1)
    m = m-1      ; }}}
```

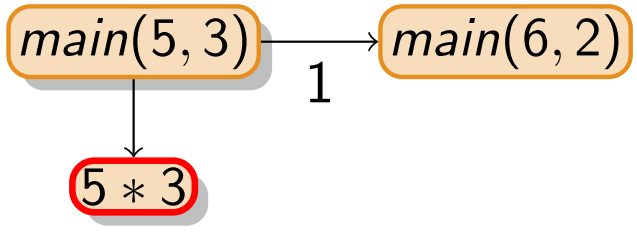
```
//@requires m >= 1
void rpop(int m){
  if (top!=null && *){
  //pop
    top=top.next ;
    //@acquire(m)
}}
```

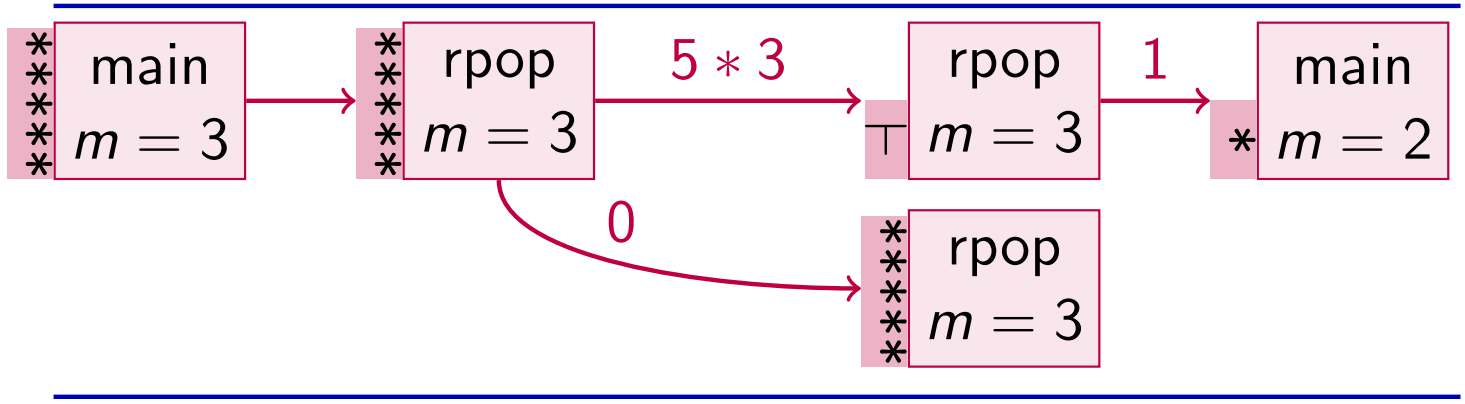
while

Classical Approach to Cost Analysis: working example



?



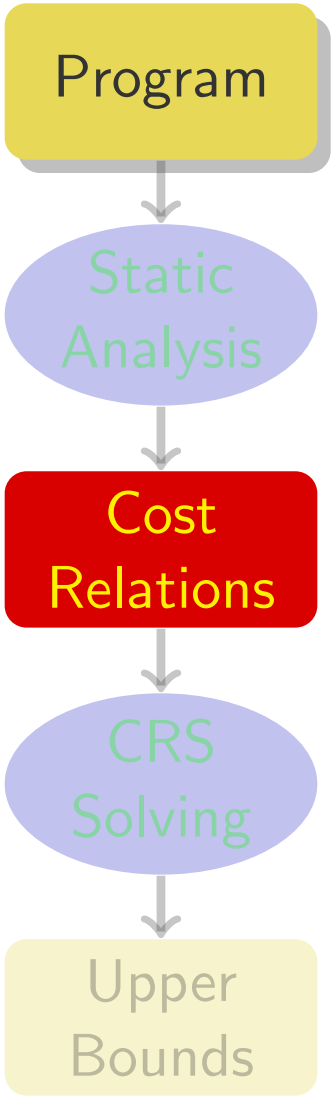



```
//@requires m >= 0
void main(int m) {
  while( m > 0 ) {
    rpop(m) ;
    top=new Node(*,top);
    //@acquire(1)
    m = m-1      ; }}}
```

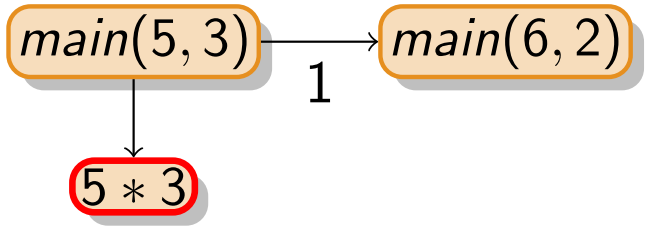
```
//@requires m >= 1
void rpop(int m){
  if (top!=null && *){
  //pop
    top=top.next ;
    //@acquire(m)
}}
```

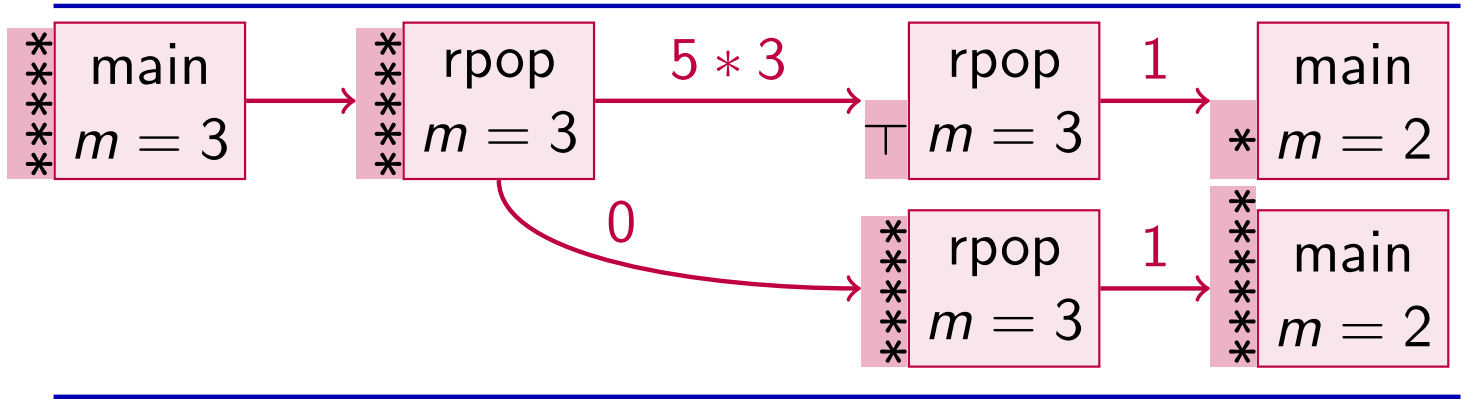
while

Classical Approach to Cost Analysis: working example



?



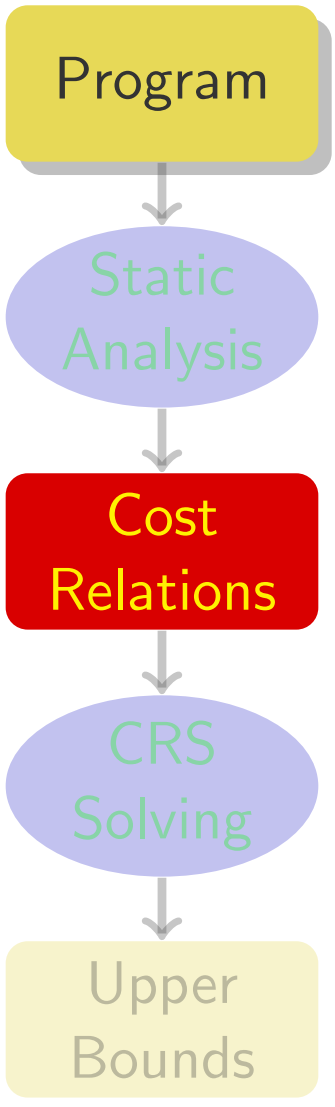



```
//@requires m >= 0
void main(int m) {
  while( m > 0 ) {
    rpop(m) ;
    top=new Node(*,top);
    //@acquire(1)
    m = m-1      ; }}}
```

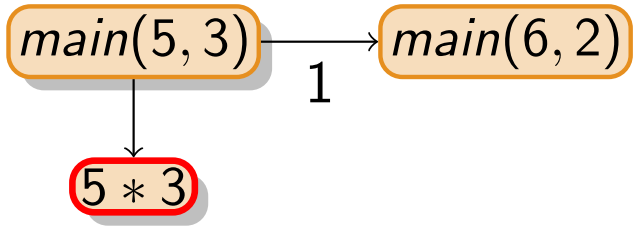
```
//@requires m >= 1
void rpop(int m){
  if (top!=null && *){
  //pop
    top=top.next ;
    //@acquire(m)
}}
```

while

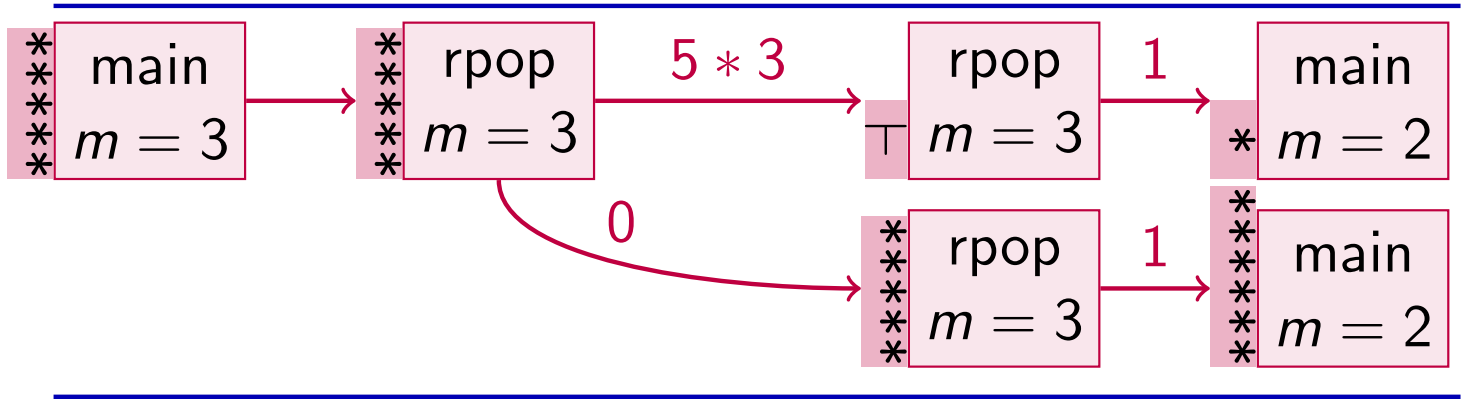
Classical Approach to Cost Analysis: working example



?



This CR evaluation matches no program execution

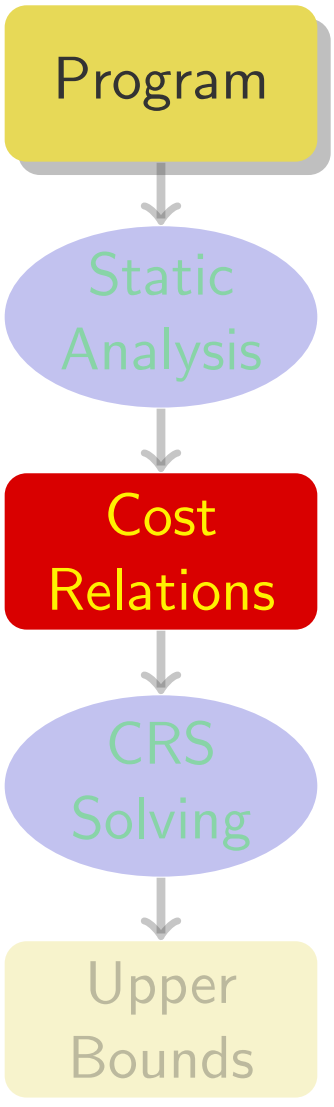



```
//@requires m >= 0
void main(int m) {
  while( m > 0 ) {
    rpop(m) ;
    top=new Node(*,top);
    //@acquire(1)
    m = m-1      ; }}}
```

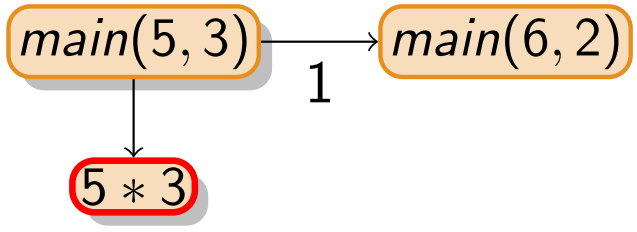
```
//@requires m >= 1
void rpop(int m){
  if (top!=null && *){
  //pop
    top=top.next ;
    //@acquire(m)
}}
```

while

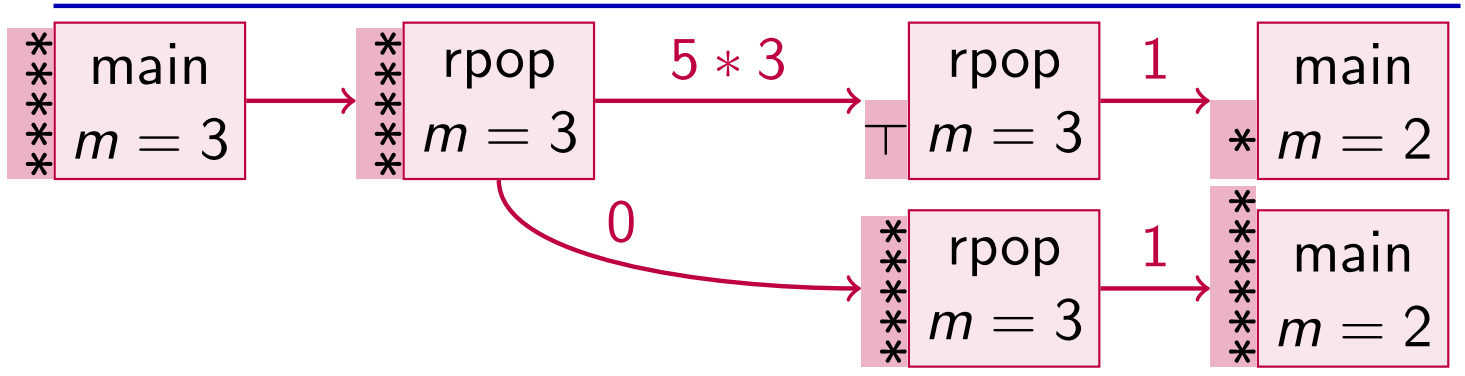
Classical Approach to Cost Analysis: working example



?



The cost of *rpop* depends on the value of its output




```
//@requires m >= 0
void main(int m) {
  while( m > 0 ) {
    rpop(m) ;
    top=new Node(*,top);
    //@acquire(1)
    m = m-1      ; }}}
```

```
//@requires m >= 1
void rpop(int m){
  if (top!=null && *){
  //pop
    top=top.next ;
    //@acquire(m)
}}
```

while

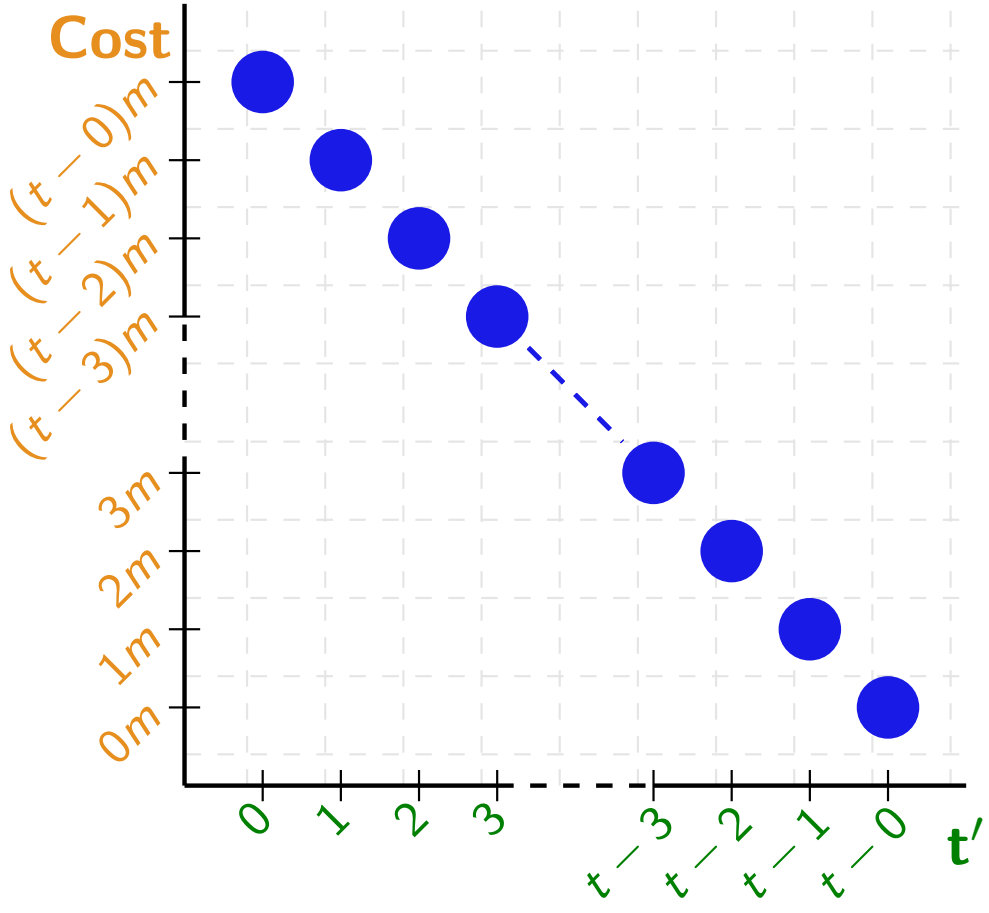
The output-cost correlation

The cost of *rpop* depends on the value of its output

```
//@requires m >= 1
void rpop(int m){
  if/while (top!=null && *){
    //pop
    top=top.next ;
    //@acquire(m)
}}
```

The output-cost correlation

The cost of *rpop* depends on the value of its output

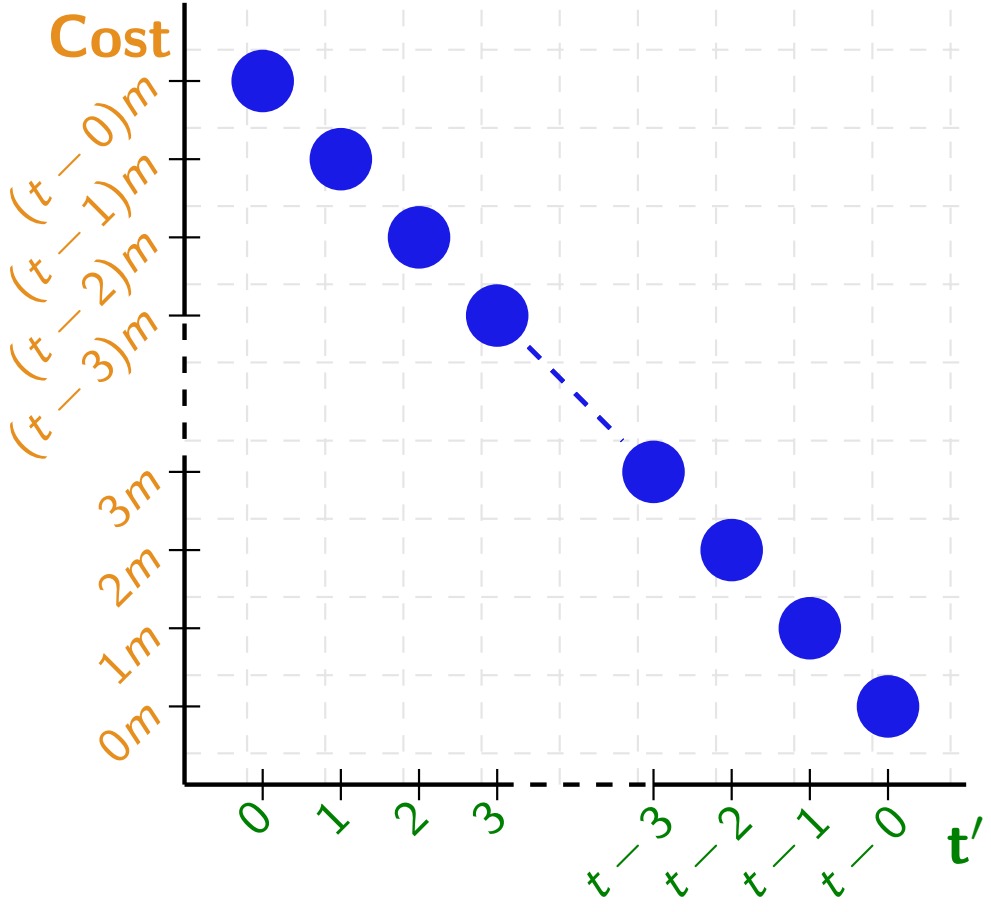



```
//@requires m >= 1
void rpop(int m){
  if/while (top!=null && *){
      //pop
      top=top.next ;
      //@acquire(m)
}}
```

The output-cost correlation

$$main(t, m) = 0 \qquad \{m = 0, t \geq 0\}$$

$$main(t, m) = \begin{matrix} rpop(t, m) + 1 + \\ main(t' + 1, m - 1) \end{matrix} \quad \left\{ \begin{matrix} m \geq 1, t \geq 0 \\ t \geq t' \geq 0 \end{matrix} \right\}$$



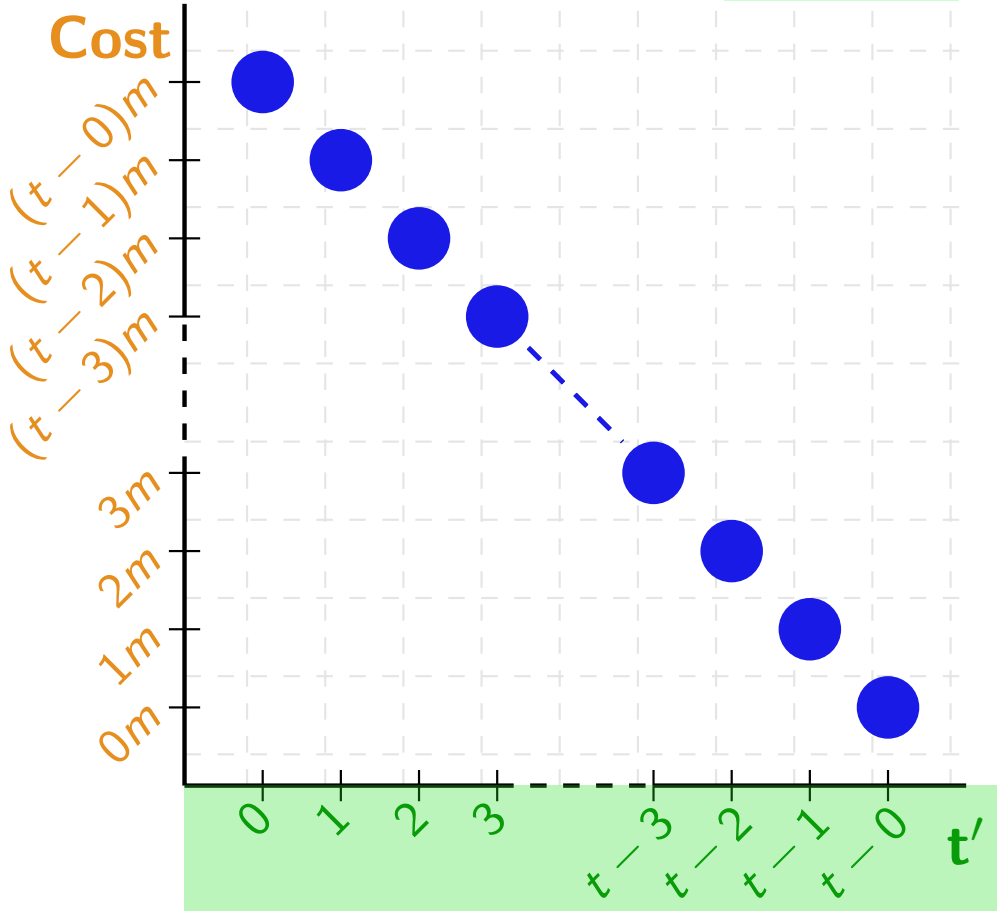
The cost of *rpop* depends on the value of its output

```
//@requires m >= 1
void rpop(int m){
  if/while (top!=null && *){
    //pop
    top=top.next ;
    //@acquire(m)
}}
```

The output-cost correlation

$$main(t,m) = 0 \quad \{m = 0, t \geq 0\}$$

$$main(t,m) = \begin{matrix} rpop(t,m) + 1 + \\ main(t' + 1, m - 1) \end{matrix} \quad \left\{ \begin{matrix} m \geq 1, t \geq 0 \\ t \geq t' \geq 0 \end{matrix} \right\}$$



The cost of *rpop* depends on the value of its output

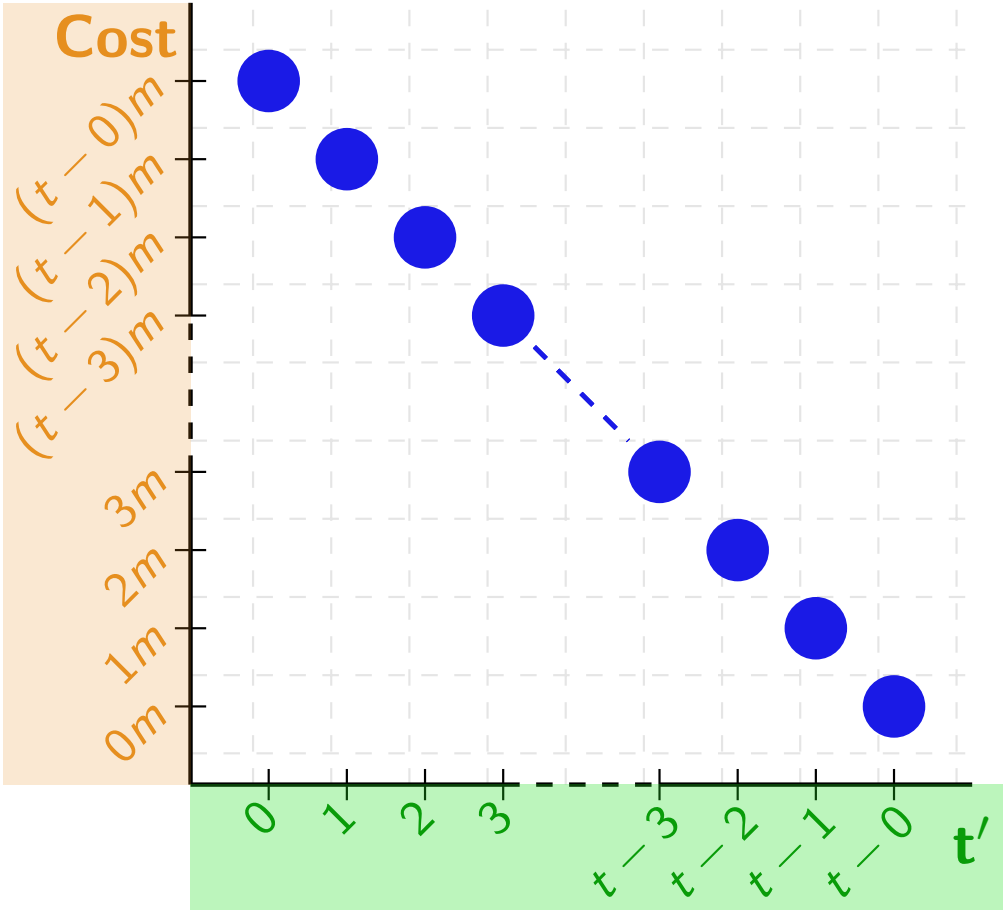
- $t \geq t' \geq 0$ is a precise postcondition

```
//@requires m >= 1
void rpop(int m){
  if/while (top!=null && *){
      //pop
      top=top.next ;
      //@acquire(m)
}}
```

The output-cost correlation

$$main(t,m) = 0 \quad \{m = 0, t \geq 0\}$$

$$main(t,m) = \begin{matrix} rpop(t,m) + 1 + \\ main(t'+1, m-1) \end{matrix} \quad \left\{ \begin{matrix} m \geq 1, t \geq 0 \\ t \geq t' \geq 0 \end{matrix} \right\}$$



The cost of *rpop* depends on the value of its output

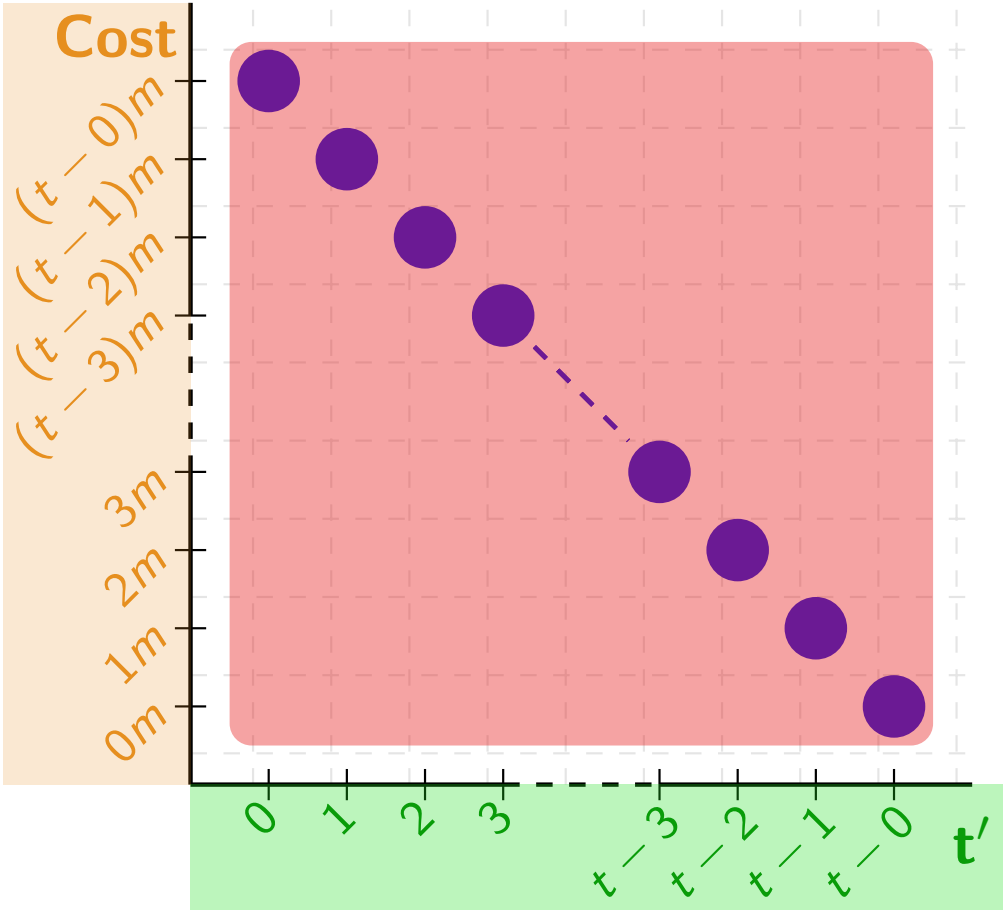
- $t \geq t' \geq 0$ is a precise postcondition
- $0 \leq c \leq t * m$ is a precise cost relation

```
//@requires m >= 1
void rpop(int m){
  if/while (top!=null && *){
    //pop
    top=top.next ;
    //@acquire(m)
}}
```

The output-cost correlation

$$main(t,m) = 0 \quad \{m = 0, t \geq 0\}$$

$$main(t,m) = \begin{matrix} rpop(t,m) + 1 + \\ main(t'+1, m-1) \end{matrix} \quad \left\{ \begin{matrix} m \geq 1, t \geq 0 \\ t \geq t' \geq 0 \end{matrix} \right\}$$



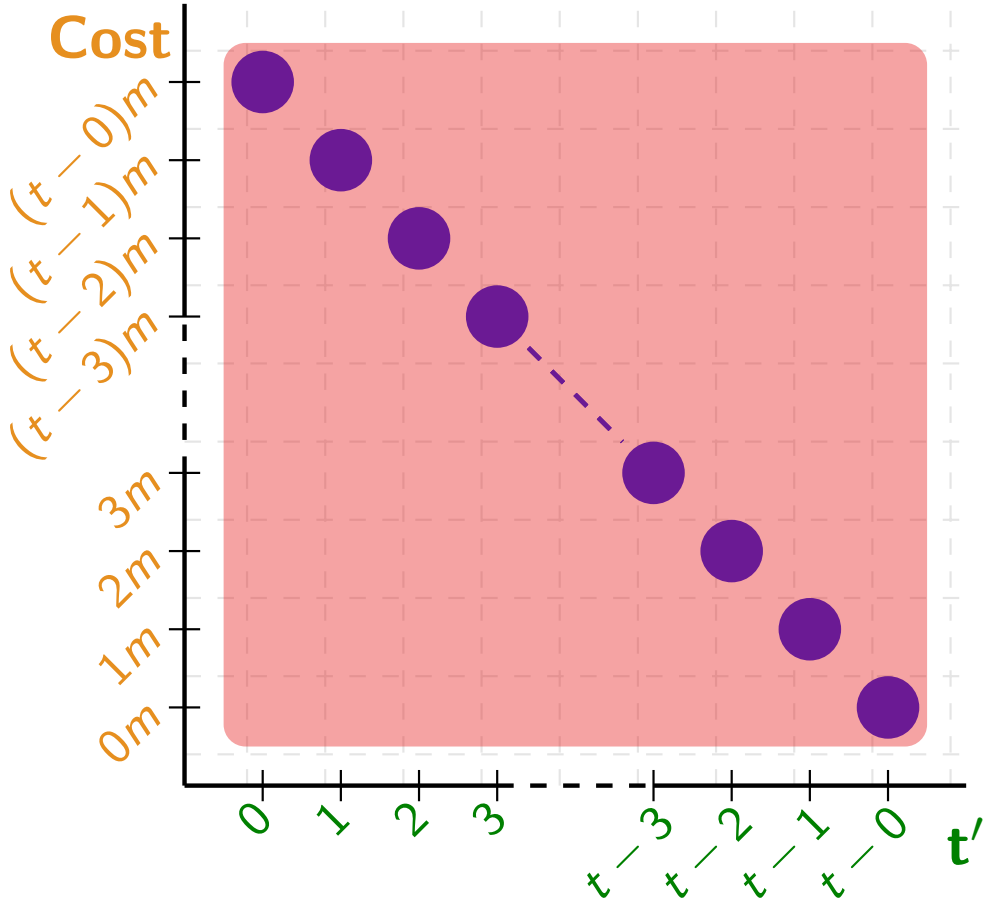
The cost of *rpop* depends on the value of its output

- $t \geq t' \geq 0$ is a precise postcondition
- $0 \leq c \leq t * m$ is a precise cost relation
- separating them introduces spurious evaluations

```
//@requires m >= 1
void rpop(int m){
  if/while (top!=null && *){
    //pop
    top=top.next ;
    //@acquire(m)
}}
```

The output-cost correlation

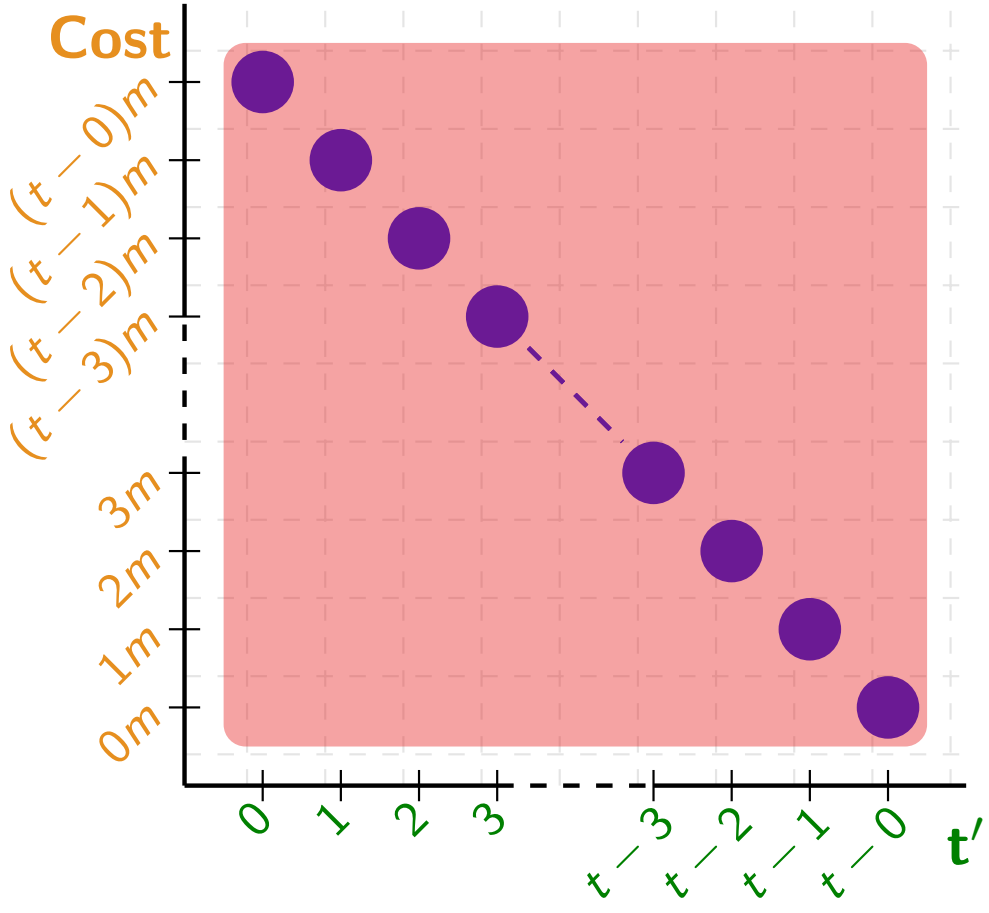
The cost of *rpop* depends on the value of its output




```
//@requires m >= 1
void rpop(int m){
  if/while (top!=null && *){
    //pop
    top=top.next ;
    //@acquire(m)
}}
```

The output-cost correlation

$$rpop^u(t, m) = m * t$$



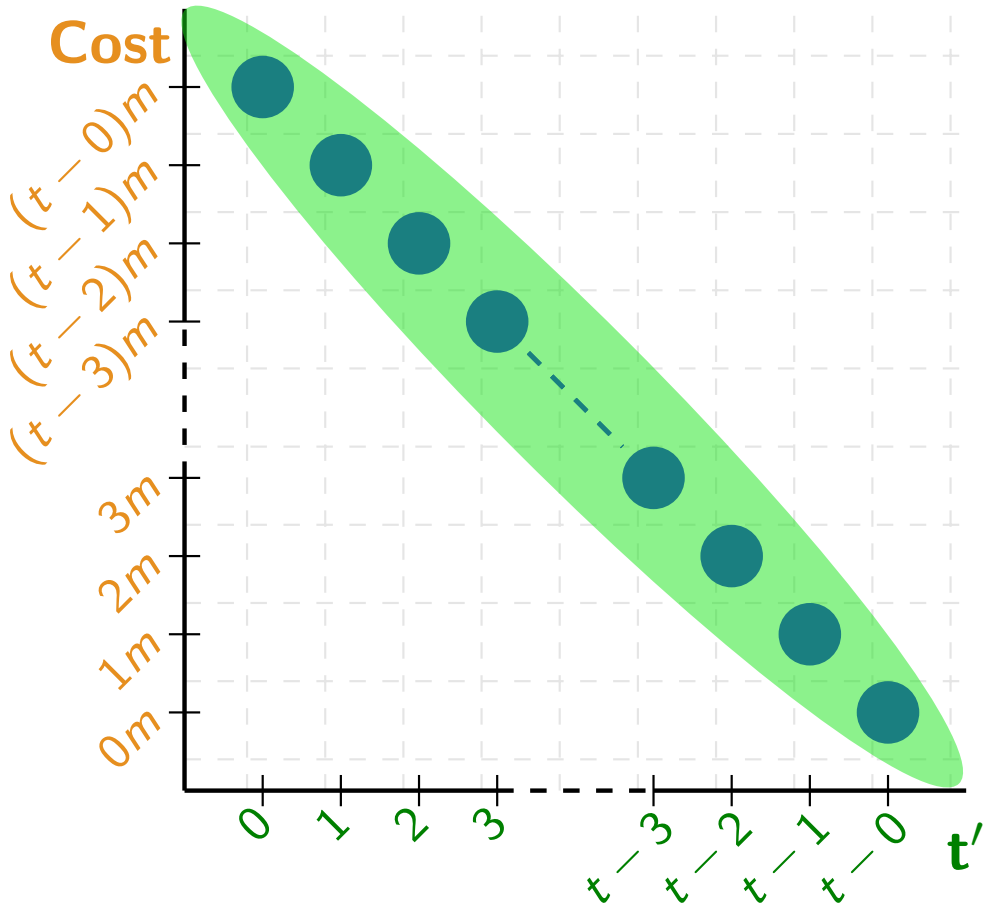
The cost of *rpop* depends on the value of its output

- $c \leq m * t$ is best cost UB that takes *rpop* inputs

```
//@requires m >= 1
void rpop(int m){
  if/while (top!=null && *){
      //pop
      top=top.next ;
      //@acquire(m)
}}
```

The output-cost correlation

$$rpop^u(t, m, t') = m(t - t')$$



The cost of *rpop* depends on the value of its output

- $c \leq m * t$ is best cost UB that takes *rpop* inputs
- $c = m * (t - t')$ is an exact cost that takes both *rpop* inputs **and outputs**

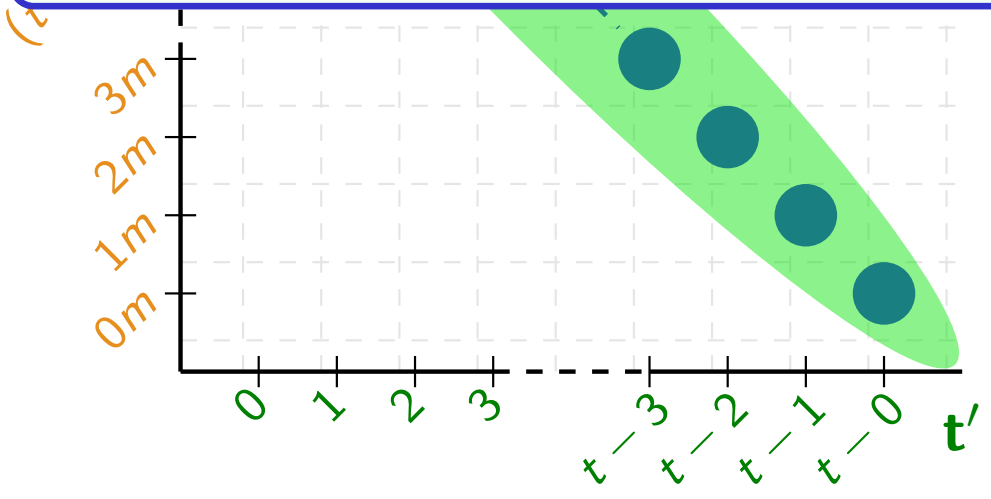
```
//@requires m >= 1
void rpop(int m){
  if/while (top!=null && *){
    //pop
    top=top.next ;
    //@acquire(m)
}}
```

The output-cost correlation

$$rpop^u(t, m, t') = m(t - t')$$

Net Cost Upper Bound

Every *rpop* execution that starts with s, m and terminates at t' consumes at most $m * (t - t')$ resources



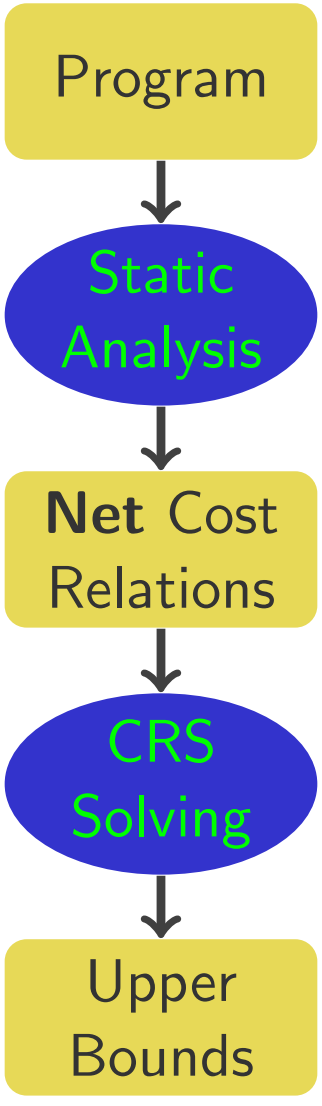
The cost of *rpop* depends on the value of its output

$c \leq m * t$ is best cost UB that takes *rpop* inputs

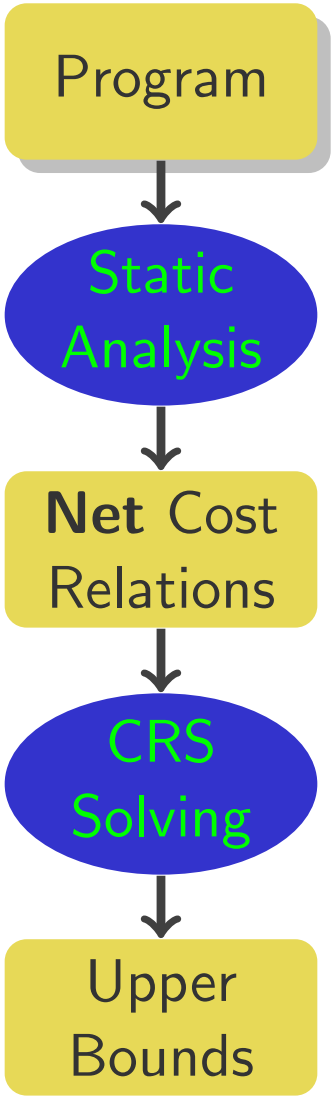
$c = m * (t - t')$ is an exact cost that takes both *rpop* inputs **and outputs**

```
//@requires m >= 1
void rpop(int m){
  if/while (top!=null && *){
    //pop
    top=top.next ;
    //@acquire(m)
}}
```

Classical Approach to Cost Analysis: repaired example



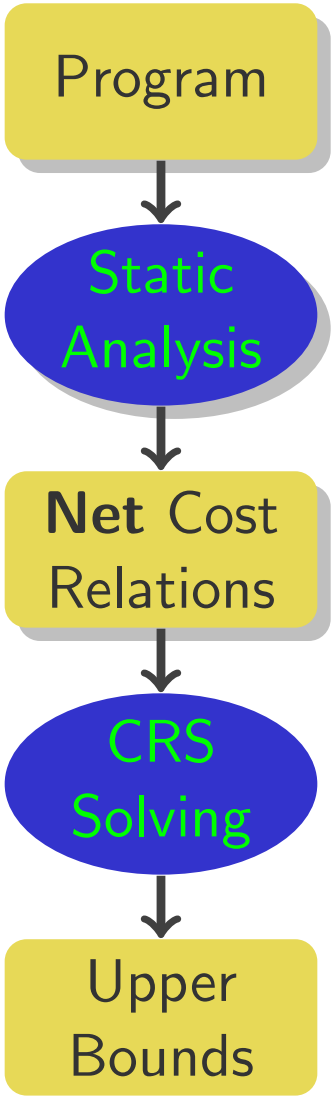
Classical Approach to Cost Analysis: repaired example




```
//@requires m >= 1
void rpop(int m){
  if (top != null && *){
    //pop
    top=top.next ;
    //@acquire(m)
}}
```

```
//@requires m >= 0
void main(int m) {
  while( m > 0 ) {
    rpop(m)  ;
    // push(*);
    top = new
     Node(*,top);
    //@acquire(1)
    m = m-1      ;
}}}
```

Classical Approach to Cost Analysis: repaired example

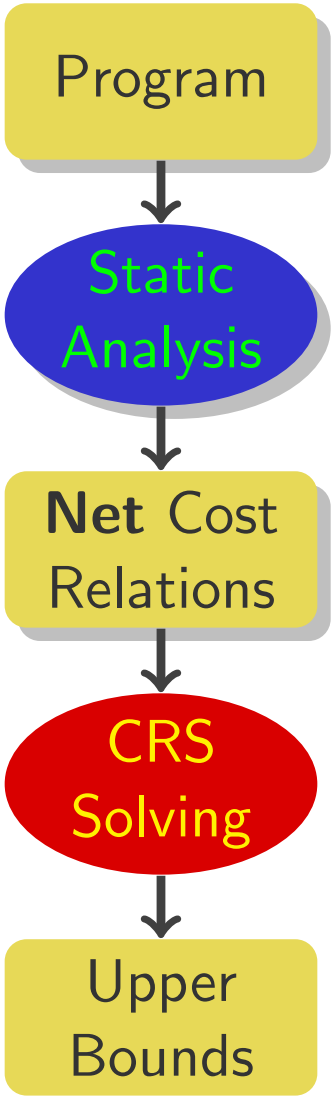



```
//@requires m >= 1
void rpop(int m){
  if (top != null && *){
    //pop
    top=top.next ;
    //@acquire(m)
}}
```

```
//@requires m >= 0
void main(int m) {
  while( m > 0 ) {
    rpop(m)  ;
    // push(*);
    top = new
     Node(*,top);
    //@acquire(1)
    m = m-1     ;
}}}
```

$$
\begin{aligned}
main(t, m, t') &= 0 && \{m = 0, t \geq 0, t' = t\} \\
main(t, m, t_1) &= rpop(t, m, t') + 1 + main(t' + 1, m - 1, t_1) && \left\{\begin{matrix} m \geq 1, t \geq 0 \\ t \geq t' \geq 0 \end{matrix}\right\} \\
rpop(t, m, t') &= 0 && \{m \geq 1, t \geq 0, t' = t\} \\
rpop(t, m, t') &= m + rpop(t - 1, m, t') && \{m \geq 1, t \geq 1\}
\end{aligned}
$$

Classical Approach to Cost Analysis: repaired example




```
//@requires m >= 1
void rpop(int m){
  if (top != null && *){
    //pop
    top=top.next ;
    //@acquire(m)
}}
```

```
//@requires m >= 0
void main(int m) {
  while( m > 0 ) {
    rpop(m)  ;
    // push(*);
    top = new
     Node(*,top);
```

Net Cost Upper Bound

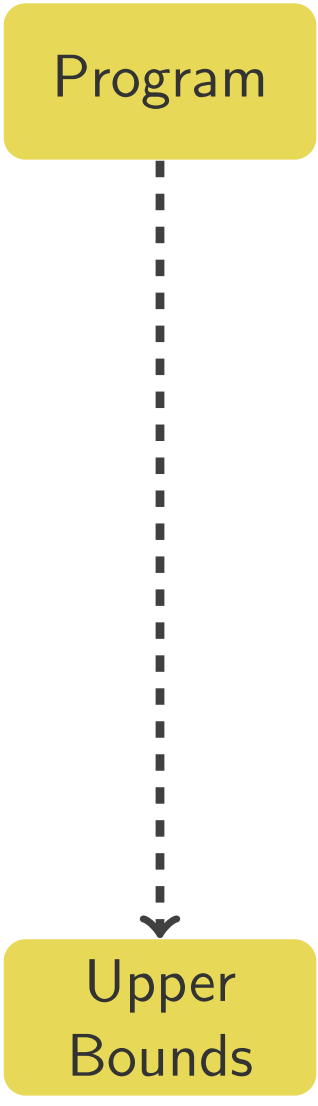
Existing CR Solvers do not obtain precise bounds for net cost equations

$$main(t, m, t_1) = \begin{matrix} rpop(t, m, t') + 1 + \\ main(t' + 1, m - 1, t_1) \end{matrix} \quad \begin{Bmatrix} m \geq 1, t \geq 0 \\ t \geq t' \geq 0 \end{Bmatrix}$$

$$rpop(t, m, t') = 0 \quad \{m \geq 1, t \geq 0, t' = t\}$$

$$rpop(t, m, t') = m + rpop(t - 1, m, t') \quad \{m \geq 1, t \geq 1\}$$

New Approach to Cost Analysis

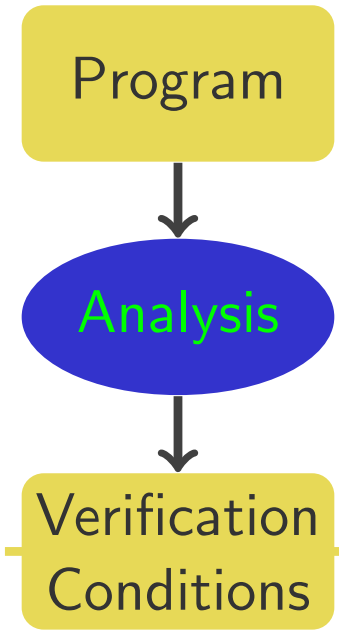


New Approach to Cost Analysis

Our approach uses the **Verification Conditions** to split cost analysis in two independent phases

New Approach to Cost Analysis

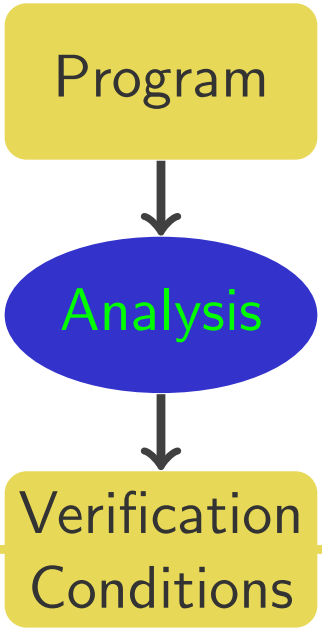
New Approach to Cost Analysis



An **analysis** abstracts the program semantics into a system of inductive **Verification Conditions**

Upper Bounds

New Approach to Cost Analysis

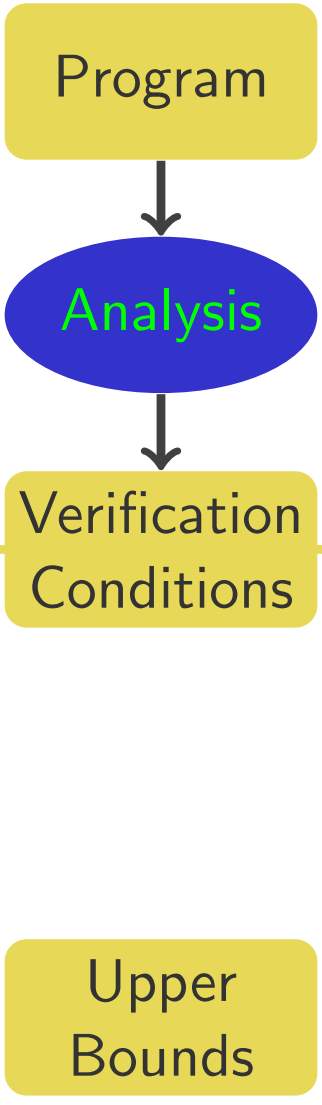


An **analysis** abstracts the program semantics into a system of inductive **Verification Conditions**

Verification Conditions describe properties a function must hold to be a valid net cost bound

Upper Bounds

New Approach to Cost Analysis

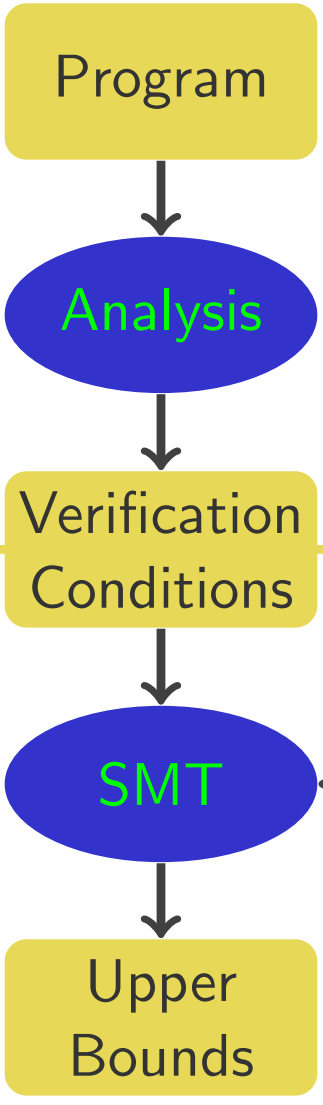


An **analysis** abstracts the program semantics into a system of inductive **Verification Conditions**

Verification Conditions describe properties a function must hold to be a valid net cost bound

Verification Conditions are formulas in First Order logic, universally quantified over the program variables

New Approach to Cost Analysis



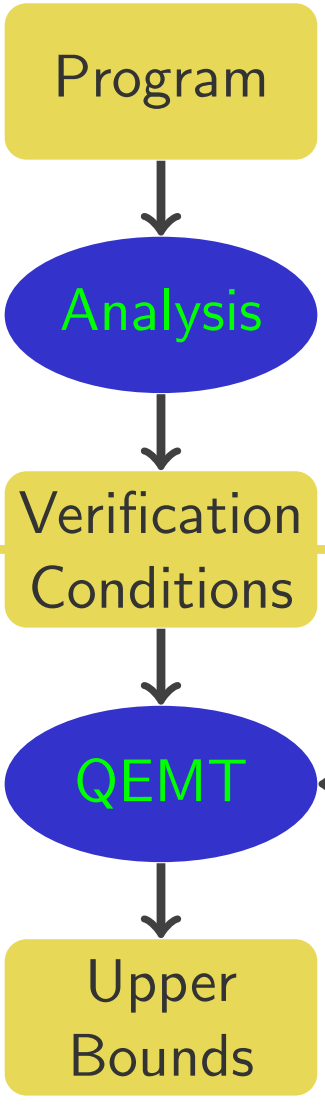
An **analysis** abstracts the program semantics into a system of inductive **Verification Conditions**

Verification Conditions describe properties a function must hold to be a valid net cost bound

Verification Conditions are formulas in First Order logic, universally quantified over the program variables

We can **Verify** a set of candidate Upper Bounds by checking the **Validity Modulo Theory** of the VCs

New Approach to Cost Analysis



An **analysis** abstracts the program semantics into a system of inductive **Verification Conditions**

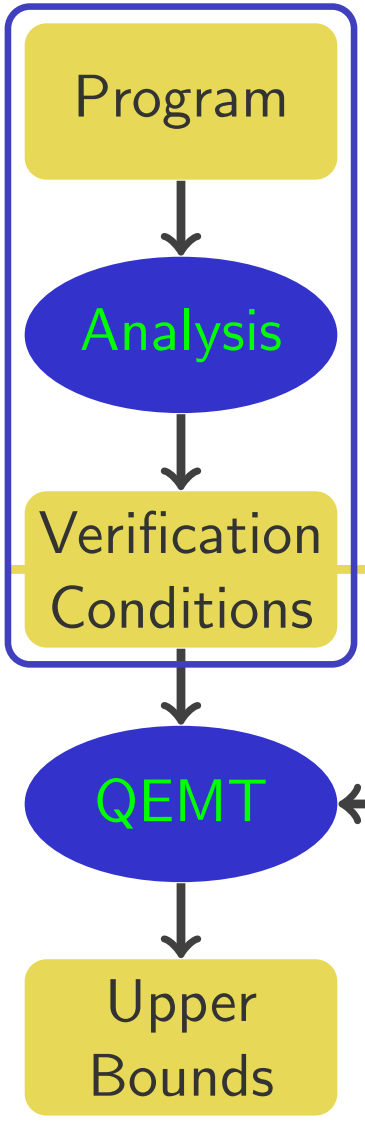
Verification Conditions describe properties a function must hold to be a valid net cost bound

Verification Conditions are formulas in First Order logic, universally quantified over the program variables

Template Upper Bound

With some template UBs, we can **synthesise** concrete bounds using **Quantifier Elimination** modulo Theory

New Approach to Cost Analysis



An **analysis** abstracts the program semantics into a system of inductive **Verification Conditions**

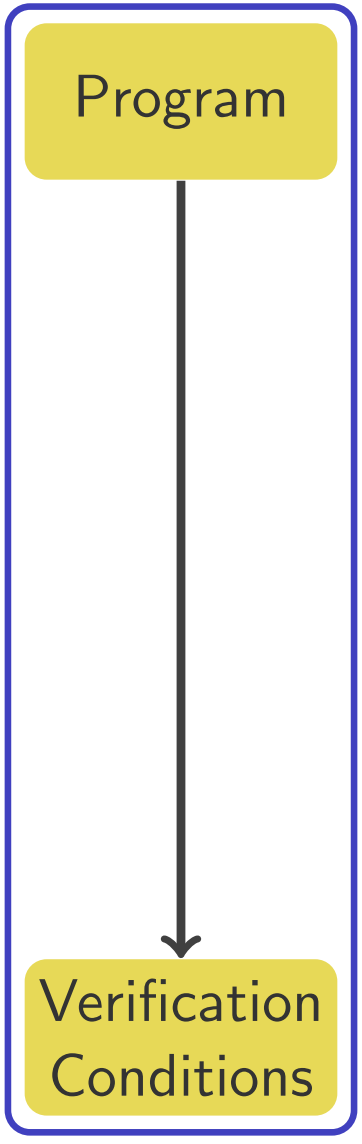
Verification Conditions describe properties a function must hold to be a valid net cost bound

Verification Conditions are formulas in First Order logic, universally quantified over the program variables

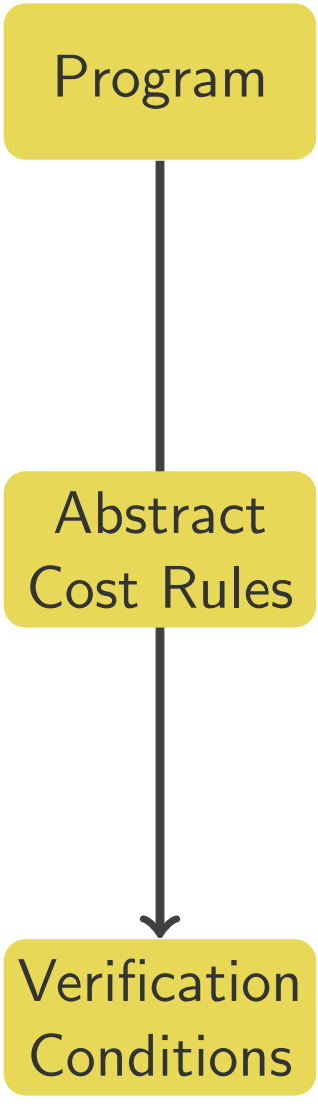
Template Upper Bound

With some template UBs, we can **synthesise** concrete bounds using **Quantifier Elimination** modulo Theory

New Approach to Cost Analysis

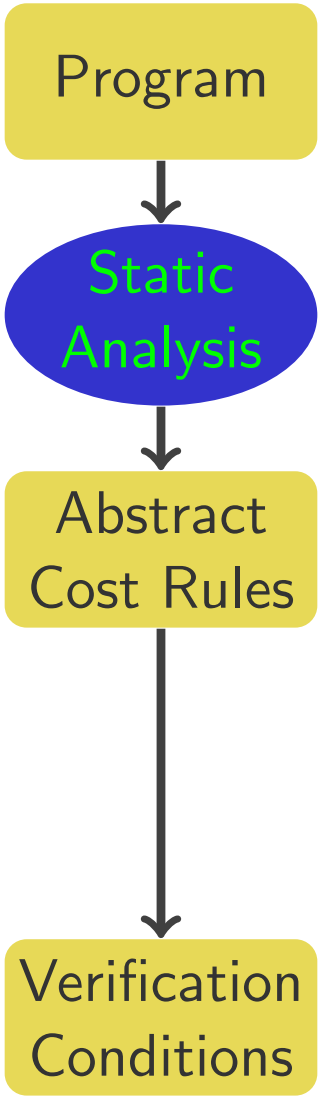


New Approach to Cost Analysis



We use **Abstract Cost Rules** as an intermediate abstraction of the program

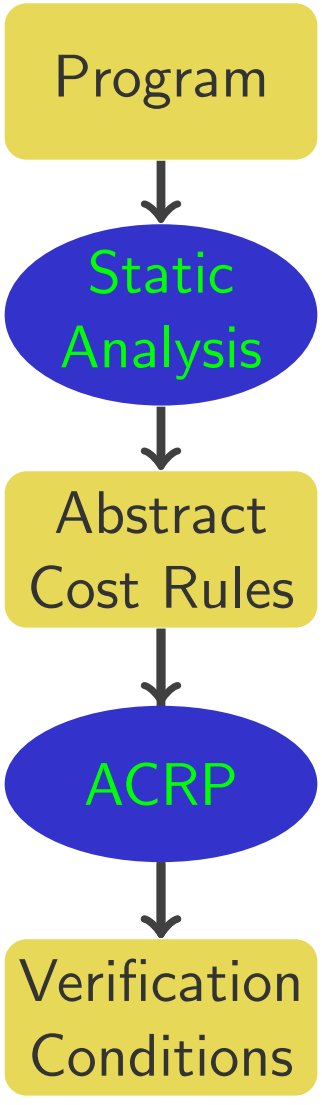
New Approach to Cost Analysis



The existing static analysis generates, **Abstract Cost Rules** as an intermediate representation

We use **Abstract Cost Rules** as an intermediate abstraction of the program

New Approach to Cost Analysis

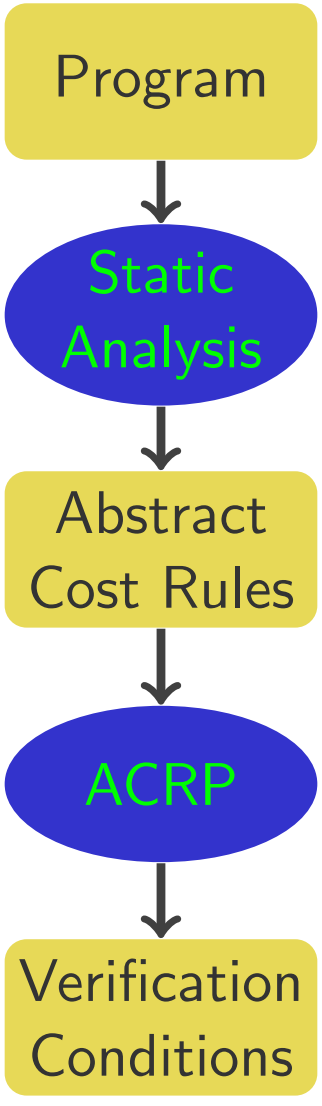


The existing static analysis generates, **Abstract Cost Rules** as an intermediate representation

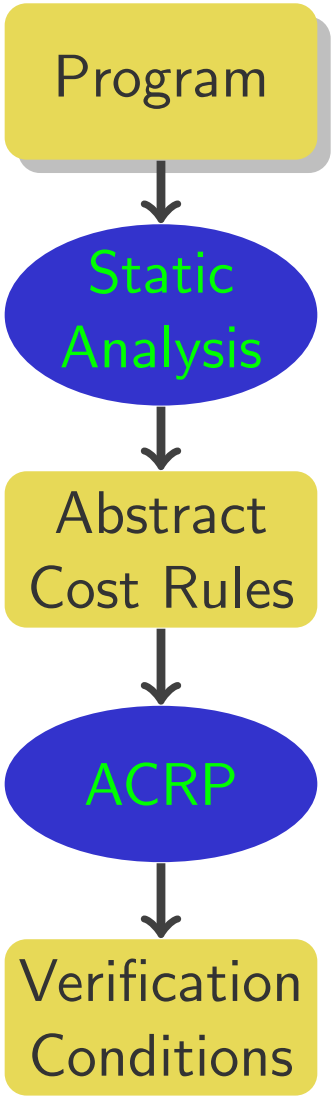
We use **Abstract Cost Rules** as an intermediate abstraction of the program

An Abstract Cost Rules Processor **(ACRP)** transforms each ACR into a VC

New Approach to Cost Analysis



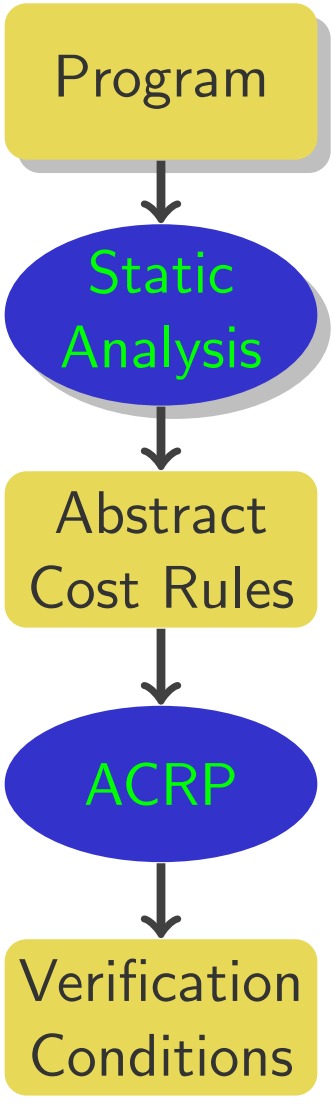
New Approach to Cost Analysis




```
//@requires m >= 0
void main(int m) {
  while( m > 0 ) {
    rpop(m)  ;
    // push(*);
    top = new
     Node(*,top);
    //@acquire(1)
    m = m-1      ;
}}}
```

```
//@requires m >= 1
void rpop(int m){
  while(top!=null && *){
    //pop
    top=top.next ;
    //@acquire(m)
}}
```

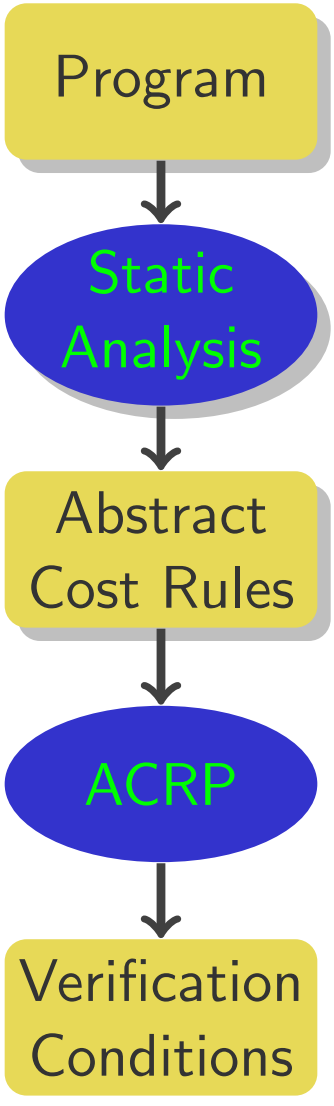
New Approach to Cost Analysis




```
//@requires m >= 0
void main(int m) {
  while( m > 0 ) {
    rpop(m)  ;
    // push(*);
    top = new
     Node(*,top);
    //@acquire(1)
    m = m-1      ;
}}}
```

```
//@requires m >= 1
void rpop(int m){
  while(top!=null && *){
    //pop
    top=top.next ;
    //@acquire(m)
}}
```

New Approach to Cost Analysis



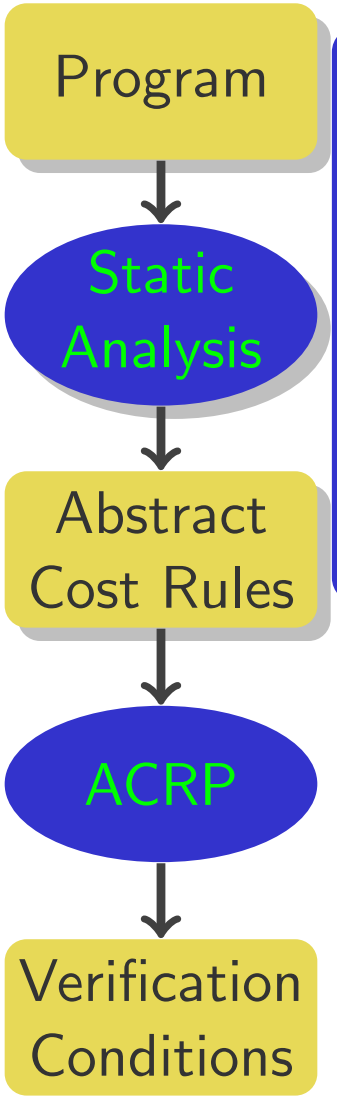

```
//@requires m >= 0
void main(int m) {
  while( m > 0 ) {
    rpop(m)  ;
    // push(*);
    top = new
     Node(*,top);
    //@acquire(1)
    m = m-1      ;
}}}
```

```
//@requires m >= 1
void rpop(int m){
  while(top!=null && *){
    //pop
    top=top.next ;
    //@acquire(m)
}}
```

$$main([t, 0], [t]) \leftarrow$$
$$\quad t \geq 0.$$
$$main([t, m], [t_1]) \leftarrow$$
$$\quad t \geq 0,$$
$$\quad m \geq 1,$$
$$\quad rpop([t, m], [t']),$$
$$\quad \mathrm{acq}(1),$$
$$\quad main([t' + 1, m - 1], [t_1]).$$

$$rpop([t, m], [t]) \leftarrow$$
$$\quad t \geq 0,$$
$$\quad m \geq 1.$$
$$rpop([t, m], [t']) \leftarrow$$
$$\quad t \geq 1,$$
$$\quad m \geq 1,$$
$$\quad \mathrm{acq}(m),$$
$$\quad rpop([t - 1, m], [t']).$$

New Approach to Cost Analysis



Abstract Cost Rules

An ACR has the form $p(\bar{x}|\bar{y}) \longleftarrow b_1, \ldots b_n$.

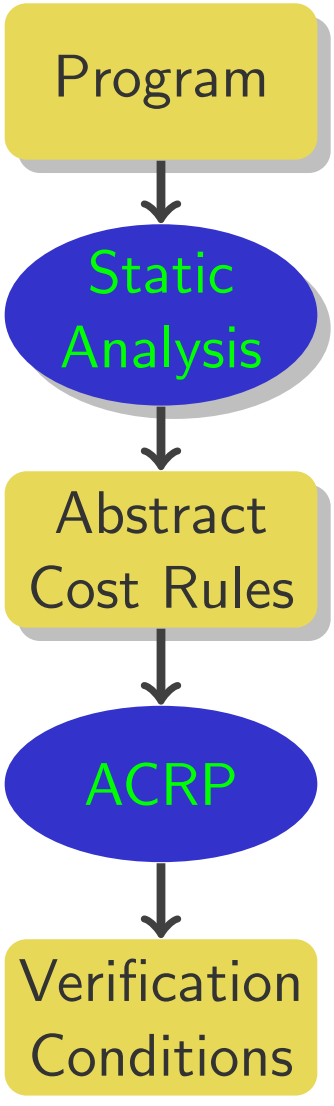
There are three kinds of abstract codes b_i

Constraint	φ	Operations & guards
Cost	$\mathtt{acq}(e)$	Resource consumption
Calls	$q(\bar{x}|\bar{z})$	Control Flow

$$
\begin{array}{l}
main([t,0],[t]) \leftarrow \\
\quad t \geq 0. \\
main([t,m],[t_1]) \leftarrow \\
\quad t \geq 0, \\
\quad m \geq 1, \\
\quad rpop([t,m],[t']), \\
\quad \mathtt{acq}(1), \\
\quad main([t'+1, m-1],[t_1]).
\end{array}
$$

$$
\begin{array}{l}
rpop([t,m],[t]) \leftarrow \\
\quad t \geq 0, \\
\quad m \geq 1. \\
rpop([t,m],[t']) \leftarrow \\
\quad t \geq 1, \\
\quad m \geq 1, \\
\quad \mathtt{acq}(m), \\
\quad rpop([t-1,m],[t']).
\end{array}
$$

New Approach to Cost Analysis



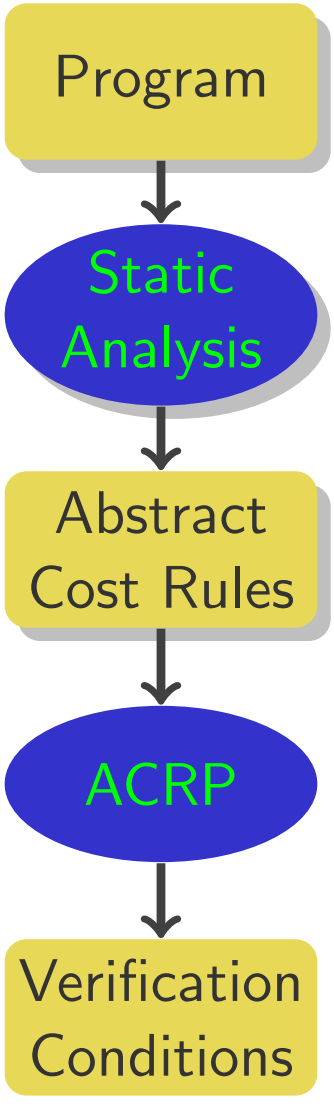

```
//@requires m >= 0
void main(int m) {
  while( m > 0 ) {
    rpop(m)  ;
    // push(*);
    top = new
     Node(*,top);
    //@acquire(1)
    m = m-1      ;
}}}
```

```
//@requires m >= 1
void rpop(int m){
  while(top!=null && *){
    //pop
    top=top.next ;
    //@acquire(m)
}}
```

$$main([t, 0], [t]) \leftarrow$$
$$\quad t \geq 0.$$
$$main([t, m], [t_1]) \leftarrow$$
$$\quad t \geq 0,$$
$$\quad m \geq 1,$$
$$\quad rpop([t, m], [t']),$$
$$\quad \mathrm{acq}(1),$$
$$\quad main([t' + 1, m - 1], [t_1]).$$

$$rpop([t, m], [t]) \leftarrow$$
$$\quad t \geq 0,$$
$$\quad m \geq 1.$$
$$rpop([t, m], [t']) \leftarrow$$
$$\quad t \geq 1,$$
$$\quad m \geq 1,$$
$$\quad \mathrm{acq}(m),$$
$$\quad rpop([t - 1, m], [t']).$$

New Approach to Cost Analysis



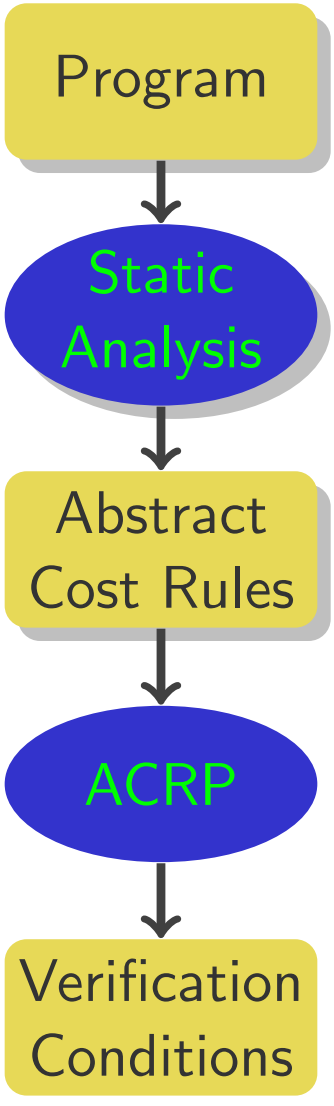

```
//@requires m >= 0
void main(int m) {
  while( m > 0 ) {
    rpop(m)  ;
    // push(*);
    top = new
     Node(*,top);
    //@acquire(1)
    m = m-1      ;
}}}
```

```
//@requires m >= 1
void rpop(int m){
  while(top!=null && *){
    //pop
    top=top.next ;
    //@acquire(m)
}}
```

$main([t, 0], [t]) \leftarrow$
$\quad t \geq 0.$
$main([t, m], [t_1]) \leftarrow$
$\quad t \geq 0,$
$\quad m \geq 1,$
$\quad rpop([t, m], [t']),$
$\quad \mathrm{acq}(1),$
$\quad main([t' + 1, m - 1], [t_1]).$

$rpop([t, m], [t]) \leftarrow$
$\quad t \geq 0,$
$\quad m \geq 1.$
$rpop([t, m], [t']) \leftarrow$
$\quad t \geq 1,$
$\quad m \geq 1,$
$\quad \mathrm{acq}(m),$
$\quad rpop([t - 1, m], [t']).$

New Approach to Cost Analysis



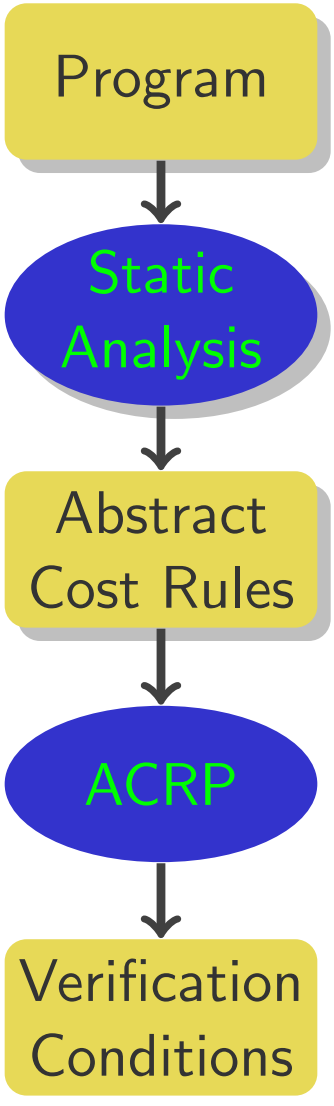

```
//@requires m >= 0
void main(int m) {
  while( m > 0 ) {
    rpop(m)  ;
    // push(*);
    top = new
     Node(*,top);
    //@acquire(1)
    m = m-1      ;
}}}
```

```
//@requires m >= 1
void rpop(int m){
  while(top!=null && *){
    //pop
    top=top.next ;
    //@acquire(m)
}}
```

$$main([t, 0], [t]) \leftarrow$$
$$\quad t \geq 0.$$
$$main([t, m], [t_1]) \leftarrow$$
$$\quad t \geq 0,$$
$$\quad m \geq 1,$$
$$\quad rpop([t, m], [t']),$$
$$\quad \mathrm{acq}(1),$$
$$\quad main([t' + 1, m - 1], [t_1]).$$

$$rpop([t, m], [t]) \leftarrow$$
$$\quad t \geq 0,$$
$$\quad m \geq 1.$$
$$rpop([t, m], [t']) \leftarrow$$
$$\quad t \geq 1,$$
$$\quad m \geq 1,$$
$$\quad \mathrm{acq}(m),$$
$$\quad rpop([t - 1, m], [t']).$$

New Approach to Cost Analysis



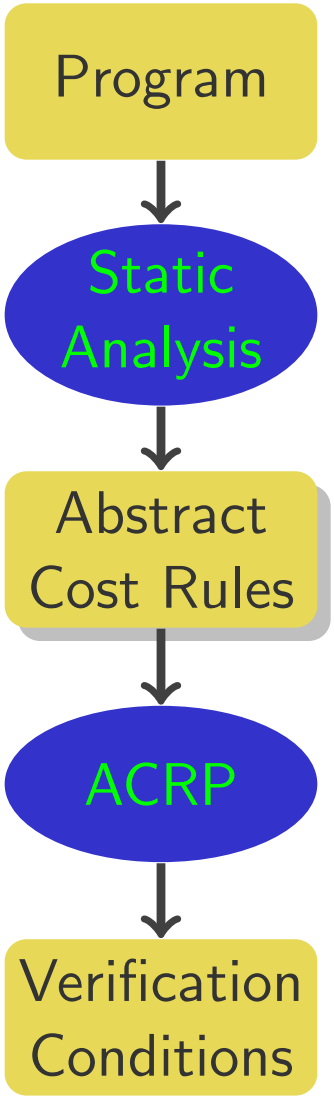

```
//@requires m >= 0
void main(int m) {
  while( m > 0 ) {
    rpop(m)  ;
    // push(*);
    top = new
     Node(*,top);
    //@acquire(1)
    m = m-1      ;
}}}
```

```
//@requires m >= 1
void rpop(int m){
  while(top!=null && *){
    //pop
    top=top.next ;
    //@acquire(m)
}}
```

$$
\begin{aligned}
&main([t, 0], [t]) \leftarrow \\
&\quad t \geq 0. \\
&main([t, m], [t_1]) \leftarrow \\
&\quad t \geq 0, \\
&\quad m \geq 1, \\
&\quad rpop([t, m], [t']), \\
&\quad \boxed{acq(1),} \\
&\quad main([t' + 1, m - 1], [t_1]).
\end{aligned}
$$

$$
\begin{aligned}
&rpop([t, m], [t]) \leftarrow \\
&\quad t \geq 0, \\
&\quad m \geq 1. \\
&rpop([t, m], [t']) \leftarrow \\
&\quad t \geq 1, \\
&\quad m \geq 1, \\
&\quad \boxed{acq(m),} \\
&\quad rpop([t - 1, m], [t']).
\end{aligned}
$$

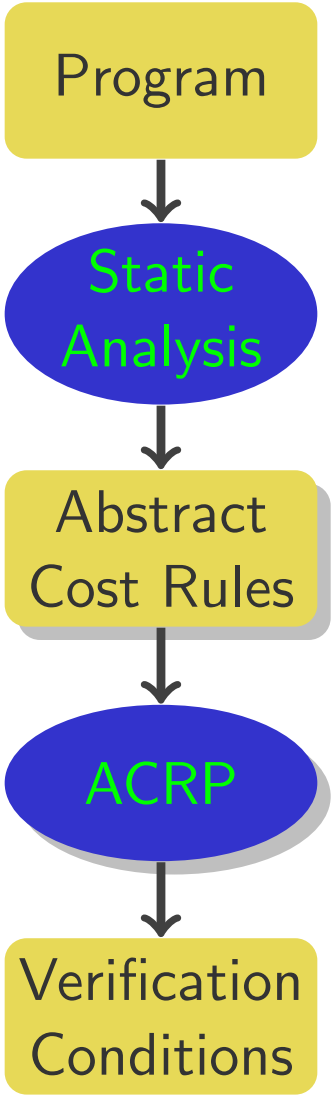
New Approach to Cost Analysis



$$main([t, 0], [t]) \leftarrow$$
$$\quad t \geq 0.$$
$$main([t, m], [t_1]) \leftarrow$$
$$\quad t \geq 0,$$
$$\quad m \geq 1,$$
$$\quad rpop([t, m], [t']),$$
$$\quad \mathtt{acq}(1),$$
$$\quad main([t' + 1, m - 1], [t_1]).$$

$$rpop([t, m], [t]) \leftarrow$$
$$\quad t \geq 0,$$
$$\quad m \geq 1.$$
$$rpop([t, m], [t']) \leftarrow$$
$$\quad t \geq 1,$$
$$\quad m \geq 1,$$
$$\quad \mathtt{acq}(m),$$
$$\quad rpop([t - 1, m], [t']).$$

New Approach to Cost Analysis

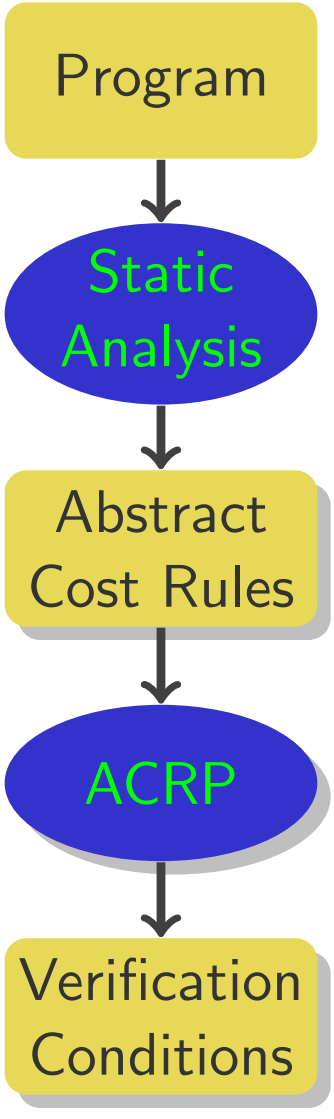


$$\frac{\forall * . \wedge_{i=1}^{n} \varphi_i \rightarrow \tilde{f}_p(\bar{x}, \bar{y}) \geq \sum_{i=1}^{n} \tilde{b}_i}{p(\bar{x}|\bar{y}) \longleftarrow b_1, \ldots b_n}$$

$main([t, 0], [t]) \leftarrow$
$\quad t \geq 0.$
$main([t, m], [t_1]) \leftarrow$
$\quad t \geq 0,$
$\quad m \geq 1,$
$\quad rpop([t, m], [t']),$
$\quad \mathrm{acq}(1),$
$\quad main([t' + 1, m - 1], [t_1]).$

$rpop([t, m], [t]) \leftarrow$
$\quad t \geq 0,$
$\quad m \geq 1.$
$rpop([t, m], [t']) \leftarrow$
$\quad t \geq 1,$
$\quad m \geq 1,$
$\quad \mathrm{acq}(m),$
$\quad rpop([t - 1, m], [t']).$

New Approach to Cost Analysis



$$\forall * .\ t \geq 0 \wedge m = 0 \Rightarrow \tilde{f}_{main}(t, m, t) \geq 0$$
$$\forall * .\ t \geq 0 \wedge m \geq 1 \Rightarrow \tilde{f}_{main}(t, m, t_1) \geq \tilde{f}_{rpop}(t, m, t') + 1 + \tilde{f}_{main}(t' + 1, m - 1, t_1)$$

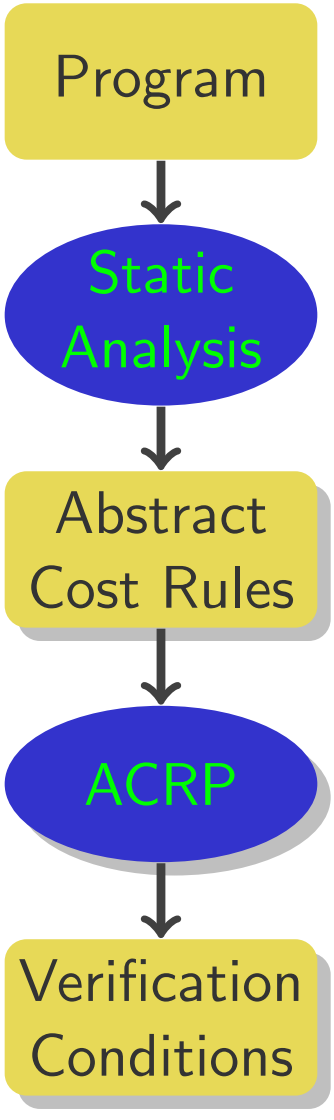
$$\forall * .\ t \geq 0 \wedge m \geq 1 \Rightarrow \tilde{f}_{rpop}(t, m, t) \geq 0$$
$$\forall * .\ t \geq 1 \wedge m \geq 1 \Rightarrow \tilde{f}_{rpop}(t, m, t') \geq m + \tilde{f}_{rpop}(t - 1, m, t')$$

$$\frac{\forall * .\ \wedge_{i=1}^{n} \varphi_i \rightarrow \tilde{f}_p(\bar{x}, \bar{y}) \geq \sum_{i=1}^{n} \tilde{b}_i}{p(\bar{x}|\bar{y}) \longleftarrow b_1, \ldots b_n}$$

$main([t, 0], [t]) \leftarrow$
  $t \geq 0.$
$main([t, m], [t_1]) \leftarrow$
  $t \geq 0,$
  $m \geq 1,$
  $rpop([t, m], [t']),$
  $\mathrm{acq}(1),$
  $main([t' + 1, m - 1], [t_1]).$

$rpop([t, m], [t]) \leftarrow$
  $t \geq 0,$
  $m \geq 1.$
$rpop([t, m], [t']) \leftarrow$
  $t \geq 1,$
  $m \geq 1,$
  $\mathrm{acq}(m),$
  $rpop([t - 1, m], [t']).$

New Approach to Cost Analysis



$$\forall * .\ t \geq 0 \wedge m = 0 \Rightarrow \tilde{f}_{main}(t, m, t) \geq 0$$

$$\forall * .\ t \geq 0 \wedge m \geq 1 \Rightarrow \tilde{f}_{main}(t, m, t_1) \geq \tilde{f}_{rpop}(t, m, t') + 1 + \tilde{f}_{main}(t' + 1, m - 1, t_1)$$

$$\forall * .\ t \geq 0 \wedge m \geq 1 \Rightarrow \tilde{f}_{rpop}(t, m, t) \geq 0$$

$$\forall * .\ t \geq 1 \wedge m \geq 1 \Rightarrow \tilde{f}_{rpop}(t, m, t') \geq m + \tilde{f}_{rpop}(t - 1, m, t')$$

$$\frac{\forall * .\ \wedge_{i=1}^{n} \varphi_i \rightarrow \tilde{f}_p(\bar{x}, \bar{y}) \geq \sum_{i=1}^{n} \tilde{b}_i}{p(\bar{x}|\bar{y}) \longleftarrow b_1, \ldots b_n}$$

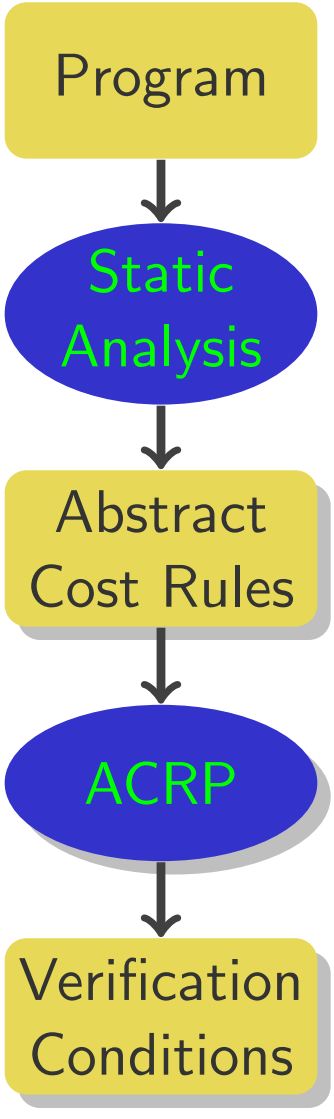
$main([t, 0], [t]) \leftarrow$
  $t \geq 0.$

$main([t, m], [t_1]) \leftarrow$
  $t \geq 0,$
  $m \geq 1,$
  $rpop([t, m], [t']),$
  $\mathrm{acq}(1),$
  $main([t' + 1, m - 1], [t_1]).$

$rpop([t, m], [t]) \leftarrow$
  $t \geq 0,$
  $m \geq 1.$

$rpop([t, m], [t']) \leftarrow$
  $t \geq 1,$
  $m \geq 1,$
  $\mathrm{acq}(m),$
  $rpop([t - 1, m], [t']).$

New Approach to Cost Analysis



$$\forall * .\ t \geq 0 \wedge m = 0 \Rightarrow \tilde{f}_{main}(t, m, t) \geq 0$$

$$\forall * .\ t \geq 0 \wedge m \geq 1 \Rightarrow \tilde{f}_{main}(t, m, t_1) \geq \tilde{f}_{rpop}(t, m, t') + 1 + \tilde{f}_{main}(t' + 1, m - 1, t_1)$$

$$\forall * .\ t \geq 0 \wedge m \geq 1 \Rightarrow \tilde{f}_{rpop}(t, m, t) \geq 0$$

$$\forall * .\ t \geq 1 \wedge m \geq 1 \Rightarrow \tilde{f}_{rpop}(t, m, t') \geq m + \tilde{f}_{rpop}(t - 1, m, t')$$

$$\frac{\forall * .\ \wedge_{i=1}^{n} \varphi_i \rightarrow \tilde{f}_p(\bar{x}, \bar{y}) \geq \sum_{i=1}^{n} \tilde{b}_i}{p(\bar{x}|\bar{y}) \longleftarrow b_1, \ldots b_n}$$

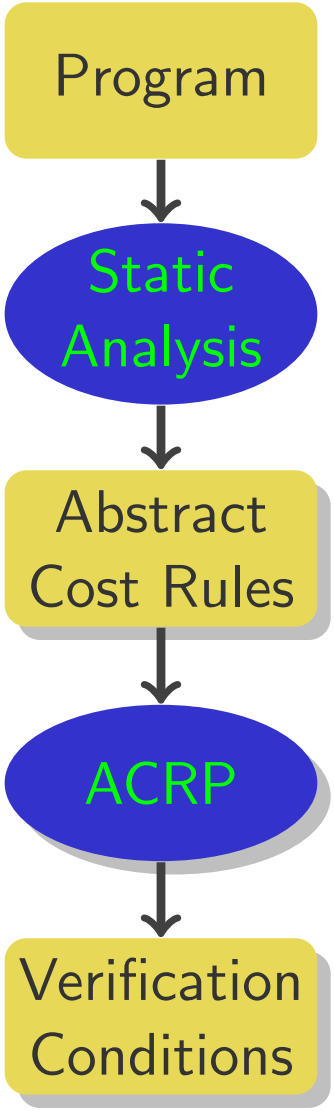
$main([t, 0], [t]) \leftarrow$
  $t \geq 0.$

$main([t, m], [t_1]) \leftarrow$
  $t \geq 0,$
  $m \geq 1,$
  $rpop([t, m], [t']),$
  $\mathrm{acq}(1),$
  $main([t' + 1, m - 1], [t_1]).$

$rpop([t, m], [t]) \leftarrow$
  $t \geq 0,$
  $m \geq 1.$

$rpop([t, m], [t']) \leftarrow$
  $t \geq 1,$
  $m \geq 1,$
  $\mathrm{acq}(m),$
  $rpop([t - 1, m], [t']).$

New Approach to Cost Analysis



$$\forall * . \; t \geq 0 \wedge m = 0 \Rightarrow \tilde{f}_{main}(t, m, t) \geq 0$$
$$\forall * . \; t \geq 0 \wedge m \geq 1 \Rightarrow \tilde{f}_{main}(t, m, t_1) \geq \tilde{f}_{rpop}(t, m, t') + 1 + \tilde{f}_{main}(t' + 1, m - 1, t_1)$$

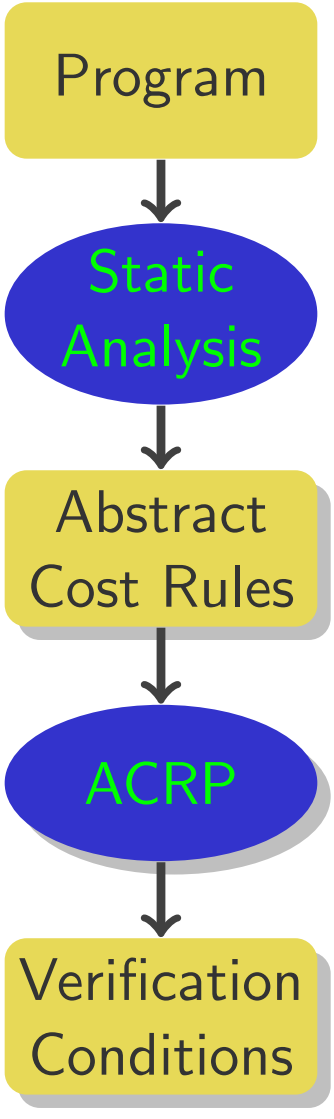
$$\forall * . \; t \geq 0 \wedge m \geq 1 \Rightarrow \tilde{f}_{rpop}(t, m, t) \geq 0$$
$$\forall * . \; t \geq 1 \wedge m \geq 1 \Rightarrow \tilde{f}_{rpop}(t, m, t') \geq m + \tilde{f}_{rpop}(t - 1, m, t')$$

$$\frac{\forall * . \wedge_{i=1}^{n} \varphi_i \rightarrow \tilde{f}_p(\bar{x}, \bar{y}) \geq \sum_{i=1}^{n} \tilde{b}_i}{p(\bar{x}|\bar{y}) \longleftarrow b_1, \ldots b_n}$$

$main([t, 0], [t]) \leftarrow$
  $t \geq 0.$
$main([t, m], [t_1]) \leftarrow$
  $t \geq 0,$
  $m \geq 1,$
  $rpop([t, m], [t']),$
  $\mathrm{acq}(1),$
  $main([t' + 1, m - 1], [t_1]).$

$rpop([t, m], [t]) \leftarrow$
  $t \geq 0,$
  $m \geq 1.$
$rpop([t, m], [t']) \leftarrow$
  $t \geq 1,$
  $m \geq 1,$
  $\mathrm{acq}(m),$
  $rpop([t - 1, m], [t']).$

New Approach to Cost Analysis



$$\forall * .\ t \geq 0 \wedge m = 0 \Rightarrow \tilde{f}_{main}(t, m, t) \geq 0$$

$$\forall * .\ t \geq 0 \wedge m \geq 1 \Rightarrow \tilde{f}_{main}(t, m, t_1) \geq \tilde{f}_{rpop}(t, m, t') + 1 + \tilde{f}_{main}(t' + 1, m - 1, t_1)$$

$$\forall * .\ t \geq 0 \wedge m \geq 1 \Rightarrow \tilde{f}_{rpop}(t, m, t) \geq 0$$

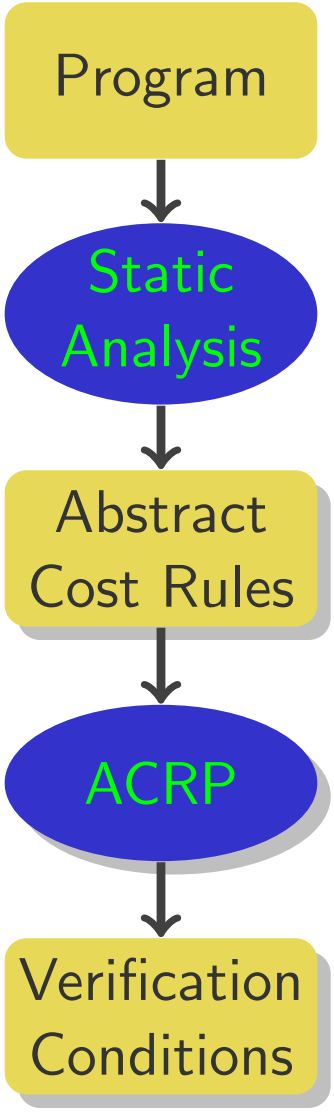
$$\forall * .\ t \geq 1 \wedge m \geq 1 \Rightarrow \tilde{f}_{rpop}(t, m, t') \geq m + \tilde{f}_{rpop}(t - 1, m, t')$$

$$\frac{\forall * .\ \wedge_{i=1}^{n} \varphi_i \rightarrow \tilde{f}_p(\bar{x}, \bar{y}) \geq \sum_{i=1}^{n} \tilde{b}_i}{p(\bar{x}|\bar{y}) \longleftarrow b_1, \ldots b_n}$$

$main([t, 0], [t]) \leftarrow$
  $t \geq 0.$
$main([t, m], [t_1]) \leftarrow$
  $t \geq 0,$
  $m \geq 1,$
  $rpop([t, m], [t']),$
  $\mathrm{acq}(1),$
  $main([t' + 1, m - 1], [t_1]).$

$rpop([t, m], [t]) \leftarrow$
  $t \geq 0,$
  $m \geq 1.$
$rpop([t, m], [t']) \leftarrow$
  $t \geq 1,$
  $m \geq 1,$
  $\mathrm{acq}(m),$
  $rpop([t - 1, m], [t']).$

New Approach to Cost Analysis



$$\begin{array}{ll}
\forall * .\ t \geq 0 \wedge m = 0 \Rightarrow & \tilde{f}_{main}(t, m, t) \geq 0 \\
\forall * .\ t \geq 0 \wedge m \geq 1 \Rightarrow & \tilde{f}_{main}(t, m, t_1) \geq \tilde{f}_{rpop}(t, m, t') + 1 + \tilde{f}_{main}(t' + 1, m - 1, t_1) \\
\forall * .\ t \geq 0 \wedge m \geq 1 \Rightarrow & \tilde{f}_{rpop}(t, m, t) \geq 0 \\
\forall * .\ t \geq 1 \wedge m \geq 1 \Rightarrow & \tilde{f}_{rpop}(t, m, t') \geq m + \tilde{f}_{rpop}(t - 1, m, t')
\end{array}$$

$$\frac{\forall * .\ \wedge_{i=1}^{n} \varphi_i \rightarrow \tilde{f}_p(\bar{x}, \bar{y}) \geq \sum_{i=1}^{n} \tilde{b}_i}{p(\bar{x}|\bar{y}) \longleftarrow b_1, \ldots b_n}$$

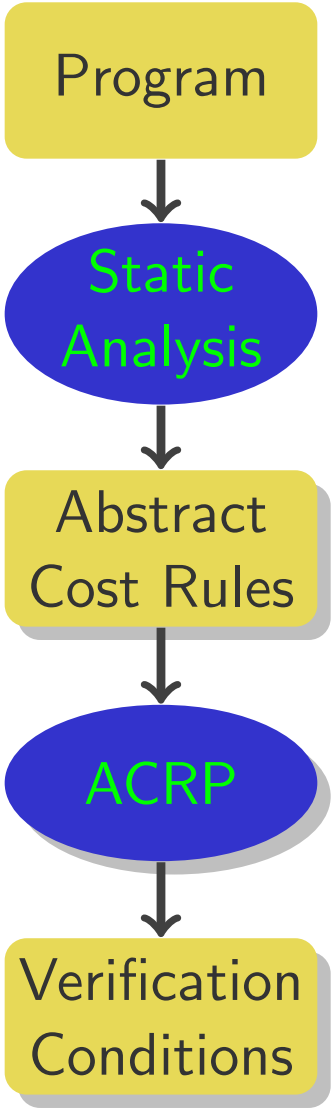
$main([t, 0], [t]) \leftarrow$
$\quad t \geq 0.$

$main([t, m], [t_1]) \leftarrow$
$\quad t \geq 0,$
$\quad m \geq 1,$
$\quad rpop([t, m], [t']),$
$\quad \mathrm{acq}(1),$
$\quad main([t' + 1, m - 1], [t_1]).$

$rpop([t, m], [t]) \leftarrow$
$\quad t \geq 0,$
$\quad m \geq 1.$

$rpop([t, m], [t']) \leftarrow$
$\quad t \geq 1,$
$\quad m \geq 1,$
$\quad \mathrm{acq}(m),$
$\quad rpop([t - 1, m], [t']).$

New Approach to Cost Analysis



$$\forall * .\ t \geq 0 \wedge m = 0 \Rightarrow \tilde{f}_{main}(t, m, t) \geq 0$$

$$\forall * .\ t \geq 0 \wedge m \geq 1 \Rightarrow \tilde{f}_{main}(t, m, t_1) \geq \tilde{f}_{rpop}(t, m, t') + 1 + \tilde{f}_{main}(t' + 1, m - 1, t_1)$$

$$\forall * .\ t \geq 0 \wedge m \geq 1 \Rightarrow \tilde{f}_{rpop}(t, m, t) \geq 0$$

$$\forall * .\ t \geq 1 \wedge m \geq 1 \Rightarrow \tilde{f}_{rpop}(t, m, t') \geq m + \tilde{f}_{rpop}(t - 1, m, t')$$

$$\frac{\forall * .\ \wedge_{i=1}^{n} \varphi_i \rightarrow \tilde{f}_p(\bar{x}, \bar{y}) \geq \sum_{i=1}^{n} \tilde{b}_i}{p(\bar{x}|\bar{y}) \longleftarrow b_1, \ldots b_n}$$

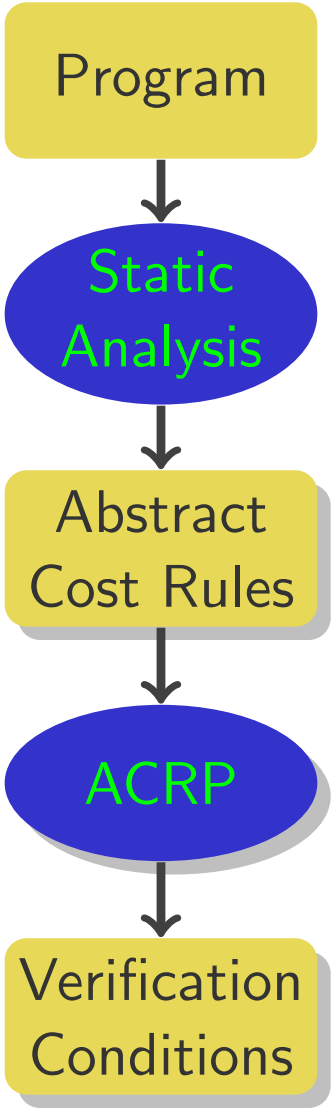
$main([t, 0], [t]) \leftarrow$
  $t \geq 0.$

$main([t, m], [t_1]) \leftarrow$
  $t \geq 0,$
  $m \geq 1,$
  $rpop([t, m], [t']),$
  $\mathrm{acq}(1),$
  $main([t' + 1, m - 1], [t_1]).$

$rpop([t, m], [t]) \leftarrow$
  $t \geq 0,$
  $m \geq 1.$

$rpop([t, m], [t']) \leftarrow$
  $t \geq 1,$
  $m \geq 1,$
  $\mathrm{acq}(m),$
  $rpop([t - 1, m], [t']).$

New Approach to Cost Analysis



$$\forall * .\ t \geq 0 \wedge m = 0 \Rightarrow \tilde{f}_{main}(t, m, t) \geq 0$$
$$\forall * .\ t \geq 0 \wedge m \geq 1 \Rightarrow \tilde{f}_{main}(t, m, t_1) \geq \tilde{f}_{rpop}(t, m, t') + 1 + \tilde{f}_{main}(t' + 1, m - 1, t_1)$$

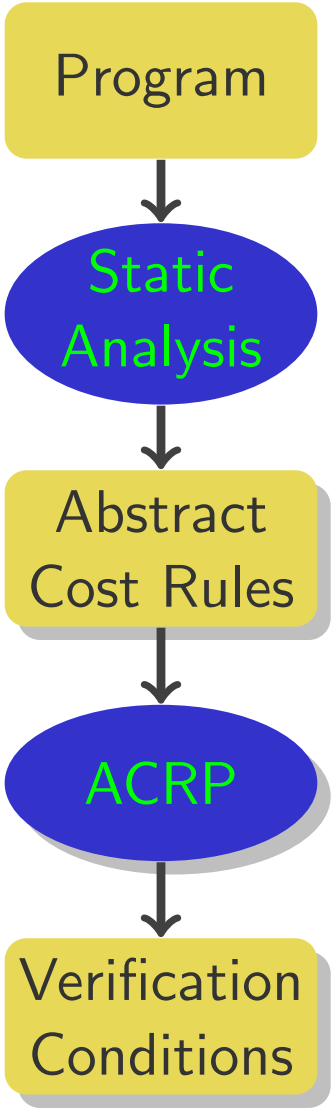
$$\forall * .\ t \geq 0 \wedge m \geq 1 \Rightarrow \tilde{f}_{rpop}(t, m, t) \geq 0$$
$$\forall * .\ t \geq 1 \wedge m \geq 1 \Rightarrow \tilde{f}_{rpop}(t, m, t') \geq m + \tilde{f}_{rpop}(t - 1, m, t')$$

$$\frac{\forall * .\ \wedge_{i=1}^{n} \varphi_i \rightarrow \tilde{f}_p(\bar{x}, \bar{y}) \geq \sum_{i=1}^{n} \tilde{b}_i}{p(\bar{x}|\bar{y}) \longleftarrow b_1, \ldots b_n}$$

$main([t, 0], [t]) \leftarrow$
  $t \geq 0.$
$main([t, m], [t_1]) \leftarrow$
  $t \geq 0,$
  $m \geq 1,$
  $rpop([t, m], [t']),$
  $\mathtt{acq}(1),$
  $main([t' + 1, m - 1], [t_1]).$

$rpop([t, m], [t]) \leftarrow$
  $t \geq 0,$
  $m \geq 1.$
$rpop([t, m], [t']) \leftarrow$
  $t \geq 1,$
  $m \geq 1,$
  $\mathtt{acq}(m),$
  $rpop([t - 1, m], [t']).$

New Approach to Cost Analysis



$$\forall * . \; t \geq 0 \wedge m = 0 \Rightarrow \tilde{f}_{main}(t, m, t) \geq 0$$

$$\forall * . \; t \geq 0 \wedge m \geq 1 \Rightarrow \tilde{f}_{main}(t, m, t_1) \geq \tilde{f}_{rpop}(t, m, t') + 1 + \tilde{f}_{main}(t' + 1, m - 1, t_1)$$

$$\forall * . \; t \geq 0 \wedge m \geq 1 \Rightarrow \tilde{f}_{rpop}(t, m, t) \geq 0$$

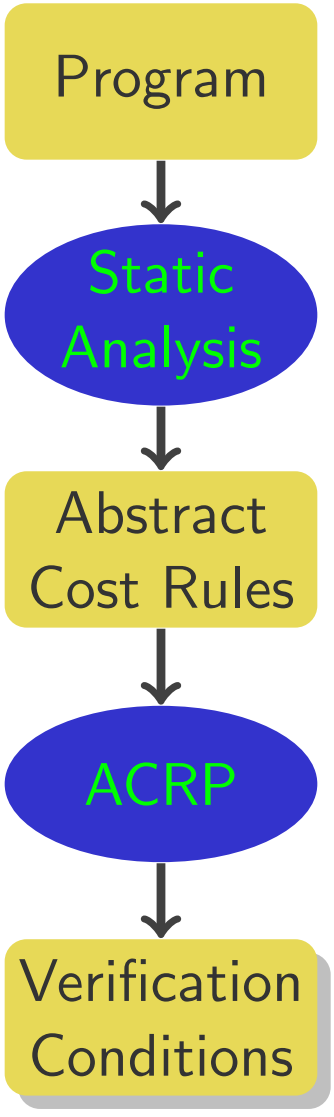
$$\forall * . \; t \geq 1 \wedge m \geq 1 \Rightarrow \tilde{f}_{rpop}(t, m, t') \geq m + \tilde{f}_{rpop}(t - 1, m, t')$$

$$\frac{\forall * . \wedge_{i=1}^{n} \varphi_i \rightarrow \tilde{f}_p(\bar{x}, \bar{y}) \geq \sum_{i=1}^{n} \tilde{b}_i}{p(\bar{x}|\bar{y}) \longleftarrow b_1, \ldots b_n}$$

$main([t, 0], [t]) \leftarrow$
  $t \geq 0.$
$main([t, m], [t_1]) \leftarrow$
  $t \geq 0,$
  $m \geq 1,$
  $rpop([t, m], [t']),$
  $\mathrm{acq}(1),$
  $main([t' + 1, m - 1], [t_1]).$

$rpop([t, m], [t]) \leftarrow$
  $t \geq 0,$
  $m \geq 1.$
$rpop([t, m], [t']) \leftarrow$
  $t \geq 1,$
  $m \geq 1,$
  $\mathrm{acq}(m),$
  $rpop([t - 1, m], [t']).$

New Approach to Cost Analysis



$$\forall * .\ t \geq 0 \wedge m = 0 \Rightarrow \tilde{f}_{main}(t, m, t) \geq 0$$

$$\forall * .\ t \geq 0 \wedge m \geq 1 \Rightarrow \tilde{f}_{main}(t, m, t_1) \geq \tilde{f}_{rpop}(t, m, t') + 1 + \tilde{f}_{main}(t' + 1, m - 1, t_1)$$

$$\forall * .\ t \geq 0 \wedge m \geq 1 \Rightarrow \tilde{f}_{rpop}(t, m, t) \geq 0$$

$$\forall * .\ t \geq 1 \wedge m \geq 1 \Rightarrow \tilde{f}_{rpop}(t, m, t') \geq m + \tilde{f}_{rpop}(t - 1, m, t')$$

New Approach to Cost Analysis

Verification Conditions

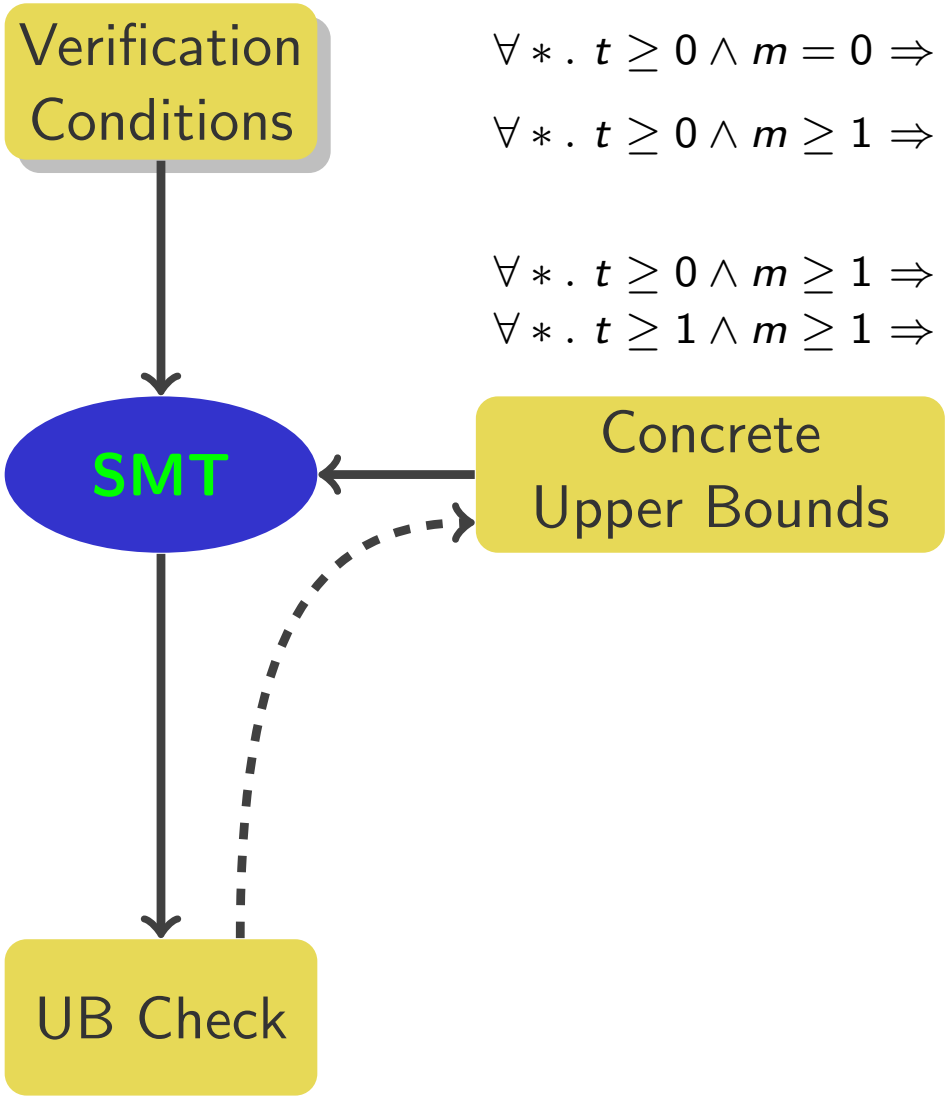
$$\forall * .\ t \geq 0 \wedge m = 0 \Rightarrow \tilde{f}_{main}(t, m, t) \geq 0$$

$$\forall * .\ t \geq 0 \wedge m \geq 1 \Rightarrow \tilde{f}_{main}(t, m, t_1) \geq \tilde{f}_{rpop}(t, m, t') + 1 + \tilde{f}_{main}(t' + 1, m - 1, t_1)$$

$$\forall * .\ t \geq 0 \wedge m \geq 1 \Rightarrow \tilde{f}_{rpop}(t, m, t) \geq 0$$

$$\forall * .\ t \geq 1 \wedge m \geq 1 \Rightarrow \tilde{f}_{rpop}(t, m, t') \geq m + \tilde{f}_{rpop}(t - 1, m, t')$$

New Approach to Cost Analysis



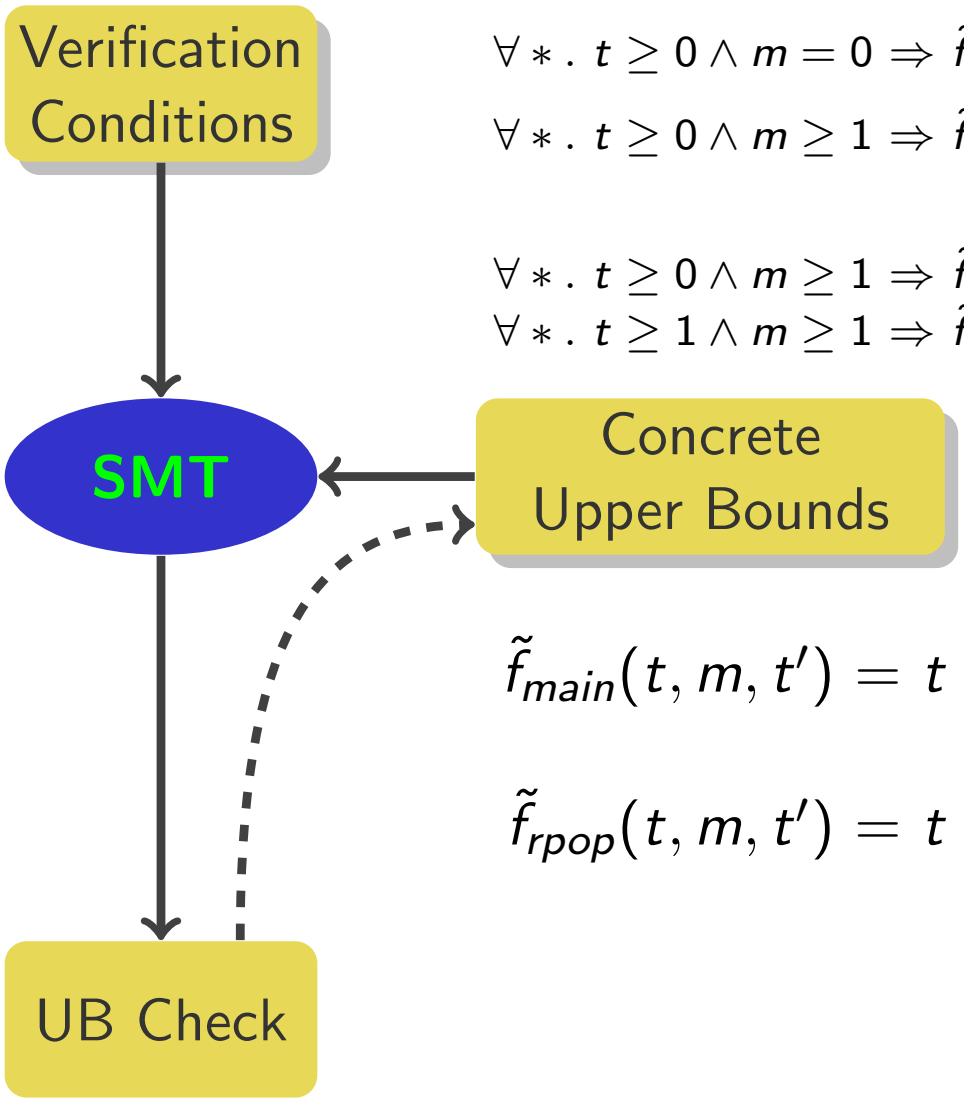
$$\forall * .\ t \geq 0 \wedge m = 0 \Rightarrow \tilde{f}_{main}(t, m, t) \geq 0$$

$$\forall * .\ t \geq 0 \wedge m \geq 1 \Rightarrow \tilde{f}_{main}(t, m, t_1) \geq \tilde{f}_{rpop}(t, m, t') + 1 + \tilde{f}_{main}(t' + 1, m - 1, t_1)$$

$$\forall * .\ t \geq 0 \wedge m \geq 1 \Rightarrow \tilde{f}_{rpop}(t, m, t) \geq 0$$

$$\forall * .\ t \geq 1 \wedge m \geq 1 \Rightarrow \tilde{f}_{rpop}(t, m, t') \geq m + \tilde{f}_{rpop}(t - 1, m, t')$$

New Approach to Cost Analysis



$$\forall * .\ t \geq 0 \wedge m = 0 \Rightarrow \tilde{f}_{main}(t, m, t) \geq 0$$

$$\forall * .\ t \geq 0 \wedge m \geq 1 \Rightarrow \tilde{f}_{main}(t, m, t_1) \geq \tilde{f}_{rpop}(t, m, t') + 1 + \tilde{f}_{main}(t' + 1, m - 1, t_1)$$

$$\forall * .\ t \geq 0 \wedge m \geq 1 \Rightarrow \tilde{f}_{rpop}(t, m, t) \geq 0$$

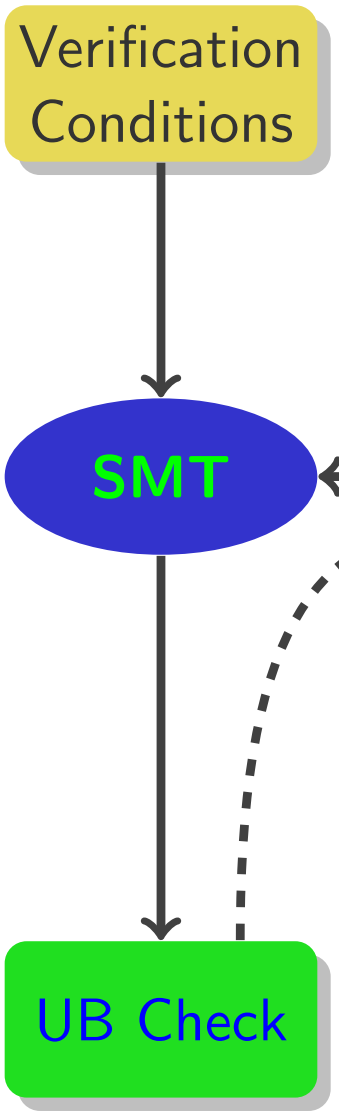
$$\forall * .\ t \geq 1 \wedge m \geq 1 \Rightarrow \tilde{f}_{rpop}(t, m, t') \geq m + \tilde{f}_{rpop}(t - 1, m, t')$$

1. We try these Input & Output Upper Bounds

$$\tilde{f}_{main}(t, m, t') = t * m + \frac{m^2 + m}{2}$$

$$\tilde{f}_{rpop}(t, m, t') = t * m - t' * m$$

New Approach to Cost Analysis



$$\forall * .\ t \geq 0 \wedge m = 0 \Rightarrow \tilde{f}_{main}(t, m, t) \geq 0$$

$$\forall * .\ t \geq 0 \wedge m \geq 1 \Rightarrow \tilde{f}_{main}(t, m, t_1) \geq \tilde{f}_{rpop}(t, m, t') + 1 + \tilde{f}_{main}(t' + 1, m - 1, t_1)$$

$$\forall * .\ t \geq 0 \wedge m \geq 1 \Rightarrow \tilde{f}_{rpop}(t, m, t) \geq 0$$

$$\forall * .\ t \geq 1 \wedge m \geq 1 \Rightarrow \tilde{f}_{rpop}(t, m, t') \geq m + \tilde{f}_{rpop}(t - 1, m, t')$$

Concrete Upper Bounds

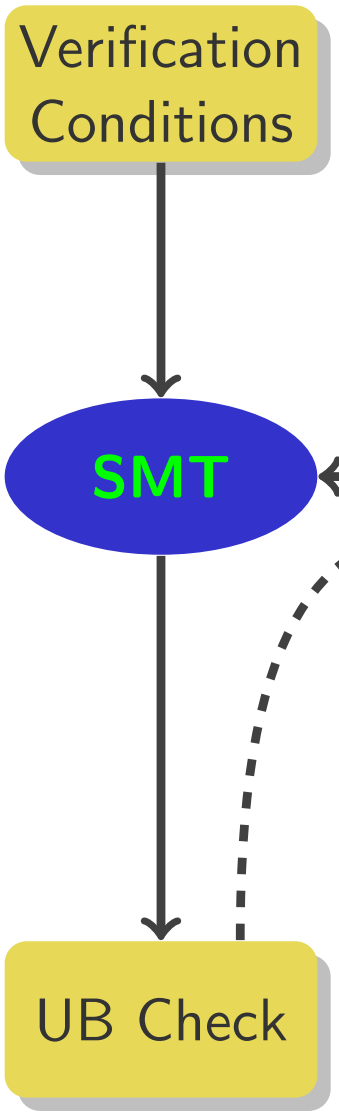
1. We try these Input & Output Upper Bounds

$$\tilde{f}_{main}(t, m, t') = t * m + \frac{m^2 + m}{2}$$

$$\tilde{f}_{rpop}(t, m, t') = t * m - t' * m$$

Valid

New Approach to Cost Analysis



$$\forall * .\ t \geq 0 \wedge m = 0 \Rightarrow \tilde{f}_{main}(t, m, t) \geq 0$$

$$\forall * .\ t \geq 0 \wedge m \geq 1 \Rightarrow \tilde{f}_{main}(t, m, t_1) \geq \tilde{f}_{rpop}(t, m, t') + 1 + \tilde{f}_{main}(t' + 1, m - 1, t_1)$$

$$\forall * .\ t \geq 0 \wedge m \geq 1 \Rightarrow \tilde{f}_{rpop}(t, m, t) \geq 0$$

$$\forall * .\ t \geq 1 \wedge m \geq 1 \Rightarrow \tilde{f}_{rpop}(t, m, t') \geq m + \tilde{f}_{rpop}(t - 1, m, t')$$

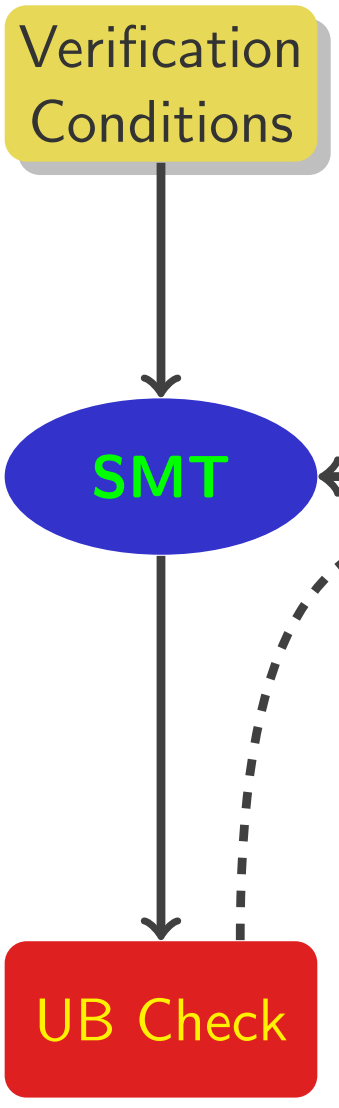
Concrete Upper Bounds

2. We try the desired Input-only Upper Bounds

$$\tilde{f}_{main}(t, m, t') = t * m + \frac{m^2 + m}{2}$$

$$\tilde{f}_{rpop}(t, m, t') = t * m$$

New Approach to Cost Analysis



$$\forall * .\ t \geq 0 \wedge m = 0 \Rightarrow \tilde{f}_{main}(t, m, t) \geq 0$$
$$\forall * .\ t \geq 0 \wedge m \geq 1 \Rightarrow \tilde{f}_{main}(t, m, t_1) \geq \tilde{f}_{rpop}(t, m, t') + 1 + \tilde{f}_{main}(t' + 1, m - 1, t_1)$$

$$\forall * .\ t \geq 0 \wedge m \geq 1 \Rightarrow \tilde{f}_{rpop}(t, m, t) \geq 0$$
$$\forall * .\ t \geq 1 \wedge m \geq 1 \Rightarrow \tilde{f}_{rpop}(t, m, t') \geq m + \tilde{f}_{rpop}(t - 1, m, t')$$

Concrete Upper Bounds

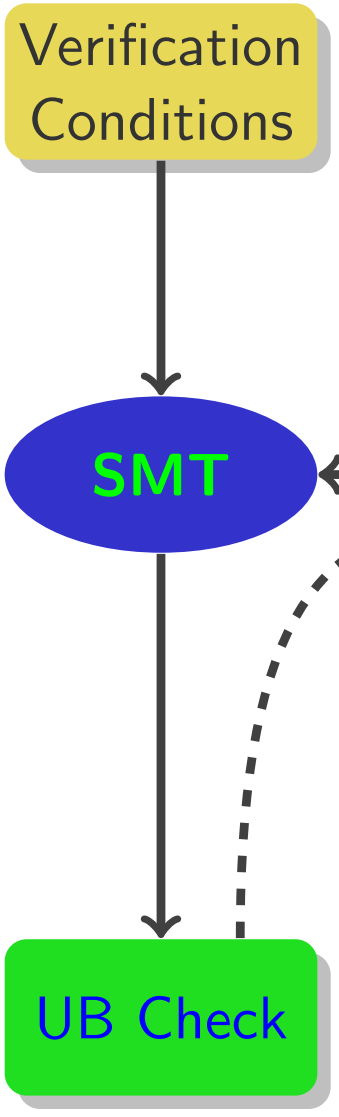
2. We try the desired Input-only Upper Bounds

$$\tilde{f}_{main}(t, m, t') = t * m + \frac{m^2 + m}{2}$$

$$\tilde{f}_{rpop}(t, m, t') = t * m$$

Invalid

New Approach to Cost Analysis



$$\forall * .\ t \geq 0 \wedge m = 0 \Rightarrow \tilde{f}_{main}(t, m, t) \geq 0$$
$$\forall * .\ t \geq 0 \wedge m \geq 1 \Rightarrow \tilde{f}_{main}(t, m, t_1) \geq \tilde{f}_{rpop}(t, m, t') + 1 + \tilde{f}_{main}(t' + 1, m - 1, t_1)$$

$$\forall * .\ t \geq 0 \wedge m \geq 1 \Rightarrow \tilde{f}_{rpop}(t, m, t) \geq 0$$
$$\forall * .\ t \geq 1 \wedge m \geq 1 \Rightarrow \tilde{f}_{rpop}(t, m, t') \geq m + \tilde{f}_{rpop}(t - 1, m, t')$$

Concrete Upper Bounds

2. We try the desired Input-only Upper Bounds

$$\tilde{f}_{main}(t, m, t') = t * m + \frac{m^2 + m}{2}$$

$$\tilde{f}_{rpop}(t, m, t') = t * m$$

Valid

New Approach to Cost Analysis

Verification Conditions

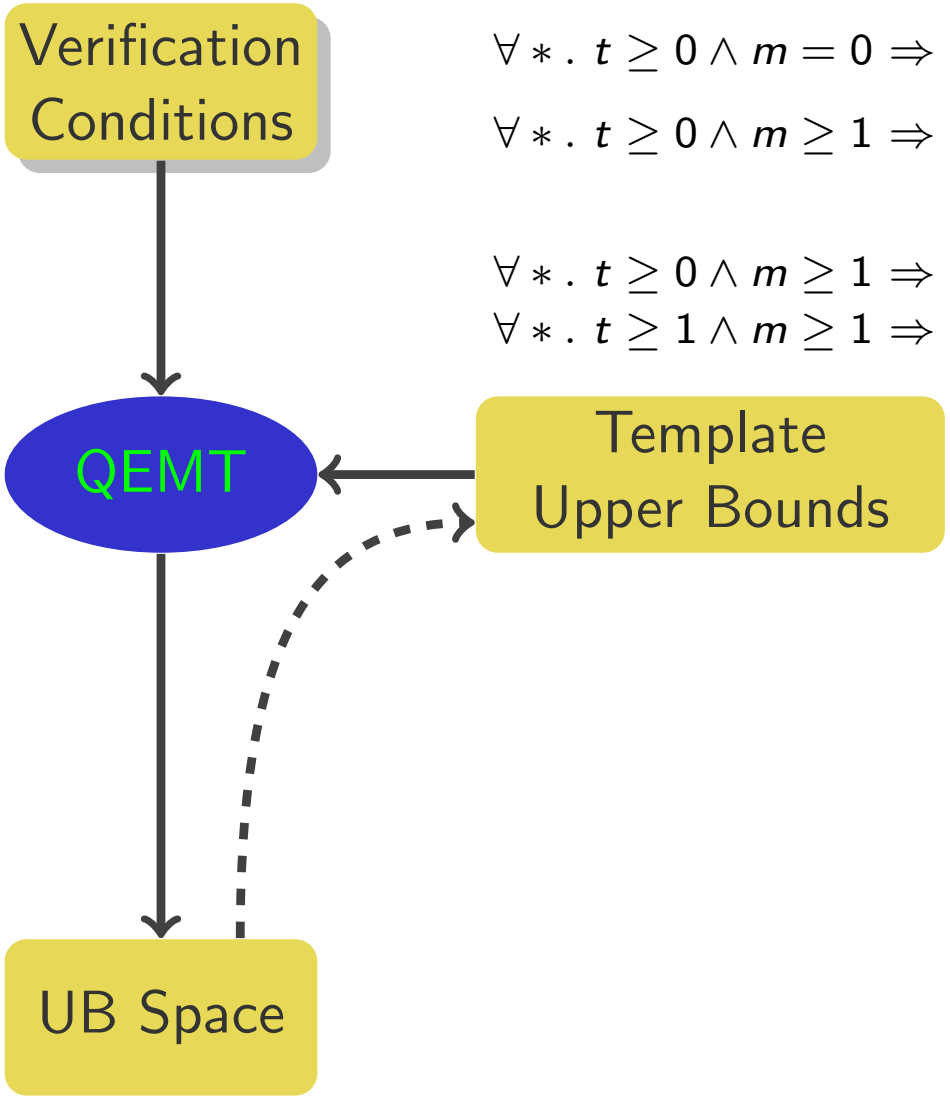
$$\forall * .\ t \geq 0 \wedge m = 0 \Rightarrow \tilde{f}_{main}(t, m, t) \geq 0$$

$$\forall * .\ t \geq 0 \wedge m \geq 1 \Rightarrow \tilde{f}_{main}(t, m, t_1) \geq \tilde{f}_{rpop}(t, m, t') + 1 + \tilde{f}_{main}(t' + 1, m - 1, t_1)$$

$$\forall * .\ t \geq 0 \wedge m \geq 1 \Rightarrow \tilde{f}_{rpop}(t, m, t) \geq 0$$

$$\forall * .\ t \geq 1 \wedge m \geq 1 \Rightarrow \tilde{f}_{rpop}(t, m, t') \geq m + \tilde{f}_{rpop}(t - 1, m, t')$$

New Approach to Cost Analysis



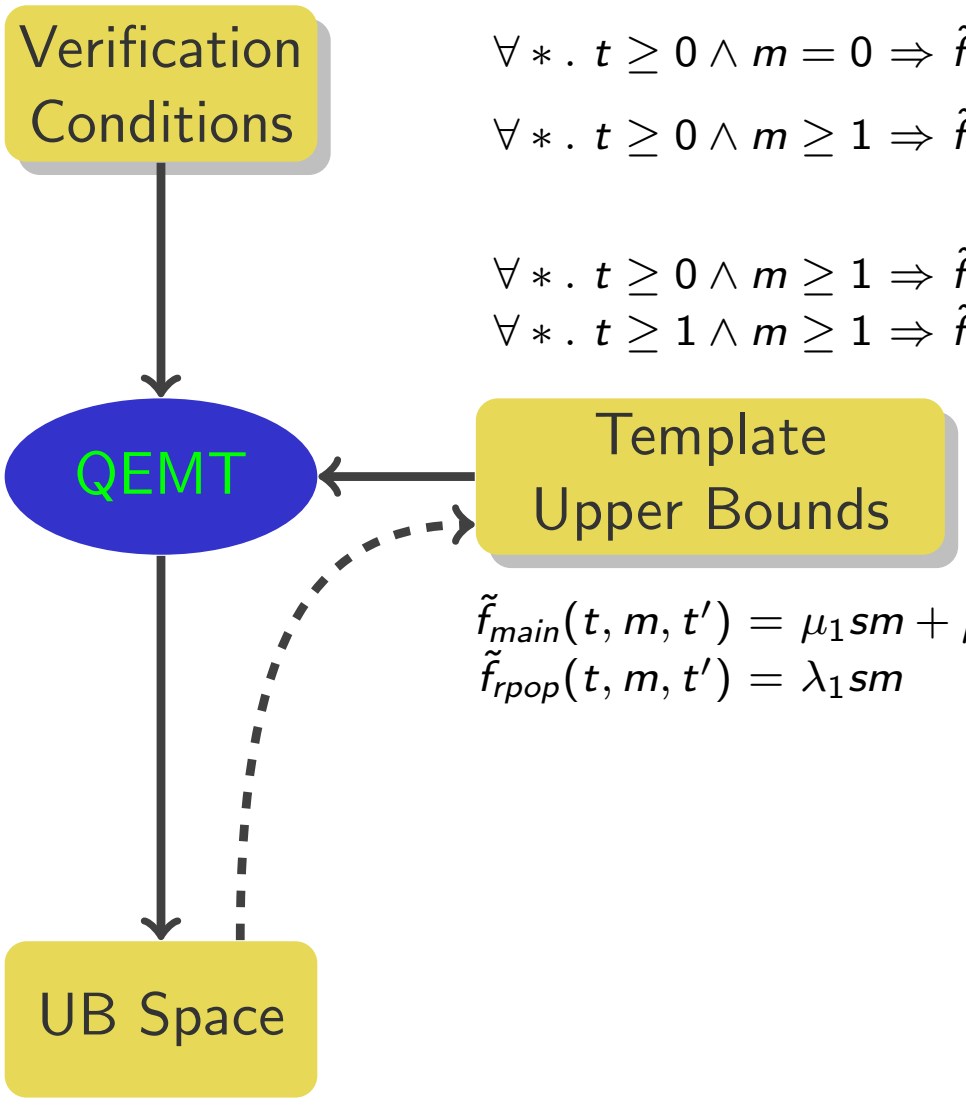
$$\forall * .\ t \geq 0 \wedge m = 0 \Rightarrow \tilde{f}_{main}(t, m, t) \geq 0$$

$$\forall * .\ t \geq 0 \wedge m \geq 1 \Rightarrow \tilde{f}_{main}(t, m, t_1) \geq \tilde{f}_{rpop}(t, m, t') + 1 + \tilde{f}_{main}(t' + 1, m - 1, t_1)$$

$$\forall * .\ t \geq 0 \wedge m \geq 1 \Rightarrow \tilde{f}_{rpop}(t, m, t) \geq 0$$

$$\forall * .\ t \geq 1 \wedge m \geq 1 \Rightarrow \tilde{f}_{rpop}(t, m, t') \geq m + \tilde{f}_{rpop}(t - 1, m, t')$$

New Approach to Cost Analysis



$$\forall * .\ t \geq 0 \wedge m = 0 \Rightarrow \tilde{f}_{main}(t, m, t) \geq 0$$

$$\forall * .\ t \geq 0 \wedge m \geq 1 \Rightarrow \tilde{f}_{main}(t, m, t_1) \geq \tilde{f}_{rpop}(t, m, t') + 1 + \tilde{f}_{main}(t' + 1, m - 1, t_1)$$

$$\forall * .\ t \geq 0 \wedge m \geq 1 \Rightarrow \tilde{f}_{rpop}(t, m, t) \geq 0$$

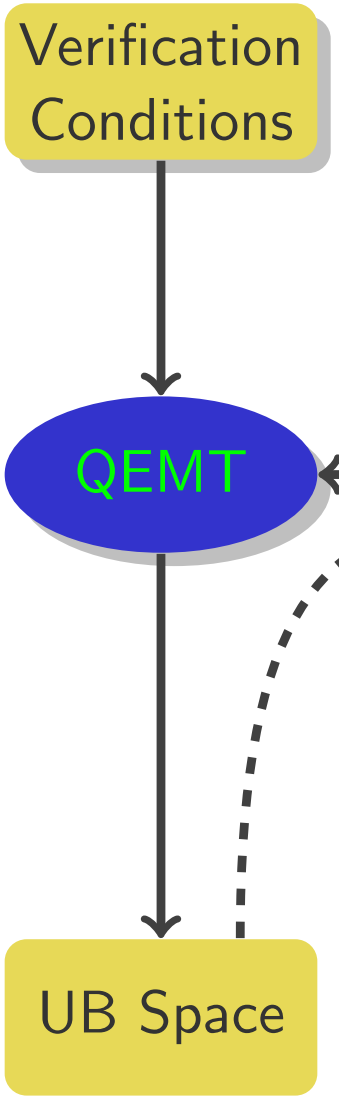
$$\forall * .\ t \geq 1 \wedge m \geq 1 \Rightarrow \tilde{f}_{rpop}(t, m, t') \geq m + \tilde{f}_{rpop}(t - 1, m, t')$$

1. We use these templates for Input & Output UBs

$$\tilde{f}_{main}(t, m, t') = \mu_1 sm + \mu_2 m^2 + \mu_3 mt' + \mu_4 s + \mu_5 m + \mu_6 t' + \mu_0$$

$$\tilde{f}_{rpop}(t, m, t') = \lambda_1 sm + \lambda_2 mt' + \lambda_3 s + \lambda_4 m + \lambda_5 t' + \lambda_0$$

New Approach to Cost Analysis



$$\forall * .\ t \geq 0 \wedge m = 0 \Rightarrow \tilde{f}_{main}(t, m, t) \geq 0$$

$$\forall * .\ t \geq 0 \wedge m \geq 1 \Rightarrow \tilde{f}_{main}(t, m, t_1) \geq \tilde{f}_{rpop}(t, m, t') + 1 + \tilde{f}_{main}(t' + 1, m - 1, t_1)$$

$$\forall * .\ t \geq 0 \wedge m \geq 1 \Rightarrow \tilde{f}_{rpop}(t, m, t) \geq 0$$

$$\forall * .\ t \geq 1 \wedge m \geq 1 \Rightarrow \tilde{f}_{rpop}(t, m, t') \geq m + \tilde{f}_{rpop}(t - 1, m, t')$$

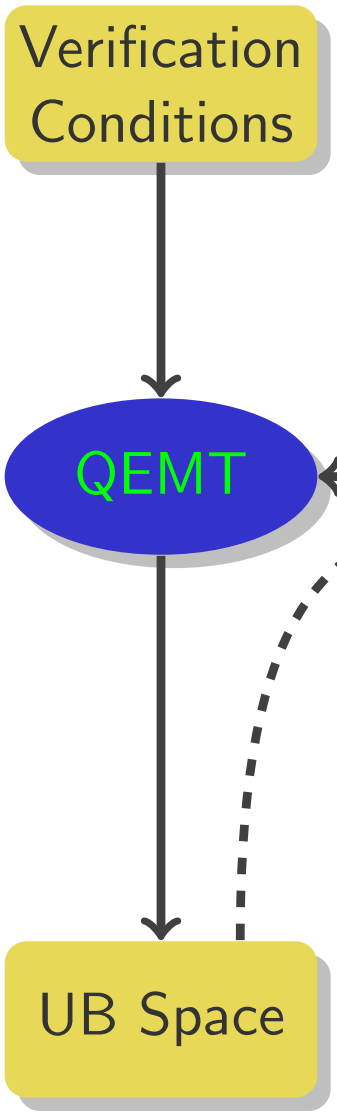
Template Upper Bounds

2. **Eliminate $\forall$ Quantifiers** on program variables

$$\tilde{f}_{main}(t, m, t') = \mu_1 sm + \mu_2 m^2 + \mu_3 mt' + \mu_4 s + \mu_5 m + \mu_6 t' + \mu_0$$

$$\tilde{f}_{rpop}(t, m, t') = \lambda_1 sm + \lambda_2 mt' + \lambda_3 s + \lambda_4 m + \lambda_5 t' + \lambda_0$$

New Approach to Cost Analysis



$$\forall * .\ t \geq 0 \wedge m = 0 \Rightarrow \tilde{f}_{main}(t, m, t) \geq 0$$

$$\forall * .\ t \geq 0 \wedge m \geq 1 \Rightarrow \tilde{f}_{main}(t, m, t_1) \geq \tilde{f}_{rpop}(t, m, t') + 1 + \tilde{f}_{main}(t' + 1, m - 1, t_1)$$

$$\forall * .\ t \geq 0 \wedge m \geq 1 \Rightarrow \tilde{f}_{rpop}(t, m, t) \geq 0$$

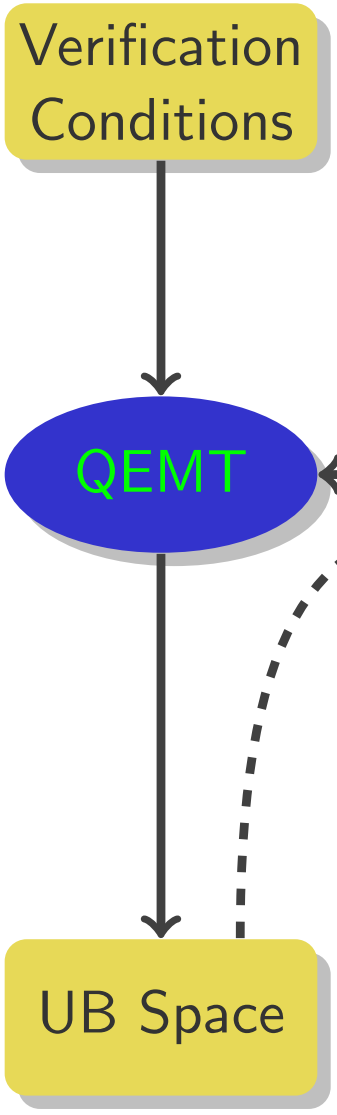
$$\forall * .\ t \geq 1 \wedge m \geq 1 \Rightarrow \tilde{f}_{rpop}(t, m, t') \geq m + \tilde{f}_{rpop}(t - 1, m, t')$$

Quantifier Elimination of $\forall \bar{w}.\Phi[\bar{w}, \bar{u}]$ gives a constraint $\psi[\bar{u}]$ on template parameters

$$\tilde{f}_{main}(t, m, t') = \mu_1 sm + \mu_2 m^2 + \mu_3 mt' + \mu_4 s + \mu_5 m + \mu_6 t' + \mu_0$$

$$\tilde{f}_{rpop}(t, m, t') = \lambda_1 sm + \lambda_2 mt' + \lambda_3 s + \lambda_4 m + \lambda_5 t' + \lambda_0$$

New Approach to Cost Analysis



$$\forall * .\ t \geq 0 \wedge m = 0 \Rightarrow \tilde{f}_{main}(t, m, t) \geq 0$$

$$\forall * .\ t \geq 0 \wedge m \geq 1 \Rightarrow \tilde{f}_{main}(t, m, t_1) \geq \tilde{f}_{rpop}(t, m, t') + 1 + \tilde{f}_{main}(t' + 1, m - 1, t_1)$$

$$\forall * .\ t \geq 0 \wedge m \geq 1 \Rightarrow \tilde{f}_{rpop}(t, m, t) \geq 0$$

$$\forall * .\ t \geq 1 \wedge m \geq 1 \Rightarrow \tilde{f}_{rpop}(t, m, t') \geq m + \tilde{f}_{rpop}(t - 1, m, t')$$

$$\tilde{f}_{main}(t, m, t') = \mu_1 sm + \mu_2 m^2 + \mu_3 mt' + \mu_4 s + \mu_5 m + \mu_6 t' + \mu_0$$

$$\tilde{f}_{rpop}(t, m, t') = \lambda_1 sm + \lambda_2 mt' + \lambda_3 s + \lambda_4 m + \lambda_5 t' + \lambda_0$$

$$\mu_0 \geq 0 \quad \mu_4 = \lambda_1 - \lambda_5 \quad \mu_5 + \mu_2 \geq \mu_4 + \lambda_0 + \lambda_4 + 1$$

$$\mu_3 = 0 \quad \mu_6 \geq \lambda_5 - \lambda_1 \quad 2 \cdot \mu_2 \geq \lambda_1 + \lambda_4$$

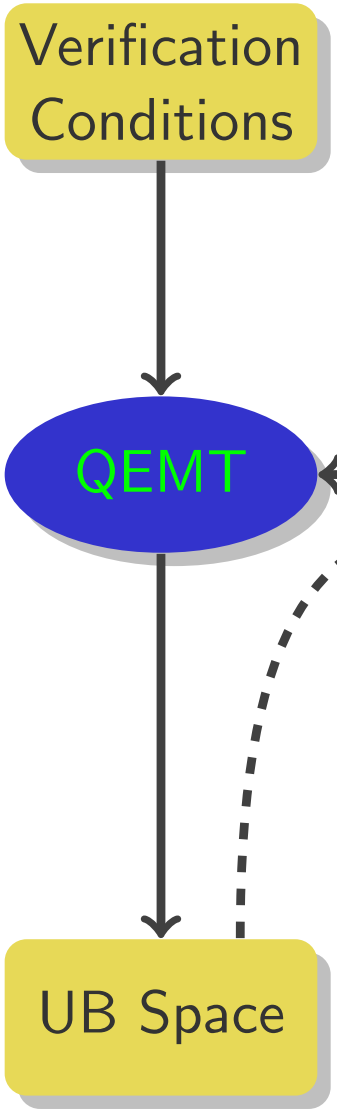
$$\mu_1 = \lambda_1$$

$$\lambda_4 \geq 0 \quad \lambda_0 + \lambda_4 \geq 0$$

$$\lambda_1 \geq 1 \quad \lambda_1 \geq \lambda_3 + \lambda_5 \quad \lambda_1 + \lambda_3 \geq 1$$

$$\lambda_2 = -\lambda_1 \quad \lambda_3 + \lambda_5 \geq 0$$

New Approach to Cost Analysis



$$\forall * .\ t \geq 0 \wedge m = 0 \Rightarrow \tilde{f}_{main}(t, m, t) \geq 0$$
$$\forall * .\ t \geq 0 \wedge m \geq 1 \Rightarrow \tilde{f}_{main}(t, m, t_1) \geq \tilde{f}_{rpop}(t, m, t') + 1 + \tilde{f}_{main}(t' + 1, m - 1, t_1)$$

$$\forall * .\ t \geq 0 \wedge m \geq 1 \Rightarrow \tilde{f}_{rpop}(t, m, t) \geq 0$$
$$\forall * .\ t \geq 1 \wedge m \geq 1 \Rightarrow \tilde{f}_{rpop}(t, m, t') \geq m + \tilde{f}_{rpop}(t - 1, m, t')$$

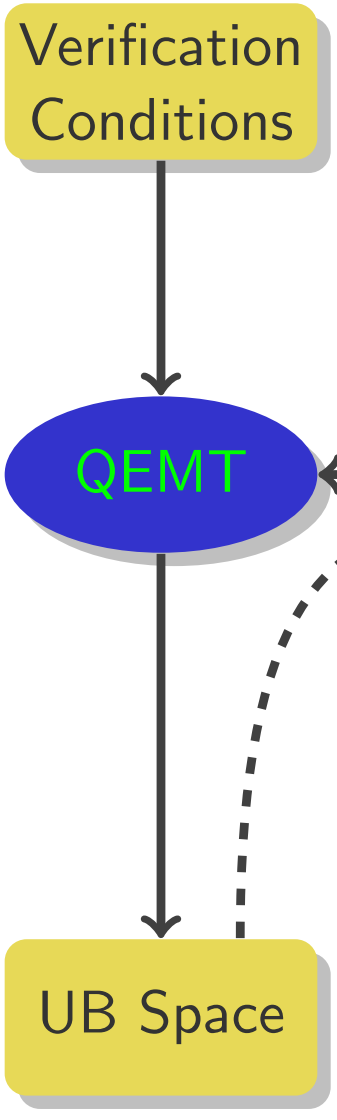
3. Solve the template constraint to get a particular instance

$$\tilde{f}_{main}(t, m, t') = \mu_1 sm + \mu_2 m^2 + \mu_3 mt' + \mu_4 s + \mu_5 m + \mu_6 t' + \mu_0$$
$$\tilde{f}_{rpop}(t, m, t') = \lambda_1 sm + \lambda_2 mt' + \lambda_3 s + \lambda_4 m + \lambda_5 t' + \lambda_0$$

$$\mu_0 \geq 0 \quad \mu_4 = \lambda_1 - \lambda_5 \quad \mu_5 + \mu_2 \geq \mu_4 + \lambda_0 + \lambda_4 + 1$$
$$\mu_3 = 0 \quad \mu_6 \geq \lambda_5 - \lambda_1 \quad 2 \cdot \mu_2 \geq \lambda_1 + \lambda_4$$
$$\mu_1 = \lambda_1$$

$$\lambda_4 \geq 0 \quad \lambda_0 + \lambda_4 \geq 0$$
$$\lambda_1 \geq 1 \quad \lambda_1 \geq \lambda_3 + \lambda_5 \quad \lambda_1 + \lambda_3 \geq 1$$
$$\lambda_2 = -\lambda_1 \quad \lambda_3 + \lambda_5 \geq 0$$

New Approach to Cost Analysis



$\forall *.\ t \geq 0 \wedge m = 0 \Rightarrow \tilde{f}_{main}(t, m, t) \geq 0$

$$\mu_1 = 1 \quad \mu_2 = \frac{1}{2} \quad \mu_3 = 0 \quad \mu_4 = 0 \quad \mu_5 = \frac{1}{2} \quad \mu_6 = 0 \quad \mu_0 = 0$$
$$\lambda_1 = 1 \quad \lambda_2 = -1 \quad \lambda_3 = 0 \quad \lambda_4 = 0 \quad \lambda_5 = 0 \quad \lambda_0 = 0$$

$\forall *.\ t \geq 0 \wedge m \geq 1 \Rightarrow \tilde{f}_{rpop}(t, m, t) \geq 0$

$\forall *.\ t \geq 1 \wedge m \geq 1 \Rightarrow \tilde{f}_{rpop}(t, m, t') \geq m + \tilde{f}_{rpop}(t-1, m, t')$

Template Upper Bounds

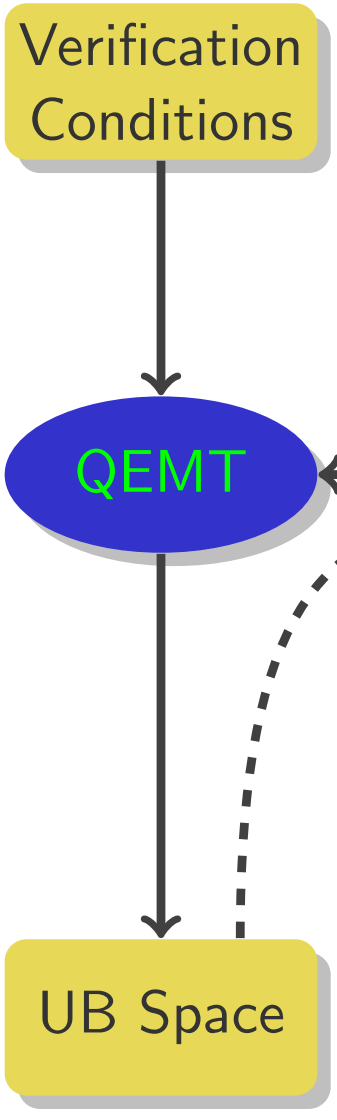
3. Solve the template constraint to get a particular instance

$$\tilde{f}_{main}(t, m, t') = \mu_1 sm + \mu_2 m^2 + \mu_3 mt' + \mu_4 s + \mu_5 m + \mu_6 t' + \mu_0$$
$$\tilde{f}_{rpop}(t, m, t') = \lambda_1 sm + \lambda_2 mt' + \lambda_3 s + \lambda_4 m + \lambda_5 t' + \lambda_0$$

$$\mu_0 \geq 0 \quad \mu_4 = \lambda_1 - \lambda_5 \quad \mu_5 + \mu_2 \geq \mu_4 + \lambda_0 + \lambda_4 + 1$$
$$\mu_3 = 0 \quad \mu_6 \geq \lambda_5 - \lambda_1 \quad 2 \cdot \mu_2 \geq \lambda_1 + \lambda_4$$
$$\mu_1 = \lambda_1$$

$$\lambda_4 \geq 0 \quad \lambda_0 + \lambda_4 \geq 0$$
$$\lambda_1 \geq 1 \quad \lambda_1 \geq \lambda_3 + \lambda_5 \quad \lambda_1 + \lambda_3 \geq 1$$
$$\lambda_2 = -\lambda_1 \quad \lambda_3 + \lambda_5 \geq 0$$

New Approach to Cost Analysis



Template Upper Bounds

$$\forall *.\ t \geq 0 \wedge m = 0 \Rightarrow \tilde{f}_{main}(t, m, t) \geq 0$$

$$\mu_1 = 1 \quad \mu_2 = \frac{1}{2} \quad \mu_3 = 0 \quad \mu_4 = 0 \quad \mu_5 = \frac{1}{2} \quad \mu_6 = 0 \quad \mu_0 = 0$$

$$\lambda_1 = 1 \quad \lambda_2 = -1 \quad \lambda_3 = 0 \quad \lambda_4 = 0 \quad \lambda_5 = 0 \quad \lambda_0 = 0$$

$$\forall *.\ t \geq 0 \wedge m \geq 1 \Rightarrow f_{rpop}(t, m, t) \geq 0$$

$$\forall *.\ t \geq 1 \wedge m \geq 1 \Rightarrow \tilde{f}_{rpop}(t, m, t') \geq m + \tilde{f}_{rpop}(t-1, m, t')$$

3. Solve the template constraint to get a particular instance

$$\tilde{f}_{main}(t, m, t') = \mu_1 sm + \mu_2 m^2 + \mu_3 mt' + \mu_4 s + \mu_5 m + \mu_6 t' + \mu_0$$

$$\tilde{f}_{rpop}(t, m, t') = \lambda_1 sm + \lambda_2 mt' + \lambda_3 s + \lambda_4 m + \lambda_5 t' + \lambda_0$$

$$\mu_0 \geq 0 \quad \mu_4 = \lambda_1 - \lambda_5 \quad \mu_5 + \mu_2 \geq \mu_4 + \lambda_0 + \lambda_4 + 1$$

$$\mu_3 = 0$$

$$\mu_1 = \lambda$$

$$\lambda_4 \geq 0$$

$$\lambda_1 \geq 1$$

$$\lambda_2 = -\lambda_1$$

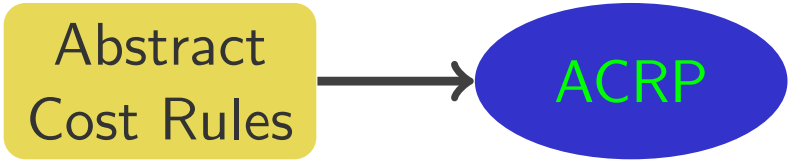
$$\tilde{f}_{main}(t, m, t') = t * m + \frac{m^2+m}{2}$$

$$\tilde{f}_{rpop}(t, m, t') = t * m - t' * m$$

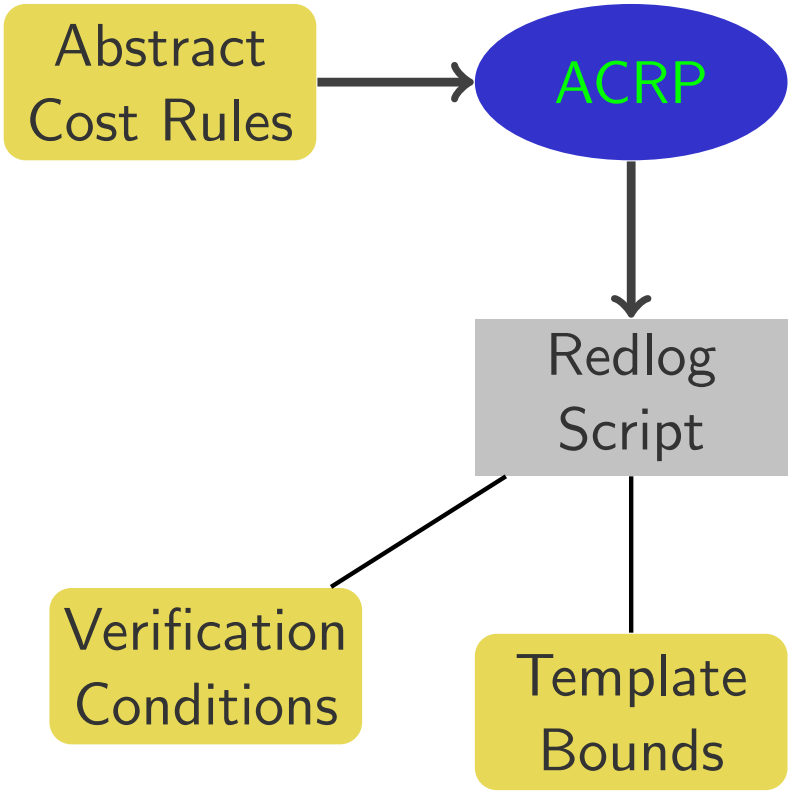
Our prototype

Abstract
Cost Rules

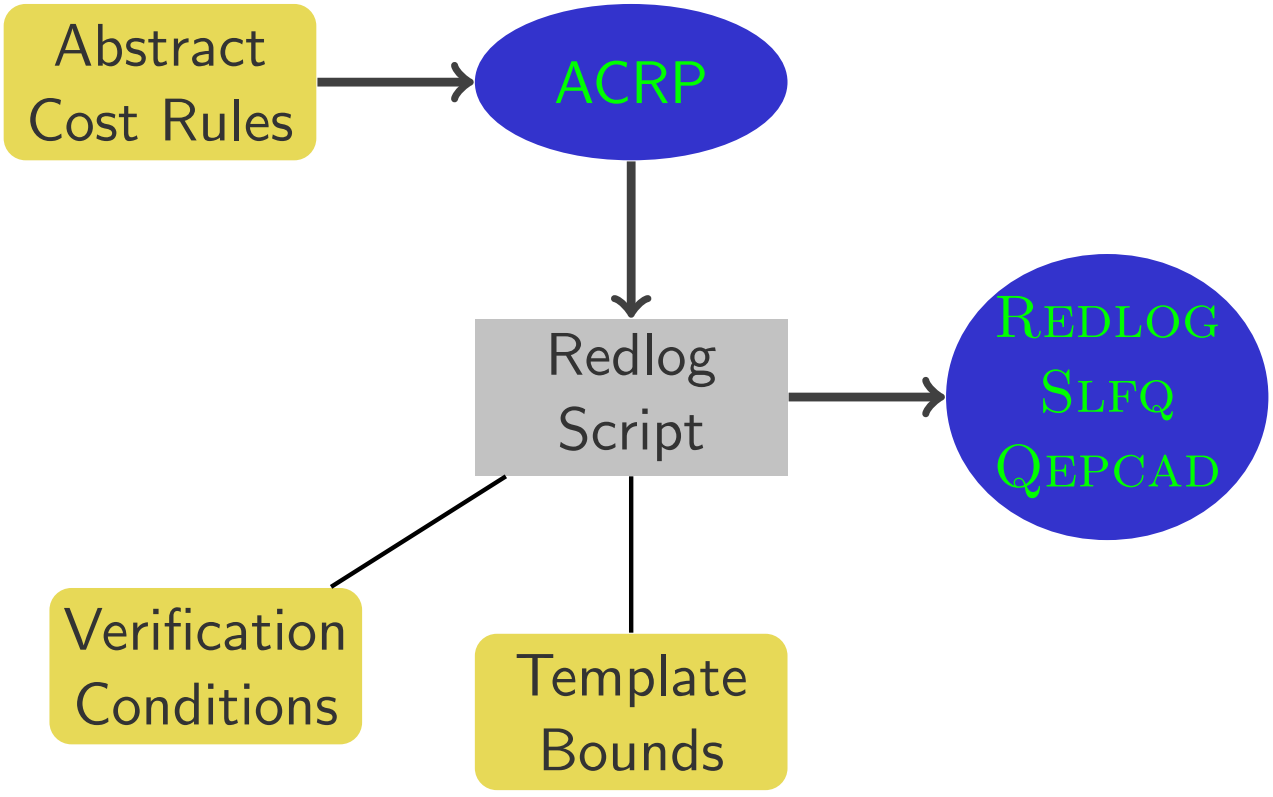
Our prototype



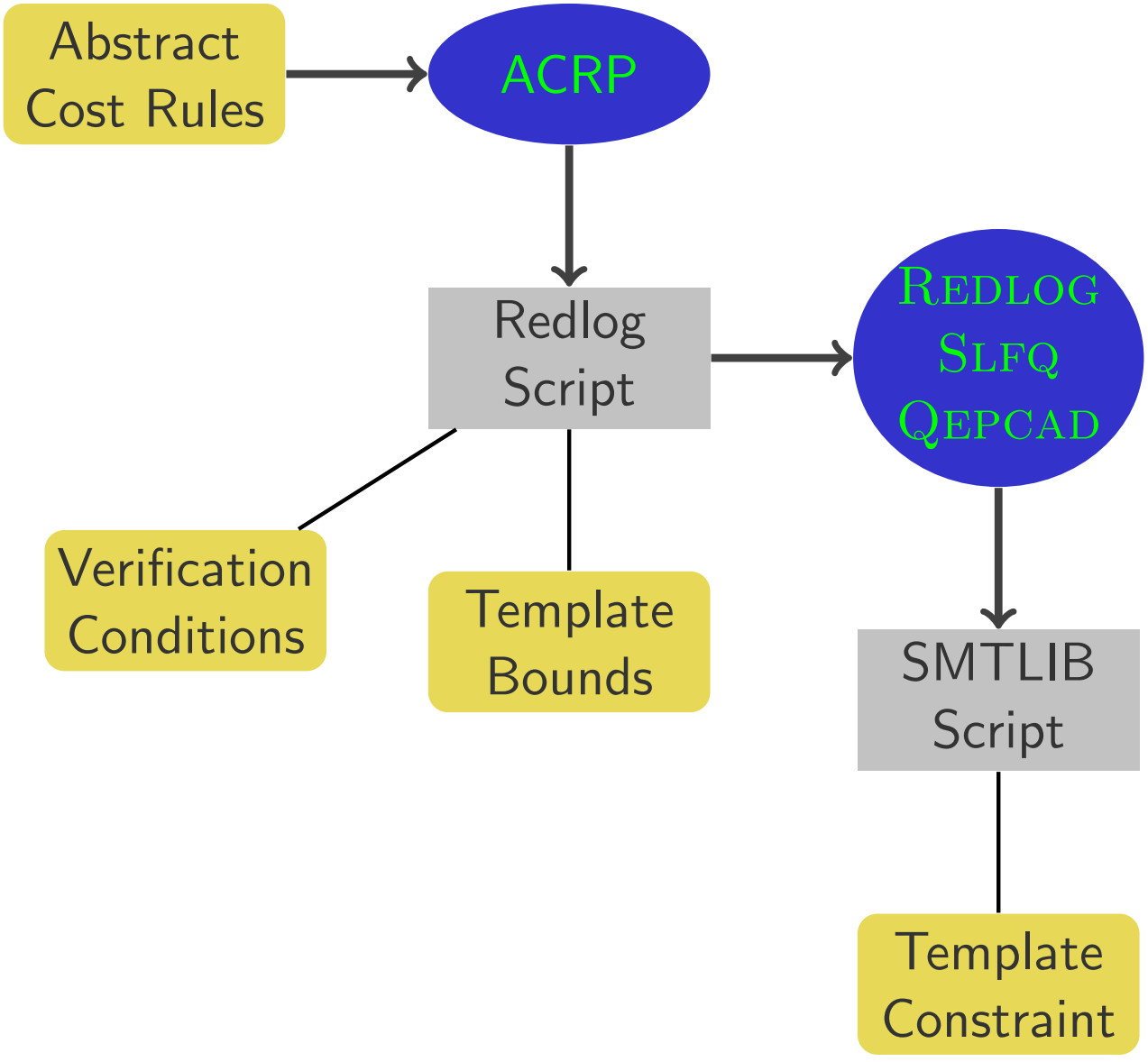
Our prototype



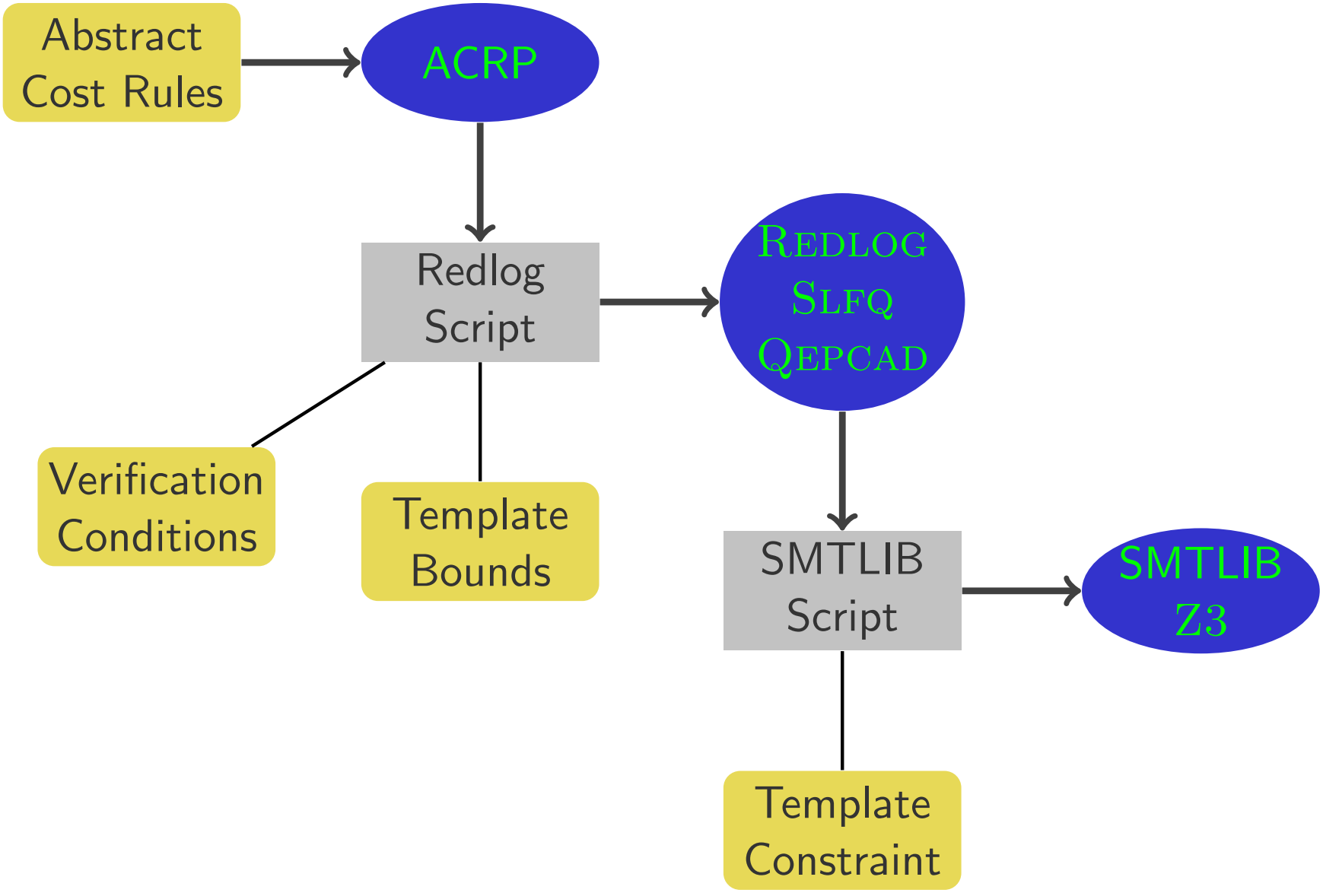
Our prototype



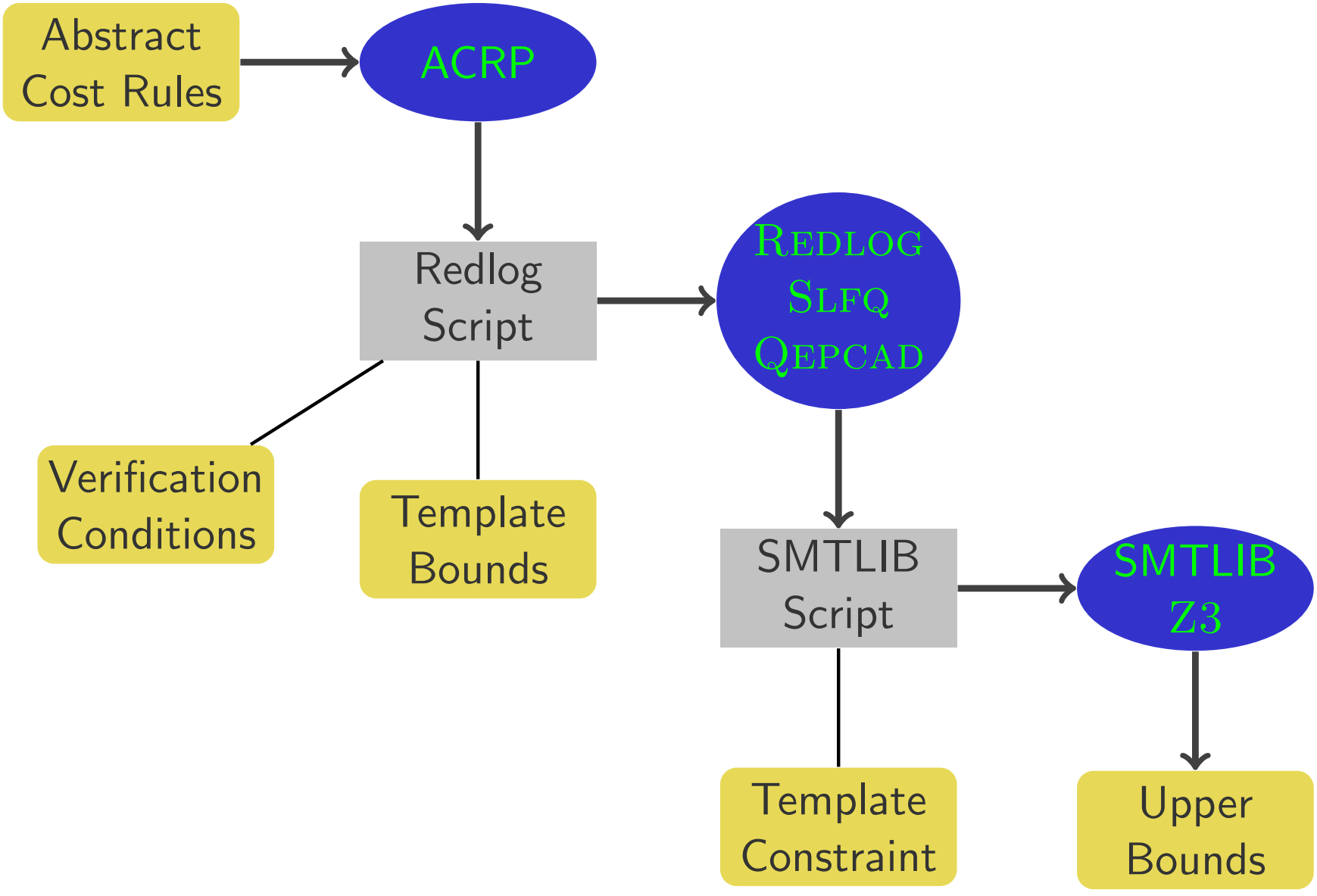
Our prototype



Our prototype



Our prototype



Conclusions

We wanted to understand the differences between different approaches to cost analysis: Classical and Amortised

- Some programs show an Output-Cost codependence
- The amortised approach to cost analysis handles them with precission
- The classical approach ignores these Out-Cost dependencies
- and is unable to get precise bounds for these examples

Our solution:

- **Net-Cost** upper bounds, that take both inputs and outputs
- An approach to *verification* of net-cost Upper Bounds
- its extension to template-based *synthesis*
- Extension to Peak cost of noncumulative resources
- Works also for synthesizing Lower Bounds