Rewriting Techniques for Programming and Proving

Horatiu Cirstea   Sergueï Lenglet

A simple game

The rules of the game:

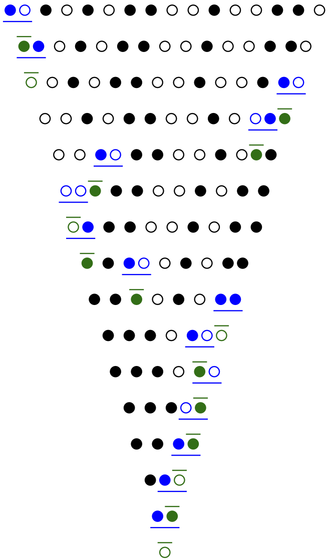
$$\begin{array}{ccc} \bullet\bullet & \longrightarrow & \circ \\ \circ\circ & \longrightarrow & \circ \\ \bullet\circ & \longrightarrow & \bullet \\ \circ\bullet & \longrightarrow & \bullet \end{array}$$

A starting point:

$$\bullet\ \circ\ \bullet\ \circ\ \bullet\ \circ\ \bullet\ \bullet\ \circ\ \circ\ \bullet\ \circ\ \circ\ \bullet\ \bullet\circ$$

Who puts the last white wins (for this starting point)

A simple game



Rules of the game

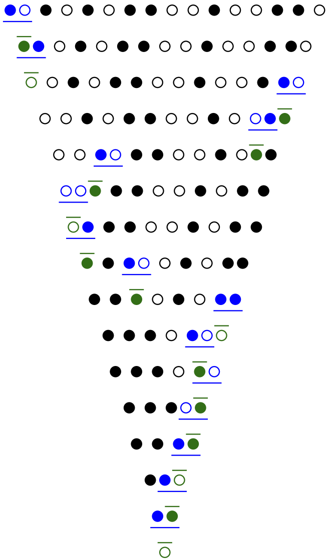
●● → ○

○○ → ○

●○ → ●

○● → ●

A simple game



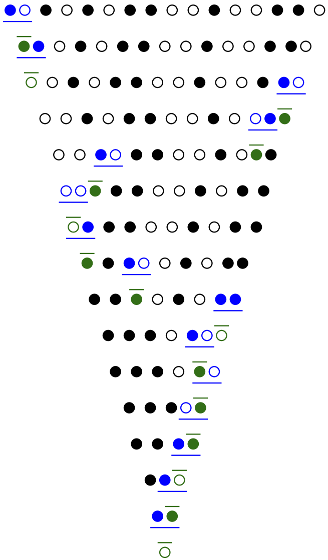
Rules of the game

●● → ○
○○ → ○
●○ → ●
○● → ●

Questions:

- May I always win?

A simple game



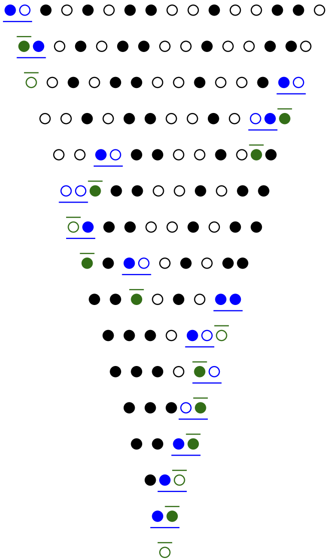
Rules of the game

●●	$\longrightarrow$	○
○○	$\longrightarrow$	○
●○	$\longrightarrow$	●
○●	$\longrightarrow$	●

Questions:

- May I always win?
- Does the game terminates?

A simple game



Rules of the game

●● → ○
○○ → ○
●○ → ●
○● → ●

Questions:

- May I always win?
- Does the game terminates?
- Do we always get the same result?

What are the basic operations that have been used?

What are the basic operations that have been used?

1– Matching

The data: $\bullet\bullet \quad \boxed{\bullet\circ}\ \circ\ \bullet\ \circ\ \bullet\bullet$

The rewrite rule: $\boxed{\bullet\circ} \to \boxed{\bullet}$

What are the basic operations that have been used?

1– Matching

The data: $\bullet\bullet \quad \boxed{\bullet\circ}\ \circ\ \bullet\ \circ\ \bullet\bullet$

The rewrite rule: $\boxed{\bullet\circ} \rightarrow \boxed{\bullet}$

2– Compute what should be substituted

The righthand side: $\boxed{\bullet}$

What are the basic operations that have been used?

1– Matching

The data: $\bullet\bullet\ \boxed{\bullet\circ}\ \circ\ \bullet\ \circ\ \bullet\bullet$

The rewrite rule: $\boxed{\bullet\circ} \rightarrow \boxed{\bullet}$

2– Compute what should be substituted

The righthand side: $\boxed{\bullet}$

3– Replacement

The new generated data: $\bullet\ \bullet\ \boxed{\bullet}\ \circ\ \bullet\ \circ\ \bullet\bullet$

Addition in Peano arithmetic

Peano gives a meaning to addition by using the following axioms:

$$
\begin{aligned}
0 + x &= x \\
s(x) + y &= s(x + y)
\end{aligned}
$$

Addition in Peano arithmetic

Peano gives a meaning to addition by using the following axioms:

$$\begin{aligned} 0 + x &= x \\ s(x) + y &= s(x + y) \end{aligned}$$

What's the result of $s(s(0)) + s(s(0))$?

Addition in Peano arithmetic

Peano gives a meaning to addition by using the following axioms:

$$\begin{aligned} 0 + x &= x \\ s(x) + y &= s(x + y) \end{aligned}$$

What's the result of $s(s(0)) + s(s(0))$?

$$s(s(0)) + s(s(0)) = s(s(0) + s(s(0)))$$

Addition in Peano arithmetic

Peano gives a meaning to addition by using the following axioms:

$$\begin{aligned} 0 + x &= x \\ s(x) + y &= s(x + y) \end{aligned}$$

What's the result of $s(s(0)) + s(s(0))$?

$$\begin{aligned} s(s(0)) + s(s(0)) &= s(s(0) + s(s(0))) \\ &= s(s(0 + s(s(0)))) \end{aligned}$$

Addition in Peano arithmetic

Peano gives a meaning to addition by using the following axioms:

$$\begin{aligned} 0 + x &= x \\ s(x) + y &= s(x + y) \end{aligned}$$

What's the result of $s(s(0)) + s(s(0))$?

$$\begin{aligned} s(s(0)) + s(s(0)) &= s(s(0) + s(s(0))) \\ &= s(s(0 + s(s(0)))) \\ &= s(s(s(s(0)))) \\ &= s(0) + s(s(s(0))) \\ &= 0 + 0 + 0 + s(s(s(s(0)))) \\ &= \ldots \end{aligned}$$

Addition in Peano arithmetic

Peano gives a meaning to addition by using the following axioms:

$$\begin{aligned} 0 + x &= x \\ s(x) + y &= s(x + y) \end{aligned}$$

What's the result of $s(s(0)) + s(s(0))$?

$$\begin{aligned} s(s(0)) + s(s(0)) &= s(s(0) + s(s(0))) \\ &= s(s(0 + s(s(0)))) \\ &= s(s(s(s(0)))) \\ &= s(0) + s(s(s(0))) \\ &= 0 + 0 + 0 + s(s(s(s(0)))) \\ &= \dots \end{aligned}$$

Is there a better result?

Addition in Peano arithmetic

Compute a result by turning the equalities into rewrite rules:

$$\begin{aligned} 0 + x &\rightarrow x \\ s(x) + y &\rightarrow s(x + y) \end{aligned}$$

Addition in Peano arithmetic

Compute a result by turning the equalities into rewrite rules:

$$
\begin{aligned}
0 + x &\rightarrow x \\
s(x) + y &\rightarrow s(x + y)
\end{aligned}
$$

$s(s(0)) + s(s(0)) \rightarrow s(s(0) + s(s(0)))$
$\rightarrow s(s(s(s(0))))$
$\rightarrow s(s(0 + s(s(0))))$

Addition in Peano arithmetic

Compute a result by turning the equalities into rewrite rules:

$$\begin{aligned} 0 + x &\rightarrow x \\ s(x) + y &\rightarrow s(x + y) \end{aligned}$$

$s(s(0)) + s(s(0)) \rightarrow s(s(0) + s(s(0))$
$\rightarrow s(s(s(s(0))))$
$\rightarrow s(s(0 + s(s(0))))$

Questions:

- Is this computation terminating?

Addition in Peano arithmetic

Compute a result by turning the equalities into rewrite rules:

$$\begin{aligned} 0 + x &\rightarrow x \\ s(x) + y &\rightarrow s(x + y) \end{aligned}$$

$s(s(0)) + s(s(0)) \rightarrow s(s(0) + s(s(0))$
$\rightarrow s(s(s(s(0))))$
$\rightarrow s(s(0 + s(s(0))))$

Questions:

- Is this computation terminating?
- Is there always a result (e.g. an expression without $+$)?

Addition in Peano arithmetic

Compute a result by turning the equalities into rewrite rules:

$$\begin{aligned} 0 + x &\rightarrow x \\ s(x) + y &\rightarrow s(x + y) \end{aligned}$$

$s(s(0)) + s(s(0)) \rightarrow s(s(0) + s(s(0))$
$\rightarrow s(s(s(s(0))))$
$\rightarrow s(s(0 + s(s(0))))$

Questions:

- Is this computation terminating?
- Is there always a result (e.g. an expression without $+$)?
- Is such a result unique?

Addition in Peano arithmetic

Compute a result by turning the equalities into rewrite rules:

$$\begin{aligned} 0 + x &\rightarrow x \\ s(x) + y &\rightarrow s(x + y) \end{aligned}$$

$s(s(0)) + s(s(0)) \rightarrow s(s(0) + s(s(0))$
$\rightarrow s(s(s(s(0))))$
$\rightarrow s(s(0 + s(s(0))))$

Questions:

- Is this computation terminating?
- Is there always a result (e.g. an expression without $+$)?
- Is such a result unique?
- How to turn equations into rewrite rules?

What are the basic operations that have been used?

What are the basic operations that have been used?

1– Matching

The data: $s(\boxed{s(0)}) + \boxed{s(s(0))}$

The rewrite rule: $s(\boxed{x}) + \boxed{y} \rightarrow \boxed{s(x+y)}$

What are the basic operations that have been used?

1– Matching

The data: $s(\boxed{s(0)}) + \boxed{s(s(0))}$

The rewrite rule: $s(\boxed{x}) + \boxed{y} \rightarrow \boxed{s(x+y)}$

2– Compute what should be substituted

The instanciated rhs: $\boxed{s(\boxed{s(0)} + \boxed{s(s(0))})}$

What are the basic operations that have been used?

1– Matching

The data: $s(\boxed{s(0)}) + \boxed{s(s(0))}$

The rewrite rule: $s(\boxed{x}) + \boxed{y} \rightarrow \boxed{s(x+y)}$

2– Compute what should be substituted

The instanciated rhs: $\boxed{s(\boxed{s(0)} + \boxed{s(s(0))})}$

3– Replacement

The new generated data: $\boxed{s(s(0)+s(s(0)))}$

Fibonacci

$$\begin{array}{lll} [\alpha] & \mathit{fib}(0) & \rightarrow & 1 \\ [\beta] & \mathit{fib}(1) & \rightarrow & 1 \\ [\gamma] & \mathit{fib}(n) & \rightarrow & \mathit{fib}(n-1) + \mathit{fib}(n-2) \end{array}$$

$\boxed{\mathrm{fib}(\boxed{3})}$

Fibonacci

$$
\begin{array}{llll}
[\alpha] & \mathit{fib}(0) & \rightarrow & 1 \\
[\beta] & \mathit{fib}(1) & \rightarrow & 1 \\
[\gamma] & \mathit{fib}(n) & \rightarrow & \mathit{fib}(n-1) + \mathit{fib}(n-2)
\end{array}
$$

$$\boxed{\text{fib}(\boxed{3})} \rightarrow \boxed{\text{fib}(\boxed{2}) + \text{fib}(\boxed{1})}$$

Fibonacci

$$\begin{array}{llcl} [\alpha] & \mathit{fib}(0) & \rightarrow & 1 \\ [\beta] & \mathit{fib}(1) & \rightarrow & 1 \\ [\gamma] & \mathit{fib}(n) & \rightarrow & \mathit{fib}(n-1) + \mathit{fib}(n-2) \end{array}$$

$$\boxed{\text{fib}(\boxed{3})} \rightarrow \boxed{\text{fib}(\boxed{2}) + \text{fib}(\boxed{1})}$$

$$\mathit{fib}(2) + \boxed{\text{fib}(1)}$$

Fibonacci

$$
\begin{array}{llcl}
[\alpha] & \mathit{fib}(0) & \rightarrow & 1 \\
[\beta] & \mathit{fib}(1) & \rightarrow & 1 \\
[\gamma] & \mathit{fib}(n) & \rightarrow & \mathit{fib}(n-1) + \mathit{fib}(n-2)
\end{array}
$$

$$\boxed{\text{fib}(\boxed{3})} \rightarrow \boxed{\text{fib}(\boxed{2}) + \text{fib}(\boxed{1})}$$

$$\mathit{fib}(2) + \boxed{\text{fib}(1)} \rightarrow \mathit{fib}(2) + \boxed{1}$$

Fibonacci

$$
\begin{array}{llcl}
[\alpha] & fib(0) & \rightarrow & 1 \\
[\beta] & fib(1) & \rightarrow & 1 \\
[\gamma] & fib(n) & \rightarrow & fib(n-1) + fib(n-2)
\end{array}
$$

$$\boxed{\mathrm{fib}(\boxed{3})} \rightarrow \boxed{\mathrm{fib}(\boxed{2}) + \mathrm{fib}(\boxed{1})}$$

$$fib(2) + \boxed{\mathrm{fib}(1)} \rightarrow fib(2) + \boxed{1}$$

$$\boxed{\mathrm{fib}(\boxed{2})} + 1$$

Fibonacci

$$\begin{array}{llcl} [\alpha] & \mathit{fib}(0) & \rightarrow & 1 \\ [\beta] & \mathit{fib}(1) & \rightarrow & 1 \\ [\gamma] & \mathit{fib}(n) & \rightarrow & \mathit{fib}(n-1) + \mathit{fib}(n-2) \end{array}$$

$$\boxed{\mathrm{fib}(\boxed{3})} \rightarrow \boxed{\mathrm{fib}(\boxed{2}) + \mathrm{fib}(\boxed{1})}$$

$$\mathit{fib}(2) + \boxed{\mathrm{fib}(1)} \rightarrow \mathit{fib}(2) + \boxed{1}$$

$$\boxed{\mathrm{fib}(\boxed{2})} + 1 \rightarrow \boxed{\mathrm{fib}(\boxed{1}) + \mathrm{fib}(\boxed{0})} + 1$$

Fibonacci

$$\begin{array}{llcl} [\alpha] & \mathit{fib}(0) & \rightarrow & 1 \\ [\beta] & \mathit{fib}(1) & \rightarrow & 1 \\ [\gamma] & \mathit{fib}(n) & \rightarrow & \mathit{fib}(n-1) + \mathit{fib}(n-2) \end{array}$$

$$\boxed{\mathrm{fib}(\boxed{3})} \rightarrow \boxed{\mathrm{fib}(\boxed{2}) + \mathrm{fib}(\boxed{1})}$$

$$\mathit{fib}(2) + \boxed{\mathrm{fib}(1)} \rightarrow \mathit{fib}(2) + \boxed{1}$$

$$\boxed{\mathrm{fib}(\boxed{2})} + 1 \rightarrow \boxed{\mathrm{fib}(\boxed{1}) + \mathrm{fib}(\boxed{0})} + 1$$

$$\boxed{\mathrm{fib}(1)} + \mathit{fib}(0) + 1$$

Fibonacci

$$\begin{array}{llll} [\alpha] & \mathit{fib}(0) & \to & 1 \\ [\beta] & \mathit{fib}(1) & \to & 1 \\ [\gamma] & \mathit{fib}(n) & \to & \mathit{fib}(n-1) + \mathit{fib}(n-2) \end{array}$$

$$\boxed{\text{fib}(\boxed{3})} \to \boxed{\text{fib}(\boxed{2}) + \text{fib}(\boxed{1})}$$

$$\mathit{fib}(2) + \boxed{\text{fib}(1)} \to \mathit{fib}(2) + \boxed{1}$$

$$\boxed{\text{fib}(\boxed{2})} + 1 \to \boxed{\text{fib}(\boxed{1}) + \text{fib}(\boxed{0})} + 1$$

$$\boxed{\text{fib}(1)} + \mathit{fib}(0) + 1 \to \boxed{1} + \mathit{fib}(0) + 1$$

Fibonacci

$$
\begin{array}{llcl}
[\alpha] & \mathit{fib}(0) & \to & 1 \\
[\beta] & \mathit{fib}(1) & \to & 1 \\
[\gamma] & \mathit{fib}(n) & \to & \mathit{fib}(n-1) + \mathit{fib}(n-2)
\end{array}
$$

$$\boxed{\text{fib}(\boxed{3})} \to \boxed{\text{fib}(\boxed{2}) + \text{fib}(\boxed{1})}$$

$$\mathit{fib}(2) + \boxed{\text{fib}(1)} \to \mathit{fib}(2) + \boxed{1}$$

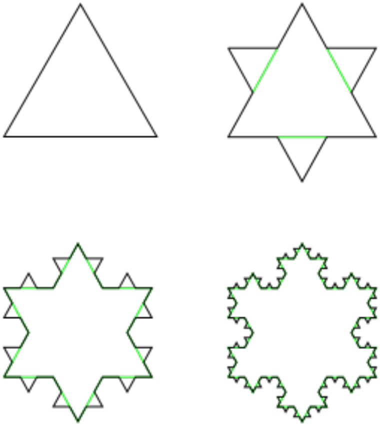
$$\boxed{\text{fib}(\boxed{2})} + 1 \to \boxed{\text{fib}(\boxed{1}) + \text{fib}(\boxed{0})} + 1$$

$$\boxed{\text{fib}(1)} + \mathit{fib}(0) + 1 \to \boxed{1} + \mathit{fib}(0) + 1$$

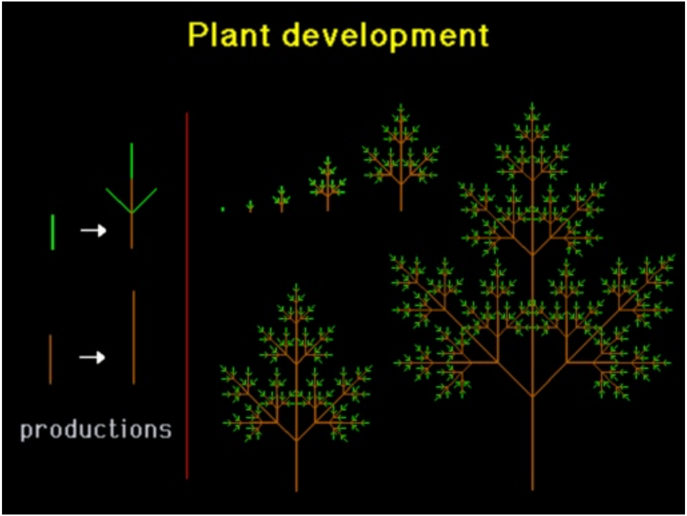
$$\dots$$

Finally $\mathit{fib}(3) = 3$, $\mathit{fib}(4) = 5$, ...

Fractal: Koch snowflake



Ecological Rewriting



http://algorithmicbotany.org/

Sorting by rewriting

```
rules for List
  X, Y : Nat ; L L' L'' : List;
   hd (X L) => X ;                         tl (X L) => L ;
   sort nil => nil .
   sort (L X L' Y L'') =>  sort (L Y L' X L'')  if Y < X .
end

sort (6 5 4 3 2 1) => ...
```

On what objects can rewriting act?

It can be defined on

- terms like $2 + \mathrm{i}(3)$ or XML documents
- strings like "What is rewriting?" (`sed` performs string rewriting)
- graphs
- sets
- multisets
- ...

References

- Rewriting, Solving, Proving
  Claude and Hélène Kirchner
- Term Rewriting System - Terese
  Marc Bezem, Jan Willem Klop, Roel de Vrijer
- Term Rewriting and All That
  Franz Baader and Tobias Nipkow