Analyzing Runtime Complexity via Innermost Runtime Complexity

Florian Frohn[1] Jürgen Giesl[1]

[1]RWTH Aachen University, Germany

May 11, 2017

Consider the following MAUDE program...

Example

```
mod BASIC-NAT is
   ...
    rl  plus   0     Y => Y .
    rl  plus   s(X)  Y => s(plus X Y) .
    rl  times  0     Y => 0 .
   crl  times  s(X)  Y => plus Z Y if times X Y => Z .
endm
```

...which can be seen as a *Conditional TRS*...

Example

$$\begin{array}{rcl} \mathbf{plus}(0, y) & \to & y \\ \mathbf{plus}(\mathsf{s}(x), y) & \to & \mathsf{s}(\mathbf{plus}(x, y)) \\ \mathbf{times}(0, y) & \to & 0 \\ \mathbf{times}(\mathsf{s}(x), y) & \to & \mathbf{plus}(z, y) \quad \Leftarrow \quad \mathbf{times}(x, y) \approx z \end{array}$$

...which can be seen as a *Conditional TRS*...

Example

$$
\begin{array}{rcl}
\mathbf{plus}(0, y) & \rightarrow & y \\
\mathbf{plus}(\mathsf{s}(x), y) & \rightarrow & \mathsf{s}(\mathbf{plus}(x, y)) \\
\mathbf{times}(0, y) & \rightarrow & 0 \\
\mathbf{times}(\mathsf{s}(x), y) & \rightarrow & \mathbf{plus}(z, y) \quad \Leftarrow \quad \mathbf{times}(x, y) \approx z
\end{array}
$$

Goal: Prove *upper* bound on *worst case* complexity

...which can be transformed to a standard TRS.

Transformation by Cynthia Kop, Aart Middeldorp, and Thomas Sternagel

"Complexity of Conditional Term Rewriting", LMCS '17

...which can be transformed to a standard TRS.

Transformation by Cynthia Kop, Aart Middeldorp, and Thomas Sternagel

"Complexity of Conditional Term Rewriting", LMCS '17

Leading Example $\mathcal{R}_{\textbf{times}}$

$$
\begin{array}{rcl}
\textbf{plus}(0, y, \top, x_2) & \to & y \\
\textbf{plus}(\mathsf{s}(x), y, x_1, \top) & \to & \mathsf{s}(\textbf{plus}(x, y, \top, \top)) \\
\textbf{times}(0, y, \top, x_2) & \to & 0 \\
\textbf{times}(\mathsf{s}(x), y, x_1, \top) & \to & \textbf{times}_2^1(\mathsf{s}(x), y, x_1, \textbf{times}(x, y, \top, \top)) \\
\textbf{times}_2^1(\mathsf{s}(x), y, x_1, z) & \to & \textbf{plus}(z, y, \top, \top)
\end{array}
$$

...which can be transformed to a standard TRS.

Transformation by Cynthia Kop, Aart Middeldorp, and Thomas Sternagel

"Complexity of Conditional Term Rewriting", LMCS '17

Leading Example $\mathcal{R}_{\textbf{times}}$

$$
\begin{array}{rcl}
\textbf{plus}(0, y, \top, x_2) & \to & y \\
\textbf{plus}(\mathsf{s}(x), y, x_1, \top) & \to & \mathsf{s}(\textbf{plus}(x, y, \top, \top)) \\
\textbf{times}(0, y, \top, x_2) & \to & 0 \\
\textbf{times}(\mathsf{s}(x), y, x_1, \top) & \to & \textbf{times}_2^1(\mathsf{s}(x), y, x_1, \textbf{times}(x, y, \top, \top)) \\
\textbf{times}_2^1(\mathsf{s}(x), y, x_1, z) & \to & \textbf{plus}(z, y, \top, \top)
\end{array}
$$

Let's analyze it using leading tools!

...which can be transformed to a standard TRS.

Transformation by Cynthia Kop, Aart Middeldorp, and Thomas Sternagel

"Complexity of Conditional Term Rewriting", LMCS '17

Leading Example $\mathcal{R}_{\mathbf{times}}$

$$
\begin{aligned}
\mathbf{plus}(0, y, \top, x_2) &\rightarrow y \\
\mathbf{plus}(\mathsf{s}(x), y, x_1, \top) &\rightarrow \mathsf{s}(\mathbf{plus}(x, y, \top, \top)) \\
\mathbf{times}(0, y, \top, x_2) &\rightarrow 0 \\
\mathbf{times}(\mathsf{s}(x), y, x_1, \top) &\rightarrow \mathbf{times}_2^1(\mathsf{s}(x), y, x_1, \mathbf{times}(x, y, \top, \top)) \\
\mathbf{times}_2^1(\mathsf{s}(x), y, x_1, z) &\rightarrow \mathbf{plus}(z, y, \top, \top)
\end{aligned}
$$

Let's analyze it using leading tools!

- TcT: timeout (60 s)

...which can be transformed to a standard TRS.

Transformation by Cynthia Kop, Aart Middeldorp, and Thomas Sternagel

"Complexity of Conditional Term Rewriting", LMCS '17

Leading Example $\mathcal{R}_{\mathbf{times}}$

$$
\begin{array}{rcl}
\mathbf{plus}(0, y, \top, x_2) & \to & y \\
\mathbf{plus}(\mathsf{s}(x), y, x_1, \top) & \to & \mathsf{s}(\mathbf{plus}(x, y, \top, \top)) \\
\mathbf{times}(0, y, \top, x_2) & \to & 0 \\
\mathbf{times}(\mathsf{s}(x), y, x_1, \top) & \to & \mathbf{times}_2^1(\mathsf{s}(x), y, x_1, \mathbf{times}(x, y, \top, \top)) \\
\mathbf{times}_2^1(\mathsf{s}(x), y, x_1, z) & \to & \mathbf{plus}(z, y, \top, \top)
\end{array}
$$

Let's analyze it using leading tools!

- TcT: timeout (60 s)
- AProVE: *full* rewriting not supported

...which can be transformed to a standard TRS.

Transformation by Cynthia Kop, Aart Middeldorp, and Thomas Sternagel

"Complexity of Conditional Term Rewriting", LMCS '17

Leading Example $\mathcal{R}_{\mathbf{times}}$

$$\begin{array}{rcl} \mathbf{plus}(0, y, \top, x_2) & \to & y \\ \mathbf{plus}(\mathsf{s}(x), y, x_1, \top) & \to & \mathsf{s}(\mathbf{plus}(x, y, \top, \top)) \\ \mathbf{times}(0, y, \top, x_2) & \to & 0 \\ \mathbf{times}(\mathsf{s}(x), y, x_1, \top) & \to & \mathbf{times}_2^1(\mathsf{s}(x), y, x_1, \mathbf{times}(x, y, \top, \top)) \\ \mathbf{times}_2^1(\mathsf{s}(x), y, x_1, z) & \to & \mathbf{plus}(z, y, \top, \top) \end{array}$$

Let's analyze it using leading tools!

- TcT: timeout (60 s)
- AProVE: *full* rewriting not supported

But: $\mathcal{O}(n^3)$ for innermost rewriting – can we exploit that?

...which can be transformed to a standard TRS.

Transformation by Cynthia Kop, Aart Middeldorp, and Thomas Sternagel

"Complexity of Conditional Term Rewriting", LMCS '17

Leading Example $\mathcal{R}_{\mathbf{times}}$

$$
\begin{array}{rcl}
\mathbf{plus}(0, y, \top, x_2) & \rightarrow & y \\
\mathbf{plus}(\mathsf{s}(x), y, x_1, \top) & \rightarrow & \mathsf{s}(\mathbf{plus}(x, y, \top, \top)) \\
\mathbf{times}(0, y, \top, x_2) & \rightarrow & 0 \\
\mathbf{times}(\mathsf{s}(x), y, x_1, \top) & \rightarrow & \mathbf{times}_2^1(\mathsf{s}(x), y, x_1, \mathbf{times}(x, y, \top, \top)) \\
\mathbf{times}_2^1(\mathsf{s}(x), y, x_1, z) & \rightarrow & \mathbf{plus}(z, y, \top, \top)
\end{array}
$$

Let's analyze it using leading tools!

- TcT: timeout (60 s)
- AProVE: *full* rewriting not supported

But: $\mathcal{O}(n^3)$ for innermost rewriting – can we exploit that?

...which can be transformed to a standard TRS.

Transformation by Cynthia Kop, Aart Middeldorp, and Thomas Sternagel

"Complexity of Conditional Term Rewriting", LMCS '17

Leading Example $\mathcal{R}_{\textbf{times}}$

$$
\begin{array}{rcl}
\mathbf{plus}(0, y) & \to & y \\
\mathbf{plus}(\mathsf{s}(x), y) & \to & \mathsf{s}(\mathbf{plus}(x, y)) \\
\mathbf{times}(0, y, \top, x_2) & \to & 0 \\
\mathbf{times}(\mathsf{s}(x), y, x_1, \top) & \to & \mathbf{times}_2^1(\mathsf{s}(x), y, x_1, \mathbf{times}(x, y, \top, \top)) \\
\mathbf{times}_2^1(\mathsf{s}(x), y, x_1, z) & \to & \mathbf{plus}(z, y)
\end{array}
$$

Let's analyze it using leading tools!

- TcT: timeout (60 s)
- AProVE: *full* rewriting not supported

But: $\mathcal{O}(n^3)$ for innermost rewriting – can we exploit that?

...which can be transformed to a standard TRS.

Transformation by Cynthia Kop, Aart Middeldorp, and Thomas Sternagel

"Complexity of Conditional Term Rewriting", LMCS '17

Leading Example $\mathcal{R}_{\textbf{times}}$

$$
\begin{aligned}
\textbf{plus}(0, y) &\to y \\
\textbf{plus}(\mathsf{s}(x), y) &\to \mathsf{s}(\textbf{plus}(x, y)) \\
\textbf{times}(0, y, \top, x_2) &\to 0 \\
\textbf{times}(\mathsf{s}(x), y, x_1, \top) &\to \textbf{plus}(\textbf{times}(x, y, \top, \top), y)
\end{aligned}
$$

Let's analyze it using leading tools!

- TcT: timeout (60 s)
- AProVE: *full* rewriting not supported

But: $\mathcal{O}(n^3)$ for innermost rewriting – can we exploit that?

...which can be transformed to a standard TRS.

Transformation by Cynthia Kop, Aart Middeldorp, and Thomas Sternagel

"Complexity of Conditional Term Rewriting", LMCS '17

Leading Example $\mathcal{R}_{\textbf{times}}$

$$
\begin{aligned}
\mathbf{plus}(0, y) &\to y \\
\mathbf{plus}(\mathsf{s}(x), y) &\to \mathsf{s}(\mathbf{plus}(x, y)) \\
\mathbf{times}(0, y) &\to 0 \\
\mathbf{times}(\mathsf{s}(x), y) &\to \mathbf{plus}(\mathbf{times}(x, y), y)
\end{aligned}
$$

Let's analyze it using leading tools!

- TcT: timeout (60 s)
- AProVE: *full* rewriting not supported

But: $\mathcal{O}(n^3)$ for innermost rewriting – can we exploit that?

Outline

1. Preliminaries
   - rc and irc
   - NDG Rewriting
2. Handling Constructor Systems
3. Handling Non-Constructor Systems
4. Experimental Results, Conclusion

rc and irc

- rc maps $n \in \mathbb{N}$ to the length of the longest rewrite sequence s.t.

rc and irc

- rc maps $n \in \mathbb{N}$ to the length of the longest rewrite sequence s.t.
  - (A) size of start term bounded by n
  - (B) start term basic

rc and irc

- rc maps $n \in \mathbb{N}$ to the length of the longest rewrite sequence s.t.
  (A) size of start term bounded by n
  (B) start term basic

Example

- $\textbf{plus}(0, \mathsf{s}(0))$

rc and irc

- rc maps $n \in \mathbb{N}$ to the length of the longest rewrite sequence s.t.
  (A) size of start term bounded by n
  (B) start term basic

Example

- $\mathbf{plus}(0, \mathsf{s}(0))$ ✓

rc and irc

- rc maps $n \in \mathbb{N}$ to the length of the longest rewrite sequence s.t.
  (A) size of start term bounded by n
  (B) start term basic

Example

- $\mathbf{plus}(0, \mathsf{s}(0))$ ✓
- $\mathbf{plus}(0, \mathbf{plus}(0, \mathsf{s}(0)))$

rc and irc

- rc maps $n \in \mathbb{N}$ to the length of the longest rewrite sequence s.t.
  (A) size of start term bounded by n
  (B) start term basic

Example

- $\mathbf{plus}(0, \mathsf{s}(0))$ ✓
- $\mathbf{plus}(0, \mathbf{plus}(0, \mathsf{s}(0)))$ ✗

rc and irc

- rc maps $n \in \mathbb{N}$ to the length of the longest rewrite sequence s.t.
  (A) size of start term bounded by n
  (B) start term basic
- irc: similar, but just considers innermost sequences

Example

- $\mathbf{plus}(0, \mathsf{s}(0))$ ✓
- $\mathbf{plus}(0, \mathbf{plus}(0, \mathsf{s}(0)))$ ✗

Outline

NDG Rewriting

By Jaco van de Pol and Hans Zantema:

"Generalized innermost rewriting" (RTA '05)

- Goal: Implement rewriting efficiently

NDG Rewriting

By Jaco van de Pol and Hans Zantema:

"Generalized innermost rewriting" (RTA '05)

- Goal: Implement rewriting efficiently
- Idea: No duplication of nested defined symbols

NDG Rewriting

By Jaco van de Pol and Hans Zantema:

"Generalized innermost rewriting" (RTA '05)

- Goal: Implement rewriting efficiently
- Idea: No duplication of nested defined symbols

Leading Example $\mathcal{R}_{\textbf{times}}$

$$\begin{array}{rcl} \textbf{plus}(0, y) & \to & y \\ \textbf{plus}(\mathsf{s}(x), y) & \to & \mathsf{s}(\textbf{plus}(x, y)) \\ \textbf{times}(0, y) & \to & 0 \\ \textbf{times}(\mathsf{s}(x), y) & \to & \textbf{plus}(\textbf{times}(x, y), y) \end{array}$$

NDG Rewriting

By Jaco van de Pol and Hans Zantema:

"Generalized innermost rewriting" (RTA '05)

- Goal: Implement rewriting efficiently
- Idea: No duplication of nested defined symbols

Leading Example $\mathcal{R}_{\textbf{times}}$

$$
\begin{array}{rcl}
\textbf{plus}(0, y) & \to & y \\
\textbf{plus}(\mathsf{s}(x), y) & \to & \mathsf{s}(\textbf{plus}(x, y)) \\
\textbf{times}(0, y) & \to & 0 \\
\textbf{times}(\mathsf{s}(x), y) & \to & \textbf{plus}(\textbf{times}(x, y), y)
\end{array}
$$

Example

- $\textbf{times}(\mathsf{s}(0), \textbf{plus}(0, 0)) \to \textbf{plus}(\textbf{times}(0, \textbf{plus}(0, 0)), \textbf{plus}(0, 0))$

NDG Rewriting

By Jaco van de Pol and Hans Zantema:

"Generalized innermost rewriting" (RTA '05)

- Goal: Implement rewriting efficiently
- Idea: No duplication of nested defined symbols

Leading Example $\mathcal{R}_{\mathbf{times}}$

$$
\begin{array}{rcl}
\mathbf{plus}(0, y) & \to & y \\
\mathbf{plus}(\mathsf{s}(x), y) & \to & \mathsf{s}(\mathbf{plus}(x, y)) \\
\mathbf{times}(0, y) & \to & 0 \\
\mathbf{times}(\mathsf{s}(x), y) & \to & \mathbf{plus}(\mathbf{times}(x, y), y)
\end{array}
$$

Example

- $\mathbf{times}(\mathsf{s}(0), \mathbf{plus}(0, 0)) \to \mathbf{plus}(\mathbf{times}(0, \mathbf{plus}(0, 0)), \mathbf{plus}(0, 0))$ ✘

NDG Rewriting

By Jaco van de Pol and Hans Zantema:

"Generalized innermost rewriting" (RTA '05)

- Goal: Implement rewriting efficiently
- Idea: No duplication of nested defined symbols

Leading Example $\mathcal{R}_{\textbf{times}}$

$$\begin{array}{rcl} \textbf{plus}(0, y) & \to & y \\ \textbf{plus}(\mathsf{s}(x), y) & \to & \mathsf{s}(\textbf{plus}(x, y)) \\ \textbf{times}(0, y) & \to & 0 \\ \textbf{times}(\mathsf{s}(x), y) & \to & \textbf{plus}(\textbf{times}(x, y), y) \end{array}$$

Example

- $\textbf{times}(\mathsf{s}(0), \textbf{plus}(0, 0)) \to \textbf{plus}(\textbf{times}(0, \textbf{plus}(0, 0)), \textbf{plus}(0, 0))$ ✘
- $\textbf{plus}(\mathsf{s}(0), \textbf{plus}(0, 0)) \to \mathsf{s}(\textbf{plus}(0, \textbf{plus}(0, 0)))$

NDG Rewriting

By Jaco van de Pol and Hans Zantema:

"Generalized innermost rewriting" (RTA '05)

- Goal: Implement rewriting efficiently
- Idea: No duplication of nested defined symbols

Leading Example $\mathcal{R}_{\textbf{times}}$

$$
\begin{array}{rcl}
\mathbf{plus}(0, y) & \to & y \\
\mathbf{plus}(\mathsf{s}(x), y) & \to & \mathsf{s}(\mathbf{plus}(x, y)) \\
\mathbf{times}(0, y) & \to & 0 \\
\mathbf{times}(\mathsf{s}(x), y) & \to & \mathbf{plus}(\mathbf{times}(x, y), y)
\end{array}
$$

Example

- $\mathbf{times}(\mathsf{s}(0), \mathbf{plus}(0, 0)) \to \mathbf{plus}(\mathbf{times}(0, \mathbf{plus}(0, 0)), \mathbf{plus}(0, 0))$ ✘
- $\mathbf{plus}(\mathsf{s}(0), \mathbf{plus}(0, 0)) \to \mathsf{s}(\mathbf{plus}(0, \mathbf{plus}(0, 0)))$ ✓

NDG Rewriting is cheap!

Theorem (Pol et. al, RTA '05)

NDG rewriting is at least as efficient as innermost rewriting.

NDG Rewriting is cheap!

Theorem (Pol et. al, RTA '05)

NDG rewriting is at least as efficient as innermost rewriting.

$\curvearrowright$ *all* sequences ndg $\implies$ innermost is the worst case

NDG Rewriting is cheap!

Theorem (Pol et. al, RTA '05)

NDG rewriting is at least as efficient as innermost rewriting.

Reminder: rc

rc maps n to the length of the longest rewrite sequence s.t.

(A) size of start term bounded by n

(B) start term basic

$\curvearrowright$ *all* sequences ndg $\implies$ innermost is the worst case

NDG Rewriting is cheap!

Theorem (Pol et. al, RTA '05)

NDG rewriting is at least as efficient as innermost rewriting.

Reminder: rc

rc maps n to the length of the longest rewrite sequence s.t.

(A) size of start term bounded by n

(B) start term basic

- ↷ *all* sequences ndg $\implies$ innermost is the worst case
- ↷ *all* sequences *starting with basic terms* ndg $\implies$ rc $=$ irc

NDG Rewriting is cheap!

Theorem (Pol et. al, RTA '05)

NDG rewriting is at least as efficient as innermost rewriting.

Reminder: rc

rc maps n to the length of the longest rewrite sequence s.t.

(A) size of start term bounded by n

(B) start term basic

- ↷ *all* sequences ndg $\implies$ innermost is the worst case
- ↷ *all* sequences *starting with basic terms* ndg $\implies$ rc $=$ irc
- ↷ **Goal: Prove that all sequences starting with basic terms are ndg**

NDG Rewriting is cheap!

Theorem (Pol et. al, RTA '05)

NDG rewriting is at least as efficient as innermost rewriting.

Reminder: rc

rc maps n to the length of the longest rewrite sequence s.t.

(A) size of start term bounded by n

(B) start term basic

- *all* sequences ndg $\Longrightarrow$ innermost is the worst case
- *all* sequences *starting with basic terms* ndg $\Longrightarrow$ rc $=$ irc
- **Goal: Prove that all sequences starting with basic terms are ndg**
  Use irc **techniques to analyze** rc

Outline

“Proving” ndg-ness by hand

Leading Example $\mathcal{R}_{\textbf{times}}$

$$
\begin{aligned}
\textbf{plus}(0, y) &\rightarrow y \\
\textbf{plus}(\mathsf{s}(x), y) &\rightarrow \mathsf{s}(\textbf{plus}(x, y)) \\
\textbf{times}(0, y) &\rightarrow 0 \\
\textbf{times}(\mathsf{s}(x), y) &\rightarrow \textbf{plus}(\textbf{times}(x, y), y)
\end{aligned}
$$

“Proving” ndg-ness by hand

Leading Example $\mathcal{R}_{\textbf{times}}$

$$
\begin{aligned}
\textbf{plus}(0, y) &\rightarrow y \\
\textbf{plus}(\mathsf{s}(x), y) &\rightarrow \mathsf{s}(\textbf{plus}(x, y)) \\
\textbf{times}(0, y) &\rightarrow 0 \\
\textbf{times}(\mathsf{s}(x), y) &\rightarrow \textbf{plus}(\textbf{times}(x, y), y)
\end{aligned}
$$

- **times**(...)

"Proving" ndg-ness by hand

Leading Example $\mathcal{R}_{\textbf{times}}$

$$\begin{array}{rcl} \textbf{plus}(0, y) & \to & y \\ \textbf{plus}(\mathsf{s}(x), y) & \to & \mathsf{s}(\textbf{plus}(x, y)) \\ \textbf{times}(0, y) & \to & 0 \\ \textbf{times}(\mathsf{s}(x), y) & \to & \textbf{plus}(\textbf{times}(x, y), y) \end{array}$$

- **times**(...)
  - nesting below **plus**' first argument

"Proving" ndg-ness by hand

Leading Example $\mathcal{R}_{\textbf{times}}$

$$\begin{aligned}
\textbf{plus}(0, y) &\to y \\
\textbf{plus}(\mathsf{s}(x), y) &\to \mathsf{s}(\textbf{plus}(x, y)) \\
\textbf{times}(0, y) &\to 0 \\
\textbf{times}(\mathsf{s}(x), y) &\to \textbf{plus}(\textbf{times}(x, y), y)
\end{aligned}$$

- **times**(...)
  - nesting below **plus**' first argument $\textbf{plus}(\textbf{times}(x, y), y)$

"Proving" ndg-ness by hand

Leading Example $\mathcal{R}_{\mathbf{times}}$

$$
\begin{aligned}
\mathbf{plus}(0, y) &\to y \\
\mathbf{plus}(\mathsf{s}(x), y) &\to \mathsf{s}(\mathbf{plus}(x, y)) \\
\mathbf{times}(0, y) &\to 0 \\
\mathbf{times}(\mathsf{s}(x), y) &\to \mathbf{plus}(\mathbf{times}(x, y), y)
\end{aligned}
$$

- **times**(...)
  - nesting below **plus**' first argument $\mathbf{plus}(\square, y)$

"Proving" ndg-ness by hand

Leading Example $\mathcal{R}_{\textbf{times}}$

$$\begin{array}{rcl} \textbf{plus}(0, y) & \to & y \\ \textbf{plus}(\mathsf{s}(x), y) & \to & \mathsf{s}(\textbf{plus}(x, y)) \\ \textbf{times}(0, y) & \to & 0 \\ \textbf{times}(\mathsf{s}(x), y) & \to & \textbf{plus}(\textbf{times}(x, y), y) \end{array}$$

- **times**(...)
  - nesting below **plus**' first argument $\textbf{plus}(\Box, y)$
  - duplication of **times**' second argument

“Proving” ndg-ness by hand

Leading Example $\mathcal{R}_{\textbf{times}}$

$$\begin{aligned}
\textbf{plus}(0, y) &\to y \\
\textbf{plus}(\mathsf{s}(x), y) &\to \mathsf{s}(\textbf{plus}(x, y)) \\
\textbf{times}(0, y) &\to 0 \\
\textbf{times}(\mathsf{s}(x), y) &\to \textbf{plus}(\textbf{times}(x, y), y)
\end{aligned}$$

- **times**(...)
  - nesting below **plus**’ first argument $\textbf{plus}(\square, y)$
  - duplication of **times**’ second argument $\textbf{times}(\mathsf{s}(x), y)$

“Proving” ndg-ness by hand

Leading Example $\mathcal{R}_{\textbf{times}}$

$$
\begin{array}{rcl}
\textbf{plus}(0, y) & \to & y \\
\textbf{plus}(\mathsf{s}(x), y) & \to & \mathsf{s}(\textbf{plus}(x, y)) \\
\textbf{times}(0, y) & \to & 0 \\
\textbf{times}(\mathsf{s}(x), y) & \to & \textbf{plus}(\textbf{times}(x, y), y)
\end{array}
$$

- **times**(...)
  - nesting below **plus**’ first argument — $\textbf{plus}(\Box, y)$
  - duplication of **times**’ second argument — $\textbf{times}(\mathsf{s}(x), \Box)$

"Proving" ndg-ness by hand

Leading Example $\mathcal{R}_{\textbf{times}}$

$$
\begin{aligned}
\textbf{plus}(0, y) &\to y \\
\textbf{plus}(\mathsf{s}(x), y) &\to \mathsf{s}(\textbf{plus}(x, y)) \\
\textbf{times}(0, y) &\to 0 \\
\textbf{times}(\mathsf{s}(x), y) &\to \textbf{plus}(\textbf{times}(x, y), y)
\end{aligned}
$$

- $\textbf{times}(\ldots)$
  - nesting below **plus**' first argument $\textbf{plus}(\square, y)$
  - duplication of **times**' second argument $\textbf{times}(\mathsf{s}(x), \square)$
- $\textbf{plus}(\ldots)$

"Proving" ndg-ness by hand

Leading Example $\mathcal{R}_{\textbf{times}}$

$$\begin{array}{rcl}
\textbf{plus}(0, y) & \to & y \\
\textbf{plus}(s(x), y) & \to & s(\textbf{plus}(x, y)) \\
\textbf{times}(0, y) & \to & 0 \\
\textbf{times}(s(x), y) & \to & \textbf{plus}(\textbf{times}(x, y), y)
\end{array}$$

- **times**(...)
  - nesting below **plus**' first argument — $\textbf{plus}(\Box, y)$
  - duplication of **times**' second argument — $\textbf{times}(s(x), \Box)$
- **plus**(...)

“Proving” ndg-ness by hand

Leading Example $\mathcal{R}_{\textbf{times}}$

$$
\begin{aligned}
\textbf{plus}(0, y) &\to y \\
\textbf{plus}(\mathsf{s}(x), y) &\to \mathsf{s}(\textbf{plus}(x, y)) \\
\textbf{times}(0, y) &\to 0 \\
\textbf{times}(\mathsf{s}(x), y) &\to \textbf{plus}(\textbf{times}(x, y), y)
\end{aligned}
$$

- **times**(...)
  - nesting below **plus**’ first argument — **plus**($\square$, y)
  - duplication of **times**’ second argument — **times**(s(x), $\square$)
- **plus**(...)

plus($\square$, y)

"Proving" ndg-ness by hand

Leading Example $\mathcal{R}_{\textbf{times}}$

$$
\begin{array}{rcl}
\textbf{plus}(0, y) & \to & y \\
\textbf{plus}(\mathsf{s}(x), y) & \to & \mathsf{s}(\textbf{plus}(x, y)) \\
\textbf{times}(0, y) & \to & 0 \\
\textbf{times}(\mathsf{s}(x), y) & \to & \textbf{plus}(\textbf{times}(x, y), y)
\end{array}
$$

- **times**(...)
  - nesting below **plus**' first argument $\textbf{plus}(\Box, y)$
  - duplication of **times**' second argument $\textbf{times}(\mathsf{s}(x), \Box)$
- **plus**(...)
  - no (further) nesting

"Proving" ndg-ness by hand

Leading Example $\mathcal{R}_{\textbf{times}}$

$$
\begin{array}{rcl}
\textbf{plus}(0, y) & \to & y \\
\textbf{plus}(\mathsf{s}(x), y) & \to & \mathsf{s}(\textbf{plus}(x, y)) \\
\textbf{times}(0, y) & \to & 0 \\
\textbf{times}(\mathsf{s}(x), y) & \to & \textbf{plus}(\textbf{times}(x, y), y)
\end{array}
$$

- **times**(...)
  - nesting below **plus**' first argument $\qquad \textbf{plus}(\square, y)$
  - duplication of **times**' second argument $\qquad \textbf{times}(\mathsf{s}(x), \square)$
- **plus**(...)
  - no (further) nesting
  - no duplication

"Proving" ndg-ness by hand

Leading Example $\mathcal{R}_{\textbf{times}}$

$$
\begin{aligned}
\textbf{plus}(0, y) &\rightarrow y \\
\textbf{plus}(\mathsf{s}(x), y) &\rightarrow \mathsf{s}(\textbf{plus}(x, y)) \\
\textbf{times}(0, y) &\rightarrow 0 \\
\textbf{times}(\mathsf{s}(x), y) &\rightarrow \textbf{plus}(\textbf{times}(x, y), y)
\end{aligned}
$$

- **times**(...)
  - nesting below **plus**' first argument — $\textbf{plus}(\square, y)$
  - duplication of **times**' second argument — $\textbf{times}(\mathsf{s}(x), \square)$
- **plus**(...)
  - no (further) nesting
  - no duplication
- $\textbf{plus}(\square, y)$ and $\textbf{times}(\mathsf{s}(x), \square)$ don't "overlap"

"Proving" ndg-ness by hand

Leading Example $\mathcal{R}_{\textbf{times}}$

$$\textbf{plus}(0, y) \to y$$

pl...

tim...

Reminder

no duplication of defined symbols

$\curvearrowright$ innermost rewriting is worst

$\curvearrowright$ rc $=$ irc

$\curvearrowright$ irc techniques applicable for rc

$\textbf{plus}(\Box, y)$

$\textbf{times}(\mathsf{s}(x), \Box)$

-
-
  - no (further) nesting
  - no duplication
- $\textbf{plus}(\Box, y)$ and $\textbf{times}(\mathsf{s}(x), \Box)$ don't "overlap"

Proving ndg-ness automatically

Representing sets of contexts

C matches D if

Proving ndg-ness automatically

Representing sets of contexts

C matches D if

- $C[x]\sigma = D$

Proving ndg-ness automatically

Representing sets of contexts

C matches D if

- $C[x]\sigma = D$
- $\Box$ in D below $\Box$ in C

Proving ndg-ness automatically

Representing sets of contexts

C matches D if

- $C[x]\sigma = D$
- $\Box$ in D below $\Box$ in C

Example

- $\mathbf{plus}(x, \Box)$ does not match $\mathbf{plus}(\mathrm{s}(\Box), y)$

Proving ndg-ness automatically

Representing sets of contexts

C matches D if

- $C[x]\sigma = D$
- $\Box$ in D below $\Box$ in C

Example

- $\mathbf{plus}(\Box, y)$ matches $\mathbf{plus}(\mathrm{s}(\Box), y)$

Proving ndg-ness automatically

Representing sets of contexts

C matches D if

- $C[x]\sigma = D$
- $\Box$ in D below $\Box$ in C

Example

- $\mathbf{plus}(\Box, y)$ matches $\mathbf{plus}(\mathsf{s}(\Box), y)$
- Intuition: $\mathbf{plus}(\Box, y)$ represents "marked" terms

Proving ndg-ness automatically

Representing sets of contexts

C matches D if

- $C[x]\sigma = D$
- $\Box$ in D below $\Box$ in C

Example

- $\mathbf{plus}(\Box, y)$ matches $\mathbf{plus}(\mathsf{s}(\Box), y)$
- Intuition: $\mathbf{plus}(\Box, y)$ represents "marked" terms
  $\mathbf{plus}(\mathbf{times}(x, z), y)$,

Proving ndg-ness automatically

Representing sets of contexts

C matches D if

- $C[x]\sigma = D$
- $\Box$ in D below $\Box$ in C

Example

- $\mathbf{plus}(\Box, y)$ matches $\mathbf{plus}(\mathrm{s}(\Box), y)$
- Intuition: $\mathbf{plus}(\Box, y)$ represents "marked" terms

  $\mathbf{plus}(\mathbf{times}(x, z), y)$,

  $\mathbf{plus}(\mathrm{s}(\mathbf{times}(0, 0)), 0)$, ...

Proving ndg-ness automatically

Representing sets of contexts

C matches D if

- $C[x]\sigma = D$
- $\square$ in D below $\square$ in C

Example

- $\mathbf{plus}(\square, y)$ matches $\mathbf{plus}(\mathsf{s}(\square), y)$
- Intuition: $\mathbf{plus}(\square, y)$ represents "marked" terms
  $\mathbf{plus}(\mathbf{times}(x, z), y)$,
  $\mathbf{plus}(\mathsf{s}(\mathbf{times}(0, 0)), 0)$, ...

Goal: compute sets of contexts *Dup* and *Def*

Proving ndg-ness automatically

Representing sets of contexts

C matches D if

- $C[x]\sigma = D$
- $\Box$ in D below $\Box$ in C

Example

- **plus**$(\Box, y)$ matches **plus**$(\mathsf{s}(\Box), y)$
- Intuition: **plus**$(\Box, y)$ represents "marked" terms
  plus(**times**(x, z), y),
  plus(s(**times**$(0, 0)$), 0), ...

Goal: compute sets of contexts *Dup* and *Def*
Dup and *Def* don't overlap $\curvearrowright$ rc $=$ irc

Proving ndg-ness automatically

Representing sets of contexts

C matches D if

- $C[x]\sigma = D$
- $\Box$ in D below $\Box$ in C

Overlapping contexts

C and D overlap if both match some E

Example

- $\mathbf{plus}(\Box, y)$ matches $\mathbf{plus}(\mathsf{s}(\Box), y)$
- Intuition: $\mathbf{plus}(\Box, y)$ represents "marked" terms
  $\mathbf{plus}(\mathbf{times}(x, z), y)$,
  $\mathbf{plus}(\mathsf{s}(\mathbf{times}(0, 0)), 0)$, ...

Goal: compute sets of contexts *Dup* and *Def*
Dup and *Def* don't overlap $\curvearrowright$ rc $=$ irc

The easy one: Computing *Dup*

Algorithm

Example

Leading Example $\mathcal{R}_{\textbf{times}}$

$$
\begin{aligned}
\textbf{plus}(0, y) &\rightarrow y \\
\textbf{plus}(\mathsf{s}(x), y) &\rightarrow \mathsf{s}(\textbf{plus}(x, y)) \\
\textbf{times}(0, y) &\rightarrow 0 \\
\textbf{times}(\mathsf{s}(x), y) &\rightarrow \textbf{plus}(\textbf{times}(x, y), y)
\end{aligned}
$$

The easy one: Computing *Dup*

Algorithm

- collect left-hand sides of rules with non-linear right-hand sides

Example

Leading Example $\mathcal{R}_{\textbf{times}}$

$$
\begin{aligned}
\textbf{plus}(0, y) &\rightarrow y \\
\textbf{plus}(\mathsf{s}(x), y) &\rightarrow \mathsf{s}(\textbf{plus}(x, y)) \\
\textbf{times}(0, y) &\rightarrow 0 \\
\textbf{times}(\mathsf{s}(x), y) &\rightarrow \textbf{plus}(\textbf{times}(x, y), y)
\end{aligned}
$$

The easy one: Computing *Dup*

Algorithm

- collect left-hand sides of rules with non-linear right-hand sides

Example

Leading Example $\mathcal{R}_{\textbf{times}}$

$$
\begin{aligned}
\textbf{plus}(0, y) &\rightarrow y \\
\textbf{plus}(\mathsf{s}(x), y) &\rightarrow \mathsf{s}(\textbf{plus}(x, y)) \\
\textbf{times}(0, y) &\rightarrow 0 \\
\textbf{times}(\mathsf{s}(x), y) &\rightarrow \textbf{plus}(\textbf{times}(x, y), y)
\end{aligned}
$$

The easy one: Computing *Dup*

Algorithm

- collect left-hand sides of rules with non-linear right-hand sides
- replace occurrences of duplicated variables in left-hand sides with $\square$

Example

Leading Example $\mathcal{R}_{\textbf{times}}$

$$\begin{array}{rcl} \textbf{plus}(0, y) & \to & y \\ \textbf{plus}(\mathsf{s}(x), y) & \to & \mathsf{s}(\textbf{plus}(x, y)) \\ \textbf{times}(0, y) & \to & 0 \\ \textbf{times}(\mathsf{s}(x), y) & \to & \textbf{plus}(\textbf{times}(x, y), y) \end{array}$$

The easy one: Computing *Dup*

Algorithm

- collect left-hand sides of rules with non-linear right-hand sides
- replace occurrences of duplicated variables in left-hand sides with $\square$

Example

Leading Example $\mathcal{R}_{\mathbf{times}}$

$$
\begin{aligned}
\mathbf{plus}(0, y) &\to y \\
\mathbf{plus}(\mathsf{s}(x), y) &\to \mathsf{s}(\mathbf{plus}(x, y)) \\
\mathbf{times}(0, y) &\to 0 \\
\mathbf{times}(\mathsf{s}(x), y) &\to \mathbf{plus}(\mathbf{times}(x, y), y)
\end{aligned}
$$

The easy one: Computing *Dup*

Algorithm

- collect left-hand sides of rules with non-linear right-hand sides
- replace occurrences of duplicated variables in left-hand sides with $\Box$

Example

$Dup = \{\mathbf{times}(\mathsf{s}(x), \Box)\}$

Leading Example $\mathcal{R}_{\mathbf{times}}$

$$
\begin{aligned}
\mathbf{plus}(0, y) &\to y \\
\mathbf{plus}(\mathsf{s}(x), y) &\to \mathsf{s}(\mathbf{plus}(x, y)) \\
\mathbf{times}(0, y) &\to 0 \\
\mathbf{times}(\mathsf{s}(x), y) &\to \mathbf{plus}(\mathbf{times}(x, y), y)
\end{aligned}
$$

The easy one: Computing *Dup*

Reminder

Dup and *Def* don't overlap

$\curvearrowright$ no duplication of defined symbols

$\curvearrowright$ rc = irc

$\curvearrowright$ irc techniques applicable for rc

Algorithm

- collect left-har
- replace occurre

Example

$Dup = \{\mathbf{times}(\mathsf{s}(x), \Box)\}$

Leading Example $\mathcal{R}_{\mathbf{times}}$

$$
\begin{aligned}
\mathbf{plus}(0, y) &\to y \\
\mathbf{plus}(\mathsf{s}(x), y) &\to \mathsf{s}(\mathbf{plus}(x, y)) \\
\mathbf{times}(0, y) &\to 0 \\
\mathbf{times}(\mathsf{s}(x), y) &\to \mathbf{plus}(\mathbf{times}(x, y), y)
\end{aligned}
$$

The hard one: Computing *Def*

Initialization

Example

Leading Example $\mathcal{R}_{\textbf{times}}$

$$
\begin{aligned}
\textbf{plus}(0, y) &\to y \\
\textbf{plus}(\mathsf{s}(x), y) &\to \mathsf{s}(\textbf{plus}(x, y)) \\
\textbf{times}(0, y) &\to 0 \\
\textbf{times}(\mathsf{s}(x), y) &\to \textbf{plus}(\textbf{times}(x, y), y)
\end{aligned}
$$

The hard one: Computing *Def*

Initialization

- collect right-hand sides with nested defined symbols

Example

Leading Example $\mathcal{R}_{\textbf{times}}$

$$
\begin{aligned}
\textbf{plus}(0, y) &\rightarrow y \\
\textbf{plus}(\mathsf{s}(x), y) &\rightarrow \mathsf{s}(\textbf{plus}(x, y)) \\
\textbf{times}(0, y) &\rightarrow 0 \\
\textbf{times}(\mathsf{s}(x), y) &\rightarrow \textbf{plus}(\textbf{times}(x, y), y)
\end{aligned}
$$

The hard one: Computing *Def*

Initialization

- collect right-hand sides with nested defined symbols

Example

Leading Example $\mathcal{R}_{\textbf{times}}$

$$\begin{array}{rcl} \textbf{plus}(0, y) & \to & y \\ \textbf{plus}(\mathsf{s}(x), y) & \to & \mathsf{s}(\textbf{plus}(x, y)) \\ \textbf{times}(0, y) & \to & 0 \\ \textbf{times}(\mathsf{s}(x), y) & \to & \textbf{plus}(\textbf{times}(x, y), y) \end{array}$$

The hard one: Computing *Def*

Initialization

- collect right-hand sides with nested defined symbols
- replace nested defined symbols with $\Box \curvearrowright C$

Example

Leading Example $\mathcal{R}_{\textbf{times}}$

$$
\begin{aligned}
\textbf{plus}(0, y) &\rightarrow y \\
\textbf{plus}(\mathsf{s}(x), y) &\rightarrow \mathsf{s}(\textbf{plus}(x, y)) \\
\textbf{times}(0, y) &\rightarrow 0 \\
\textbf{times}(\mathsf{s}(x), y) &\rightarrow \textbf{plus}(\textbf{times}(x, y), y)
\end{aligned}
$$

The hard one: Computing *Def*

Initialization

- collect right-hand sides with nested defined symbols
- replace nested defined symbols with $\Box \curvearrowright C$

Example

Leading Example $\mathcal{R}_{\mathbf{times}}$

$$
\begin{aligned}
\mathbf{plus}(0, y) &\rightarrow y \\
\mathbf{plus}(\mathsf{s}(x), y) &\rightarrow \mathsf{s}(\mathbf{plus}(x, y)) \\
\mathbf{times}(0, y) &\rightarrow 0 \\
\mathbf{times}(\mathsf{s}(x), y) &\rightarrow \mathbf{plus}(\mathbf{times}(x, y), y)
\end{aligned}
$$

The hard one: Computing *Def*

Initialization

- collect right-hand sides with nested defined symbols
- replace nested defined symbols with $\Box \curvearrowright C$

Example

$$\mathbf{plus}(\Box, y)$$

Leading Example $\mathcal{R}_{\mathbf{times}}$

$$\begin{aligned}
\mathbf{plus}(0, y) &\to y \\
\mathbf{plus}(\mathsf{s}(x), y) &\to \mathsf{s}(\mathbf{plus}(x, y)) \\
\mathbf{times}(0, y) &\to 0 \\
\mathbf{times}(\mathsf{s}(x), y) &\to \mathbf{plus}(\mathbf{times}(x, y), y)
\end{aligned}$$

The hard one: Computing *Def*

Initialization

- collect right-hand sides with nested defined symbols
- replace nested defined symbols with $\Box \curvearrowright C$
- add $\lfloor C \rfloor$ to *Def*

Example

$$\mathbf{plus}(\Box, y)$$

Leading Example $\mathcal{R}_{\mathbf{times}}$

$$\begin{aligned}
\mathbf{plus}(0, y) &\rightarrow y \\
\mathbf{plus}(\mathsf{s}(x), y) &\rightarrow \mathsf{s}(\mathbf{plus}(x, y)) \\
\mathbf{times}(0, y) &\rightarrow 0 \\
\mathbf{times}(\mathsf{s}(x), y) &\rightarrow \mathbf{plus}(\mathbf{times}(x, y), y)
\end{aligned}$$

The hard one: Computing *Def*

Initialization

- collect right-hand sides with nested defined symbols
- replace nested defined symbols with $\Box \curvearrowright C$
- add $\lfloor C \rfloor$ to *Def*

Example

$Def = \{\lfloor \mathbf{plus}(\Box, y) \rfloor\}$

Leading Example $\mathcal{R}_{\mathbf{times}}$

$$\begin{aligned}
\mathbf{plus}(0, y) &\to y \\
\mathbf{plus}(\mathsf{s}(x), y) &\to \mathsf{s}(\mathbf{plus}(x, y)) \\
\mathbf{times}(0, y) &\to 0 \\
\mathbf{times}(\mathsf{s}(x), y) &\to \mathbf{plus}(\mathbf{times}(x, y), y)
\end{aligned}$$

The hard one: Computing *Def*

Initialization

- collect right-hand sides with nested defined symbols
- replace nested defined symbols with $\Box \curvearrowright C$
- add $\lfloor C \rfloor$ to *Def*

Example

$$Def = \{\mathbf{plus}(\Box, y)\}$$

Leading Example $\mathcal{R}_{\mathbf{times}}$

$$\begin{aligned}
\mathbf{plus}(0, y) &\to y \\
\mathbf{plus}(\mathsf{s}(x), y) &\to \mathsf{s}(\mathbf{plus}(x, y)) \\
\mathbf{times}(0, y) &\to 0 \\
\mathbf{times}(\mathsf{s}(x), y) &\to \mathbf{plus}(\mathbf{times}(x, y), y)
\end{aligned}$$

The hard one: Computing *Def*

Idea: capture that nested defined symbols may be matched by variables

Fixed Point Step

Leading Example $\mathcal{R}_{\textbf{times}}$

$$\begin{aligned}
\textbf{plus}(0, y) &\to y \\
\textbf{plus}(\mathsf{s}(x), y) &\to \mathsf{s}(\textbf{plus}(x, y)) \\
\textbf{times}(0, y) &\to 0 \\
\textbf{times}(\mathsf{s}(x), y) &\to \textbf{plus}(\textbf{times}(x, y), y)
\end{aligned}$$

Example

- $\mathit{Def} = \{\textbf{plus}(\square, y)\}$

The hard one: Computing *Def*

Idea: capture that nested defined symbols may be matched by variables

Fixed Point Step

- pick a rule $\ell \to r$

Leading Example $\mathcal{R}_{\textbf{times}}$

$$\begin{aligned} \textbf{plus}(0, y) &\to y \\ \textbf{plus}(\mathsf{s}(x), y) &\to \mathsf{s}(\textbf{plus}(x, y)) \\ \textbf{times}(0, y) &\to 0 \\ \textbf{times}(\mathsf{s}(x), y) &\to \textbf{plus}(\textbf{times}(x, y), y) \end{aligned}$$

Example

- $Def = \{\textbf{plus}(\Box, y)\}$

The hard one: Computing *Def*

Idea: capture that nested defined symbols may be matched by variables

Fixed Point Step

- pick a rule $\ell \to r$

Leading Example $\mathcal{R}_{\textbf{times}}$

$$\begin{aligned} \mathbf{plus}(0, y) &\to y \\ \mathbf{plus}(\mathsf{s}(x), y) &\to \mathsf{s}(\mathbf{plus}(x, y)) \\ \mathbf{times}(0, y) &\to 0 \\ \mathbf{times}(\mathsf{s}(x), y) &\to \mathbf{plus}(\mathbf{times}(x, y), y) \end{aligned}$$

Example

- $\mathit{Def} = \{\mathbf{plus}(\square, y)\}$

The hard one: Computing *Def*

Idea: capture that nested defined symbols may be matched by variables

Fixed Point Step

- pick a rule $\ell \to r$
- replace some x in ℓ with $\square$

Leading Example $\mathcal{R}_{\textbf{times}}$

$$\begin{aligned}
\textbf{plus}(0, y) &\to y \\
\textbf{plus}(\mathsf{s}(x), y) &\to \mathsf{s}(\textbf{plus}(x, y)) \\
\textbf{times}(0, y) &\to 0 \\
\textbf{times}(\mathsf{s}(x), y) &\to \textbf{plus}(\textbf{times}(x, y), y)
\end{aligned}$$

Example

- $\mathit{Def} = \{\textbf{plus}(\square, y)\}$

The hard one: Computing *Def*

Idea: capture that nested defined symbols may be matched by variables

Fixed Point Step

- pick a rule $\ell \to r$
- replace some x in ℓ with $\Box$

Leading Example $\mathcal{R}_{\textbf{times}}$

$$\begin{aligned} \textbf{plus}(0, y) &\to y \\ \textbf{plus}(\mathsf{s}(x), y) &\to \mathsf{s}(\textbf{plus}(x, y)) \\ \textbf{times}(0, y) &\to 0 \\ \textbf{times}(\mathsf{s}(x), y) &\to \textbf{plus}(\textbf{times}(x, y), y) \end{aligned}$$

Example

- $\mathit{Def} = \{\textbf{plus}(\Box, y)\}$

The hard one: Computing *Def*

Idea: capture that nested defined symbols may be matched by variables

Fixed Point Step

- pick a rule $\ell \to r$
- replace some x in ℓ with $\square$

Leading Example $\mathcal{R}_{\textbf{times}}$

$$\begin{aligned}
\textbf{plus}(0, y) &\to y \\
\textbf{plus}(\mathsf{s}(x), y) &\to \mathsf{s}(\textbf{plus}(x, y)) \\
\textbf{times}(0, y) &\to 0 \\
\textbf{times}(\mathsf{s}(x), y) &\to \textbf{plus}(\textbf{times}(x, y), y)
\end{aligned}$$

Example

- $\mathit{Def} = \{\textbf{plus}(\square, y)\}$
- $\ell[\square] = \textbf{plus}(\mathsf{s}(\square), y)$

The hard one: Computing *Def*

Idea: capture that nested defined symbols may be matched by variables

Fixed Point Step

- pick a rule $\ell \to r$
- replace some x in ℓ with $\square$
- if $\ell[\square]$ overlaps with $D \in \mathit{Def}$

Leading Example $\mathcal{R}_{\textbf{times}}$

$$
\begin{aligned}
\textbf{plus}(0, y) &\to y \\
\textbf{plus}(\mathsf{s}(x), y) &\to \mathsf{s}(\textbf{plus}(x, y)) \\
\textbf{times}(0, y) &\to 0 \\
\textbf{times}(\mathsf{s}(x), y) &\to \textbf{plus}(\textbf{times}(x, y), y)
\end{aligned}
$$

Example

- $\mathit{Def} = \{\textbf{plus}(\square, y)\}$
- $\ell[\square] = \textbf{plus}(\mathsf{s}(\square), y)$

The hard one: Computing *Def*

Idea: capture that nested defined symbols may be matched by variables

Fixed Point Step

- pick a rule $\ell \to r$
- replace some x in ℓ with $\square$
- if $\ell[\square]$ overlaps with $D \in Def$
- pick a subterm $\mathbf{f}(\ldots x \ldots)$ of r

Leading Example $\mathcal{R}_{\textbf{times}}$

$$\begin{aligned}
\mathbf{plus}(0, y) &\to y \\
\mathbf{plus}(\mathsf{s}(x), y) &\to \mathsf{s}(\mathbf{plus}(x, y)) \\
\mathbf{times}(0, y) &\to 0 \\
\mathbf{times}(\mathsf{s}(x), y) &\to \mathbf{plus}(\mathbf{times}(x, y), y)
\end{aligned}$$

Example

- $Def = \{\mathbf{plus}(\square, y)\}$
- $\ell[\square] = \mathbf{plus}(\mathsf{s}(\square), y)$

The hard one: Computing *Def*

Idea: capture that nested defined symbols may be matched by variables

Fixed Point Step

- pick a rule $\ell \to r$
- replace some x in ℓ with $\Box$
- if $\ell[\Box]$ overlaps with $D \in \mathit{Def}$
- pick a subterm $\mathbf{f}(\ldots x \ldots)$ of r

Leading Example $\mathcal{R}_{\textbf{times}}$

$$
\begin{aligned}
\mathbf{plus}(0, y) &\to y \\
\mathbf{plus}(\mathsf{s}(x), y) &\to \mathsf{s}(\mathbf{plus}(x, y)) \\
\mathbf{times}(0, y) &\to 0 \\
\mathbf{times}(\mathsf{s}(x), y) &\to \mathbf{plus}(\mathbf{times}(x, y), y)
\end{aligned}
$$

Example

- $\mathit{Def} = \{\mathbf{plus}(\Box, y)\}$
- $\ell[\Box] = \mathbf{plus}(\mathsf{s}(\Box), y)$

The hard one: Computing *Def*

Idea: capture that nested defined symbols may be matched by variables

Fixed Point Step

- pick a rule $\ell \to r$
- replace some x in ℓ with $\Box$
- if $\ell[\Box]$ overlaps with $D \in \mathit{Def}$
- pick a subterm $\mathbf{f}(\ldots x \ldots)$ of r
- add $\lfloor \mathbf{f}(\ldots \Box \ldots) \rfloor$ to Def

Leading Example $\mathcal{R}_{\textbf{times}}$

$$
\begin{aligned}
\mathbf{plus}(0, y) &\to y \\
\mathbf{plus}(\mathsf{s}(x), y) &\to \mathsf{s}(\mathbf{plus}(x, y)) \\
\mathbf{times}(0, y) &\to 0 \\
\mathbf{times}(\mathsf{s}(x), y) &\to \mathbf{plus}(\mathbf{times}(x, y), y)
\end{aligned}
$$

Example

- $\mathit{Def} = \{\mathbf{plus}(\Box, y)\}$
- $\ell[\Box] = \mathbf{plus}(\mathsf{s}(\Box), y)$

The hard one: Computing *Def*

Idea: capture that nested defined symbols may be matched by variables

Fixed Point Step

- pick a rule $\ell \to r$
- replace some x in ℓ with $\Box$
- if $\ell[\Box]$ overlaps with $D \in \mathit{Def}$
- pick a subterm $\mathbf{f}(\ldots x \ldots)$ of r
- add $\lfloor \mathbf{f}(\ldots \Box \ldots) \rfloor$ to Def

Leading Example $\mathcal{R}_{\mathbf{times}}$

$$\begin{aligned}
\mathbf{plus}(0, y) &\to y \\
\mathbf{plus}(\mathsf{s}(x), y) &\to \mathsf{s}(\mathbf{plus}(x, y)) \\
\mathbf{times}(0, y) &\to 0 \\
\mathbf{times}(\mathsf{s}(x), y) &\to \mathbf{plus}(\mathbf{times}(x, y), y)
\end{aligned}$$

Example

- $\mathit{Def} = \{\mathbf{plus}(\Box, y), \lfloor \mathbf{plus}(\Box, y) \rfloor\}$
- $\ell[\Box] = \mathbf{plus}(\mathsf{s}(\Box), y)$

The hard one: Computing *Def*

Idea: capture that nested defined symbols may be matched by variables

Fixed Point Step

- pick a rule $\ell \to r$
- replace some x in ℓ with $\square$
- if $\ell[\square]$ overlaps with $D \in \mathit{Def}$
- pick a subterm $\mathbf{f}(\ldots x \ldots)$ of r
- add $\lfloor \mathbf{f}(\ldots \square \ldots) \rfloor$ to Def

Leading Example $\mathcal{R}_{\textbf{times}}$

$$\begin{aligned} \mathbf{plus}(0, y) &\to y \\ \mathbf{plus}(\mathsf{s}(x), y) &\to \mathsf{s}(\mathbf{plus}(x, y)) \\ \mathbf{times}(0, y) &\to 0 \\ \mathbf{times}(\mathsf{s}(x), y) &\to \mathbf{plus}(\mathbf{times}(x, y), y) \end{aligned}$$

Example

- $\mathit{Def} = \{\mathbf{plus}(\square, y), \mathbf{plus}(\square, y)\}$
- $\ell[\square] = \mathbf{plus}(\mathsf{s}(\square), y)$

The hard one: Computing *Def*

Idea: capture that nested defined symbols may be matched by variables

Fixed Point Step

- pick a rule $\ell \to r$
- replace some x in ℓ with $\Box$
- if $\ell[\Box]$ overlaps with $D \in \mathit{Def}$
- pick a subterm $\mathbf{f}(\ldots x \ldots)$ of r
- add $\lfloor \mathbf{f}(\ldots \Box \ldots) \rfloor$ to Def

Leading Example $\mathcal{R}_{\textbf{times}}$

$$\begin{aligned}
\textbf{plus}(0, y) &\to y \\
\textbf{plus}(\mathsf{s}(x), y) &\to \mathsf{s}(\textbf{plus}(x, y)) \\
\textbf{times}(0, y) &\to 0 \\
\textbf{times}(\mathsf{s}(x), y) &\to \textbf{plus}(\textbf{times}(x, y), y)
\end{aligned}$$

Example

- $\mathit{Def} = \{\textbf{plus}(\Box, y)\}$
- $\ell[\Box] = \textbf{plus}(\mathsf{s}(\Box), y)$

The hard one: Computing *Def*

Idea: capture that nested defined symbols may be matched by variables

Fixed Point Step

- pick a rule $\ell \to r$
- replace some x in ℓ with $\square$
- if $\ell[\square]$ overlaps with $D \in \mathit{Def}$
- pick a subterm $\mathbf{f}(\ldots x \ldots)$ of r
- add $\lfloor \mathbf{f}(\ldots \square \ldots) \rfloor$ to Def

Leading Example $\mathcal{R}_{\mathbf{times}}$

$$
\begin{aligned}
\mathbf{plus}(0, y) &\to y \\
\mathbf{plus}(\mathsf{s}(x), y) &\to \mathsf{s}(\mathbf{plus}(x, y)) \\
\mathbf{times}(0, y) &\to 0 \\
\mathbf{times}(\mathsf{s}(x), y) &\to \mathbf{plus}(\mathbf{times}(x, y), y)
\end{aligned}
$$

Example

- $\mathit{Def} = \{\mathbf{plus}(\square, y)\}$
- $\ell[\square] = \mathbf{plus}(\mathsf{s}(\square), y)$

$\mathit{Dup} = \{\mathbf{times}(\mathsf{s}(x), \square)\}$ and $\mathit{Def} = \{\mathbf{plus}(\square, y)\}$ don't overlap $\curvearrowright$ rc = irc!

Outline

Handling Non-Constructor Systems

Leading Example $\mathcal{R}_{\textbf{times}}$

$$
\begin{aligned}
\textbf{plus}(0, y) &\rightarrow y \\
\textbf{plus}(\mathsf{s}(x), y) &\rightarrow \mathsf{s}(\textbf{plus}(x, y)) \\
\textbf{times}(0, y) &\rightarrow 0 \\
\textbf{times}(\mathsf{s}(x), y) &\rightarrow \textbf{plus}(\textbf{times}(x, y), y)
\end{aligned}
$$

Handling Non-Constructor Systems

Leading Example $\mathcal{R}_{\textbf{times}}$

$$
\begin{array}{rcl}
\textbf{plus}(0, y) & \to & y \\
\textbf{plus}(\mathsf{s}(x), y) & \to & \mathsf{s}(\textbf{plus}(x, y)) \\
\textbf{times}(0, y) & \to & 0 \\
\textbf{times}(\mathsf{s}(x), y) & \to & \textbf{plus}(\textbf{times}(x, y), y) \\
\textbf{plus}(x, \textbf{plus}(y, z)) & \to & \textbf{plus}(\textbf{plus}(x, y), z)
\end{array}
$$

Handling Non-Constructor Systems

Leading Example $\mathcal{R}_{\mathbf{times}}$

$$
\begin{array}{rcl}
\mathbf{plus}(0, y) & \to & y \\
\mathbf{plus}(\mathsf{s}(x), y) & \to & \mathsf{s}(\mathbf{plus}(x, y)) \\
\mathbf{times}(0, y) & \to & 0 \\
\mathbf{times}(\mathsf{s}(x), y) & \to & \mathbf{plus}(\mathbf{times}(x, y), y) \\
\mathbf{plus}(x, \mathbf{plus}(y, z)) & \to & \mathbf{plus}(\mathbf{plus}(x, y), z)
\end{array}
$$

- nested defined symbols only below **plus**'s first argument

Handling Non-Constructor Systems

Leading Example $\mathcal{R}_{\mathbf{times}}$

$$\begin{aligned}
\mathbf{plus}(0, y) &\to y \\
\mathbf{plus}(\mathsf{s}(x), y) &\to \mathsf{s}(\mathbf{plus}(x, y)) \\
\mathbf{times}(0, y) &\to 0 \\
\mathbf{times}(\mathsf{s}(x), y) &\to \mathbf{plus}(\mathbf{times}(x, y), y) \\
\mathbf{plus}(x, \mathbf{plus}(y, z)) &\to \mathbf{plus}(\mathbf{plus}(x, y), z)
\end{aligned}$$

- nested defined symbols only below **plus**'s first argument

$\curvearrowright$ $\mathbf{plus}(x, \mathbf{plus}(y, z))$ not reachable from basic terms!

Handling Non-Constructor Systems

Leading Example $\mathcal{R}_{\textbf{times}}$

$$\begin{aligned}
\textbf{plus}(0, y) &\to y \\
\textbf{plus}(\mathsf{s}(x), y) &\to \mathsf{s}(\textbf{plus}(x, y)) \\
\textbf{times}(0, y) &\to 0 \\
\textbf{times}(\mathsf{s}(x), y) &\to \textbf{plus}(\textbf{times}(x, y), y)
\end{aligned}$$

- nested defined symbols only below **plus**'s first argument

↷ $\textbf{plus}(x, \textbf{plus}(y, z))$ not reachable from basic terms!

Handling Non-Constructor Systems

Leading Example $\mathcal{R}_{\textbf{times}}$

$$
\begin{aligned}
\textbf{plus}(0, y) &\rightarrow y \\
\textbf{plus}(\mathsf{s}(x), y) &\rightarrow \mathsf{s}(\textbf{plus}(x, y)) \\
\textbf{times}(0, y) &\rightarrow 0 \\
\textbf{times}(\mathsf{s}(x), y) &\rightarrow \textbf{plus}(\textbf{times}(x, y), y)
\end{aligned}
$$

- nested defined symbols only below **plus**'s first argument

↷ $\textbf{plus}(x, \textbf{plus}(y, z))$ not reachable from basic terms!

- information *which* defined symbols can be nested often crucial

Handling Non-Constructor Systems

Leading Example $\mathcal{R}_{\mathbf{times}}$

$$\begin{aligned}
\mathbf{plus}(0, y) &\rightarrow y \\
\mathbf{plus}(\mathsf{s}(x), y) &\rightarrow \mathsf{s}(\mathbf{plus}(x, y)) \\
\mathbf{times}(0, y) &\rightarrow 0 \\
\mathbf{times}(\mathsf{s}(x), y) &\rightarrow \mathbf{plus}(\mathbf{times}(x, y), y)
\end{aligned}$$

- nested defined symbols only below **plus**'s first argument

↷ $\mathbf{plus}(x, \mathbf{plus}(y, z))$ not reachable from basic terms!

- information *which* defined symbols can be nested often crucial

↷ similar fixed point algorithm

Experimental Results, Conclusion

Experiments on the *TPDB*:

TcT	AProVE	TcT preproc	AProVE & TcT
209	270	299	308

Experimental Results, Conclusion

Experiments on the *TPDB*:

TcT	AProVE	TcT preproc	AProVE & TcT
209	270	299	308

- powerful sufficient criterion for rc = irc

Experimental Results, Conclusion

Experiments on the *TPDB*:

TcT	AProVE	TcT preproc	AProVE & TcT
209	270	299	308

- powerful sufficient criterion for $\mathrm{rc} = \mathrm{irc}$
- easy to automate

Experimental Results, Conclusion

Experiments on the *TPDB*:

TcT	AProVE	TcT preproc	AProVE & TcT
209	270	299	308

- powerful sufficient criterion for rc = irc
- easy to automate

$\curvearrowright$ future irc techniques applicable for rc

Experimental Results, Conclusion

Experiments on the *TPDB*:

TcT	AProVE	TcT preproc	AProVE & TcT	AProVE++
209	270	299	308	324

- powerful sufficient criterion for rc = irc
- easy to automate

↷ future irc techniques applicable for rc

Experimental Results, Conclusion

Experiments on the *TPDB*:

TcT	AProVE	TcT preproc	AProVE & TcT	AProVE++
209	270	299	308	324

- powerful sufficient criterion for rc = irc
- easy to automate

↷ future irc techniques applicable for rc

- significant improvement of the state of the art