Behavioural differential equations and coinduction for binary trees

Alexandra Silva and Jan Rutten

CWI, The Netherlands

WoLLIC 2007

Motivation

- Previous work on streams and formal power series

  Behavioural differential equations: a coinductive calculus of streams, automata, and power series

  Elements of stream calculus (an extensive exercise in coinduction)

  showed that coinduction and behavioural differential equations are useful for building a calculus
- We want to investigate if the same approach is useful for other infinite structures, *e.g.* infinite binary trees

What will we show?

We will show how to…

- … define infinite binary trees through *behavioural differential equations*
- … develop a calculus for binary trees *à la formal power series*
- … calculate closed expressions for infinite binary trees
- … use closed expressions to prove properties about infinite binary trees

Infinite binary trees coalgebraically

Infinite binary trees

$$T_A = \{\sigma : \{L, R\}^* \to A\}$$

Recall: A formal power series is a function $\sigma : X^* \to A$ where X is the set of variables (or input symbols) and A is a semiring.

For A semiring, the set T_A coincides with the formal power series over $X = 2 = \{L, R\}$.

Infinite binary trees coalgebraically

Infinite binary trees

$$T_A = \{\sigma : \{L, R\}^* \to A\}$$

Recall: A formal power series is a function $\sigma : X^* \to A$ where X is the set of variables (or input symbols) and A is a semiring.

For A semiring, the set T_A coincides with the formal power series over $X = 2 = \{L, R\}$.

Infinite binary trees coalgebraically

Infinite binary trees

$$T_A = \{\sigma : \{L, R\}^* \to A\}$$

Recall: A formal power series is a function $\sigma : X^* \to A$ where X is the set of variables (or input symbols) and A is a semiring.

For A semiring, the set T_A coincides with the formal power series over $X = 2 = \{L, R\}$.

Infinite binary trees coalgebraically

$(A^{B^*}, < o, t >)$ is the final coalgebra for functor $G(X) = A \times X^B$.

$$< o, t > \quad : \quad A^{B^*} \to A \times (A^{B^*})^B$$
$$< o, t > (f) \quad = \quad < f(\varepsilon), \lambda b \cdot \lambda w \cdot f(bw) >$$

$(T_A = A^{2^*}, \alpha)$ is the final coalgebra of $F(X) = A \times X^2$

$$\alpha(\sigma) = < \sigma(\varepsilon), \sigma_L, \sigma_R >$$

Infinite binary trees coalgebraically

$(A^{B^*}, < o, t >)$ is the final coalgebra for functor $G(X) = A \times X^B$.

$$\begin{aligned} < o, t > \quad &: \quad A^{B^*} \rightarrow A \times (A^{B^*})^B \\ < o, t > (f) \quad &= \quad < f(\varepsilon), \lambda b \cdot \lambda w \cdot f(bw) > \end{aligned}$$

$(T_A = A^{2^*}, \alpha)$ is the final coalgebra of $F(X) = A \times X^2$

$$\alpha(\sigma) = < \sigma(\varepsilon), \sigma_L, \sigma_R >$$

Infinite binary trees coalgebraically

$(A^{B^*}, < o, t >)$ is the final coalgebra for functor $G(X) = A \times X^B$.

$$< o, t > \quad : \quad A^{B^*} \to A \times (A^{B^*})^B$$
$$< o, t > (f) \quad = \quad < f(\varepsilon), \lambda b \cdot \lambda w \cdot f(bw) >$$

$(T_A = A^{2^*}, \alpha)$ is the final coalgebra of $F(X) = A \times X^2$

$$\alpha(\sigma) = < \sigma(\varepsilon), \sigma_L, \sigma_R >$$

Infinite binary trees coalgebraically

$(A^{B^*}, < o, t >)$ is the final coalgebra for functor $G(X) = A \times X^B$.

$$
\begin{aligned}
< o, t > \quad &: \quad A^{B^*} \to A \times (A^{B^*})^B \\
< o, t > (f) \quad &= \quad < f(\varepsilon), \lambda b \cdot \lambda w \cdot f(bw) >
\end{aligned}
$$

$(T_A = A^{2^*}, \alpha)$ is the final coalgebra of $F(X) = A \times X^2$

$$\alpha(\sigma) = < \sigma(\varepsilon), \sigma_L, \sigma_R >$$

Infinite binary trees coalgebraically

(T_A, α) final coalgebra: every tree $\sigma \in T_A$ is uniquely determined by defining $\sigma(\varepsilon)$, σ_L and σ_R.

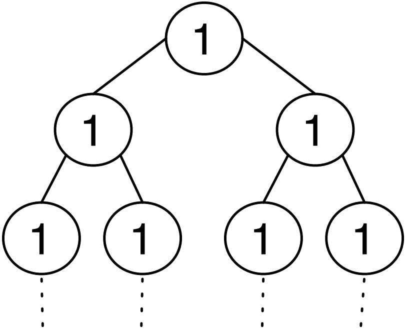
Infinite binary trees coalgebraically

(T_A, α) final coalgebra: every tree $\sigma \in T_A$ is uniquely determined by defining $\sigma(\varepsilon)$, σ_L and σ_R.

$\sigma \in T_A$ will be defined by means of **behavioural differential equations**:

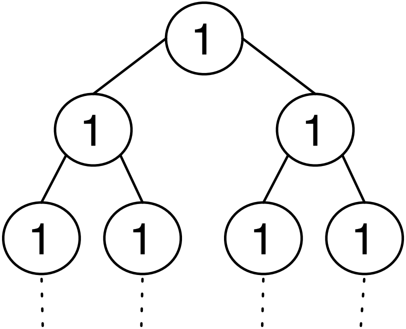
$$\begin{array}{rcll} \sigma(\varepsilon) & = & c & \textit{initial value} \\ \sigma_L & = & \textit{left\_exp} & \textit{left derivative} \\ \sigma_R & = & \textit{right\_exp} & \textit{right derivative} \end{array}$$

Example I



$$\begin{aligned} ones(\varepsilon) &= 1 \\ ones_L &= ones \\ ones_R &= ones \end{aligned}$$

Example I



$$
\begin{array}{rcl}
ones(\varepsilon) & = & 1 \\
ones_L & = & ones \\
ones_R & = & ones
\end{array}
$$

Behavioural Differential Equations – format

In which conditions a system of behavioural differential equations defining a tree $\sigma = f(\sigma_1, \ldots, \sigma_n)$

$$
\begin{array}{lcll}
f(\sigma_1, \ldots, \sigma_n)(\varepsilon) & = & c & \textit{initial value} \\
(f(\sigma_1, \ldots, \sigma_n))_L & = & \textit{left\_exp} & \textit{left derivative} \\
(f(\sigma_1, \ldots, \sigma_n))_R & = & \textit{right\_exp} & \textit{right derivative}
\end{array}
$$

is well formed?

We proved that if:

1. c is calculated only involving $\sigma_1(\varepsilon), \ldots, \sigma_n(\varepsilon)$
2. left_exp and right_exp only depend on $\sigma_1, \ldots, \sigma_n, (\sigma_1)_L, \ldots, (\sigma_n)_L, (\sigma_1)_R, \ldots, (\sigma_n)_R$ and function symbols

the system is well-formed, *i.e.* uniquely determines $\sigma = f(\sigma_1, \ldots, \sigma_n)$.

Behavioural Differential Equations – format

In which conditions a system of behavioural differential equations defining a tree $\sigma = f(\sigma_1, \ldots, \sigma_n)$

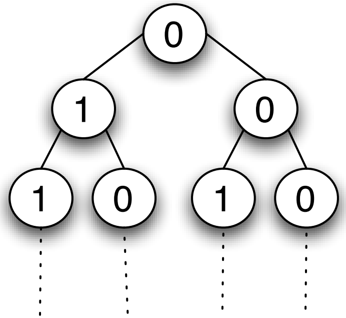
$$
\begin{array}{rcll}
f(\sigma_1, \ldots, \sigma_n)(\varepsilon) & = & c & \textit{initial value} \\
(f(\sigma_1, \ldots, \sigma_n))_L & = & \textit{left\_exp} & \textit{left derivative} \\
(f(\sigma_1, \ldots, \sigma_n))_R & = & \textit{right\_exp} & \textit{right derivative}
\end{array}
$$

is well formed?
We proved that if:

1. c is calculated only involving $\sigma_1(\varepsilon), \ldots, \sigma_n(\varepsilon)$
2. left_exp and right_exp only depend on $\sigma_1, \ldots, \sigma_n, (\sigma_1)_L, \ldots, (\sigma_n)_L, (\sigma_1)_R, \ldots, (\sigma_n)_R$ and function symbols

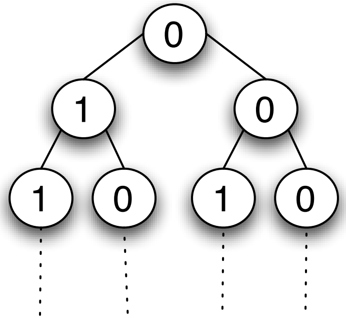
the system is well-formed, *i.e.* uniquely determines $\sigma = f(\sigma_1, \ldots, \sigma_n)$.

Examples II



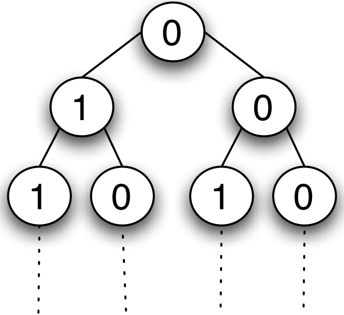
$$\sigma(\varepsilon) = 0$$
$$\sigma_L = \sigma + [1]$$
$$\sigma_R = \sigma$$

Examples II



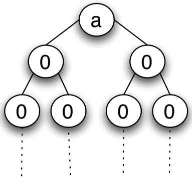
$$\begin{aligned} \sigma(\varepsilon) &= 0 \\ \sigma_L &= \sigma + [1] \\ \sigma_R &= \sigma \end{aligned}$$

Examples II



$$\begin{aligned} \sigma(\varepsilon) &= 0 \\ \sigma_L &= \sigma + [1] \\ \sigma_R &= \sigma \end{aligned}$$

$[a]$ denotes



Examples III – The Thue-Morse sequence

- Obtained from the parities of the counts of 1's in the binary representation of non negative integers.
- 0,1,1,0,1,0,0,1, . . .
- Can be obtained by the substitution map $\{0 \to 01; 1 \to 10\}$:

$$0 \to 01 \to 0110 \to 01101001 \to \dots$$

- Tree representation (at level k, we have 2^k digits)

Examples III – The Thue-Morse sequence

- Obtained from the parities of the counts of 1's in the binary representation of non negative integers.
- 0,1,1,0,1,0,0,1, . . .
- Can be obtained by the substitution map $\{0 \to 01; 1 \to 10\}$:

$$0 \to 01 \to 0110 \to 01101001 \to \ldots$$

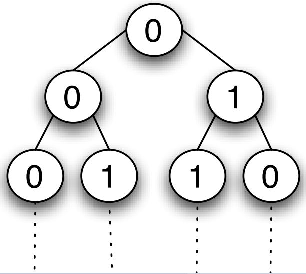
- Tree representation (at level k, we have 2^k digits)

Examples III – The Thue-Morse sequence

- Obtained from the parities of the counts of 1's in the binary representation of non negative integers.
- 0,1,1,0,1,0,0,1, ...
- Can be obtained by the substitution map $\{0 \to 01; 1 \to 10\}$:

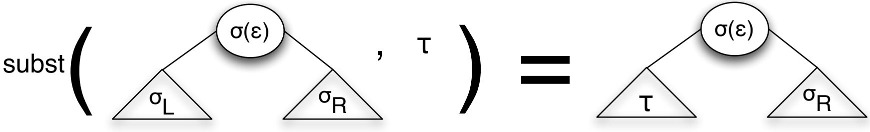
$$0 \to 01 \to 0110 \to 01101001 \to \dots$$

- Tree representation (at level k, we have 2^k digits)



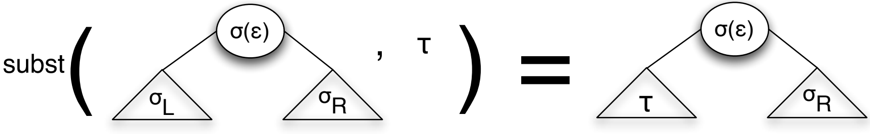
$$\begin{array}{rcl} thue(\varepsilon) & = & 0 \\ thue_L & = & thue \\ thue_R & = & thue + ones \end{array}$$

Examples IV – Substitution operation



$$
\begin{aligned}
(subst(\sigma,\tau))(\varepsilon) &= \sigma(\varepsilon)\\
(subst(\sigma,\tau))_L &= \tau\\
(subst(\sigma,\tau))_R &= \sigma_R
\end{aligned}
$$

Examples IV – Substitution operation



$$
\begin{array}{rcl}
(\mathit{subst}(\sigma,\tau))(\varepsilon) & = & \sigma(\varepsilon) \\
(\mathit{subst}(\sigma,\tau))_L & = & \tau \\
(\mathit{subst}(\sigma,\tau))_R & = & \sigma_R
\end{array}
$$

Operations on trees

From formal power series we inherit several definitions of operations:

Name	Sum	Product
Initial value	$(\sigma+\tau)(\varepsilon)=\sigma(\varepsilon)+\tau(\varepsilon)$	$(\sigma \times \tau)(\varepsilon) = \sigma(\varepsilon) \times \tau(\varepsilon)$
Left der.	$(\sigma + \tau)_L = \sigma_L + \tau_L$	$(\sigma \times \tau)_L = \sigma_L \times \tau + \sigma(\varepsilon) \times \tau_L$
Right der	$(\sigma + \tau)_R = \sigma_R + \tau_R$	$(\sigma \times \tau)_R = \sigma_R \times \tau + \sigma(\varepsilon) \times \tau_R$

Fundamental Theorem

For all infinite binary trees $\sigma \in T_A$:

$$\sigma = \sigma(\varepsilon) + L \times \sigma_L + R \times \sigma_R$$

where

$$\begin{array}{ll} L(\varepsilon) = 0 & R(\varepsilon) = 0 \\ L_L = [1] & R_L = [0] \\ L_R = [0] & R_R = [1] \end{array}$$

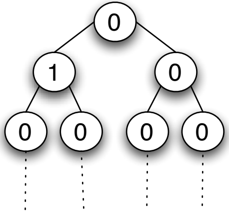
Fundamental Theorem

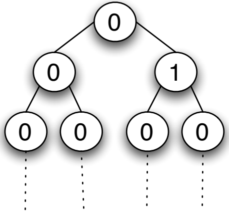
For all infinite binary trees $\sigma \in T_A$:

$$\sigma = \sigma(\varepsilon) + L \times \sigma_L + R \times \sigma_R$$

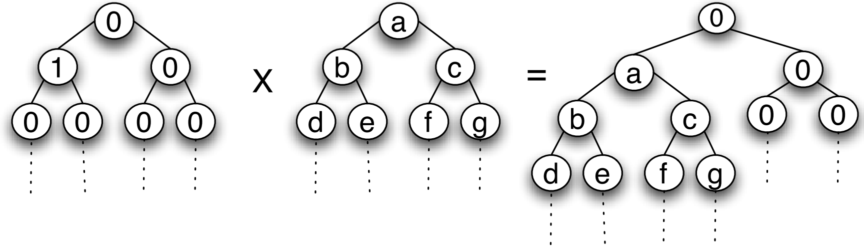
where

$$L(\varepsilon) = 0 \qquad R(\varepsilon) = 0$$
$$L_L = [1] \qquad R_L = [0]$$
$$L_R = [0] \qquad R_R = [1]$$



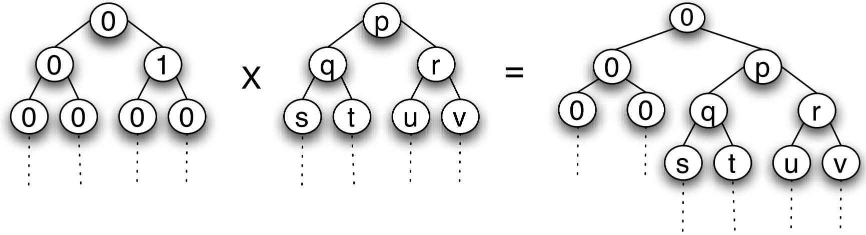


$L \times \sigma_L$

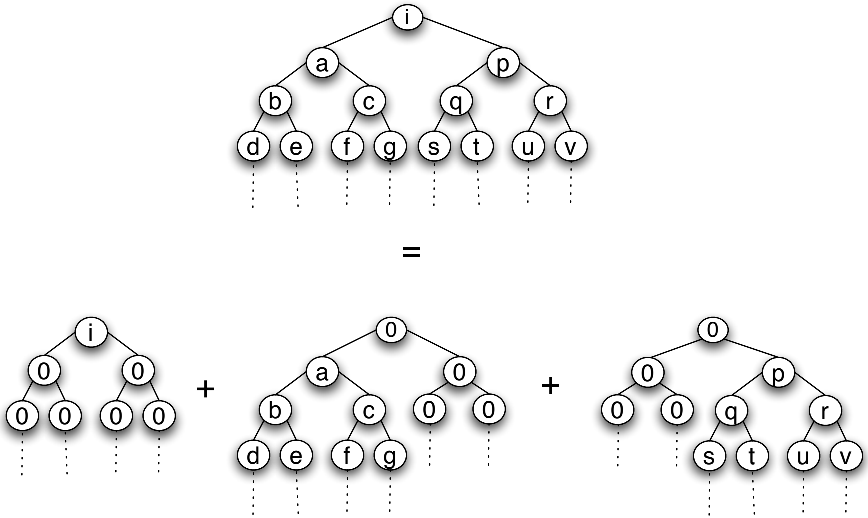


$R \times \sigma_R$

Similarly:



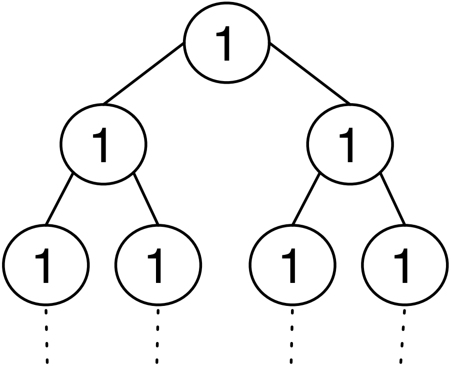
$\sigma = \sigma(\varepsilon) + L \times \sigma_L + R \times \sigma_R$



But. . . What can we do with this theorem?

?

Examples Revisited I



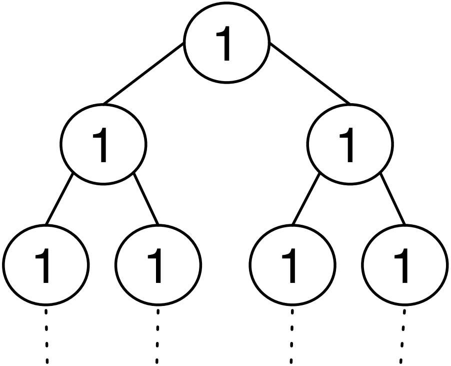
$$\begin{aligned} ones(\varepsilon) &= 1 \\ ones_L &= ones \\ ones_R &= ones \end{aligned}$$

Closed Formula I

$$\begin{aligned} & ones = 1 + L \times ones + R \times ones \\ \Leftrightarrow \quad & (1 - L - R) \times ones = 1 \\ \Leftrightarrow \quad & ones = \frac{1}{1-L-R} \end{aligned}$$

$\dfrac{x}{y}$ stands for $y^{-1} \times x$

Examples Revisited I



$$
\begin{array}{rcl}
ones(\varepsilon) & = & 1 \\
ones_L & = & ones \\
ones_R & = & ones
\end{array}
$$

Closed Formula I

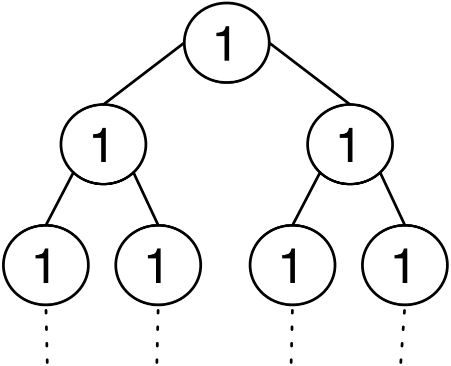
$$ones = 1 + L \times ones + R \times ones$$

$$\Leftrightarrow \quad (1 - L - R) \times ones = 1$$

$$\Leftrightarrow \quad ones = \frac{1}{1-L-R}$$

$\frac{x}{y}$ stands for $y^{-1} \times x$

Examples Revisited I



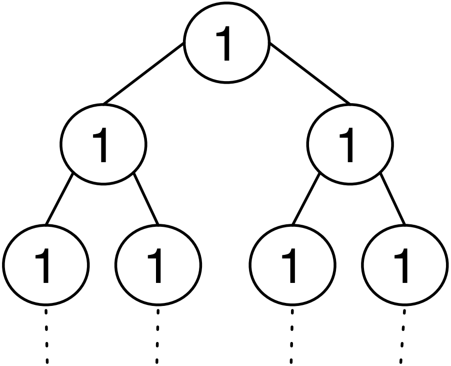
$$ones(\varepsilon) = 1$$
$$ones_L = ones$$
$$ones_R = ones$$

Closed Formula I

$$ones = 1 + L \times ones + R \times ones$$
$$\Leftrightarrow \quad (1 - L - R) \times ones = 1$$
$$\Leftrightarrow \quad ones = \frac{1}{1-L-R}$$

$\frac{x}{y}$ stands for $y^{-1} \times x$

Examples Revisited I



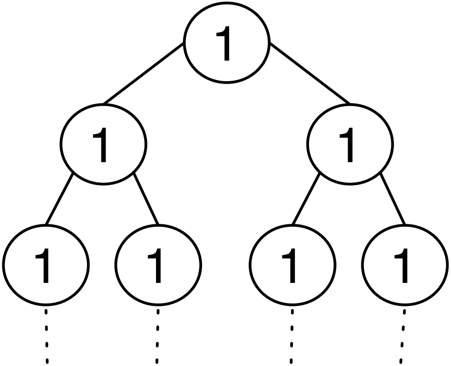
$$\begin{array}{rcl} ones(\varepsilon) & = & 1 \\ ones_L & = & ones \\ ones_R & = & ones \end{array}$$

Closed Formula I

$$\begin{array}{rc} & ones = 1 + L \times ones + R \times ones \\ \Leftrightarrow & (1 - L - R) \times ones = 1 \\ \Leftrightarrow & ones = \frac{1}{1-L-R} \end{array}$$

$\frac{x}{y}$ stands for $y^{-1} \times x$

Examples Revisited I



$$\begin{array}{rcl} ones(\varepsilon) & = & 1 \\ ones_L & = & ones \\ ones_R & = & ones \end{array}$$

Closed Formula I

$$ones = 1 + L \times ones + R \times ones$$

$$\Leftrightarrow \quad (1 - L - R) \times ones = 1$$

$$\Leftrightarrow \quad ones = \frac{1}{1-L-R}$$

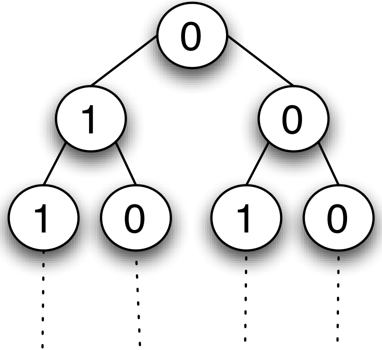
$\frac{x}{y}$ stands for $y^{-1} \times x$

Inverse operation

The inverse of a tree – σ^{-1} – is defined formally so that $\sigma \times \sigma^{-1} = 1$.

$$\begin{array}{rcl} \sigma^{-1}(\varepsilon) & = & (\sigma(\varepsilon))^{-1} \\ (\sigma^{-1})_L & = & (\sigma(\varepsilon))^{-1} \times \sigma_L \times \sigma^{-1} \\ (\sigma^{-1})_R & = & (\sigma(\varepsilon))^{-1} \times \sigma_R \times \sigma^{-1} \end{array}$$

Examples Revisited II

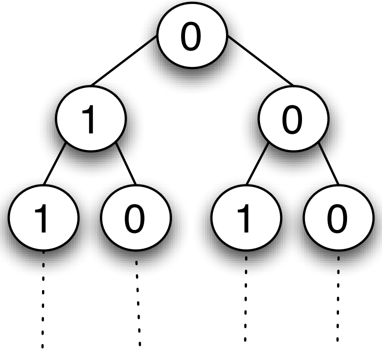


$$\begin{aligned}\sigma(\varepsilon) &= 0\\ \sigma_L &= \sigma + 1\\ \sigma_R &= \sigma\end{aligned}$$

Closed Formula II

$$\begin{aligned} & \sigma = 0 + L \times (\sigma + 1) + R \times \sigma \\ \Leftrightarrow \quad & (1 - L - R)\sigma = L \\ \Leftrightarrow \quad & \sigma = \frac{L}{1-L-R}\end{aligned}$$

Examples Revisited II



$$\begin{array}{rcl} \sigma(\varepsilon) & = & 0 \\ \sigma_L & = & \sigma + 1 \\ \sigma_R & = & \sigma \end{array}$$

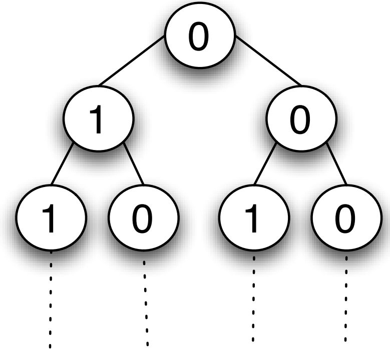
Closed Formula II

$$\sigma = 0 + L \times (\sigma + 1) + R \times \sigma$$

$$\Leftrightarrow \quad (1 - L - R)\sigma = L$$

$$\Leftrightarrow \quad \sigma = \frac{L}{1 - L - R}$$

Examples Revisited II

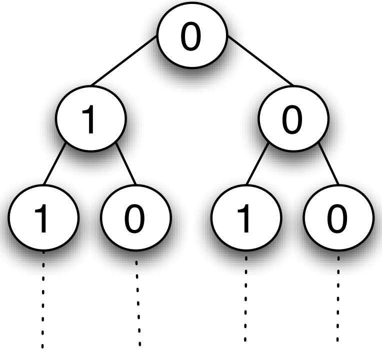


$$\begin{aligned} \sigma(\varepsilon) &= 0 \\ \sigma_L &= \sigma + 1 \\ \sigma_R &= \sigma \end{aligned}$$

Closed Formula II

$$\begin{aligned} & \sigma = 0 + L \times (\sigma + 1) + R \times \sigma \\ \Leftrightarrow \quad & (1 - L - R)\sigma = L \\ \Leftrightarrow \quad & \sigma = \frac{L}{1 - L - R} \end{aligned}$$

Examples Revisited II

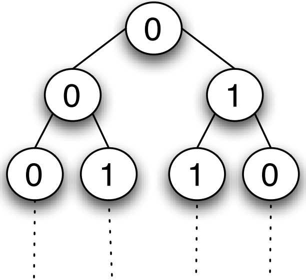


$$\begin{aligned} \sigma(\varepsilon) &= 0 \\ \sigma_L &= \sigma + 1 \\ \sigma_R &= \sigma \end{aligned}$$

Closed Formula II

$$\begin{aligned} & \sigma = 0 + L \times (\sigma + 1) + R \times \sigma \\ \Leftrightarrow \quad & (1 - L - R)\sigma = L \\ \Leftrightarrow \quad & \sigma = \frac{L}{1-L-R} \end{aligned}$$

Examples revisited III – The Thue-Morse sequence



$$\begin{array}{rcl} \sigma(\varepsilon) & = & 0 \\ \sigma_L & = & \sigma \\ \sigma_R & = & \sigma + \mathit{ones} \end{array}$$

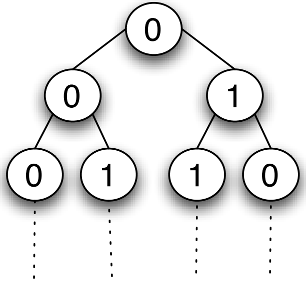
Closed Formula III

$$\sigma = 0 + L \times \sigma + R \times (\sigma + \mathit{ones})$$

$$\Leftrightarrow \quad (1 - L - R)\sigma = R \times \frac{1}{1-L-R}$$

$$\Leftrightarrow \quad \sigma = \frac{1}{1-L-R} \times R \times \frac{1}{1-L-R}$$

Examples revisited III – The Thue-Morse sequence



$$\begin{array}{rcl} \sigma(\varepsilon) & = & 0 \\ \sigma_L & = & \sigma \\ \sigma_R & = & \sigma + \mathit{ones} \end{array}$$

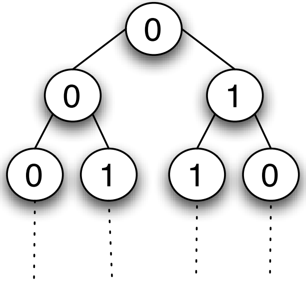
Closed Formula III

$$\sigma = 0 + L \times \sigma + R \times (\sigma + \mathit{ones})$$

$$\Leftrightarrow \quad (1 - L - R)\sigma = R \times \frac{1}{1-L-R}$$

$$\Leftrightarrow \quad \sigma = \frac{1}{1-L-R} \times R \times \frac{1}{1-L-R}$$

Examples revisited III – The Thue-Morse sequence

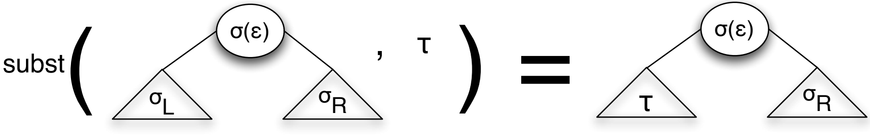


$$\sigma(\varepsilon) = 0$$
$$\sigma_L = \sigma$$
$$\sigma_R = \sigma + \mathit{ones}$$

Closed Formula III

$$\sigma = 0 + L \times \sigma + R \times (\sigma + \mathit{ones})$$
$$\Leftrightarrow \quad (1 - L - R)\sigma = R \times \frac{1}{1-L-R}$$
$$\Leftrightarrow \quad \sigma = \frac{1}{1-L-R} \times R \times \frac{1}{1-L-R}$$

Examples revisited IV – Substitution



$$\begin{aligned}(\mathit{subst}(\sigma,\tau))(\varepsilon) &= \sigma(\varepsilon)\\ (\mathit{subst}(\sigma,\tau))_L &= \tau\\ (\mathit{subst}(\sigma,\tau))_R &= \sigma_R\end{aligned}$$

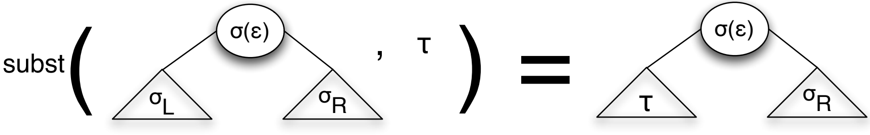
Closed Formula IV

$$\mathit{subst}(\sigma,\tau) = \sigma(\varepsilon) + L \times \tau + R \times \sigma_R$$

$$\Leftrightarrow \quad \{\, \sigma - L \times \sigma_L = \sigma(\varepsilon) + R \times \sigma_R \}$$

$$\mathit{subst}(\sigma,\tau) = \sigma - L \times (\sigma_L - \tau)$$

Examples revisited IV – Substitution



$$
\begin{array}{rcl}
(subst(\sigma,\tau))(\varepsilon) & = & \sigma(\varepsilon) \\
(subst(\sigma,\tau))_L & = & \tau \\
(subst(\sigma,\tau))_R & = & \sigma_R
\end{array}
$$

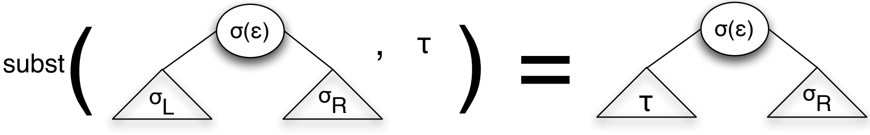
Closed Formula IV

$$subst(\sigma,\tau) = \sigma(\varepsilon) + L \times \tau + R \times \sigma_R$$

$$\Leftrightarrow \quad \{\sigma - L \times \sigma_L = \sigma(\varepsilon) + R \times \sigma_R\}$$

$$subst(\sigma,\tau) = \sigma - L \times (\sigma_L - \tau)$$

Examples revisited IV – Substitution



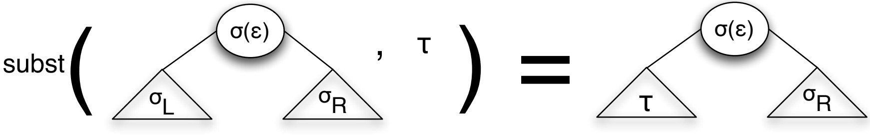
$$
\begin{aligned}
(\mathit{subst}(\sigma,\tau))(\varepsilon) &= \sigma(\varepsilon)\\
(\mathit{subst}(\sigma,\tau))_L &= \tau\\
(\mathit{subst}(\sigma,\tau))_R &= \sigma_R
\end{aligned}
$$

Closed Formula IV

$$\mathit{subst}(\sigma,\tau) = \sigma(\varepsilon) + L \times \tau + R \times \sigma_R$$

$$\Leftrightarrow \quad \{\,\sigma - L \times \sigma_L = \sigma(\varepsilon) + R \times \sigma_R\}$$

$$\mathit{subst}(\sigma,\tau) = \sigma - L \times (\sigma_L - \tau)$$



$$
\begin{array}{lcl}
(\mathit{subst}(\sigma,\tau))(\varepsilon) & = & \sigma(\varepsilon) \\
(\mathit{subst}(\sigma,\tau))_L & = & \tau \\
(\mathit{subst}(\sigma,\tau))_R & = & \sigma_R
\end{array}
$$

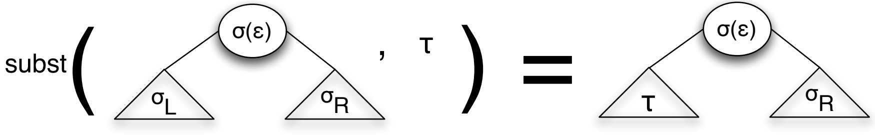
Closed Formula IV

$$\mathit{subst}(\sigma,\tau) = \sigma(\varepsilon) + L \times \tau + R \times \sigma_R$$

$$\Leftrightarrow \quad \{\, \sigma - L \times \sigma_L = \sigma(\varepsilon) + R \times \sigma_R \}$$

$$\mathit{subst}(\sigma,\tau) = \sigma - L \times (\sigma_L - \tau)$$

Examples revisited IV – Substitution



$$
\begin{aligned}
(\mathit{subst}(\sigma,\tau))(\varepsilon) &= \sigma(\varepsilon)\\
(\mathit{subst}(\sigma,\tau))_L &= \tau\\
(\mathit{subst}(\sigma,\tau))_R &= \sigma_R
\end{aligned}
$$

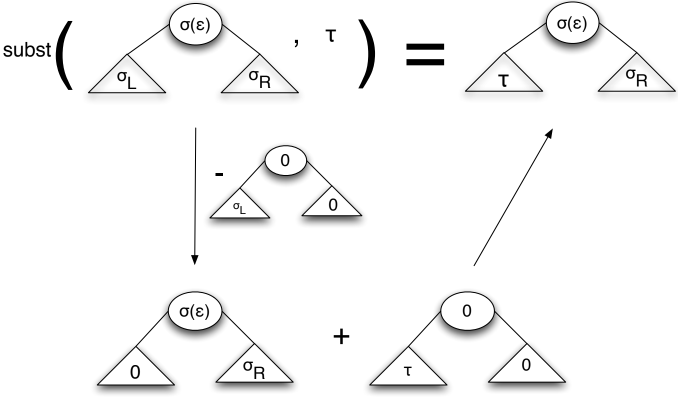
Closed Formula IV

$$
\mathit{subst}(\sigma,\tau) = \sigma(\varepsilon) + L \times \tau + R \times \sigma_R
$$

$$
\Leftrightarrow \qquad \{\, \sigma - L \times \sigma_L = \sigma(\varepsilon) + R \times \sigma_R \}
$$

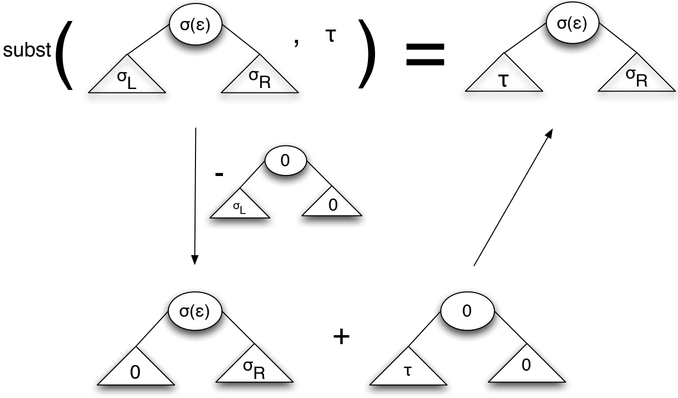
$$
\mathit{subst}(\sigma,\tau) = \sigma - L \times (\sigma_L - \tau)
$$

$subst(\sigma, \tau) = \sigma - L \times (\sigma_L - \tau)$



Easily generalizes: $subst(\sigma, \tau, P) = \sigma - P(\sigma_P - \tau)$

$subst(\sigma, \tau) = \sigma - L \times (\sigma_L - \tau)$



Easily generalizes: $subst(\sigma, \tau, P) = \sigma - P(\sigma_P - \tau)$

Proving properties

subst$(\sigma, \sigma_P, P) = \sigma$ and *subst*$(subst(\sigma, \tau, P), \sigma_P, P) = \sigma$

- **Property 1**

$$subst(\sigma, \sigma_P, P) = \sigma - P(\sigma_P - \sigma_P) = \sigma$$

- **Property 2**

$$\begin{aligned} & subst(subst(\sigma, \tau, P), \sigma_P, P) & \\ = \; & subst(\sigma - P(\sigma_P - \tau), \sigma_P, P) & \text{Def. of } subst \\ = \; & \sigma - P(\sigma_P - \tau) - P((\sigma - P(\sigma_P - \tau))_P - \sigma_P) & \text{Def. of } subst \\ = \; & \sigma - P(\sigma_P - \tau) - P(\tau - \sigma_P) & \\ = \; & \sigma & \end{aligned}$$

Proving properties

subst$(\sigma, \sigma_P, P) = \sigma$ and *subst*$(\textit{subst}(\sigma, \tau, P), \sigma_P, P) = \sigma$

- **Property 1**

$$\textit{subst}(\sigma, \sigma_P, P) = \sigma - P(\sigma_P - \sigma_P) = \sigma$$

- **Property 2**

$$\begin{aligned} & \textit{subst}(\textit{subst}(\sigma, \tau, P), \sigma_P, P) & \\ = \; & \textit{subst}(\sigma - P(\sigma_P - \tau), \sigma_P, P) & \text{Def. of } \textit{subst} \\ = \; & \sigma - P(\sigma_P - \tau) - P((\sigma - P(\sigma_P - \tau))_P - \sigma_P) & \text{Def. of } \textit{subst} \\ = \; & \sigma - P(\sigma_P - \tau) - P(\tau - \sigma_P) & \\ = \; & \sigma & \end{aligned}$$

Proving properties

$subst(\sigma, \sigma_P, P) = \sigma$ and $subst(subst(\sigma, \tau, P), \sigma_P, P) = \sigma$

- **Property 1**

$$subst(\sigma, \sigma_P, P) = \sigma - P(\sigma_P - \sigma_P) = \sigma$$

- **Property 2**

$$\begin{array}{lll} & subst(subst(\sigma, \tau, P), \sigma_P, P) & \\ = & subst(\sigma - P(\sigma_P - \tau), \sigma_P, P) & \text{Def. of } subst \\ = & \sigma - P(\sigma_P - \tau) - P((\sigma - P(\sigma_P - \tau))_P - \sigma_P) & \text{Def. of } subst \\ = & \sigma - P(\sigma_P - \tau) - P(\tau - \sigma_P) & \\ = & \sigma & \end{array}$$

Conclusions

- Modelling binary trees as formal power series we have derived a calculus
- We have defined various non-trivial trees by means of simple differential equations
- We have showed how to compute compact closed formulae for them
- We have also illustrated that this can be used to prove properties about trees

Conclusions

- Modelling binary trees as formal power series we have derived a calculus
- We have defined various non-trivial trees by means of simple differential equations
- We have showed how to compute compact closed formulae for them
- We have also illustrated that this can be used to prove properties about trees

Conclusions

- Modelling binary trees as formal power series we have derived a calculus
- We have defined various non-trivial trees by means of simple differential equations
- We have showed how to compute compact closed formulae for them
- We have also illustrated that this can be used to prove properties about trees

Conclusions

- Modelling binary trees as formal power series we have derived a calculus
- We have defined various non-trivial trees by means of simple differential equations
- We have showed how to compute compact closed formulae for them
- We have also illustrated that this can be used to prove properties about trees

Future work

- Behavioural differential equations are closely related to lazy functional programming implementations.
- In particular, we would like to study the relation between closed expressions and elimination of corecursion
- The closed formula that we have obtained for the Thue-Morse sequence suggests a possible use of coinduction and differential equations in the area of automatic sequences.