On First-order Cons-free Term Rewriting and PTIME

Cynthia Kop

2 Apr, 2016

Overview

Overview

In 2001, Jones:

In 2001, Jones:

- defined a minimal functional programming language operating on lists

In 2001, Jones:

- defined a minimal functional programming language operating on lists
- introduced *cons-free* programs, which cannot generate new data

In 2001, Jones:

- defined a minimal functional programming language operating on lists
- introduced *cons-free* programs, which cannot generate new data
- showed that programs of data order k can simulate arbitrary deterministic Turing Machines in $\mathcal{O}(\mathtt{exp}_2^k(n))$

In 2001, Jones:

- defined a minimal functional programming language operating on lists
- introduced *cons-free* programs, which cannot generate new data
- showed that programs of data order k can simulate arbitrary deterministic Turing Machines in $\mathcal{O}(\mathtt{exp}_2^k(n))$
- showed that arbitrary programs of data order k can be simulated by deterministic Turing Machines in $\mathcal{O}(\mathtt{exp}_2^k(n))$

In 2001, Jones:

- defined a minimal functional programming language operating on lists
- introduced *cons-free* programs, which cannot generate new data
- showed that programs of data order k can simulate arbitrary deterministic Turing Machines in $\mathcal{O}(\mathtt{exp}_2^k(n))$
- showed that arbitrary programs of data order k can be simulated by deterministic Turing Machines in $\mathcal{O}(\mathtt{exp}_2^k(n))$
- $\Longrightarrow$ proved that programs of data order k *characterise* $\mathrm{EXPTIME}^k$

In 2001, Jones:

- defined a minimal functional programming language operating on lists
- introduced *cons-free* programs, which cannot generate new data
- showed that programs of data order k can simulate arbitrary deterministic Turing Machines in $\mathcal{O}(\mathtt{exp}_2^k(n))$
- showed that arbitrary programs of data order k can be simulated by deterministic Turing Machines in $\mathcal{O}(\mathtt{exp}_2^k(n))$
- $\Longrightarrow$ proved that programs of data order k *characterise* EXPTIME^k
- (proved several other characterisations as well)

In 2001, Jones:

- defined a minimal functional programming language operating on lists
- introduced *cons-free* programs, which cannot generate new data
- showed that programs of data order k can simulate arbitrary deterministic Turing Machines in $\mathcal{O}(\exp_2^k(n))$
- showed that arbitrary programs of data order k can be simulated by deterministic Turing Machines in $\mathcal{O}(\exp_2^k(n))$
- $\Longrightarrow$ proved that programs of data order k *characterise* EXPTIME^k
- (proved several other characterisations as well)

In 2001, Jones:

- defined a minimal functional programming language operating on lists
- introduced *cons-free* programs, which cannot generate new data
- showed that programs of data order k can simulate arbitrary deterministic Turing Machines in $\mathcal{O}(\exp_2^k(n))$
- showed that arbitrary programs of data order k can be simulated by deterministic Turing Machines in $\mathcal{O}(\exp_2^k(n))$
- $\Longrightarrow$ proved that programs of data order k *characterise* EXPTIME^k
- (proved several other characterisations as well)

Why not term rewriting?

Term rewriting:

Term rewriting:

- is the core calculus underlying functional programming

Term rewriting:

- is the core calculus underlying functional programming
- has a small definition (including higher-order versions)

Term rewriting:

- is the core calculus underlying functional programming
- has a small definition (including higher-order versions)
- already avoids complications present in real-world languages

Term rewriting:

- is the core calculus underlying functional programming
- has a small definition (including higher-order versions)
- already avoids complications present in real-world languages
- natively includes non-determinism

Term rewriting:

- is the core calculus underlying functional programming
- has a small definition (including higher-order versions)
- already avoids complications present in real-world languages
- natively includes non-determinism
- is supported by a large community with decades of theory

Term rewriting:

- is the core calculus underlying functional programming
- has a small definition (including higher-order versions)
- already avoids complications present in real-world languages
- natively includes non-determinism
- is supported by a large community with decades of theory
- allows for fine-tuning in many ways

Term rewriting:

- is the core calculus underlying functional programming
- has a small definition (including higher-order versions)
- already avoids complications present in real-world languages
- natively includes non-determinism
- is supported by a large community with decades of theory
- allows for fine-tuning in many ways

Let's see what comes out!

Overview

Terms: trees built from variables and function symbols.
Rules: generate a relation between terms.

Terms: trees built from variables and function symbols.
Rules: generate a relation between terms.

Example:
Function symbols:

$$\begin{array}{rclrcl} 0 & : & [] \Rightarrow \mathsf{N} & \mathsf{Add} & : & [\mathsf{N} \times \mathsf{N}] \Rightarrow \mathsf{N} \\ \mathsf{S} & : & [\mathsf{N}] \Rightarrow \mathsf{N} & \mathsf{Mul} & : & [\mathsf{N} \times \mathsf{N}] \Rightarrow \mathsf{N} \end{array}$$

Terms: trees built from variables and function symbols.
Rules: generate a relation between terms.

Example:
Function symbols:

$$\begin{array}{lclclcl} \mathsf{0} & : & [] \Rightarrow \mathsf{N} & \mathsf{Add} & : & [\mathsf{N} \times \mathsf{N}] \Rightarrow \mathsf{N} \\ \mathsf{S} & : & [\mathsf{N}] \Rightarrow \mathsf{N} & \mathsf{Mul} & : & [\mathsf{N} \times \mathsf{N}] \Rightarrow \mathsf{N} \end{array}$$

Rules:

$$\begin{array}{rcl} \mathsf{Add}(x, \mathsf{0}) & \to & x \\ \mathsf{Add}(\mathsf{0}, y) & \to & y \\ \mathsf{Add}(x, \mathsf{S}(y)) & \to & \mathsf{S}(\mathsf{Add}(x, y)) \\ \mathsf{Mul}(x, \mathsf{0}) & \to & \mathsf{0} \\ \mathsf{Mul}(x, \mathsf{S}(y)) & \to & \mathsf{Add}(\mathsf{Mul}(x, y), x) \end{array}$$

Terms: trees built from variables and function symbols.
Rules: generate a relation between terms.

Example:
Function symbols:

$$\begin{array}{lclclcl} \mathsf{0} & : & [] \Rightarrow \mathsf{N} & \mathsf{Add} & : & [\mathsf{N} \times \mathsf{N}] \Rightarrow \mathsf{N} \\ \mathsf{S} & : & [\mathsf{N}] \Rightarrow \mathsf{N} & \mathsf{Mul} & : & [\mathsf{N} \times \mathsf{N}] \Rightarrow \mathsf{N} \end{array}$$

Rules:

$$\begin{array}{rcl} \mathsf{Add}(x, \mathsf{0}) & \to & x \\ \mathsf{Add}(\mathsf{0}, y) & \to & y \\ \mathsf{Add}(x, \mathsf{S}(y)) & \to & \mathsf{S}(\mathsf{Add}(x, y)) \\ \mathsf{Mul}(x, \mathsf{0}) & \to & \mathsf{0} \\ \mathsf{Mul}(x, \mathsf{S}(y)) & \to & \mathsf{Add}(\mathsf{Mul}(x, y), x) \end{array}$$

Example reduction:
$\mathsf{Add}(\mathsf{S}(\mathsf{0}), \mathsf{S}(\mathsf{0}))$

Terms: trees built from variables and function symbols.
Rules: generate a relation between terms.

Example:
Function symbols:

$$\begin{array}{lclclcl} 0 & : & [] \Rightarrow \mathsf{N} & \mathsf{Add} & : & [\mathsf{N} \times \mathsf{N}] \Rightarrow \mathsf{N} \\ \mathsf{S} & : & [\mathsf{N}] \Rightarrow \mathsf{N} & \mathsf{Mul} & : & [\mathsf{N} \times \mathsf{N}] \Rightarrow \mathsf{N} \end{array}$$

Rules:

$$\begin{array}{rcl} \mathsf{Add}(x, 0) & \to & x \\ \mathsf{Add}(0, y) & \to & y \\ \mathsf{Add}(x, \mathsf{S}(y)) & \to & \mathsf{S}(\mathsf{Add}(x, y)) \\ \mathsf{Mul}(x, 0) & \to & 0 \\ \mathsf{Mul}(x, \mathsf{S}(y)) & \to & \mathsf{Add}(\mathsf{Mul}(x, y), x) \end{array}$$

Example reduction:
$\mathsf{Add}(\mathsf{S}(0), \mathsf{S}(0))$

Terms: trees built from variables and function symbols.
Rules: generate a relation between terms.

Example:
Function symbols:

$$\begin{array}{lllllll} 0 & : & [] \Rightarrow \mathsf{N} & \mathsf{Add} & : & [\mathsf{N} \times \mathsf{N}] \Rightarrow \mathsf{N} \\ \mathsf{S} & : & [\mathsf{N}] \Rightarrow \mathsf{N} & \mathsf{Mul} & : & [\mathsf{N} \times \mathsf{N}] \Rightarrow \mathsf{N} \end{array}$$

Rules:

$$\begin{array}{rcl} \mathsf{Add}(x, 0) & \to & x \\ \mathsf{Add}(0, y) & \to & y \\ \mathsf{Add}(x, \mathsf{S}(y)) & \to & \mathsf{S}(\mathsf{Add}(x, y)) \\ \mathsf{Mul}(x, 0) & \to & 0 \\ \mathsf{Mul}(x, \mathsf{S}(y)) & \to & \mathsf{Add}(\mathsf{Mul}(x, y), x) \end{array}$$

Example reduction:
$\mathsf{Add}(\mathsf{S}(0), \mathsf{S}(0)) \to \mathsf{S}(\mathsf{Add}(\mathsf{S}(0), 0))$

Terms: trees built from variables and function symbols.
Rules: generate a relation between terms.

Example:
Function symbols:

$$\begin{array}{lllllll} 0 & : & [] \Rightarrow \mathsf{N} & \mathsf{Add} & : & [\mathsf{N} \times \mathsf{N}] \Rightarrow \mathsf{N} \\ \mathsf{S} & : & [\mathsf{N}] \Rightarrow \mathsf{N} & \mathsf{Mul} & : & [\mathsf{N} \times \mathsf{N}] \Rightarrow \mathsf{N} \end{array}$$

Rules:

$$\begin{array}{rcl} \mathsf{Add}(x, 0) & \to & x \\ \mathsf{Add}(0, y) & \to & y \\ \mathsf{Add}(x, \mathsf{S}(y)) & \to & \mathsf{S}(\mathsf{Add}(x, y)) \\ \mathsf{Mul}(x, 0) & \to & 0 \\ \mathsf{Mul}(x, \mathsf{S}(y)) & \to & \mathsf{Add}(\mathsf{Mul}(x, y), x) \end{array}$$

Example reduction:
$\mathsf{Add}(\mathsf{S}(0), \mathsf{S}(0)) \to \mathsf{S}(\mathsf{Add}(\mathsf{S}(0), 0))$

Terms: trees built from variables and function symbols.
Rules: generate a relation between terms.

Example:
Function symbols:

$$\begin{array}{lllllll} 0 & : & [] \Rightarrow \mathsf{N} & \mathsf{Add} & : & [\mathsf{N} \times \mathsf{N}] \Rightarrow \mathsf{N} \\ \mathsf{S} & : & [\mathsf{N}] \Rightarrow \mathsf{N} & \mathsf{Mul} & : & [\mathsf{N} \times \mathsf{N}] \Rightarrow \mathsf{N} \end{array}$$

Rules:

$$\begin{array}{rcl} \mathsf{Add}(x, 0) & \rightarrow & x \\ \mathsf{Add}(0, y) & \rightarrow & y \\ \mathsf{Add}(x, \mathsf{S}(y)) & \rightarrow & \mathsf{S}(\mathsf{Add}(x, y)) \\ \mathsf{Mul}(x, 0) & \rightarrow & 0 \\ \mathsf{Mul}(x, \mathsf{S}(y)) & \rightarrow & \mathsf{Add}(\mathsf{Mul}(x, y), x) \end{array}$$

Example reduction:
$\mathsf{Add}(\mathsf{S}(0), \mathsf{S}(0)) \rightarrow \mathsf{S}(\mathsf{Add}(\mathsf{S}(0), 0))$

Terms: trees built from variables and function symbols.
Rules: generate a relation between terms.

Example:
Function symbols:

$$\begin{array}{lclclcl} 0 & : & [] \Rightarrow \mathsf{N} & \mathsf{Add} & : & [\mathsf{N} \times \mathsf{N}] \Rightarrow \mathsf{N} \\ \mathsf{S} & : & [\mathsf{N}] \Rightarrow \mathsf{N} & \mathsf{Mul} & : & [\mathsf{N} \times \mathsf{N}] \Rightarrow \mathsf{N} \end{array}$$

Rules:

$$\begin{array}{rcl} \mathsf{Add}(x,0) & \to & x \\ \mathsf{Add}(0,y) & \to & y \\ \mathsf{Add}(x,\mathsf{S}(y)) & \to & \mathsf{S}(\mathsf{Add}(x,y)) \\ \mathsf{Mul}(x,0) & \to & 0 \\ \mathsf{Mul}(x,\mathsf{S}(y)) & \to & \mathsf{Add}(\mathsf{Mul}(x,y),x) \end{array}$$

Example reduction:
$\mathsf{Add}(\mathsf{S}(0),\mathsf{S}(0)) \to \mathsf{S}(\mathsf{Add}(\mathsf{S}(0),0)) \to \mathsf{S}(\mathsf{S}(0))$

Peculiarities

Note that:

Peculiarities

Note that:

- rule selection is non-deterministic (unless otherwise specified)

Peculiarities

Note that:

- rule selection is non-deterministic (unless otherwise specified)
- position selection is non-deterministic (unless otherwise specified)

Peculiarities

Note that:

- rule selection is non-deterministic (unless otherwise specified)
- position selection is non-deterministic (unless otherwise specified)

$$\begin{array}{rcl} \mathsf{Add}(x,0) & \to & x \\ \mathsf{Add}(0,y) & \to & y \\ \mathsf{Add}(x,\mathsf{S}(y)) & \to & \mathsf{S}(\mathsf{Add}(x,y)) \\ \mathsf{Mul}(x,0) & \to & 0 \\ \mathsf{Mul}(x,\mathsf{S}(y)) & \to & \mathsf{Add}(\mathsf{Mul}(x,y),x) \end{array}$$

Example reduction:

$\mathsf{Mul}(\mathsf{S}(0),\mathsf{S}(0))$

Peculiarities

Note that:

- rule selection is non-deterministic (unless otherwise specified)
- position selection is non-deterministic (unless otherwise specified)

$$\begin{array}{rcl} \mathsf{Add}(x,0) & \to & x \\ \mathsf{Add}(0,y) & \to & y \\ \mathsf{Add}(x,\mathsf{S}(y)) & \to & \mathsf{S}(\mathsf{Add}(x,y)) \\ \mathsf{Mul}(x,0) & \to & 0 \\ \mathsf{Mul}(x,\mathsf{S}(y)) & \to & \mathsf{Add}(\mathsf{Mul}(x,y),x) \end{array}$$

Example reduction:

$\mathsf{Mul}(\mathsf{S}(0),\mathsf{S}(0))$

Peculiarities

Note that:

- rule selection is non-deterministic (unless otherwise specified)
- position selection is non-deterministic (unless otherwise specified)

$$\begin{array}{rcl} \mathsf{Add}(x,0) & \to & x \\ \mathsf{Add}(0,y) & \to & y \\ \mathsf{Add}(x,\mathsf{S}(y)) & \to & \mathsf{S}(\mathsf{Add}(x,y)) \\ \mathsf{Mul}(x,0) & \to & 0 \\ \mathsf{Mul}(x,\mathsf{S}(y)) & \to & \mathsf{Add}(\mathsf{Mul}(x,y),x) \end{array}$$

Example reduction:

$\mathsf{Mul}(\mathsf{S}(0),\mathsf{S}(0)) \to \mathsf{Add}(\mathsf{Mul}(\mathsf{S}(0),0),\mathsf{S}(0))$

Peculiarities

Note that:

- rule selection is non-deterministic (unless otherwise specified)
- position selection is non-deterministic (unless otherwise specified)

$$
\begin{array}{rcl}
\mathsf{Add}(x,0) & \to & x \\
\mathsf{Add}(0,y) & \to & y \\
\mathsf{Add}(x,\mathsf{S}(y)) & \to & \mathsf{S}(\mathsf{Add}(x,y)) \\
\mathsf{Mul}(x,0) & \to & 0 \\
\mathsf{Mul}(x,\mathsf{S}(y)) & \to & \mathsf{Add}(\mathsf{Mul}(x,y),x)
\end{array}
$$

Example reduction:
$\mathsf{Mul}(\mathsf{S}(0),\mathsf{S}(0)) \to \mathsf{Add}(\mathsf{Mul}(\mathsf{S}(0),0),\mathsf{S}(0))$

Peculiarities

Note that:

- rule selection is non-deterministic (unless otherwise specified)
- position selection is non-deterministic (unless otherwise specified)

$$\begin{array}{rcl} \mathsf{Add}(x, 0) & \rightarrow & x \\ \mathsf{Add}(0, y) & \rightarrow & y \\ \mathsf{Add}(x, \mathsf{S}(y)) & \rightarrow & \mathsf{S}(\mathsf{Add}(x, y)) \\ \mathsf{Mul}(x, 0) & \rightarrow & 0 \\ \mathsf{Mul}(x, \mathsf{S}(y)) & \rightarrow & \mathsf{Add}(\mathsf{Mul}(x, y), x) \end{array}$$

Example reduction:

$\mathsf{Mul}(\mathsf{S}(0), \mathsf{S}(0)) \rightarrow \mathsf{Add}(\mathsf{Mul}(\mathsf{S}(0), 0), \mathsf{S}(0))$

Peculiarities

Note that:

- rule selection is non-deterministic (unless otherwise specified)
- position selection is non-deterministic (unless otherwise specified)

$$\begin{array}{rcl} \mathsf{Add}(x,0) & \to & x \\ \mathsf{Add}(0,y) & \to & y \\ \mathsf{Add}(x,\mathsf{S}(y)) & \to & \mathsf{S}(\mathsf{Add}(x,y)) \\ \mathsf{Mul}(x,0) & \to & 0 \\ \mathsf{Mul}(x,\mathsf{S}(y)) & \to & \mathsf{Add}(\mathsf{Mul}(x,y),x) \end{array}$$

Example reduction:

$\mathsf{Mul}(\mathsf{S}(0),\mathsf{S}(0)) \to \mathsf{Add}(\mathsf{Mul}(\mathsf{S}(0),0),\mathsf{S}(0))$

Peculiarities

Note that:

- rule selection is non-deterministic (unless otherwise specified)
- position selection is non-deterministic (unless otherwise specified)

$$\begin{array}{rcl} \mathsf{Add}(x, 0) & \to & x \\ \mathsf{Add}(0, y) & \to & y \\ \mathsf{Add}(x, \mathsf{S}(y)) & \to & \mathsf{S}(\mathsf{Add}(x, y)) \\ \mathsf{Mul}(x, 0) & \to & 0 \\ \mathsf{Mul}(x, \mathsf{S}(y)) & \to & \mathsf{Add}(\mathsf{Mul}(x, y), x) \end{array}$$

Example reduction:

$\mathsf{Mul}(\mathsf{S}(0), \mathsf{S}(0)) \to \mathsf{Add}(\mathsf{Mul}(\mathsf{S}(0), 0), \mathsf{S}(0)) \to$
$\mathsf{S}(\mathsf{Add}(\mathsf{Mul}(\mathsf{S}(0), 0), 0))$

Peculiarities

Note that:

- rule selection is non-deterministic (unless otherwise specified)
- position selection is non-deterministic (unless otherwise specified)

$$
\begin{array}{rcl}
\mathsf{Add}(x, 0) & \rightarrow & x \\
\mathsf{Add}(0, y) & \rightarrow & y \\
\mathsf{Add}(x, \mathsf{S}(y)) & \rightarrow & \mathsf{S}(\mathsf{Add}(x, y)) \\
\mathsf{Mul}(x, 0) & \rightarrow & 0 \\
\mathsf{Mul}(x, \mathsf{S}(y)) & \rightarrow & \mathsf{Add}(\mathsf{Mul}(x, y), x)
\end{array}
$$

Example reduction:

$\mathsf{Mul}(\mathsf{S}(0), \mathsf{S}(0)) \rightarrow \mathsf{Add}(\mathsf{Mul}(\mathsf{S}(0), 0), \mathsf{S}(0)) \rightarrow$
$\mathsf{S}(\mathsf{Add}(\mathsf{Mul}(\mathsf{S}(0), 0), 0))$

Peculiarities

Note that:

- rule selection is non-deterministic (unless otherwise specified)
- position selection is non-deterministic (unless otherwise specified)

$$
\begin{array}{rcl}
\mathsf{Add}(x, 0) & \to & x \\
\mathsf{Add}(0, y) & \to & y \\
\mathsf{Add}(x, \mathsf{S}(y)) & \to & \mathsf{S}(\mathsf{Add}(x, y)) \\
\mathsf{Mul}(x, 0) & \to & 0 \\
\mathsf{Mul}(x, \mathsf{S}(y)) & \to & \mathsf{Add}(\mathsf{Mul}(x, y), x)
\end{array}
$$

Example reduction:

$\mathsf{Mul}(\mathsf{S}(0), \mathsf{S}(0)) \to \mathsf{Add}(\mathsf{Mul}(\mathsf{S}(0), 0), \mathsf{S}(0)) \to \mathsf{S}(\mathsf{Add}(\mathsf{Mul}(\mathsf{S}(0), 0), 0))$

Peculiarities

Note that:

- rule selection is non-deterministic (unless otherwise specified)
- position selection is non-deterministic (unless otherwise specified)

$$\begin{array}{rcl} \mathsf{Add}(x, 0) & \to & x \\ \mathsf{Add}(0, y) & \to & y \\ \mathsf{Add}(x, \mathsf{S}(y)) & \to & \mathsf{S}(\mathsf{Add}(x, y)) \\ \mathsf{Mul}(x, 0) & \to & 0 \\ \mathsf{Mul}(x, \mathsf{S}(y)) & \to & \mathsf{Add}(\mathsf{Mul}(x, y), x) \end{array}$$

Example reduction:

$\mathsf{Mul}(\mathsf{S}(0), \mathsf{S}(0)) \to \mathsf{Add}(\mathsf{Mul}(\mathsf{S}(0), 0), \mathsf{S}(0)) \to \mathsf{S}(\mathsf{Add}(\mathsf{Mul}(\mathsf{S}(0), 0), 0)) \to \mathsf{S}(\mathsf{Add}(0, 0))$

Peculiarities

Note that:

- rule selection is non-deterministic (unless otherwise specified)
- position selection is non-deterministic (unless otherwise specified)

$$
\begin{array}{rcl}
\mathsf{Add}(x,0) & \to & x \\
\mathsf{Add}(0,y) & \to & y \\
\mathsf{Add}(x,\mathsf{S}(y)) & \to & \mathsf{S}(\mathsf{Add}(x,y)) \\
\mathsf{Mul}(x,0) & \to & 0 \\
\mathsf{Mul}(x,\mathsf{S}(y)) & \to & \mathsf{Add}(\mathsf{Mul}(x,y),x)
\end{array}
$$

Example reduction:

$\mathsf{Mul}(\mathsf{S}(0),\mathsf{S}(0)) \to \mathsf{Add}(\mathsf{Mul}(\mathsf{S}(0),0),\mathsf{S}(0)) \to$
$\mathsf{S}(\mathsf{Add}(\mathsf{Mul}(\mathsf{S}(0),0),0)) \to \mathsf{S}(\mathsf{Add}(0,0))$

Overview

Extending cons-free

Extending cons-free

- in Jones: programs not containing the list constructor ("cons-free")

Extending cons-free

- in Jones: programs not containing the list constructor ("cons-free")
- in practice: no new data is created

Extending cons-free

Function symbols:

$$\begin{array}{lcllcl} 0 & : & [] \Rightarrow \mathsf{N} & \mathsf{Add} & : & [\mathsf{N} \times \mathsf{N}] \Rightarrow \mathsf{N} \\ \mathsf{S} & : & [\mathsf{N}] \Rightarrow \mathsf{N} & \mathsf{Mul} & : & [\mathsf{N} \times \mathsf{N}] \Rightarrow \mathsf{N} \end{array}$$

Rules:

$$\begin{array}{rcl} \mathsf{Add}(x, 0) & \to & x \\ \mathsf{Add}(0, y) & \to & y \\ \mathsf{Add}(x, \mathsf{S}(y)) & \to & \mathsf{S}(\mathsf{Add}(x, y)) \\ \mathsf{Mul}(x, 0) & \to & 0 \\ \mathsf{Mul}(x, \mathsf{S}(y)) & \to & \mathsf{Add}(\mathsf{Mul}(x, y), x) \end{array}$$

Extending cons-free

Function symbols:

$$\begin{array}{lllllll} 0 & : & [] \Rightarrow \mathsf{N} & \mathsf{Add} & : & [\mathsf{N} \times \mathsf{N}] \Rightarrow \mathsf{N} \\ \mathsf{S} & : & [\mathsf{N}] \Rightarrow \mathsf{N} & \mathsf{Mul} & : & [\mathsf{N} \times \mathsf{N}] \Rightarrow \mathsf{N} \end{array}$$

Rules:

$$\begin{array}{rcl} \mathsf{Add}(x, 0) & \to & x \\ \mathsf{Add}(0, y) & \to & y \\ \mathsf{Add}(x, \mathsf{S}(y)) & \to & \mathsf{S}(\mathsf{Add}(x, y)) \\ \mathsf{Mul}(x, 0) & \to & 0 \\ \mathsf{Mul}(x, \mathsf{S}(y)) & \to & \mathsf{Add}(\mathsf{Mul}(x, y), x) \end{array}$$

Extending cons-free

Function symbols:

$$\begin{array}{lllllll} 0 & : & [] \Rightarrow \mathsf{N} & \mathsf{Add} & : & [\mathsf{N} \times \mathsf{N}] \Rightarrow \mathsf{N} \\ \mathsf{S} & : & [\mathsf{N}] \Rightarrow \mathsf{N} & \mathsf{Mul} & : & [\mathsf{N} \times \mathsf{N}] \Rightarrow \mathsf{N} \end{array}$$

Rules:

$$\begin{array}{rcl} \mathsf{Add}(x, 0) & \to & x \\ \mathsf{Add}(0, y) & \to & y \\ \mathsf{Add}(x, \mathsf{S}(y)) & \to & \mathsf{S}(\mathsf{Add}(x, y)) \\ \mathsf{Mul}(x, 0) & \to & 0 \\ \mathsf{Mul}(x, \mathsf{S}(y)) & \to & \mathsf{Add}(\mathsf{Mul}(x, y), x) \end{array}$$

Extending cons-free

Function symbols:

$$\begin{array}{lllllll} 0 & : & [] \Rightarrow \mathsf{N} & \mathsf{Add} & : & [\mathsf{N} \times \mathsf{N}] \Rightarrow \mathsf{N} \\ \mathsf{S} & : & [\mathsf{N}] \Rightarrow \mathsf{N} & \mathsf{Mul} & : & [\mathsf{N} \times \mathsf{N}] \Rightarrow \mathsf{N} \end{array}$$

Rules:

$$\begin{array}{rcl} \mathsf{Add}(x, 0) & \to & x \\ \mathsf{Add}(0, y) & \to & y \\ \mathsf{Add}(x, \mathsf{S}(y)) & \to & \underline{\mathsf{S}}(\mathsf{Add}(x, y)) \\ \mathsf{Mul}(x, 0) & \to & 0 \\ \mathsf{Mul}(x, \mathsf{S}(y)) & \to & \mathsf{Add}(\mathsf{Mul}(x, y), x) \end{array}$$

Cons-free Term Rewriting:

- symbols are split into data and function symbols

Cons-free Term Rewriting:

- symbols are split into data and function symbols
- right-hand sides create no new data:

Cons-free Term Rewriting:

- symbols are split into data and function symbols
- right-hand sides create no new data:
  - right-hand sides *may* contain terms exclusively built of constructors (e.g. $\mathsf{True}, \mathsf{False}, \mathsf{S}(0)$)

Cons-free Term Rewriting:

- symbols are split into data and function symbols
- right-hand sides create no new data:
  - right-hand sides *may* contain terms exclusively built of constructors (e.g. $\mathsf{True}, \mathsf{False}, \mathsf{S}(0)$)
  - right-hand sides *may* contain subterms of their left-hand side (to allow for pattern matching)

Cons-free Term Rewriting:

- symbols are split into data and function symbols
- right-hand sides create no new data:
  - right-hand sides *may* contain terms exclusively built of constructors (e.g. $\mathsf{True}, \mathsf{False}, \mathsf{S}(0)$)
  - right-hand sides *may* contain subterms of their left-hand side (to allow for pattern matching)
  - right-hand sides *may not* contain data symbols otherwise

Cons-free Term Rewriting:

- symbols are split into data and function symbols
- right-hand sides create no new data:
  - right-hand sides *may* contain terms exclusively built of constructors (e.g. $\mathsf{True}, \mathsf{False}, \mathsf{S}(0)$)
  - right-hand sides *may* contain subterms of their left-hand side (to allow for pattern matching)
  - right-hand sides *may not* contain data symbols otherwise

$$\begin{array}{rcl} \mathsf{Hd}(h;t) & \to & h \\ \mathsf{Tl}(h;t) & \to & t \end{array}$$

Cons-free Term Rewriting:

- symbols are split into data and function symbols
- right-hand sides create no new data:
  - right-hand sides *may* contain terms exclusively built of constructors (e.g. $\mathsf{True}, \mathsf{False}, \mathsf{S}(0)$)
  - right-hand sides *may* contain subterms of their left-hand side (to allow for pattern matching)
  - right-hand sides *may not* contain data symbols otherwise

$$
\begin{array}{rcl}
\mathsf{Evenlength}(l) & \to & \mathsf{Helper}(l, \mathsf{True}) \\
\mathsf{Helper}([], b) & \to & b \\
\mathsf{Helper}(h;t, \mathsf{True}) & \to & \mathsf{Helper}(t, \mathsf{False}) \\
\mathsf{Helper}(h;t, \mathsf{False}) & \to & \mathsf{Helper}(t, \mathsf{True})
\end{array}
$$

Cons-free Term Rewriting:

- symbols are split into data and function symbols
- right-hand sides create no new data:
  - right-hand sides *may* contain terms exclusively built of constructors (e.g. $\mathsf{True}, \mathsf{False}, \mathsf{S}(0)$)
  - right-hand sides *may* contain subterms of their left-hand side (to allow for pattern matching)
  - right-hand sides *may not* contain data symbols otherwise

$$\begin{array}{rcl} \mathsf{Half}(0) & \to & 0 \\ \mathsf{Half}(\mathsf{S}(0)) & \to & 0 \\ \mathsf{Half}(\mathsf{S}(\mathsf{S}(x))) & \to & \mathsf{S}(\mathsf{Half}(x)) \end{array}$$

Cons-free Term Rewriting:

- symbols are split into data and function symbols
- right-hand sides create no new data:
  - right-hand sides *may* contain terms exclusively built of constructors (e.g. $\mathsf{True}, \mathsf{False}, \mathsf{S}(0)$)
  - right-hand sides *may* contain subterms of their left-hand side (to allow for pattern matching)
  - right-hand sides *may not* contain data symbols otherwise

$$\begin{array}{rcl} \mathsf{Half}(0) & \to & 0 \\ \mathsf{Half}(\mathsf{S}(0)) & \to & 0 \\ \mathsf{Half}(\mathsf{S}(\mathsf{S}(x))) & \to & \mathsf{S}(\mathsf{Half}(x)) \end{array}$$

Cons-free Term Rewriting:

- symbols are split into data and function symbols
- right-hand sides create no new data:
  - right-hand sides *may* contain terms exclusively built of constructors (e.g. $\mathsf{True}, \mathsf{False}, \mathsf{S}(0)$)
  - right-hand sides *may* contain subterms of their left-hand side (to allow for pattern matching)
  - right-hand sides *may not* contain data symbols otherwise

$$\begin{array}{rcl} \mathsf{Half}(x) & \rightarrow & \mathsf{Helper}(x, x) \\ \mathsf{Helper}(0, y) & \rightarrow & y \\ \mathsf{Helper}(\mathsf{S}(0), \mathsf{S}(y)) & \rightarrow & y \\ \mathsf{Helper}(\mathsf{S}(\mathsf{S}(x)), \mathsf{S}(y)) & \rightarrow & \mathsf{Helper}(x, y) \end{array}$$

Cons-free Term Rewriting:

- symbols are split into data and function symbols
- right-hand sides create no new data:
  - right-hand sides *may* contain terms exclusively built of constructors (e.g. $\mathsf{True}, \mathsf{False}, \mathsf{S}(0)$)
  - right-hand sides *may* contain subterms of their left-hand side (to allow for pattern matching)
  - right-hand sides *may not* contain data symbols otherwise

$$\mathsf{Succ}(x) \quad \rightarrow \quad \mathsf{S}(x)$$

Cons-free Term Rewriting:

- symbols are split into data and function symbols
- right-hand sides create no new data:
  - right-hand sides *may* contain terms exclusively built of constructors (e.g. $\mathsf{True}, \mathsf{False}, \mathsf{S}(0)$)
  - right-hand sides *may* contain subterms of their left-hand side (to allow for pattern matching)
  - right-hand sides *may not* contain data symbols otherwise

$$\mathsf{Succ}(x) \quad \rightarrow \quad \mathsf{S}(x)$$

Cons-free Term Rewriting:

- symbols are split into data and function symbols
- right-hand sides create no new data:
  - right-hand sides *may* contain terms exclusively built of constructors (e.g. $\mathsf{True}, \mathsf{False}, \mathsf{S}(0)$)
  - right-hand sides *may* contain subterms of their left-hand side (to allow for pattern matching)
  - right-hand sides *may not* contain data symbols otherwise

$$\begin{array}{rcl} \mathsf{Last}(h;[]) & \to & h \\ \mathsf{Last}(h_1;h_2;tl) & \to & \mathsf{Last}(h_2;tl) \end{array}$$

Cons-free Term Rewriting:

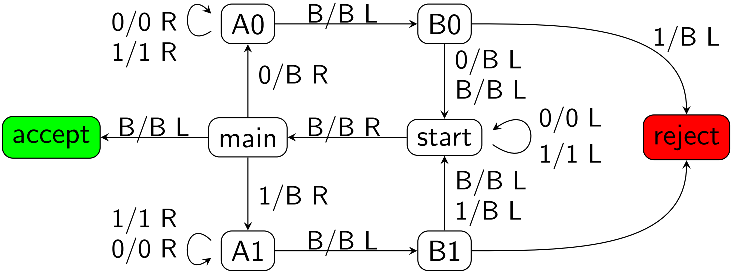
- symbols are split into data and function symbols
- right-hand sides create no new data:
    - right-hand sides *may* contain terms exclusively built of constructors (e.g. $\mathsf{True}, \mathsf{False}, \mathsf{S}(0)$)
    - right-hand sides *may* contain subterms of their left-hand side (to allow for pattern matching)
    - right-hand sides *may not* contain data symbols otherwise

$$\begin{array}{rcl} \mathsf{Last}(h; []) & \to & h \\ \mathsf{Last}(h_1;h_2;tl) & \to & \mathsf{Last}(h_2;tl) \end{array}$$

Claim: we can simulate any PTIME Turing Machine

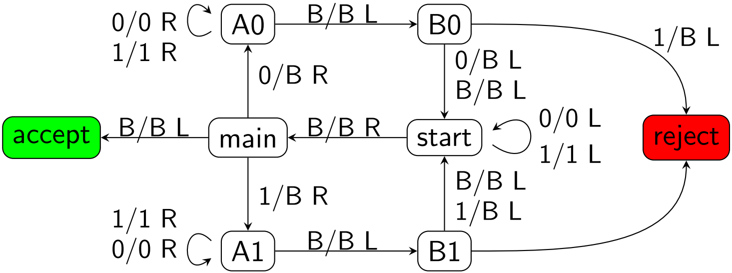
Claim: we can simulate any PTIME Turing Machine

Example:



Claim: we can simulate any PTIME Turing Machine

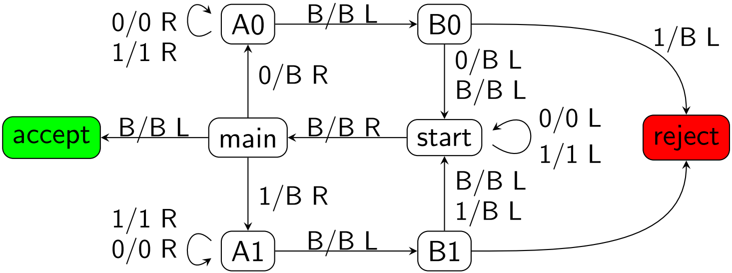
Example:



Runs in: $< 2 \cdot (n+1)^2$ steps

Claim: we can simulate any PTIME Turing Machine

Example:

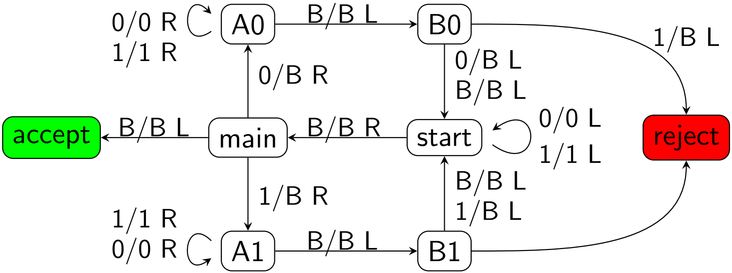


Runs in: $< 2 \cdot (n+1)^2$ steps

$$\mathsf{Transition}(\mathsf{Start}, 0) \quad \rightarrow \quad \mathsf{X}(\mathsf{Start}, 0, \mathsf{L})$$

Claim: we can simulate any PTIME Turing Machine

Example:

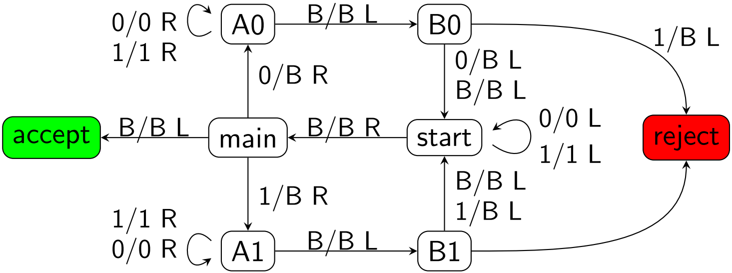


Runs in: $< 2 \cdot (n+1)^2$ steps

$$\begin{array}{lcl} \mathsf{Transition}(\mathsf{Start}, 0) & \to & \mathsf{X}(\mathsf{Start}, 0, \mathsf{L}) \\ \mathsf{Transition}(\mathsf{Start}, 1) & \to & \mathsf{X}(\mathsf{Start}, 1, \mathsf{L}) \\ \mathsf{Transition}(\mathsf{Start}, \mathsf{B}) & \to & \mathsf{X}(\mathsf{Main}, \mathsf{B}, \mathsf{R}) \end{array}$$

Claim: we can simulate any PTIME Turing Machine

Example:

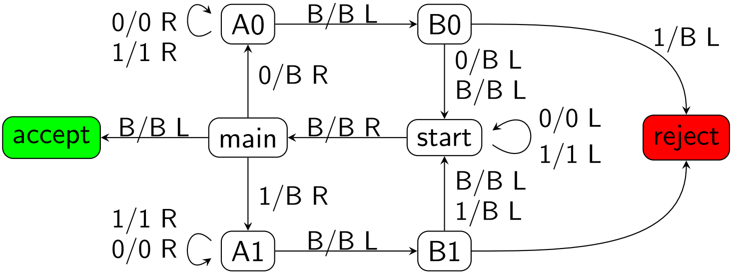


Runs in: $< 2 \cdot (n+1)^2$ steps

$$
\begin{array}{rcl}
\mathsf{Transition}(\mathsf{Start}, 0) & \to & \mathsf{X}(\mathsf{Start}, 0, \mathsf{L}) \\
\mathsf{Transition}(\mathsf{Start}, 1) & \to & \mathsf{X}(\mathsf{Start}, 1, \mathsf{L}) \\
\mathsf{Transition}(\mathsf{Start}, \mathsf{B}) & \to & \mathsf{X}(\mathsf{Main}, \mathsf{B}, \mathsf{R}) \\
\mathsf{Transition}(\mathsf{Main}, 0) & \to & \mathsf{X}(\mathsf{A0}, \mathsf{B}, \mathsf{R})
\end{array}
$$

Claim: we can simulate any PTIME Turing Machine

Example:

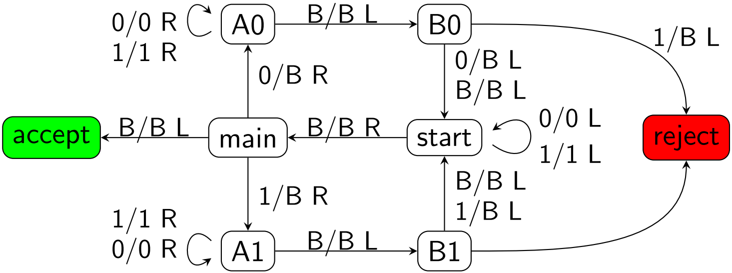


Runs in: $< 2 \cdot (n+1)^2$ steps

$$
\begin{array}{rcl}
\mathsf{Transition}(\mathsf{Start}, 0) & \to & \mathsf{X}(\mathsf{Start}, 0, \mathsf{L}) \\
\mathsf{Transition}(\mathsf{Start}, 1) & \to & \mathsf{X}(\mathsf{Start}, 1, \mathsf{L}) \\
\mathsf{Transition}(\mathsf{Start}, \mathsf{B}) & \to & \mathsf{X}(\mathsf{Main}, \mathsf{B}, \mathsf{R}) \\
\mathsf{Transition}(\mathsf{Main}, 0) & \to & \mathsf{X}(\mathsf{A0}, \mathsf{B}, \mathsf{R}) \\
\mathsf{Transition}(\mathsf{Accept}, x) & \to & \mathsf{X}(\mathsf{Accept}, x, \mathsf{N})
\end{array}
$$

Claim: we can simulate any PTIME Turing Machine

Example:



Runs in: $< 2 \cdot (n+1)^2$ steps

$$\begin{array}{rcl}
\mathsf{Transition}(\mathsf{Start},0) & \to & \mathsf{X}(\mathsf{Start},0,\mathsf{L}) \\
\mathsf{Transition}(\mathsf{Start},1) & \to & \mathsf{X}(\mathsf{Start},1,\mathsf{L}) \\
\mathsf{Transition}(\mathsf{Start},\mathsf{B}) & \to & \mathsf{X}(\mathsf{Main},\mathsf{B},\mathsf{R}) \\
\mathsf{Transition}(\mathsf{Main},0) & \to & \mathsf{X}(\mathsf{A0},\mathsf{B},\mathsf{R}) \\
\mathsf{Transition}(\mathsf{Accept},x) & \to & \mathsf{X}(\mathsf{Accept},x,\mathsf{N}) \\
& \dots &
\end{array}$$

Simulation: Turing Machine running in $< 2 \cdot (n+1)^2$ steps.

$$\begin{array}{rcl} \mathsf{Transition}(\mathsf{Start}, 0) & \to & \mathsf{X}(\mathsf{Start}, 0, \mathsf{L}) \\ \mathsf{Transition}(\mathsf{Start}, 1) & \to & \mathsf{X}(\mathsf{Start}, 1, \mathsf{L}) \\ \mathsf{Transition}(\mathsf{Start}, \mathsf{B}) & \to & \mathsf{X}(\mathsf{Main}, \mathsf{B}, \mathsf{R}) \\ \mathsf{Transition}(\mathsf{Main}, 0) & \to & \mathsf{X}(\mathsf{A0}, \mathsf{B}, \mathsf{R}) \\ \mathsf{Transition}(\mathsf{Accept}, x) & \to & \mathsf{X}(\mathsf{Accept}, x, \mathsf{S}) \\ & \dots & \end{array}$$

Simulation: Turing Machine running in $< 2 \cdot (n+1)^2$ steps.

$$\begin{array}{rcl}
\mathsf{Transition}(\mathsf{Start}, 0) & \to & \mathsf{X}(\mathsf{Start}, 0, \mathsf{L}) \\
\mathsf{Transition}(\mathsf{Start}, 1) & \to & \mathsf{X}(\mathsf{Start}, 1, \mathsf{L}) \\
\mathsf{Transition}(\mathsf{Start}, \mathsf{B}) & \to & \mathsf{X}(\mathsf{Main}, \mathsf{B}, \mathsf{R}) \\
\mathsf{Transition}(\mathsf{Main}, 0) & \to & \mathsf{X}(\mathsf{A0}, \mathsf{B}, \mathsf{R}) \\
\mathsf{Transition}(\mathsf{Accept}, x) & \to & \mathsf{X}(\mathsf{Accept}, x, \mathsf{S}) \\
& \dots &
\end{array}$$

Representation: $(l_1, l_2, l_3) \Longrightarrow |l_1| \cdot (n+1)^2 + |l_2| \cdot (n+1) + |l_3|$

Simulation: Turing Machine running in $< 2 \cdot (n+1)^2$ steps.

$$\begin{array}{rcl} \mathsf{Transition}(\mathsf{Start}, 0) & \to & \mathsf{X}(\mathsf{Start}, 0, \mathsf{L}) \\ \mathsf{Transition}(\mathsf{Start}, 1) & \to & \mathsf{X}(\mathsf{Start}, 1, \mathsf{L}) \\ \mathsf{Transition}(\mathsf{Start}, \mathsf{B}) & \to & \mathsf{X}(\mathsf{Main}, \mathsf{B}, \mathsf{R}) \\ \mathsf{Transition}(\mathsf{Main}, 0) & \to & \mathsf{X}(\mathsf{A0}, \mathsf{B}, \mathsf{R}) \\ \mathsf{Transition}(\mathsf{Accept}, x) & \to & \mathsf{X}(\mathsf{Accept}, x, \mathsf{S}) \\ & \dots & \end{array}$$

Representation: $(l_1, l_2, l_3) \Longrightarrow |l_1| \cdot (n+1)^2 + |l_2| \cdot (n+1) + |l_3|$

$$\mathsf{TransitionAt}(i, t_1, t_2, t_3) \quad \to$$

Simulation: Turing Machine running in $< 2 \cdot (n+1)^2$ steps.

$$
\begin{array}{rcl}
\mathsf{Transition}(\mathsf{Start}, 0) & \to & \mathsf{X}(\mathsf{Start}, 0, \mathsf{L}) \\
\mathsf{Transition}(\mathsf{Start}, 1) & \to & \mathsf{X}(\mathsf{Start}, 1, \mathsf{L}) \\
\mathsf{Transition}(\mathsf{Start}, \mathsf{B}) & \to & \mathsf{X}(\mathsf{Main}, \mathsf{B}, \mathsf{R}) \\
\mathsf{Transition}(\mathsf{Main}, 0) & \to & \mathsf{X}(\mathsf{A0}, \mathsf{B}, \mathsf{R}) \\
\mathsf{Transition}(\mathsf{Accept}, x) & \to & \mathsf{X}(\mathsf{Accept}, x, \mathsf{S}) \\
 & \dots &
\end{array}
$$

Representation: $(l_1, l_2, l_3) \Longrightarrow |l_1| \cdot (n+1)^2 + |l_2| \cdot (n+1) + |l_3|$

$$
\begin{array}{rcl}
\mathsf{TransitionAt}(i, t_1, t_2, t_3) & \to & \mathsf{Transition}(\mathsf{State}(i, t_1, t_2, t_3), \\
 & & \qquad \mathsf{CurTape}(i, t_1, t_2, t_3))
\end{array}
$$

Simulation: Turing Machine running in $< 2 \cdot (n+1)^2$ steps.

$$\begin{array}{rcl}
\mathsf{Transition}(\mathsf{Start}, 0) & \to & \mathsf{X}(\mathsf{Start}, 0, \mathsf{L}) \\
\mathsf{Transition}(\mathsf{Start}, 1) & \to & \mathsf{X}(\mathsf{Start}, 1, \mathsf{L}) \\
\mathsf{Transition}(\mathsf{Start}, \mathsf{B}) & \to & \mathsf{X}(\mathsf{Main}, \mathsf{B}, \mathsf{R}) \\
\mathsf{Transition}(\mathsf{Main}, 0) & \to & \mathsf{X}(\mathsf{A0}, \mathsf{B}, \mathsf{R}) \\
\mathsf{Transition}(\mathsf{Accept}, x) & \to & \mathsf{X}(\mathsf{Accept}, x, \mathsf{S}) \\
& \dots &
\end{array}$$

Representation: $(l_1, l_2, l_3) \Longrightarrow |l_1| \cdot (n+1)^2 + |l_2| \cdot (n+1) + |l_3|$

$$\begin{array}{rcl}
\mathsf{TransitionAt}(i, t_1, t_2, t_3) & \to & \mathsf{Transition}(\mathsf{State}(i, t_1, t_2, t_3), \\
& & \qquad \mathsf{CurTape}(i, t_1, t_2, t_3)) \\
\mathsf{State}(i, [], [], []) & \to &
\end{array}$$

Simulation: Turing Machine running in $< 2 \cdot (n+1)^2$ steps.

$$
\begin{array}{rcl}
\mathsf{Transition}(\mathsf{Start}, 0) & \to & \mathsf{X}(\mathsf{Start}, 0, \mathsf{L}) \\
\mathsf{Transition}(\mathsf{Start}, 1) & \to & \mathsf{X}(\mathsf{Start}, 1, \mathsf{L}) \\
\mathsf{Transition}(\mathsf{Start}, \mathsf{B}) & \to & \mathsf{X}(\mathsf{Main}, \mathsf{B}, \mathsf{R}) \\
\mathsf{Transition}(\mathsf{Main}, 0) & \to & \mathsf{X}(\mathsf{A0}, \mathsf{B}, \mathsf{R}) \\
\mathsf{Transition}(\mathsf{Accept}, x) & \to & \mathsf{X}(\mathsf{Accept}, x, \mathsf{S}) \\
& \dots &
\end{array}
$$

Representation: $(l_1, l_2, l_3) \Longrightarrow |l_1| \cdot (n+1)^2 + |l_2| \cdot (n+1) + |l_3|$

$$
\begin{array}{rcl}
\mathsf{TransitionAt}(i, t_1, t_2, t_3) & \to & \mathsf{Transition}(\mathsf{State}(i, t_1, t_2, t_3), \\
& & \qquad \mathsf{CurTape}(i, t_1, t_2, t_3)) \\
\mathsf{State}(i, [], [], []) & \to & \mathsf{Start}
\end{array}
$$

Simulation: Turing Machine running in $< 2 \cdot (n+1)^2$ steps.

$$
\begin{array}{rcl}
\mathsf{Transition}(\mathsf{Start}, 0) & \to & \mathsf{X}(\mathsf{Start}, 0, \mathsf{L}) \\
\mathsf{Transition}(\mathsf{Start}, 1) & \to & \mathsf{X}(\mathsf{Start}, 1, \mathsf{L}) \\
\mathsf{Transition}(\mathsf{Start}, \mathsf{B}) & \to & \mathsf{X}(\mathsf{Main}, \mathsf{B}, \mathsf{R}) \\
\mathsf{Transition}(\mathsf{Main}, 0) & \to & \mathsf{X}(\mathsf{A0}, \mathsf{B}, \mathsf{R}) \\
\mathsf{Transition}(\mathsf{Accept}, x) & \to & \mathsf{X}(\mathsf{Accept}, x, \mathsf{S}) \\
 & \dots &
\end{array}
$$

Representation: $(l_1, l_2, l_3) \Longrightarrow |l_1| \cdot (n+1)^2 + |l_2| \cdot (n+1) + |l_3|$

$$
\begin{array}{rcl}
\mathsf{TransitionAt}(i, t_1, t_2, t_3) & \to & \mathsf{Transition}(\mathsf{State}(i, t_1, t_2, t_3), \\
 & & \qquad \mathsf{CurTape}(i, t_1, t_2, t_3)) \\
\mathsf{State}(i, [], [], []) & \to & \mathsf{Start} \\
\mathsf{State}(i, t_1, t_2, x;y) & \to & \\
\mathsf{State}(i, t_1, x;y, []) & \to & \\
\mathsf{State}(i, x;y, [], []) & \to &
\end{array}
$$

Simulation: Turing Machine running in $< 2 \cdot (n+1)^2$ steps.

$$\begin{array}{rcl}
\mathsf{Transition}(\mathsf{Start}, 0) & \to & \mathsf{X}(\mathsf{Start}, 0, \mathsf{L}) \\
\mathsf{Transition}(\mathsf{Start}, 1) & \to & \mathsf{X}(\mathsf{Start}, 1, \mathsf{L}) \\
\mathsf{Transition}(\mathsf{Start}, \mathsf{B}) & \to & \mathsf{X}(\mathsf{Main}, \mathsf{B}, \mathsf{R}) \\
\mathsf{Transition}(\mathsf{Main}, 0) & \to & \mathsf{X}(\mathsf{A0}, \mathsf{B}, \mathsf{R}) \\
\mathsf{Transition}(\mathsf{Accept}, x) & \to & \mathsf{X}(\mathsf{Accept}, x, \mathsf{S}) \\
& \dots &
\end{array}$$

Representation: $(l_1, l_2, l_3) \Longrightarrow |l_1| \cdot (n+1)^2 + |l_2| \cdot (n+1) + |l_3|$

$$\begin{array}{rcl}
\mathsf{TransitionAt}(i, t_1, t_2, t_3) & \to & \mathsf{Transition}(\mathsf{State}(i, t_1, t_2, t_3), \\
& & \qquad \mathsf{CurTape}(i, t_1, t_2, t_3)) \\
\mathsf{State}(i, [], [], []) & \to & \mathsf{Start} \\
\mathsf{State}(i, t_1, t_2, x;y) & \to & \\
\mathsf{State}(i, t_1, x;y, []) & \to & \\
\mathsf{State}(i, x;y, [], []) & \to &
\end{array}$$

Simulation: Turing Machine running in $< 2 \cdot (n+1)^2$ steps.

$$\begin{array}{rcl}
\mathsf{Transition}(\mathsf{Start}, 0) & \to & \mathsf{X}(\mathsf{Start}, 0, \mathsf{L}) \\
\mathsf{Transition}(\mathsf{Start}, 1) & \to & \mathsf{X}(\mathsf{Start}, 1, \mathsf{L}) \\
\mathsf{Transition}(\mathsf{Start}, \mathsf{B}) & \to & \mathsf{X}(\mathsf{Main}, \mathsf{B}, \mathsf{R}) \\
\mathsf{Transition}(\mathsf{Main}, 0) & \to & \mathsf{X}(\mathsf{A0}, \mathsf{B}, \mathsf{R}) \\
\mathsf{Transition}(\mathsf{Accept}, x) & \to & \mathsf{X}(\mathsf{Accept}, x, \mathsf{S}) \\
& \dots &
\end{array}$$

Representation: $(l_1, l_2, l_3) \Longrightarrow |l_1| \cdot (n+1)^2 + |l_2| \cdot (n+1) + |l_3|$

$$\begin{array}{rcl}
\mathsf{TransitionAt}(i, t_1, t_2, t_3) & \to & \mathsf{Transition}(\mathsf{State}(i, t_1, t_2, t_3), \\
& & \qquad \mathsf{CurTape}(i, t_1, t_2, t_3)) \\
\mathsf{State}(i, [], [], []) & \to & \mathsf{Start} \\
\mathsf{State}(i, t_1, t_2, x;y) & \to & \mathsf{Fst}(\mathsf{TransitionAt}(i, t_1, t_2, y)) \\
\mathsf{State}(i, t_1, x;y, []) & \to & \mathsf{Fst}(\mathsf{TransitionAt}(i, t_1, y, i)) \\
\mathsf{State}(i, x;y, [], []) & \to & \mathsf{Fst}(\mathsf{TransitionAt}(i, y, i, i))
\end{array}$$

Conclusion:

Conclusion: any algorithm in PTIME can be implemented by a cons-free TRS

Conclusion: any algorithm in PTIME can be implemented by a cons-free TRS

Recall: cons-free programs with data order 0 characterise PTIME

Conclusion: any algorithm in PTIME can be implemented by a cons-free TRS

Recall: cons-free programs with data order 0 characterise PTIME

Question:

does (first-order) term rewriting characterise PTIME?

Conclusion: any algorithm in PTIME can be implemented by a cons-free TRS

Recall: cons-free programs with data order 0 characterise PTIME

Question:

does (first-order) term rewriting characterise PTIME?

Answer:

NO

Conclusion: any algorithm in PTIME can be implemented by a deterministic cons-free TRS with call-by-value reduction

Recall: cons-free programs with data order 0 characterise PTIME

Question:

does (first-order) term rewriting characterise PTIME?

Answer:

NO

Conclusion: any algorithm in PTIME can be implemented by a deterministic cons-free TRS with call-by-value reduction

Recall: cons-free programs with data order 0 characterise PTIME

Question:

does (first-order) deterministic term rewriting characterise PTIME?

Conclusion: any algorithm in PTIME can be implemented by a deterministic cons-free TRS with call-by-value reduction

Recall: cons-free programs with data order 0 characterise PTIME

Question:

does (first-order) deterministic term rewriting characterise PTIME?

Answer:

YES

Conclusion: any algorithm in PTIME can be implemented by a deterministic cons-free TRS with call-by-value reduction

Recall: cons-free programs with data order 0 characterise PTIME

Question:

does (first-order) call-by-value term rewriting characterise PTIME?

Answer:

Conclusion: any algorithm in PTIME can be implemented by a deterministic cons-free TRS with call-by-value reduction

Recall: cons-free programs with data order 0 characterise PTIME

Question:

does (first-order) call-by-value term rewriting characterise PTIME?

Answer:

YES

Overview

Call-by-value term rewriting characterises PTIME

Call-by-value term rewriting characterises PTIME

- any algorithm in PTIME can be implemented by a cons-free TRS with call-by-value reduction

Call-by-value term rewriting characterises PTIME

- any algorithm in PTIME can be implemented by a cons-free TRS with call-by-value reduction
- for any cons-free TRS with call-by-value reduction, there is an algorithm in PTIME which, given $s_1, \dots, s_n$, determines whether $\mathsf{Start}(s_1, \dots, s_n) \to^* \mathsf{True}$

Call-by-value term rewriting characterises PTIME

- any algorithm in PTIME can be implemented by a cons-free TRS with call-by-value reduction
- for any cons-free TRS with call-by-value reduction, there is an algorithm in PTIME which, given $s_1, \dots, s_n$, determines whether $\mathsf{Start}(s_1, \dots, s_n) \to^* \mathsf{True}$

Constructing an algorithm

Constructing an algorithm

Rules:

$$\begin{array}{rcl}
\mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\
\mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\
\mathsf{tl}(x;y) & \to & y
\end{array}
\qquad
\begin{array}{rcl}
\mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s,\mathsf{tl}(t)) \\
\mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\
\mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t)
\end{array}$$

Constructing an algorithm

Rules:

$$\begin{array}{rcl}
\mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\
\mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\
\mathsf{tl}(x;y) & \to & y
\end{array}
\qquad
\begin{array}{rcl}
\mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\
\mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\
\mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t)
\end{array}$$

Start term: $\mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[])$

Constructing an algorithm

Rules:

$$\begin{array}{rcl} \mathsf{Subseq}([], t) & \to & \mathsf{True} \\ \mathsf{Subseq}(s, []) & \to & \mathsf{False} \\ \mathsf{tl}(x;y) & \to & y \end{array} \qquad \begin{array}{rcl} \mathsf{Subseq}(s, t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\ \mathsf{Subseq}(0;x, 0;t) & \to & \mathsf{Subseq}(s, t) \\ \mathsf{Subseq}(1;x, 1;t) & \to & \mathsf{Subseq}(s, t) \end{array}$$

Start term: $\mathsf{Subseq}(0;0;1;[], 0;1;0;1;[])$

Constructing an algorithm

Rules:

$$\begin{array}{rcl}
\mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\
\mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\
\mathsf{tl}(x;y) & \to & y
\end{array}
\qquad
\begin{array}{rcl}
\mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\
\mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\
\mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t)
\end{array}$$

Start term: $\mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[])$

Let $\mathcal{B} =$

Constructing an algorithm

Rules:

$$\begin{array}{rcl} \mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\ \mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\ \mathsf{tl}(x;y) & \to & y \end{array} \qquad \begin{array}{rcl} \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s,\mathsf{tl}(t)) \\ \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\ \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t) \end{array}$$

Start term: $\mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[])$

Let $\mathcal{B} =$

$\{0;0;1;[]$

Constructing an algorithm

Rules:

$$\begin{array}{rcl} \mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\ \mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\ \mathsf{tl}(x;y) & \to & y \end{array} \qquad \begin{array}{rcl} \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\ \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\ \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t) \end{array}$$

Start term: $\mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[])$

Let $\mathcal{B} =$

$\{0;0;1;[],\ \ 0;1;[]$

Constructing an algorithm

Rules:

$$\begin{array}{rcl} \mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\ \mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\ \mathsf{tl}(x;y) & \to & y \end{array} \qquad \begin{array}{rcl} \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\ \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\ \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t) \end{array}$$

Start term: $\mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[])$

Let $\mathcal{B} =$

$\{0;0;1;[],\ 0;1;[],\ 1;[]$

Constructing an algorithm

Rules:

$$\begin{array}{rcl} \mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\ \mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\ \mathsf{tl}(x;y) & \to & y \end{array} \qquad \begin{array}{rcl} \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s,\mathsf{tl}(t)) \\ \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\ \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t) \end{array}$$

Start term: $\mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[])$

Let $\mathcal{B} =$

$\{0;0;1;[],\ 0;1;[],\ 1;[],\ []$

Constructing an algorithm

Rules:

$$\begin{array}{rcl} \mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\ \mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\ \mathsf{tl}(x;y) & \to & y \end{array} \qquad \begin{array}{rcl} \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s,\mathsf{tl}(t)) \\ \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\ \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t) \end{array}$$

Start term: $\mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[])$

Let $\mathcal{B} =$
$\{0;0;1;[],\ 0;1;[],\ 1;[],\ [],\ 0;1;0;1;[]$

Constructing an algorithm

Rules:

$$\begin{array}{rcl} \mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\ \mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\ \mathsf{tl}(x;y) & \to & y \end{array} \quad \begin{array}{rcl} \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s,\mathsf{tl}(t)) \\ \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\ \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t) \end{array}$$

Start term: $\mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[])$

Let $\mathcal{B} =$

$\{0;0;1;[],\ 0;1;[],\ 1;[],\ [],\ 0;1;0;1;[],\ 1;0;1;[]$

Constructing an algorithm

Rules:

$$\begin{array}{rcl} \mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\ \mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\ \mathsf{tl}(x;y) & \to & y \end{array} \qquad \begin{array}{rcl} \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s,\mathsf{tl}(t)) \\ \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\ \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t) \end{array}$$

Start term: $\mathsf{Subseq}(0;0;1;[],\ 0;1;\color{red}{0;1;[]})$

Let $\mathcal{B} =$

$\{0;0;1;[],\ \color{red}{0;1;[]},\ 1;[],\ [],\ 0;1;0;1;[],\ 1;0;1;[]$

Constructing an algorithm

Rules:

$$\begin{array}{rcl} \mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\ \mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\ \mathsf{tl}(x;y) & \to & y \end{array} \qquad \begin{array}{rcl} \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s,\mathsf{tl}(t)) \\ \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\ \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t) \end{array}$$

Start term: $\mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[])$

Let $\mathcal{B} =$

$\{0;0;1;[],\ 0;1;[],\ 1;[],\ [],\ 0;1;0;1;[],\ 1;0;1;[]$

Constructing an algorithm

Rules:

$$\begin{array}{rcl}
\mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\
\mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\
\mathsf{tl}(x;y) & \to & y
\end{array}
\qquad
\begin{array}{rcl}
\mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\
\mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\
\mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t)
\end{array}$$

Start term: $\mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[])$

Let $\mathcal{B} =$

$\{0;0;1;[],\ 0;1;[],\ 1;[],\ [],\ 0;1;0;1;[],\ 1;0;1;[]$

Constructing an algorithm

Rules:

$$\begin{array}{rcl} \mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\ \mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\ \mathsf{tl}(x;y) & \to & y \end{array} \qquad \begin{array}{rcl} \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\ \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\ \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t) \end{array}$$

Start term: $\mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[])$

Let $\mathcal{B} =$

$\{0;0;1;[],\ 0;1;[],\ 1;[],\ [],\ 0;1;0;1;[],\ 1;0;1;[],\ \mathsf{True}$

Constructing an algorithm

Rules:

$$\begin{array}{rcl rcl} \mathsf{Subseq}([],\ t) & \to & \mathsf{True} & \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\ \mathsf{Subseq}(s,\ []) & \to & \mathsf{False} & \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\ \mathsf{tl}(x;y) & \to & y & \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t) \end{array}$$

Start term: $\mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[])$

Let $\mathcal{B} =$
$\{0;0;1;[],\ 0;1;[],\ 1;[],\ [],\ 0;1;0;1;[],\ 1;0;1;[],\ \mathsf{True},\ \mathsf{False}\}$

Constructing an algorithm

Rules:

$$\begin{array}{rcl} \mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\ \mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\ \mathsf{tl}(x;y) & \to & y \end{array} \qquad \begin{array}{rcl} \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\ \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\ \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t) \end{array}$$

Start term: $\mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[])$

Let $\mathcal{B} =$
$\{0;0;1;[],\ 0;1;[],\ 1;[],\ [],\ 0;1;0;1;[],\ 1;0;1;[],\ \mathsf{True},\ \mathsf{False}\}$

Make a list:

Constructing an algorithm

Rules:

$$\begin{array}{rcl} \mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\ \mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\ \mathsf{tl}(x;y) & \to & y \end{array} \qquad \begin{array}{rcl} \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s,\mathsf{tl}(t)) \\ \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\ \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t) \end{array}$$

Start term: $\mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[])$

Let $\mathcal{B} =$
$\{0;0;1;[],\ 0;1;[],\ 1;[],\ [],\ 0;1;0;1;[],\ 1;0;1;[],\ \mathsf{True},\ \mathsf{False}\}$

Make a list:

$$\mathsf{Subseq}(0;0;1;[],\ 0;0;1;[]) \quad \to^* \quad \{\}$$

Constructing an algorithm

Rules:

$$\begin{array}{rcl} \mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\ \mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\ \mathsf{tl}(x;y) & \to & y \end{array} \qquad \begin{array}{rcl} \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\ \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\ \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t) \end{array}$$

Start term: $\mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[])$

Let $\mathcal{B} =$
$\{0;0;1;[],\ 0;1;[],\ 1;[],\ [],\ 0;1;0;1;[],\ 1;0;1;[],\ \mathsf{True},\ \mathsf{False}\}$

Make a list:

$$\begin{array}{rcl} \mathsf{Subseq}(0;0;1;[],\ 0;0;1;[]) & \to^* & \{\} \\ \mathsf{Subseq}(0;0;1;[],\ 0;1;[]) & \to^* & \{\} \end{array}$$

Constructing an algorithm

Rules:

$$\begin{array}{rcl} \mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\ \mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\ \mathsf{tl}(x;y) & \to & y \end{array} \qquad \begin{array}{rcl} \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s,\mathsf{tl}(t)) \\ \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\ \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t) \end{array}$$

Start term: $\mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[])$

Let $\mathcal{B} =$

$\{0;0;1;[],\ 0;1;[],\ 1;[],\ [],\ 0;1;0;1;[],\ 1;0;1;[],\ \mathsf{True},\ \mathsf{False}\}$

Make a list:

$$\begin{array}{rcl} \mathsf{Subseq}(0;0;1;[],\ 0;0;1;[]) & \to^* & \{\} \\ \mathsf{Subseq}(0;0;1;[],\ 0;1;[]) & \to^* & \{\} \\ & \dots & \end{array}$$

Constructing an algorithm

Rules:

$$\begin{array}{rcl} \mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\ \mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\ \mathsf{tl}(x;y) & \to & y \end{array} \qquad \begin{array}{rcl} \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\ \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\ \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t) \end{array}$$

Start term: $\mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[])$
Let $\mathcal{B} =$
$\{0;0;1;[],\ 0;1;[],\ 1;[],\ [],\ 0;1;0;1;[],\ 1;0;1;[],\ \mathsf{True},\ \mathsf{False}\}$
Make a list:

$$\begin{array}{rcl} \mathsf{Subseq}(0;0;1;[],\ 0;0;1;[]) & \to^* & \{\} \\ \mathsf{Subseq}(0;0;1;[],\ 0;1;[]) & \to^* & \{\} \\ & \dots & \\ \mathsf{Subseq}(0;1;[],\ 0;0;1;[]) & \to^* & \{\} \end{array}$$

Constructing an algorithm

Rules:

$$\begin{array}{rcl} \mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\ \mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\ \mathsf{tl}(x;y) & \to & y \end{array} \quad \begin{array}{rcl} \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\ \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\ \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t) \end{array}$$

Start term: $\mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[])$

Let $\mathcal{B} =$
$\{0;0;1;[],\ 0;1;[],\ 1;[],\ [],\ 0;1;0;1;[],\ 1;0;1;[],\ \mathsf{True},\ \mathsf{False}\}$

Make a list:

$$\begin{array}{rcl} \mathsf{Subseq}(0;0;1;[],\ 0;0;1;[]) & \to^* & \{\} \\ \mathsf{Subseq}(0;0;1;[],\ 0;1;[]) & \to^* & \{\} \\ & \dots & \\ \mathsf{Subseq}(0;1;[],\ 0;0;1;[]) & \to^* & \{\} \\ & \dots & \end{array}$$

$$\begin{array}{rcl} \mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\ \mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\ \mathsf{tl}(x;y) & \to & y \end{array} \qquad \begin{array}{rcl} \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\ \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\ \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t) \end{array}$$

Statements:

$$\begin{array}{rl} \mathsf{tl}(0;1;0;1;[]) & \to^* \\ \mathsf{tl}(1;0;1;[]) & \to^* \\ \dots & \\ \mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* \\ \mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* \\ \mathsf{Subseq}(0;0;1;[],\ []) & \to^* \\ \mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* \\ \mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* \\ \mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* \\ \mathsf{Subseq}(1;[],\ 1;[]) & \to^* \\ \mathsf{Subseq}([],\ []) & \to^* \\ \dots & \end{array}$$

$$\begin{array}{rcl} \mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\ \mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\ \mathsf{tl}(x;y) & \to & y \end{array} \qquad \begin{array}{rcl} \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s,\mathsf{tl}(t)) \\ \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\ \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t) \end{array}$$

Statements:

$$\begin{array}{rl} \mathsf{tl}(0;1;0;1;[]) & \to^* \\ \mathsf{tl}(1;0;1;[]) & \to^* \\ \dots & \\ \mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* \\ \mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* \\ \mathsf{Subseq}(0;0;1;[],\ []) & \to^* \\ \mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* \\ \mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* \\ \mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* \\ \mathsf{Subseq}(1;[],\ 1;[]) & \to^* \\ \mathsf{Subseq}([],\ []) & \to^* \\ \dots & \end{array}$$

$$\begin{array}{rcl} \mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\ \mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\ \mathsf{tl}(x;y) & \to & y \end{array} \qquad \begin{array}{rcl} \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\ \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\ \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t) \end{array}$$

Statements:

$$\begin{array}{rl} \mathsf{tl}(0;1;0;1;[]) & \to^* \\ \mathsf{tl}(1;0;1;[]) & \to^* \\ \dots & \\ \mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* \\ \mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* \\ \mathsf{Subseq}(0;0;1;[],\ []) & \to^* \\ \mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* \\ \mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* \\ \mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* \\ \mathsf{Subseq}(1;[],\ 1;[]) & \to^* \\ \mathsf{Subseq}([],\ []) & \to^* \\ \dots & \end{array}$$

$$\begin{array}{rcl} \mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\ \mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\ \mathsf{tl}(x;y) & \to & y \end{array} \qquad \begin{array}{rcl} \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s,\mathsf{tl}(t)) \\ \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\ \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t) \end{array}$$

Statements:

$$\begin{array}{rcl} \mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\ \mathsf{tl}(1;0;1;[]) & \to^* & \\ & \dots & \\ \mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \\ \mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \\ \mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \\ \mathsf{Subseq}([],\ []) & \to^* & \\ & \dots & \end{array}$$

$$
\begin{array}{rcl@{\qquad}rcl}
\mathsf{Subseq}([],\ t) & \to & \mathsf{True} & \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s,\mathsf{tl}(t)) \\
\mathsf{Subseq}(s,\ []) & \to & \mathsf{False} & \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\
\mathsf{tl}(x;y) & \to & y & \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t)
\end{array}
$$

Statements:

$$
\begin{array}{rcl}
\mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\
\mathsf{tl}(1;0;1;[]) & \to^* & \\
 & \dots & \\
\mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \\
\mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \\
\mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \\
\mathsf{Subseq}([],\ []) & \to^* & \\
 & \dots &
\end{array}
$$

$$\begin{array}{rclrcl}
\mathsf{Subseq}([],\ t) & \to & \mathsf{True} & \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\
\mathsf{Subseq}(s,\ []) & \to & \mathsf{False} & \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\
\mathsf{tl}(x;y) & \to & y & \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t)
\end{array}$$

Statements:

$$\begin{array}{rcl}
\mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\
\mathsf{tl}(1;0;1;[]) & \to^* & \\
 & \dots & \\
\mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \\
\mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \\
\mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \\
\mathsf{Subseq}([],\ []) & \to^* & \\
 & \dots &
\end{array}$$

$$
\begin{array}{rcl}
\mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\
\mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\
\color{red}{\mathsf{tl}(x;y)} & \color{red}{\to} & \color{red}{y}
\end{array}
\qquad
\begin{array}{rcl}
\mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s,\mathsf{tl}(t)) \\
\mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\
\mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t)
\end{array}
$$

Statements:

$$
\begin{array}{rcl}
\mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\
\color{blue}{\mathsf{tl}(1;0;1;[])} & \color{blue}{\to^*} & \color{blue}{0;1;[]} \\
 & \ldots & \\
\mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \\
\mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \\
\mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \\
\mathsf{Subseq}([],\ []) & \to^* & \\
 & \ldots &
\end{array}
$$

$$
\begin{array}{rcl}
\mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\
\mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\
\mathsf{tl}(x;y) & \to & y
\end{array}
\qquad
\begin{array}{rcl}
\mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s,\mathsf{tl}(t)) \\
\mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\
\mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t)
\end{array}
$$

Statements:

$$
\begin{array}{rcl}
\mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\
\mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[]
\end{array}
$$

...

$$
\begin{array}{rl}
\mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* \\
\mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* \\
\mathsf{Subseq}(0;0;1;[],\ []) & \to^* \\
\mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* \\
\mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* \\
\mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* \\
\mathsf{Subseq}(1;[],\ 1;[]) & \to^* \\
\mathsf{Subseq}([],\ []) & \to^*
\end{array}
$$

...

$$
\begin{array}{rcl}
\mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\
\mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\
\mathsf{tl}(x;y) & \to & y
\end{array}
\qquad
\begin{array}{rcl}
\mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\
\mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\
\mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t)
\end{array}
$$

Statements:

$$
\begin{array}{rcl}
\mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\
\mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\
& \dots & \\
\mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \\
\mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \\
\mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \\
\mathsf{Subseq}([],\ []) & \to^* & \\
& \dots &
\end{array}
$$

$$\begin{array}{rcl} \mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\ \mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\ \mathsf{tl}(x;y) & \to & y \end{array} \qquad \begin{array}{rcl} \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s,\mathsf{tl}(t)) \\ \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\ \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t) \end{array}$$

Statements:

$$\begin{array}{rcl} \mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\ \mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\ & \dots & \\ \mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \\ \mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \\ \mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \\ \mathsf{Subseq}([],\ []) & \to^* & \\ & \dots & \end{array}$$

$$\begin{array}{rcl} \mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\ \mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\ \mathsf{tl}(x;y) & \to & y \end{array} \qquad \begin{array}{rcl} \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s,\mathsf{tl}(t)) \\ \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\ \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t) \end{array}$$

Statements:

$$\begin{array}{rcl} \mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\ \mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\ & \dots & \\ \mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \\ \mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \\ \mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \\ \mathsf{Subseq}([],\ []) & \to^* & \\ & \dots & \end{array}$$

$$\begin{array}{rcl@{\qquad}rcl}
\mathsf{Subseq}([],\ t) & \to & \mathsf{True} & \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\
\mathsf{Subseq}(s,\ []) & \to & \mathsf{False} & \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\
\mathsf{tl}(x;y) & \to & y & \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t)
\end{array}$$

Statements:

$$\begin{array}{rcl}
\mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\
\mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\
& \dots & \\
\mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \\
\mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \\
\mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \\
\mathsf{Subseq}([],\ []) & \to^* & \\
& \dots &
\end{array}$$

$$
\begin{array}{rclrcl}
\mathsf{Subseq}([],\ t) & \to & \mathsf{True} & \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\
\mathsf{Subseq}(s,\ []) & \to & \mathsf{False} & \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\
\mathsf{tl}(x;y) & \to & y & \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t)
\end{array}
$$

Statements:

$$
\begin{array}{rcl}
\mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\
\mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\
& \dots & \\
\mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \\
\mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \\
\mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \\
\mathsf{Subseq}([],\ []) & \to^* & \\
& \dots &
\end{array}
$$

$$
\begin{array}{rcl rcl}
\mathsf{Subseq}([],\ t) & \to & \mathsf{True} & \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\
\mathsf{Subseq}(s,\ []) & \to & \mathsf{False} & \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\
\mathsf{tl}(x;y) & \to & y & \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t)
\end{array}
$$

Statements:

$$
\begin{array}{rcl}
\mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\
\mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\
 & \dots & \\
\mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \\
\mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \\
\mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \\
\mathsf{Subseq}([],\ []) & \to^* & \\
 & \dots &
\end{array}
$$

$$
\begin{array}{rcl@{\qquad}rcl}
\mathsf{Subseq}([],\ t) & \to & \mathsf{True} & \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\
\mathsf{Subseq}(s,\ []) & \to & \mathsf{False} & \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\
\mathsf{tl}(x;y) & \to & y & \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t)
\end{array}
$$

Statements:

$$
\begin{array}{rcl}
\mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\
\mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\
& \cdots & \\
\mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \\
\mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \\
\mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \\
\mathsf{Subseq}([],\ []) & \to^* & \\
& \cdots &
\end{array}
$$

$$
\begin{array}{rcl}
\mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\
\mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\
\mathsf{tl}(x;y) & \to & y
\end{array}
\qquad
\begin{array}{rcl}
\mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\
\mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\
\mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t)
\end{array}
$$

Statements:

$$
\begin{array}{rcl}
\mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\
\mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\
& \dots & \\
\mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \\
\mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \\
\mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \\
\mathsf{Subseq}([],\ []) & \to^* & \\
& \dots &
\end{array}
$$

$$\begin{array}{rclrcl}
\mathsf{Subseq}([],\ t) & \to & \mathsf{True} & \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\
\mathsf{Subseq}(s,\ []) & \to & \mathsf{False} & \mathsf{Subseq}(0; x,\ 0; t) & \to & \mathsf{Subseq}(s,\ t) \\
\mathsf{tl}(x; y) & \to & y & \mathsf{Subseq}(1; x,\ 1; t) & \to & \mathsf{Subseq}(s,\ t)
\end{array}$$

Statements:

$$\begin{array}{rcl}
\mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\
\mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\
& \dots & \\
\mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \\
\mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \\
\mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \\
\mathsf{Subseq}([],\ []) & \to^* & \\
& \dots &
\end{array}$$

$$
\begin{array}{rcl rcl}
\mathsf{Subseq}([],\ t) & \to & \mathsf{True} & \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\
\mathsf{Subseq}(s,\ []) & \to & \mathsf{False} & \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\
\mathsf{tl}(x;y) & \to & y & \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t)
\end{array}
$$

Statements:

$$
\begin{array}{rcl}
\mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\
\mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\
& \dots & \\
\mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \\
\mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \\
\mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \\
\mathsf{Subseq}([],\ []) & \to^* & \\
& \dots &
\end{array}
$$

$$
\begin{array}{rcl}
\mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\
\mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\
\mathsf{tl}(x;y) & \to & y
\end{array}
\qquad
\begin{array}{rcl}
\mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s,\mathsf{tl}(t)) \\
\mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\
\color{red}{\mathsf{Subseq}(1;x,\ 1;t)} & \color{red}{\to} & \color{red}{\mathsf{Subseq}(s,\ t)}
\end{array}
$$

Statements:

$$
\begin{array}{rcl}
\mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\
\mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\
& \dots & \\
\color{blue}{\mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[])} & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \\
\mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \\
\mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \\
\mathsf{Subseq}([],\ []) & \to^* & \\
& \dots &
\end{array}
$$

$$\begin{array}{rcl} \mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\ \mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\ \mathsf{tl}(x;y) & \to & y \end{array} \qquad \begin{array}{rcl} \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s,\mathsf{tl}(t)) \\ \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\ \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t) \end{array}$$

Statements:

$$\begin{array}{rcl} \mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\ \mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\ & \dots & \\ \mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \\ \mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \\ \mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \\ \mathsf{Subseq}([],\ []) & \to^* & \\ & \dots & \end{array}$$

$$\begin{array}{rcl} \mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\ \mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\ \mathsf{tl}(x;y) & \to & y \end{array} \qquad \begin{array}{rcl} \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\ \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\ \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t) \end{array}$$

Statements:

$$\begin{array}{rcl} \mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\ \mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\ & \dots & \\ \mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \\ \mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \\ \mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \\ \mathsf{Subseq}([],\ []) & \to^* & \\ & \dots & \end{array}$$

$$\begin{array}{rcl} \mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\ \mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\ \mathsf{tl}(x;y) & \to & y \end{array} \qquad \begin{array}{rcl} \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\ \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\ \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t) \end{array}$$

Statements:

$$\begin{array}{rcl} \mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\ \mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\ & \dots & \\ \mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \\ \mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \\ \mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \\ \mathsf{Subseq}([],\ []) & \to^* & \\ & \dots & \end{array}$$

$$\begin{array}{rcl} \mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\ \mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\ \mathsf{tl}(x;y) & \to & y \end{array} \qquad \begin{array}{rcl} \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s,\mathsf{tl}(t)) \\ \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\ \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t) \end{array}$$

Statements:

$$\begin{array}{rcl} \mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\ \mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\ & \dots & \\ \mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \mathsf{False} \\ \mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \\ \mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \\ \mathsf{Subseq}([],\ []) & \to^* & \\ & \dots & \end{array}$$

$$\begin{array}{rcl} \mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\ \mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\ \mathsf{tl}(x;y) & \to & y \end{array} \qquad \begin{array}{rcl} \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s,\mathsf{tl}(t)) \\ \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\ \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t) \end{array}$$

Statements:

$$\begin{array}{rcl} \mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\ \mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\ & \dots & \\ \mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \mathsf{False} \\ \mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \\ \mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \\ \mathsf{Subseq}([],\ []) & \to^* & \\ & \dots & \end{array}$$

$$
\begin{array}{rcl@{\qquad}rcl}
\mathsf{Subseq}([],\ t) & \to & \mathsf{True} & \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\
\mathsf{Subseq}(s,\ []) & \to & \mathsf{False} & \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\
\mathsf{tl}(x;y) & \to & y & \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t)
\end{array}
$$

Statements:

$$
\begin{array}{rcl}
\mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\
\mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\
 & \dots & \\
\mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \mathsf{False} \\
\mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \\
\mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \\
\mathsf{Subseq}([],\ []) & \to^* & \\
 & \dots &
\end{array}
$$

$$
\begin{array}{rclrcl}
\mathsf{Subseq}([],\ t) & \to & \mathsf{True} & \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\
\mathsf{Subseq}(s,\ []) & \to & \mathsf{False} & \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\
\mathsf{tl}(x;y) & \to & y & \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t)
\end{array}
$$

Statements:

$$
\begin{array}{rcl}
\mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\
\mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\
& \dots & \\
\mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \mathsf{False} \\
\color{blue}{\mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[])} & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \\
\mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \\
\mathsf{Subseq}([],\ []) & \to^* & \\
& \dots &
\end{array}
$$

$$
\begin{array}{rcl}
\mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\
\mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\
\mathsf{tl}(x;y) & \to & y
\end{array}
\qquad
\begin{array}{rcl}
\mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s,\mathsf{tl}(t)) \\
\mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\
\mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t)
\end{array}
$$

Statements:

$$
\begin{array}{rcl}
\mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\
\mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\
 & \dots & \\
\mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \mathsf{False} \\
\mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \\
\mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \\
\mathsf{Subseq}([],\ []) & \to^* & \\
 & \dots &
\end{array}
$$

$$\begin{array}{rcl}
\mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\
\mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\
\mathsf{tl}(x;y) & \to & y
\end{array}
\qquad
\begin{array}{rcl}
\mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s,\mathsf{tl}(t)) \\
\mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\
\mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t)
\end{array}$$

Statements:

$$\begin{array}{rcl}
\mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\
\mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\
 & \dots & \\
\mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \mathsf{False} \\
\mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \\
\mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \\
\mathsf{Subseq}([],\ []) & \to^* & \\
 & \dots &
\end{array}$$

$$
\begin{array}{rcl@{\qquad}rcl}
\mathsf{Subseq}([],\ t) & \to & \mathsf{True} & \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\
\mathsf{Subseq}(s,\ []) & \to & \mathsf{False} & \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\
\mathsf{tl}(x;y) & \to & y & \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t)
\end{array}
$$

Statements:

$$
\begin{array}{rcl}
\mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\
\mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\
& \dots & \\
\mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \mathsf{False} \\
\mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \\
\mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \\
\mathsf{Subseq}([],\ []) & \to^* & \\
& \dots &
\end{array}
$$

$$\begin{array}{rclrcl}
\mathsf{Subseq}([],\ t) & \to & \mathsf{True} & \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\
\mathsf{Subseq}(s,\ []) & \to & \mathsf{False} & \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\
\mathsf{tl}(x;y) & \to & y & \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t)
\end{array}$$

Statements:

$$\begin{array}{rcl}
\mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\
\mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\
& \dots & \\
\mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \mathsf{False} \\
\mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \\
\mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \\
\mathsf{Subseq}([],\ []) & \to^* & \\
& \dots &
\end{array}$$

$$
\begin{array}{rcl}
\color{red}{\mathsf{Subseq}([],\ t)} & \color{red}{\to} & \color{red}{\mathsf{True}} \\
\color{red}{\mathsf{Subseq}(s,\ [])} & \color{red}{\to} & \color{red}{\mathsf{False}} \\
\mathsf{tl}(x;y) & \to & y
\end{array}
\qquad
\begin{array}{rcl}
\mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\
\mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\
\mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t)
\end{array}
$$

Statements:

$$
\begin{array}{rcl}
\mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\
\mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\
 & \dots & \\
\mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \mathsf{False} \\
\mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \\
\mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \\
\color{blue}{\mathsf{Subseq}([],\ [])} & \to^* & \\
 & \dots &
\end{array}
$$

$$
\begin{array}{rclrcl}
\mathsf{Subseq}([],\ t) & \to & \mathsf{True} & \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\
\mathsf{Subseq}(s,\ []) & \to & \mathsf{False} & \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\
\mathsf{tl}(x;y) & \to & y & \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t)
\end{array}
$$

Statements:

$$
\begin{array}{rcl}
\mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\
\mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\
& \dots & \\
\mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \mathsf{False} \\
\mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \\
\mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \\
\mathsf{Subseq}([],\ []) & \to^* & \mathsf{True}, \mathsf{False} \\
& \dots &
\end{array}
$$

$$
\begin{array}{rcl}
\mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\
\mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\
\mathsf{tl}(x;y) & \to & y
\end{array}
\qquad
\begin{array}{rcl}
\mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\
\mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\
\mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t)
\end{array}
$$

Statements:

$$
\begin{array}{rcl}
\mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\
\mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\
 & \dots & \\
\mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \mathsf{False} \\
\mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \\
\mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \\
\mathsf{Subseq}([],\ []) & \to^* & \mathsf{True}, \mathsf{False}
\end{array}
$$

...

$$
\begin{array}{rcl}
\mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\
\mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\
\mathsf{tl}(x;y) & \to & y
\end{array}
\qquad
\begin{array}{rcl}
\mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\
\mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\
\mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t)
\end{array}
$$

Statements:

$$
\begin{array}{rcl}
\color{blue}{\mathsf{tl}(0;1;0;1;[])} & \to^* & 1;0;1;[] \\
\mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\
& \ldots & \\
\mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \mathsf{False} \\
\mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \\
\mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \\
\mathsf{Subseq}([],\ []) & \to^* & \mathsf{True}, \mathsf{False} \\
& \ldots &
\end{array}
$$

$$
\begin{array}{rcl@{\qquad}rcl}
\mathsf{Subseq}([],\ t) & \to & \mathsf{True} & \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\
\mathsf{Subseq}(s,\ []) & \to & \mathsf{False} & \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\
\mathsf{tl}(x;y) & \to & y & \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t)
\end{array}
$$

Statements:

$$
\begin{array}{rcl}
\mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\
\mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\
& \dots & \\
\mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \mathsf{False} \\
\mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \\
\mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \\
\mathsf{Subseq}([],\ []) & \to^* & \mathsf{True}, \mathsf{False} \\
& \dots &
\end{array}
$$

$$
\begin{array}{rcl}
\mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\
\mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\
\mathsf{tl}(x;y) & \to & y
\end{array}
\qquad
\begin{array}{rcl}
\mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\
\mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\
\mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t)
\end{array}
$$

Statements:

$$
\begin{array}{rcl}
\mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\
\mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\
 & \dots & \\
\mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \mathsf{False} \\
\mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \\
\mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \\
\mathsf{Subseq}([],\ []) & \to^* & \mathsf{True}, \mathsf{False}
\end{array}
$$

...

$$\begin{array}{rcl} \mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\ \mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\ \mathsf{tl}(x;y) & \to & y \end{array} \qquad \begin{array}{rcl} \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\ \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\ \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t) \end{array}$$

Statements:

$$\begin{array}{rcl}
\mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\
\mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\
& \dots & \\
\color{blue}{\mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[])} & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \mathsf{False} \\
\mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \\
\mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \\
\mathsf{Subseq}([],\ []) & \to^* & \mathsf{True}, \mathsf{False} \\
& \dots &
\end{array}$$

$$
\begin{array}{rcl}
\mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\
\mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\
\mathsf{tl}(x;y) & \to & y
\end{array}
\qquad
\begin{array}{rcl}
\mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s,\mathsf{tl}(t)) \\
\mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\
\mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t)
\end{array}
$$

Statements:

$$
\begin{array}{rcl}
\mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\
\mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\
 & \dots & \\
\mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \mathsf{False} \\
\mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \\
\mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \\
\mathsf{Subseq}([],\ []) & \to^* & \mathsf{True}, \mathsf{False} \\
 & \dots &
\end{array}
$$

$$
\begin{array}{rcl}
\mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\
\mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\
\mathsf{tl}(x;y) & \to & y
\end{array}
\qquad
\begin{array}{rcl}
\mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s,\mathsf{tl}(t)) \\
\mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\
\mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t)
\end{array}
$$

Statements:

$$
\begin{array}{rcl}
\mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\
\mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\
& \dots & \\
\mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \mathsf{False} \\
\mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \\
\mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \\
\mathsf{Subseq}([],\ []) & \to^* & \mathsf{True}, \mathsf{False} \\
& \dots &
\end{array}
$$

$$\begin{array}{rcl} \mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\ \mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\ \mathsf{tl}(x;y) & \to & y \end{array} \qquad \begin{array}{rcl} \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s,\mathsf{tl}(t)) \\ \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\ \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t) \end{array}$$

Statements:

$$\begin{array}{rcl} \mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\ \mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\ & \dots & \\ \mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \mathsf{False} \\ \mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \\ \mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \\ \mathsf{Subseq}([],\ []) & \to^* & \mathsf{True},\mathsf{False} \\ & \dots & \end{array}$$

$$\begin{array}{rcl} \mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\ \mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\ \mathsf{tl}(x;y) & \to & y \end{array} \qquad \begin{array}{rcl} \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s,\mathsf{tl}(t)) \\ \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\ \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t) \end{array}$$

Statements:

$$\begin{array}{rcl} \mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\ \mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\ & \dots & \\ \mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \mathsf{False} \\ \mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \\ \mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \\ \mathsf{Subseq}([],\ []) & \to^* & \mathsf{True}, \mathsf{False} \\ & \dots & \end{array}$$

$$\begin{array}{rcl} \mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\ \mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\ \mathsf{tl}(x;y) & \to & y \end{array} \qquad \begin{array}{rcl} \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\ \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\ \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t) \end{array}$$

Statements:

$$\begin{array}{rcl} \mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\ \mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\ & \dots & \\ \mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \mathsf{False} \\ \mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \\ \mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \\ \mathsf{Subseq}([],\ []) & \to^* & \mathsf{True}, \mathsf{False} \\ & \dots & \end{array}$$

$$\begin{array}{rcl} \mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\ \mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\ \mathsf{tl}(x;y) & \to & y \end{array} \qquad \begin{array}{rcl} \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\ \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\ \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t) \end{array}$$

Statements:

$$\begin{array}{rcl} \mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\ \mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\ & \dots & \\ \mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \mathsf{False} \\ \mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \\ \mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \\ \mathsf{Subseq}([],\ []) & \to^* & \mathsf{True}, \mathsf{False} \\ & \dots & \end{array}$$

$$\begin{array}{rcl} \mathsf{Subseq}([\,],\ t) & \to & \mathsf{True} \\ \mathsf{Subseq}(s,\ [\,]) & \to & \mathsf{False} \\ \mathsf{tl}(x;y) & \to & y \end{array} \qquad \begin{array}{rcl} \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s,\mathsf{tl}(t)) \\ \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\ \color{red}{\mathsf{Subseq}(1;x,\ 1;t)} & \color{red}{\to} & \color{red}{\mathsf{Subseq}(s,\ t)} \end{array}$$

Statements:

$$\begin{array}{rcl} \mathsf{tl}(0;1;0;1;[\,]) & \to^* & 1;0;1;[\,] \\ \mathsf{tl}(1;0;1;[\,]) & \to^* & 0;1;[\,] \\ & \dots & \\ \mathsf{Subseq}(0;0;1;[\,],\ 0;1;0;1;[\,]) & \to^* & \\ \mathsf{Subseq}(0;0;1;[\,],\ 1;0;1;[\,]) & \to^* & \\ \mathsf{Subseq}(0;0;1;[\,],\ [\,]) & \to^* & \mathsf{False} \\ \mathsf{Subseq}(1;0;1;[\,],\ 0;1;0;1;[\,]) & \to^* & \\ \mathsf{Subseq}(0;1;[\,],\ 1;0;1;[\,]) & \to^* & \\ \mathsf{Subseq}(0;1;[\,],\ 0;1;[\,]) & \to^* & \\ \color{blue}{\mathsf{Subseq}(1;[\,],\ 1;[\,])} & \to^* & \\ \mathsf{Subseq}([\,],\ [\,]) & \to^* & \mathsf{True},\mathsf{False} \\ & \dots & \end{array}$$

$$\begin{array}{rcl} \mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\ \mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\ \mathsf{tl}(x;y) & \to & y \end{array} \qquad \begin{array}{rcl} \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\ \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\ \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t) \end{array}$$

Statements:

$$\begin{array}{rcl} \mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\ \mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\ & \dots & \\ \mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \mathsf{False} \\ \mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \\ \mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \\ \mathsf{Subseq}([],\ []) & \to^* & \mathsf{True}, \mathsf{False} \\ & \dots & \end{array}$$

$$
\begin{array}{rcl@{\qquad}rcl}
\mathsf{Subseq}([],\ t) & \to & \mathsf{True} & \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\
\mathsf{Subseq}(s,\ []) & \to & \mathsf{False} & \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\
\mathsf{tl}(x;y) & \to & y & \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t)
\end{array}
$$

Statements:

$$
\begin{array}{rcl}
\mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\
\mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\
& \dots & \\
\mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \mathsf{False} \\
\mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \\
\mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}([],\ []) & \to^* & \mathsf{True}, \mathsf{False} \\
& \dots &
\end{array}
$$

$$
\begin{array}{rcl}
\mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\
\mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\
\mathsf{tl}(x;y) & \to & y
\end{array}
\qquad
\begin{array}{rcl}
\mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s,\mathsf{tl}(t)) \\
\mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\
\mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t)
\end{array}
$$

Statements:

$$
\begin{array}{rcl}
\mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\
\mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\
 & \dots & \\
\mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \mathsf{False} \\
\mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \\
\mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}([],\ []) & \to^* & \mathsf{True}, \mathsf{False} \\
 & \dots &
\end{array}
$$

$$
\begin{array}{rcl}
\mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\
\mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\
\mathsf{tl}(x;y) & \to & y
\end{array}
\qquad
\begin{array}{rcl}
\mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\
\mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\
\mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t)
\end{array}
$$

Statements:

$$
\begin{array}{rcl}
\mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\
\mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\
& \cdots & \\
\mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \mathsf{False} \\
\mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \\
\mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}([],\ []) & \to^* & \mathsf{True}, \mathsf{False} \\
& \cdots &
\end{array}
$$

$$\begin{array}{rclrcl}
\mathsf{Subseq}([],\ t) & \to & \mathsf{True} & \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\
\mathsf{Subseq}(s,\ []) & \to & \mathsf{False} & \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\
\mathsf{tl}(x;y) & \to & y & \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t)
\end{array}$$

Statements:

$$\begin{array}{rcl}
\mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\
\mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\
& \dots & \\
\mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \mathsf{False} \\
\mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \\
\mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}([],\ []) & \to^* & \mathsf{True}, \mathsf{False} \\
& \dots &
\end{array}$$

$$
\begin{array}{rclrcl}
\mathsf{Subseq}([],\ t) & \to & \mathsf{True} & \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\
\mathsf{Subseq}(s,\ []) & \to & \mathsf{False} & \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\
\mathsf{tl}(x;y) & \to & y & \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t)
\end{array}
$$

Statements:

$$
\begin{array}{rcl}
\mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\
\mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\
& \dots & \\
\mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \mathsf{False} \\
\mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \\
\mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}([],\ []) & \to^* & \mathsf{True}, \mathsf{False}
\end{array}
$$

...

$$\begin{array}{rcl}
\mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\
\mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\
\mathsf{tl}(x;y) & \to & y
\end{array}
\qquad
\begin{array}{rcl}
\mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\
\mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\
\mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t)
\end{array}$$

Statements:

$$\begin{array}{rcl}
\mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\
\mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\
 & \ldots & \\
\mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \mathsf{False} \\
\mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \\
\mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}([],\ []) & \to^* & \mathsf{True}, \mathsf{False} \\
 & \ldots &
\end{array}$$

$$\begin{array}{rcl}\mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\ \mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\ \mathsf{tl}(x;y) & \to & y\end{array} \qquad \begin{array}{rcl}\mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s,\mathsf{tl}(t)) \\ \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\ \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t)\end{array}$$

Statements:

$$\begin{array}{rcl}\mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\ \mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\ & \dots & \\ \mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \mathsf{False} \\ \mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \\ \mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \mathsf{True},\mathsf{False} \\ \mathsf{Subseq}([],\ []) & \to^* & \mathsf{True},\mathsf{False} \\ & \dots & \end{array}$$

$$
\begin{array}{rcl@{\qquad}rcl}
\mathsf{Subseq}([],\ t) & \to & \mathsf{True} & \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\
\mathsf{Subseq}(s,\ []) & \to & \mathsf{False} & \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\
\mathsf{tl}(x;y) & \to & y & \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t)
\end{array}
$$

Statements:

$$
\begin{array}{rcl}
\mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\
\mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\
 & \dots & \\
\mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \mathsf{False} \\
\mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \\
\mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}([],\ []) & \to^* & \mathsf{True}, \mathsf{False} \\
 & \dots &
\end{array}
$$

$$
\begin{array}{rcl rcl}
\mathsf{Subseq}([],\ t) & \to & \mathsf{True} & \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\
\mathsf{Subseq}(s,\ []) & \to & \mathsf{False} & \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\
\mathsf{tl}(x;y) & \to & y & \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t)
\end{array}
$$

Statements:

$$
\begin{array}{rcl}
\mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\
\mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\
 & \ldots & \\
\mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \mathsf{False} \\
\mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \\
\mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}([],\ []) & \to^* & \mathsf{True}, \mathsf{False} \\
 & \ldots &
\end{array}
$$

$$\begin{array}{rcl} \mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\ \mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\ \mathsf{tl}(x;y) & \to & y \end{array} \qquad \begin{array}{rcl} \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\ \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\ \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t) \end{array}$$

Statements:

$$\begin{array}{rcl} \mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\ \mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\ & \dots & \\ \mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \\ \color{blue}{\mathsf{Subseq}(0;0;1;[],\ 1;0;1;[])} & \to^* & \\ \mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \mathsf{False} \\ \mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \\ \mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}([],\ []) & \to^* & \mathsf{True}, \mathsf{False} \\ & \dots & \end{array}$$

$$\begin{array}{rcl} \mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\ \mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\ \mathsf{tl}(x;y) & \to & y \end{array} \qquad \begin{array}{rcl} \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s,\mathsf{tl}(t)) \\ \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\ \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t) \end{array}$$

Statements:

$$\begin{array}{rcl} \mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\ \mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\ & \dots & \\ \mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \mathsf{False} \\ \mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \\ \mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}([],\ []) & \to^* & \mathsf{True}, \mathsf{False} \\ & \dots & \end{array}$$

$$
\begin{array}{rcl}
\mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\
\mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\
\mathsf{tl}(x;y) & \to & y
\end{array}
\qquad
\begin{array}{rcl}
\mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\
\mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\
\mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t)
\end{array}
$$

Statements:

$$
\begin{array}{rcl}
\mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\
\mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\
& \dots & \\
\mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \mathsf{False} \\
\mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \\
\mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}([],\ []) & \to^* & \mathsf{True}, \mathsf{False} \\
& \dots &
\end{array}
$$

$$\begin{array}{rcl} \mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\ \mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\ \mathsf{tl}(x;y) & \to & y \end{array} \qquad \begin{array}{rcl} \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s,\mathsf{tl}(t)) \\ \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\ \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t) \end{array}$$

Statements:

$$\begin{array}{rcl} \mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\ \mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\ & \dots & \\ \mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \mathsf{False} \\ \mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \\ \mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}([],\ []) & \to^* & \mathsf{True}, \mathsf{False} \\ & \dots & \end{array}$$

$$
\begin{array}{rcl}
\mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\
\mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\
\mathsf{tl}(x;y) & \to & y
\end{array}
\qquad
\begin{array}{rcl}
\mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s,\mathsf{tl}(t)) \\
\mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\
\mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t)
\end{array}
$$

Statements:

$$
\begin{array}{rcl}
\mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\
\mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\
 & \dots & \\
\mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \mathsf{False} \\
\mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \\
\color{blue}{\mathsf{Subseq}(0;1;[],\ 0;1;[])} & \to^* & \\
\mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}([],\ []) & \to^* & \mathsf{True}, \mathsf{False} \\
 & \dots &
\end{array}
$$

$$\begin{array}{rcl} \mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\ \mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\ \mathsf{tl}(x;y) & \to & y \end{array} \qquad \begin{array}{rcl} \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\ \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\ \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t) \end{array}$$

Statements:

$$\begin{array}{rcl} \mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\ \mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\ & \ldots & \\ \mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \mathsf{False} \\ \mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}([],\ []) & \to^* & \mathsf{True}, \mathsf{False} \\ & \ldots & \end{array}$$

$$
\begin{array}{rclrcl}
\mathsf{Subseq}([],\ t) & \to & \mathsf{True} & \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\
\mathsf{Subseq}(s,\ []) & \to & \mathsf{False} & \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\
\mathsf{tl}(x;y) & \to & y & \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t)
\end{array}
$$

Statements:

$$
\begin{array}{rcl}
\mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\
\mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\
& \ldots & \\
\mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \mathsf{False} \\
\mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\color{blue}{\mathsf{Subseq}(1;[],\ 1;[])} & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}([],\ []) & \to^* & \mathsf{True}, \mathsf{False} \\
& \ldots &
\end{array}
$$

$$\begin{array}{rcl} \mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\ \mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\ \mathsf{tl}(x;y) & \to & y \end{array} \qquad \begin{array}{rcl} \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s,\mathsf{tl}(t)) \\ \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\ \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t) \end{array}$$

Statements:

$$\begin{array}{rcl} \mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\ \mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\ & \dots & \\ \mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \mathsf{False} \\ \mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \mathsf{True},\mathsf{False} \\ \mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \mathsf{True},\mathsf{False} \\ \mathsf{Subseq}([],\ []) & \to^* & \mathsf{True},\mathsf{False} \\ & \dots & \end{array}$$

$$\begin{array}{rcl}
\mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\
\mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\
\mathsf{tl}(x;y) & \to & y
\end{array}
\qquad
\begin{array}{rcl}
\mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\
\mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\
\mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t)
\end{array}$$

Statements:

$$\begin{array}{rcl}
\mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\
\mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\
& \dots & \\
\mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \mathsf{False} \\
\mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}([],\ []) & \to^* & \mathsf{True}, \mathsf{False}
\end{array}$$

...

$$
\begin{array}{rcl}
\mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\
\mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\
\mathsf{tl}(x;y) & \to & y
\end{array}
\qquad
\begin{array}{rcl}
\mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\
\mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\
\mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t)
\end{array}
$$

Statements:

$$
\begin{array}{rcl}
\color{blue}{\mathsf{tl}(0;1;0;1;[])} & \to^* & 1;0;1;[] \\
\mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\
 & \ldots & \\
\mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \mathsf{False} \\
\mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}([],\ []) & \to^* & \mathsf{True}, \mathsf{False} \\
 & \ldots &
\end{array}
$$

$$\begin{array}{rcl} \mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\ \mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\ \mathsf{tl}(x;y) & \to & y \end{array} \qquad \begin{array}{rcl} \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\ \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\ \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t) \end{array}$$

Statements:

$$\begin{array}{rcl} \mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\ \mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\ & \dots & \\ \mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \mathsf{False} \\ \mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}([],\ []) & \to^* & \mathsf{True}, \mathsf{False} \\ & \dots & \end{array}$$

$$
\begin{array}{rcl@{\qquad}rcl}
\mathsf{Subseq}([],\ t) & \to & \mathsf{True} & \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\
\mathsf{Subseq}(s,\ []) & \to & \mathsf{False} & \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\
\mathsf{tl}(x;y) & \to & y & \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t)
\end{array}
$$

Statements:

$$
\begin{array}{rcl}
\mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\
\mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\
 & \dots & \\
\mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \mathsf{False} \\
\mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}([],\ []) & \to^* & \mathsf{True}, \mathsf{False}
\end{array}
$$

...

$$
\begin{array}{rcl}
\mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\
\mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\
\mathsf{tl}(x;y) & \to & y
\end{array}
\qquad
\begin{array}{rcl}
\mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\
\mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\
\mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t)
\end{array}
$$

Statements:

$$
\begin{array}{rcl}
\mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\
\mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\
 & \dots & \\
\mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \mathsf{False} \\
\mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}([],\ []) & \to^* & \mathsf{True}, \mathsf{False} \\
 & \dots &
\end{array}
$$

$$\begin{array}{rcl} \mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\ \mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\ \mathsf{tl}(x;y) & \to & y \end{array} \qquad \begin{array}{rcl} \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s,\mathsf{tl}(t)) \\ \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\ \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t) \end{array}$$

Statements:

$$\begin{array}{rcl} \mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\ \mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\ & \dots & \\ \mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \mathsf{False} \\ \mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}([],\ []) & \to^* & \mathsf{True}, \mathsf{False} \\ & \dots & \end{array}$$

$$
\begin{array}{rcl@{\qquad}rcl}
\mathsf{Subseq}([],\ t) & \to & \mathsf{True} & \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\
\mathsf{Subseq}(s,\ []) & \to & \mathsf{False} & \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\
\mathsf{tl}(x;y) & \to & y & \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t)
\end{array}
$$

Statements:

$$
\begin{array}{rcl}
\mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\
\mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\
 & \dots & \\
\mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \mathsf{False} \\
\mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \mathsf{False} \\
\mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}([],\ []) & \to^* & \mathsf{True}, \mathsf{False} \\
 & \dots &
\end{array}
$$

$$
\begin{array}{rcl rcl}
\mathsf{Subseq}([],\ t) & \to & \mathsf{True} & \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\
\mathsf{Subseq}(s,\ []) & \to & \mathsf{False} & \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\
\mathsf{tl}(x;y) & \to & y & \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t)
\end{array}
$$

Statements:

$$
\begin{array}{rcl}
\mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\
\mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\
& \dots & \\
\mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \mathsf{False} \\
\mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \mathsf{False} \\
\mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}([],\ []) & \to^* & \mathsf{True}, \mathsf{False} \\
& \dots &
\end{array}
$$

$$\begin{array}{rcl} \mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\ \mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\ \mathsf{tl}(x;y) & \to & y \end{array} \qquad \begin{array}{rcl} \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\ \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\ \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t) \end{array}$$

Statements:

$$\begin{array}{rcl} \mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\ \mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\ & \dots & \\ \mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \mathsf{False} \\ \mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \mathsf{False} \\ \mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}([],\ []) & \to^* & \mathsf{True}, \mathsf{False} \\ & \dots & \end{array}$$

$$
\begin{array}{rclrcl}
\mathsf{Subseq}([],\ t) & \to & \mathsf{True} & \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\
\mathsf{Subseq}(s,\ []) & \to & \mathsf{False} & \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\
\mathsf{tl}(x;y) & \to & y & \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t)
\end{array}
$$

Statements:

$$
\begin{array}{rcl}
\mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\
\mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\
 & \dots & \\
\mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \mathsf{False} \\
\mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \mathsf{False} \\
\mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}([],\ []) & \to^* & \mathsf{True}, \mathsf{False} \\
 & \dots &
\end{array}
$$

$$\begin{array}{rcl} \mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\ \mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\ \mathsf{tl}(x;y) & \to & y \end{array} \qquad \begin{array}{rcl} \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\ \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\ \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t) \end{array}$$

Statements:

$$\begin{array}{rcl} \mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\ \mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\ & \dots & \\ \mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \mathsf{False} \\ \mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \mathsf{False} \\ \mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}([],\ []) & \to^* & \mathsf{True}, \mathsf{False} \\ & \dots & \end{array}$$

$$
\begin{array}{rcl rcl}
\mathsf{Subseq}([],\ t) & \to & \mathsf{True} & \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\
\mathsf{Subseq}(s,\ []) & \to & \mathsf{False} & \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\
\mathsf{tl}(x;y) & \to & y & \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t)
\end{array}
$$

Statements:

$$
\begin{array}{rcl}
\mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\
\mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\
& \dots & \\
\mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \mathsf{False} \\
\mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \mathsf{False} \\
\mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\color{blue}{\mathsf{Subseq}(0;1;[],\ 0;1;[])} & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}([],\ []) & \to^* & \mathsf{True}, \mathsf{False} \\
& \dots &
\end{array}
$$

$$
\begin{array}{rcl@{\qquad}rcl}
\mathsf{Subseq}([],\ t) & \to & \mathsf{True} & \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\
\mathsf{Subseq}(s,\ []) & \to & \mathsf{False} & \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\
\mathsf{tl}(x;y) & \to & y & \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t)
\end{array}
$$

Statements:

$$
\begin{array}{rcl}
\mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\
\mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\
& \dots & \\
\mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \mathsf{False} \\
\mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \mathsf{False} \\
\mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}([],\ []) & \to^* & \mathsf{True}, \mathsf{False} \\
& \dots &
\end{array}
$$

$$
\begin{array}{rcl@{\qquad}rcl}
\mathsf{Subseq}([],\ t) & \to & \mathsf{True} & \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\
\mathsf{Subseq}(s,\ []) & \to & \mathsf{False} & \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\
\mathsf{tl}(x;y) & \to & y & \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t)
\end{array}
$$

Statements:

$$
\begin{array}{rcl}
\mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\
\mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\
 & \dots & \\
\mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \mathsf{False} \\
\mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \mathsf{False} \\
\mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}([],\ []) & \to^* & \mathsf{True}, \mathsf{False} \\
 & \dots &
\end{array}
$$

$$\begin{array}{rcl} \mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\ \mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\ \mathsf{tl}(x;y) & \to & y \end{array} \qquad \begin{array}{rcl} \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s,\mathsf{tl}(t)) \\ \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\ \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t) \end{array}$$

Statements:

$$\begin{array}{rcl} \mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\ \mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\ & \dots & \\ \mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \mathsf{False} \\ \mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \mathsf{False} \\ \mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}([],\ []) & \to^* & \mathsf{True}, \mathsf{False} \end{array}$$

...

$$\begin{array}{rcl} \mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\ \mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\ \mathsf{tl}(x;y) & \to & y \end{array} \qquad \begin{array}{rcl} \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s,\mathsf{tl}(t)) \\ \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\ \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t) \end{array}$$

Statements:

$$\begin{array}{rcl} \mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\ \mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\ & \dots & \\ \mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \mathsf{False} \\ \mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \mathsf{False} \\ \mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}([],\ []) & \to^* & \mathsf{True}, \mathsf{False} \\ & \dots & \end{array}$$

$$\begin{array}{rcl} \mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\ \mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\ \mathsf{tl}(x;y) & \to & y \end{array} \qquad \begin{array}{rcl} \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\ \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\ \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t) \end{array}$$

Statements:

$$\begin{array}{rcl} \mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\ \mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\ & \dots & \\ \mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \mathsf{False} \\ \mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \mathsf{False} \\ \mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}([],\ []) & \to^* & \mathsf{True}, \mathsf{False} \\ & \dots & \end{array}$$

$$
\begin{array}{rcl@{\qquad}rcl}
\mathsf{Subseq}([],\ t) & \to & \mathsf{True} & \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\
\mathsf{Subseq}(s,\ []) & \to & \mathsf{False} & \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\
\mathsf{tl}(x;y) & \to & y & \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t)
\end{array}
$$

Statements:

$$
\begin{array}{rcl}
\mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\
\mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\
& \dots & \\
\mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \mathsf{False} \\
\mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \mathsf{False} \\
\mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}([],\ []) & \to^* & \mathsf{True}, \mathsf{False}
\end{array}
$$

. . .

$$\begin{array}{rcl} \mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\ \mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\ \mathsf{tl}(x;y) & \to & y \end{array} \qquad \begin{array}{rcl} \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\ \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\ \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t) \end{array}$$

Statements:

$$\begin{array}{rcl} \mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\ \mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\ & \ldots & \\ \color{blue}{\mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[])} & \to^* & \\ \mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \mathsf{False} \\ \mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \mathsf{False} \\ \mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}([],\ []) & \to^* & \mathsf{True}, \mathsf{False} \\ & \ldots & \end{array}$$

$$\begin{array}{rcl} \mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\ \mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\ \mathsf{tl}(x;y) & \to & y \end{array} \qquad \begin{array}{rcl} \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s,\mathsf{tl}(t)) \\ \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\ \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t) \end{array}$$

Statements:

$$\begin{array}{rcl} \mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\ \mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\ & \dots & \\ \mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \mathsf{False} \\ \mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \mathsf{False} \\ \mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}([],\ []) & \to^* & \mathsf{True}, \mathsf{False} \\ & \dots & \end{array}$$

$$\begin{array}{rcl} \mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\ \mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\ \mathsf{tl}(x;y) & \to & y \end{array} \qquad \begin{array}{rcl} \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\ \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\ \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t) \end{array}$$

Statements:

$$\begin{array}{rcl} \mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\ \mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\ & \dots & \\ \mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \mathsf{False} \\ \mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \mathsf{False} \\ \mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}([],\ []) & \to^* & \mathsf{True}, \mathsf{False} \\ & \dots & \end{array}$$

$$\begin{array}{rcl} \mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\ \mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\ \mathsf{tl}(x;y) & \to & y \end{array} \qquad \begin{array}{rcl} \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\ \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\ \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t) \end{array}$$

Statements:

$$\begin{array}{rcl} \mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\ \mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\ & \dots & \\ \mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \mathsf{False} \\ \mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \mathsf{False} \\ \mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \\ \mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}([],\ []) & \to^* & \mathsf{True}, \mathsf{False} \\ & \dots & \end{array}$$

$$\begin{array}{rcl@{\qquad}rcl}
\mathsf{Subseq}([],\ t) & \to & \mathsf{True} & \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\
\mathsf{Subseq}(s,\ []) & \to & \mathsf{False} & \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\
\mathsf{tl}(x;y) & \to & y & \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t)
\end{array}$$

Statements:

$$\begin{array}{rcl}
\mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\
\mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\
 & \dots & \\
\mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \mathsf{False} \\
\mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \mathsf{False} \\
\mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \\
\mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}([],\ []) & \to^* & \mathsf{True}, \mathsf{False} \\
 & \dots &
\end{array}$$

$$\begin{array}{rcl} \mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\ \mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\ \mathsf{tl}(x;y) & \to & y \end{array} \qquad \begin{array}{rcl} \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\ \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\ \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t) \end{array}$$

Statements:

$$\begin{array}{rcl} \mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\ \mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\ & \dots & \\ \mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \mathsf{False} \\ \mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \mathsf{False} \\ \mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}([],\ []) & \to^* & \mathsf{True}, \mathsf{False} \\ & \dots & \end{array}$$

$$
\begin{array}{rcl}
\mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\
\mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\
\mathsf{tl}(x;y) & \to & y
\end{array}
\qquad
\begin{array}{rcl}
\mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\
\mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\
\mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t)
\end{array}
$$

Statements:

$$
\begin{array}{rcl}
\mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\
\mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\
 & \dots & \\
\mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \mathsf{False} \\
\mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \mathsf{False} \\
\mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}([],\ []) & \to^* & \mathsf{True}, \mathsf{False} \\
 & \dots &
\end{array}
$$

$$\begin{array}{rcl}
\mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\
\mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\
\mathsf{tl}(x;y) & \to & y
\end{array}
\qquad
\begin{array}{rcl}
\mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s,\mathsf{tl}(t)) \\
\mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\
\mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t)
\end{array}$$

Statements:

$$\begin{array}{rcl}
\mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\
\mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\
& \dots & \\
\mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \mathsf{True},\mathsf{False} \\
\mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \mathsf{False} \\
\mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \mathsf{False} \\
\mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \mathsf{True},\mathsf{False} \\
\mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \mathsf{True},\mathsf{False} \\
\mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \mathsf{True},\mathsf{False} \\
\mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \mathsf{True},\mathsf{False} \\
\mathsf{Subseq}([],\ []) & \to^* & \mathsf{True},\mathsf{False} \\
& \dots &
\end{array}$$

$$\begin{array}{rcl} \mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\ \mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\ \mathsf{tl}(x;y) & \to & y \end{array} \qquad \begin{array}{rcl} \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s,\mathsf{tl}(t)) \\ \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\ \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t) \end{array}$$

Statements:

$$\begin{array}{rcl} \mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\ \mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\ & \dots & \\ \mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \mathsf{True},\mathsf{False} \\ \mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \mathsf{False} \\ \mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \mathsf{False} \\ \mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \mathsf{True},\mathsf{False} \\ \mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \mathsf{True},\mathsf{False} \\ \mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \mathsf{True},\mathsf{False} \\ \mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \mathsf{True},\mathsf{False} \\ \mathsf{Subseq}([],\ []) & \to^* & \mathsf{True},\mathsf{False} \\ & \dots & \end{array}$$

$$
\begin{array}{rcl@{\qquad}rcl}
\mathsf{Subseq}([],\ t) & \to & \mathsf{True} & \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\
\mathsf{Subseq}(s,\ []) & \to & \mathsf{False} & \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\
\mathsf{tl}(x;y) & \to & y & \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t)
\end{array}
$$

Statements:

$$
\begin{array}{rcl}
\mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\
\mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\
 & \dots & \\
\mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \mathsf{False} \\
\mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \mathsf{False} \\
\mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}([],\ []) & \to^* & \mathsf{True}, \mathsf{False} \\
 & \dots &
\end{array}
$$

$$\begin{array}{rcl} \mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\ \mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\ \mathsf{tl}(x;y) & \to & y \end{array} \qquad \begin{array}{rcl} \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\ \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\ \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t) \end{array}$$

Statements:

$$\begin{array}{rcl} \mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\ \mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\ & \dots & \\ \mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \mathsf{False} \\ \mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \mathsf{False} \\ \mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}([],\ []) & \to^* & \mathsf{True}, \mathsf{False} \end{array}$$

...

$$\begin{array}{rcl} \mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\ \mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\ \mathsf{tl}(x;y) & \to & y \end{array} \qquad \begin{array}{rcl} \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\ \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\ \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t) \end{array}$$

Statements:

$$\begin{array}{rcl} \mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\ \mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\ & \dots & \\ \mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \mathsf{False} \\ \mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \mathsf{False} \\ \mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}([],\ []) & \to^* & \mathsf{True}, \mathsf{False} \\ & \dots & \end{array}$$

$$\begin{array}{rcl} \mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\ \mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\ \mathsf{tl}(x;y) & \to & y \end{array} \qquad \begin{array}{rcl} \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\ \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\ \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t) \end{array}$$

Statements:

$$\begin{array}{rcl} \mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\ \mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\ & \dots & \\ \mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \mathsf{False} \\ \mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \mathsf{False} \\ \mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}([],\ []) & \to^* & \mathsf{True}, \mathsf{False} \\ & \dots & \end{array}$$

$$\begin{array}{rcl} \mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\ \mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\ \mathsf{tl}(x;y) & \to & y \end{array} \qquad \begin{array}{rcl} \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s,\mathsf{tl}(t)) \\ \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\ \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t) \end{array}$$

Statements:

$$\begin{array}{rcl} \mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\ \mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\ & \dots & \\ \mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \mathsf{False} \\ \mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \mathsf{False} \\ \mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}([],\ []) & \to^* & \mathsf{True}, \mathsf{False} \\ & \dots & \end{array}$$

$$\begin{array}{rcl}
\mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\
\mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\
\mathsf{tl}(x;y) & \to & y
\end{array}
\qquad
\begin{array}{rcl}
\mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\
\mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\
\mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t)
\end{array}$$

Statements:

$$\begin{array}{rcl}
\mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\
\mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\
& \dots & \\
\mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \mathsf{False} \\
\mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \mathsf{False} \\
\mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}([],\ []) & \to^* & \mathsf{True}, \mathsf{False} \\
& \dots &
\end{array}$$

$$
\begin{array}{rclrcl}
\mathsf{Subseq}([],\ t) & \to & \mathsf{True} & \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\
\mathsf{Subseq}(s,\ []) & \to & \mathsf{False} & \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\
\mathsf{tl}(x;y) & \to & y & \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t)
\end{array}
$$

Statements:

$$
\begin{array}{rcl}
\mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\
\mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\
 & \dots & \\
\mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \mathsf{False} \\
\mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \mathsf{False} \\
\mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}([],\ []) & \to^* & \mathsf{True}, \mathsf{False} \\
 & \dots &
\end{array}
$$

$$
\begin{array}{rcl rcl}
\mathsf{Subseq}([],\ t) & \to & \mathsf{True} & \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\
\mathsf{Subseq}(s,\ []) & \to & \mathsf{False} & \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\
\mathsf{tl}(x;y) & \to & y & \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t)
\end{array}
$$

Statements:

$$
\begin{array}{rcl}
\mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\
\mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\
 & \dots & \\
\mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \mathsf{False} \\
\color{blue}{\mathsf{Subseq}(0;0;1;[],\ [])} & \to^* & \mathsf{False} \\
\mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}([],\ []) & \to^* & \mathsf{True}, \mathsf{False} \\
 & \dots &
\end{array}
$$

$$\begin{array}{rcl} \mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\ \mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\ \mathsf{tl}(x;y) & \to & y \end{array} \qquad \begin{array}{rcl} \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s,\mathsf{tl}(t)) \\ \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\ \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t) \end{array}$$

Statements:

$$\begin{array}{rcl} \mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\ \mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\ & \dots & \\ \mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \mathsf{False} \\ \mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \mathsf{False} \\ \mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}([],\ []) & \to^* & \mathsf{True}, \mathsf{False} \\ & \dots & \end{array}$$

$$
\begin{array}{rclrcl}
\mathsf{Subseq}([],\ t) & \to & \mathsf{True} & \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\
\mathsf{Subseq}(s,\ []) & \to & \mathsf{False} & \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\
\mathsf{tl}(x;y) & \to & y & \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t)
\end{array}
$$

Statements:

$$
\begin{array}{rcl}
\mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\
\mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\
& \dots & \\
\mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \mathsf{False} \\
\mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \mathsf{False} \\
\mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}([],\ []) & \to^* & \mathsf{True}, \mathsf{False} \\
& \dots &
\end{array}
$$

$$\begin{array}{rcl} \mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\ \mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\ \mathsf{tl}(x;y) & \to & y \end{array} \qquad \begin{array}{rcl} \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s,\mathsf{tl}(t)) \\ \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\ \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t) \end{array}$$

Statements:

$$\begin{array}{rcl} \mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\ \mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\ & \cdots & \\ \mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \mathsf{False} \\ \mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \mathsf{False} \\ \mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}([],\ []) & \to^* & \mathsf{True}, \mathsf{False} \\ & \cdots & \end{array}$$

$$\begin{array}{rcl} \mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\ \mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\ \mathsf{tl}(x;y) & \to & y \end{array} \qquad \begin{array}{rcl} \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\ \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\ \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t) \end{array}$$

Statements:

$$\begin{array}{rcl} \mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\ \mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\ & \dots & \\ \mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \mathsf{False} \\ \mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \mathsf{False} \\ \mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}([],\ []) & \to^* & \mathsf{True}, \mathsf{False} \\ & \dots & \end{array}$$

$$
\begin{array}{rcl}
\mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\
\mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\
\mathsf{tl}(x;y) & \to & y
\end{array}
\qquad
\begin{array}{rcl}
\mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\
\mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\
\mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t)
\end{array}
$$

Statements:

$$
\begin{array}{rcl}
\mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\
\mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\
 & \ldots & \\
\mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \mathsf{False} \\
\mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \mathsf{False} \\
\mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}([],\ []) & \to^* & \mathsf{True}, \mathsf{False} \\
 & \ldots &
\end{array}
$$

$$
\begin{array}{rcl}
\mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\
\mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\
\mathsf{tl}(x;y) & \to & y
\end{array}
\qquad
\begin{array}{rcl}
\mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\
\mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\
\mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t)
\end{array}
$$

Statements:

$$
\begin{array}{rcl}
\mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\
\mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\
 & \dots & \\
\mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \mathsf{False} \\
\mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \mathsf{False} \\
\mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}([],\ []) & \to^* & \mathsf{True}, \mathsf{False}
\end{array}
$$

...

$$\begin{array}{rcl} \mathsf{Subseq}([],\ t) & \to & \mathsf{True} \\ \mathsf{Subseq}(s,\ []) & \to & \mathsf{False} \\ \mathsf{tl}(x;y) & \to & y \end{array} \qquad \begin{array}{rcl} \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s,\mathsf{tl}(t)) \\ \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\ \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t) \end{array}$$

Statements:

$$\begin{array}{rcl} \mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\ \mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\ & \dots & \\ \mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \mathsf{False} \\ \mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \mathsf{False} \\ \mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\ \mathsf{Subseq}([],\ []) & \to^* & \mathsf{True}, \mathsf{False} \\ & \dots & \end{array}$$

$$
\begin{array}{rclrcl}
\mathsf{Subseq}([],\ t) & \to & \mathsf{True} & \mathsf{Subseq}(s,\ t) & \to & \mathsf{Subseq}(s, \mathsf{tl}(t)) \\
\mathsf{Subseq}(s,\ []) & \to & \mathsf{False} & \mathsf{Subseq}(0;x,\ 0;t) & \to & \mathsf{Subseq}(s,\ t) \\
\mathsf{tl}(x;y) & \to & y & \mathsf{Subseq}(1;x,\ 1;t) & \to & \mathsf{Subseq}(s,\ t)
\end{array}
$$

Statements:

$$
\begin{array}{rcl}
\mathsf{tl}(0;1;0;1;[]) & \to^* & 1;0;1;[] \\
\mathsf{tl}(1;0;1;[]) & \to^* & 0;1;[] \\
& \dots & \\
\mathsf{Subseq}(0;0;1;[],\ 0;1;0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}(0;0;1;[],\ 1;0;1;[]) & \to^* & \mathsf{False} \\
\mathsf{Subseq}(0;0;1;[],\ []) & \to^* & \mathsf{False} \\
\mathsf{Subseq}(1;0;1;[],\ 0;1;0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}(0;1;[],\ 1;0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}(0;1;[],\ 0;1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}(1;[],\ 1;[]) & \to^* & \mathsf{True}, \mathsf{False} \\
\mathsf{Subseq}([],\ []) & \to^* & \mathsf{True}, \mathsf{False} \\
& \dots &
\end{array}
$$

Algorithm: formal description

Algorithm: formal description

Given: TRS with rules $\mathcal{R}$

Algorithm: formal description

Given: TRS with rules $\mathcal{R}$

Input: start term $\mathsf{Start}(s_1, \dots, s_n)$ with $s_1, \dots, s_n$ data

Algorithm: formal description

Given: TRS with rules $\mathcal{R}$
Input: start term $\mathsf{Start}(s_1, \dots, s_n)$ with $s_1, \dots, s_n$ data
Intended output: true if $\mathsf{Start}(s_1, \dots, s_n) \to_{\mathcal{R}}^* \mathsf{True}$ or false if not

Algorithm: formal description

Given: TRS with rules $\mathcal{R}$
Input: start term $\mathsf{Start}(s_1, \ldots, s_n)$ with $s_1, \ldots, s_n$ data
Intended output: true if $\mathsf{Start}(s_1, \ldots, s_n) \to_{\mathcal{R}}^* \mathsf{True}$ or false if not

Initialisation:

- let $\mathcal{B}$ contain all subterms of s and data subterms of rule right-hand sides

Algorithm: formal description

Given: TRS with rules $\mathcal{R}$
Input: start term $\mathsf{Start}(s_1, \dots, s_n)$ with $s_1, \dots, s_n$ data
Intended output: true if $\mathsf{Start}(s_1, \dots, s_n) \to_{\mathcal{R}}^{*} \mathsf{True}$ or false if not

Initialisation:

- let $\mathcal{B}$ contain all subterms of s and data subterms of rule right-hand sides
- make a list of all statements $f(t_1, \dots, t_m) \to^{*} \{\}$ with $\vec{t} \in \mathcal{B}$

Algorithm: formal description

Given: TRS with rules $\mathcal{R}$
Input: start term $\mathsf{Start}(s_1, \dots, s_n)$ with $s_1, \dots, s_n$ data
Intended output: true if $\mathsf{Start}(s_1, \dots, s_n) \to_{\mathcal{R}}^{*}$ True or false if not

Initialisation:

- let $\mathcal{B}$ contain all subterms of s and data subterms of rule right-hand sides
- make a list of all statements $f(t_1, \dots, t_m) \to^* \{\}$ with $\vec{t} \in \mathcal{B}$

Repeat:

- find all u such that $f(t_1, \dots, t_m) \to_{\mathcal{R}} u$

Algorithm: formal description

Given: TRS with rules $\mathcal{R}$
Input: start term $\mathsf{Start}(s_1, \dots, s_n)$ with $s_1, \dots, s_n$ data
Intended output: true if $\mathsf{Start}(s_1, \dots, s_n) \to^*_{\mathcal{R}}$ True or false if not

Initialisation:

- let $\mathcal{B}$ contain all subterms of s and data subterms of rule right-hand sides
- make a list of all statements $f(t_1, \dots, t_m) \to^* \{\}$ with $\vec{t} \in \mathcal{B}$

Repeat:

- find all u such that $f(t_1, \dots, t_m) \to_{\mathcal{R}} u$
- simplify u using "confirmed steps"

Algorithm: formal description

Given: TRS with rules $\mathcal{R}$
Input: start term $\mathsf{Start}(s_1, \dots, s_n)$ with $s_1, \dots, s_n$ data
Intended output: true if $\mathsf{Start}(s_1, \dots, s_n) \to^*_{\mathcal{R}}$ True or false if not

Initialisation:

- let $\mathcal{B}$ contain all subterms of s and data subterms of rule right-hand sides
- make a list of all statements $f(t_1, \dots, t_m) \to^* \{\}$ with $\vec{t} \in \mathcal{B}$

Repeat:

- find all u such that $f(t_1, \dots, t_m) \to_{\mathcal{R}} u$
- simplify u using “confirmed steps”
- if u simplifies to data q, include q in the set

Algorithm: formal description

Given: TRS with rules $\mathcal{R}$
Input: start term $\mathsf{Start}(s_1, \dots, s_n)$ with $s_1, \dots, s_n$ data
Intended output: true if $\mathsf{Start}(s_1, \dots, s_n) \to_{\mathcal{R}}^* \mathsf{True}$ or false if not

Initialisation:

- let $\mathcal{B}$ contain all subterms of s and data subterms of rule right-hand sides
- make a list of all statements $f(t_1, \dots, t_m) \to^* \{\}$ with $\vec{t} \in \mathcal{B}$

Repeat:

- find all u such that $f(t_1, \dots, t_m) \to_{\mathcal{R}} u$
- simplify u using "confirmed steps"
- if u simplifies to data q, include q in the set

Conclusion:

- stop when no new elements are added to sets

Algorithm: formal description

Given: TRS with rules $\mathcal{R}$
Input: start term $\mathsf{Start}(s_1, \dots, s_n)$ with $s_1, \dots, s_n$ data
Intended output: true if $\mathsf{Start}(s_1, \dots, s_n) \to^*_{\mathcal{R}}$ True or false if not

Initialisation:

- let $\mathcal{B}$ contain all subterms of s and data subterms of rule right-hand sides
- make a list of all statements $f(t_1, \dots, t_m) \to^* \{\}$ with $\vec{t} \in \mathcal{B}$

Repeat:

- find all u such that $f(t_1, \dots, t_m) \to_{\mathcal{R}} u$
- simplify u using "confirmed steps"
- if u simplifies to data q, include q in the set

Conclusion:

- stop when no new elements are added to sets
- return true if and only if $\mathsf{Start}(s_1, \dots, s_n) \to^* A \cup \{\mathsf{True}\}$

Conclusion:

Conclusion:

- call-by-value term rewriting characterises PTIME

Conclusion:

- call-by-value term rewriting characterises PTIME
- innermost term rewriting characterises PTIME

Conclusion:

- call-by-value term rewriting characterises PTIME
- innermost term rewriting characterises PTIME

Note: non-determinism does not increase expressivity

Conclusion:

- call-by-value term rewriting characterises PTIME
- innermost term rewriting characterises PTIME

Note: non-determinism does not increase expressivity
(see also: Cook '71, Bonfante '06)

Overview

Recall:

Recall: any algorithm in PTIME can be implemented by a deterministic cons-free TRS with call-by-value reduction

Recall: any algorithm in PTIME can be implemented by a deterministic cons-free TRS with call-by-value reduction

Question:

does deterministic term rewriting characterise PTIME?

Recall: any algorithm in PTIME can be implemented by a deterministic cons-free TRS with call-by-value reduction

Question:

does deterministic term rewriting characterise PTIME?

Answer:

YES

Recall: any algorithm in PTIME can be implemented by a deterministic cons-free TRS with call-by-value reduction

Question:

does deterministic term rewriting characterise PTIME?

Answer:

YES

Proof:

Recall: any algorithm in PTIME can be implemented by a deterministic cons-free TRS with call-by-value reduction

Question:

does deterministic term rewriting characterise PTIME?

Answer:

YES

Proof:

suppose $\rightarrow_{\mathcal{R}}^{*}$ has unique normal forms

Recall: any algorithm in PTIME can be implemented by a deterministic cons-free TRS with call-by-value reduction

Question:

does deterministic term rewriting characterise PTIME?

Answer:

YES

Proof:

suspose $\rightarrow^*_{\mathcal{R}}$ has unique normal forms

then

Recall: any algorithm in PTIME can be implemented by a deterministic cons-free TRS with call-by-value reduction

Question:

does deterministic term rewriting characterise PTIME?

Answer:

YES

Proof:

suppose $\to_{\mathcal{R}}^{*}$ has unique normal forms

then

$s \to_{\mathcal{R}}^{*} t$ with t data if and only if $s \to_{\mathcal{R},\text{call-by-value}}^{*} t$

Recall: any algorithm in PTIME can be implemented by a deterministic cons-free TRS with call-by-value reduction

Question:

does deterministic term rewriting characterise PTIME?

Answer:

YES

Proof:

suppose $\rightarrow^*_{\mathcal{R}}$ has unique normal forms
then exists $\mathcal{R}'$ such that
$s \rightarrow^*_{\mathcal{R}} t$ with t data if and only if $s \rightarrow^*_{\mathcal{R}',\text{call-by-value}} t$

Recall: any algorithm in PTIME can be implemented by a deterministic cons-free TRS with call-by-value reduction

Question:

does deterministic term rewriting characterise PTIME?

Answer:

YES

Proof:

suppose $\to_{\mathcal{R}}^{*}$ has unique normal forms

then exists $\mathcal{R}'$ such that

$s \to_{\mathcal{R}}^{*} t$ with t data if and only if $s \to_{\mathcal{R}',\text{call-by-value}}^{*} t$

(in a cons-free TRS)

Semi-Linear Term Rewriting (adapted from [Carvalho and Simonsen '14])

Semi-Linear Term Rewriting (adapted from [Carvalho and Simonsen '14])

Definition:

a rule $f(\ell_1, \dots, \ell_n) \to r$ is semi-linear if:

Semi-Linear Term Rewriting (adapted from [Carvalho and Simonsen '14])

Definition:

a rule $f(\ell_1, \dots, \ell_n) \to r$ is semi-linear if:

for each ℓ_i:

Semi-Linear Term Rewriting (adapted from [Carvalho and Simonsen '14])

Definition:

a rule $f(\ell_1, \dots, \ell_n) \to r$ is semi-linear if:

for each ℓ_i:

if ℓ_i is a variable then ℓ_i occurs at most once in r

Semi-Linear Term Rewriting (adapted from [Carvalho and Simonsen '14])

Definition:

a rule $f(\ell_1, \ldots, \ell_n) \to r$ is semi-linear if:
for each ℓ_i:
if ℓ_i is a variable then ℓ_i occurs at most once in r

Claim:

rewriting with semi-linear rules characterises PTIME

Semi-Linear Term Rewriting (adapted from [Carvalho and Simonsen '14])

Definition:

a rule $f(\ell_1, \dots, \ell_n) \to r$ is semi-linear if:
for each ℓ_i:
if ℓ_i is a variable then ℓ_i occurs at most once in r

Claim:

rewriting with semi-linear rules characterises PTIME

Proof:

Semi-Linear Term Rewriting (adapted from [Carvalho and Simonsen '14])

Definition:

a rule $f(\ell_1, \dots, \ell_n) \to r$ is semi-linear if:
for each ℓ_i:
if ℓ_i is a variable then ℓ_i occurs at most once in r

Claim:

rewriting with semi-linear rules characterises PTIME

Proof:

- same algorithm as for innermost can be expressed semi-linear

Semi-Linear Term Rewriting (adapted from [Carvalho and Simonsen '14])

Definition:

a rule $f(\ell_1, \dots, \ell_n) \to r$ is semi-linear if:
for each ℓ_i:
if ℓ_i is a variable then ℓ_i occurs at most once in r

Claim:

rewriting with semi-linear rules characterises PTIME

Proof:

- same algorithm as for innermost can be expressed semi-linear
- if all rules in $\mathcal{R}$ are semi-linear
  then exists $\mathcal{R}'$ such that
  $s \to_{\mathcal{R}}^{*} t$ with t data if and only if $s \to_{\mathcal{R}',\text{call-by-value}}^{*} t$

Definition: a rule $f(\ell_1, \dots, \ell_n) \to r$ is semi-linear if:

for each ℓ_i:

if ℓ_i is a variable then ℓ_i occurs at most once in r

Definition: a rule $f(\ell_1, \ldots, \ell_n) \to r$ is semi-linear if:

for each ℓ_i:

if ℓ_i is a variable then ℓ_i occurs at most once in r

What does this prohibit?

Definition: a rule $f(\ell_1, \dots, \ell_n) \to r$ is semi-linear if:
for each ℓ_i:
if ℓ_i is a variable then ℓ_i occurs at most once in r

What does this prohibit?

$$\begin{array}{lcl} \text{either}(x, y) & \to & x \\ \text{either}(x, y) & \to & y \end{array}$$

Definition: a rule $f(\ell_1, \dots, \ell_n) \to r$ is semi-linear if:
for each ℓ_i:
if ℓ_i is a variable then ℓ_i occurs at most once in r

What does this prohibit?

$$\begin{array}{rcl} \mathsf{either}(x, y) & \to & x \\ \mathsf{either}(x, y) & \to & y \\ \mathsf{f}(x, n) & \to & \mathsf{ifelse}(\mathsf{equal}(x, n), \mathsf{g}(x), \mathsf{g}(x, \mathsf{pred}(n))) \end{array}$$

Definition: a rule $f(\ell_1, \dots, \ell_n) \to r$ is semi-linear if:
for each ℓ_i:
if ℓ_i is a variable then ℓ_i occurs at most once in r

What does this prohibit?

$$\begin{array}{rcl} \mathsf{either}(x, y) & \to & x \\ \mathsf{either}(x, y) & \to & y \\ \mathsf{f}(x, n) & \to & \mathsf{ifelse}(\mathsf{equal}(x, n), \mathsf{g}(x), \mathsf{g}(x, \mathsf{pred}(n))) \end{array}$$

$$\mathsf{f}(\mathsf{either}(0, \mathsf{either}(2)), 1)$$

Definition: a rule $f(\ell_1, \ldots, \ell_n) \to r$ is semi-linear if:
for each ℓ_i:
if ℓ_i is a variable then ℓ_i occurs at most once in r

What does this prohibit?

$$\begin{array}{rcl} \mathsf{either}(x, y) & \to & x \\ \mathsf{either}(x, y) & \to & y \\ \mathsf{f}(x, n) & \to & \mathsf{ifelse}(\mathsf{equal}(x, n), \mathsf{g}(x), \mathsf{g}(x, \mathsf{pred}(n))) \end{array}$$

$$\begin{array}{rl} & \mathsf{f}(\mathsf{either}(0, \mathsf{either}(2)), 1) \\ \to & \mathsf{ifelse}(\mathsf{equal}(\mathsf{either}(0, \mathsf{either}(2)), 1), \mathsf{g}(\ldots), \mathsf{f}(\ldots, \mathsf{pred}(1))) \end{array}$$

Definition: a rule $f(\ell_1, \ldots, \ell_n) \to r$ is semi-linear if:

for each ℓ_i:

if ℓ_i is a variable then ℓ_i occurs at most once in r

What does this prohibit?

$$\begin{array}{rcl} \mathsf{either}(x, y) & \to & x \\ \mathsf{either}(x, y) & \to & y \\ \mathsf{f}(x, n) & \to & \mathsf{ifelse}(\mathsf{equal}(x, n), \mathsf{g}(x), \mathsf{g}(x, \mathsf{pred}(n))) \end{array}$$

$$\begin{array}{rc} & \mathsf{f}(\mathsf{either}(0, \mathsf{either}(2)), 1) \\ \to & \mathsf{ifelse}(\mathsf{equal}(\mathsf{either}(0, \mathsf{either}(2)), 1), \mathsf{g}(\ldots), \mathsf{f}(\ldots, \mathsf{pred}(1))) \\ \to & \mathsf{ifelse}(\mathsf{equal}(2, 1), \mathsf{g}(\ldots), \mathsf{f}(\ldots, \mathsf{pred}(1))) \end{array}$$

Definition: a rule $f(\ell_1, \dots, \ell_n) \to r$ is semi-linear if:
for each ℓ_i:
if ℓ_i is a variable then ℓ_i occurs at most once in r

What does this prohibit?

$$
\begin{array}{rcl}
\mathsf{either}(x, y) & \to & x \\
\mathsf{either}(x, y) & \to & y \\
\mathsf{f}(x, n) & \to & \mathsf{ifelse}(\mathsf{equal}(x, n), \mathsf{g}(x), \mathsf{g}(x, \mathsf{pred}(n)))
\end{array}
$$

$$
\begin{array}{rc}
 & \mathsf{f}(\mathsf{either}(0, \mathsf{either}(2)), 1) \\
\to & \mathsf{ifelse}(\mathsf{equal}(\mathsf{either}(0, \mathsf{either}(2)), 1), \mathsf{g}(\dots), \mathsf{f}(\dots, \mathsf{pred}(1))) \\
\to & \mathsf{ifelse}(\mathsf{equal}(2, 1), \mathsf{g}(\dots), \mathsf{f}(\dots, \mathsf{pred}(1))) \\
\to^* & \mathsf{f}(\mathsf{either}(0, \mathsf{either}(2)), \mathsf{pred}(1))
\end{array}
$$

Definition: a rule $f(\ell_1, \dots, \ell_n) \to r$ is semi-linear if:
for each ℓ_i:
if ℓ_i is a variable then ℓ_i occurs at most once in r

What does this prohibit?

$$
\begin{array}{rcl}
\mathsf{either}(x, y) & \to & x \\
\mathsf{either}(x, y) & \to & y \\
\mathsf{f}(x, n) & \to & \mathsf{ifelse}(\mathsf{equal}(x, n), \mathsf{g}(x), \mathsf{g}(x, \mathsf{pred}(n)))
\end{array}
$$

$$
\begin{array}{rc}
 & \mathsf{f}(\mathsf{either}(0, \mathsf{either}(2)), 1) \\
\to & \mathsf{ifelse}(\mathsf{equal}(\mathsf{either}(0, \mathsf{either}(2)), 1), \mathsf{g}(\dots), \mathsf{f}(\dots, \mathsf{pred}(1))) \\
\to & \mathsf{ifelse}(\mathsf{equal}(2, 1), \mathsf{g}(\dots), \mathsf{f}(\dots, \mathsf{pred}(1))) \\
\to^* & \mathsf{f}(\mathsf{either}(0, \mathsf{either}(2)), \mathsf{pred}(1)) \\
\to & \mathsf{f}(\mathsf{either}(0, \mathsf{either}(2)), 0)
\end{array}
$$

Definition: a rule $f(\ell_1, \dots, \ell_n) \to r$ is semi-linear if:
for each ℓ_i:
if ℓ_i is a variable then ℓ_i occurs at most once in r

What does this prohibit?

$$\begin{array}{rcl} \mathsf{either}(x, y) & \to & x \\ \mathsf{either}(x, y) & \to & y \\ \mathsf{f}(x, n) & \to & \mathsf{ifelse}(\mathsf{equal}(x, n), \mathsf{g}(x), \mathsf{g}(x, \mathsf{pred}(n))) \end{array}$$

$$\begin{array}{rl} & \mathsf{f}(\mathsf{either}(0, \mathsf{either}(2)), 1) \\ \to & \mathsf{ifelse}(\mathsf{equal}(\mathsf{either}(0, \mathsf{either}(2)), 1), \mathsf{g}(\dots), \mathsf{f}(\dots, \mathsf{pred}(1))) \\ \to & \mathsf{ifelse}(\mathsf{equal}(2, 1), \mathsf{g}(\dots), \mathsf{f}(\dots, \mathsf{pred}(1))) \\ \to^* & \mathsf{f}(\mathsf{either}(0, \mathsf{either}(2)), \mathsf{pred}(1)) \\ \to & \mathsf{f}(\mathsf{either}(0, \mathsf{either}(2)), 0) \\ \to & \mathsf{ifelse}(\mathsf{equal}(\mathsf{either}(0, \mathsf{either}(2)), 0), \mathsf{g}(\dots), \mathsf{f}(\dots, \mathsf{pred}(0))) \end{array}$$

Definition: a rule $f(\ell_1, \dots, \ell_n) \to r$ is semi-linear if:
for each ℓ_i:
if ℓ_i is a variable then ℓ_i occurs at most once in r

What does this prohibit?

$$
\begin{array}{rcl}
\mathsf{either}(x, y) & \to & x \\
\mathsf{either}(x, y) & \to & y \\
\mathsf{f}(x, n) & \to & \mathsf{ifelse}(\mathsf{equal}(x, n), \mathsf{g}(x), \mathsf{g}(x, \mathsf{pred}(n)))
\end{array}
$$

$$
\begin{array}{rc}
 & \mathsf{f}(\mathsf{either}(0, \mathsf{either}(2)), 1) \\
\to & \mathsf{ifelse}(\mathsf{equal}(\mathsf{either}(0, \mathsf{either}(2)), 1), \mathsf{g}(\dots), \mathsf{f}(\dots, \mathsf{pred}(1))) \\
\to & \mathsf{ifelse}(\mathsf{equal}(2, 1), \mathsf{g}(\dots), \mathsf{f}(\dots, \mathsf{pred}(1))) \\
\to^* & \mathsf{f}(\mathsf{either}(0, \mathsf{either}(2)), \mathsf{pred}(1)) \\
\to & \mathsf{f}(\mathsf{either}(0, \mathsf{either}(2)), 0) \\
\to & \mathsf{ifelse}(\mathsf{equal}(\mathsf{either}(0, \mathsf{either}(2)), 0), \mathsf{g}(\dots), \mathsf{f}(\dots, \mathsf{pred}(0))) \\
\to & \mathsf{ifelse}(\mathsf{equal}(0, 0), \mathsf{g}(\dots), \mathsf{f}(\dots, \mathsf{pred}(0)))
\end{array}
$$

Definition: a rule $f(\ell_1, \dots, \ell_n) \to r$ is semi-linear if:
for each ℓ_i:
if ℓ_i is a variable then ℓ_i occurs at most once in r

What does this prohibit?

$$
\begin{array}{rcl}
\mathsf{either}(x, y) & \to & x \\
\mathsf{either}(x, y) & \to & y \\
\mathsf{f}(x, n) & \to & \mathsf{ifelse}(\mathsf{equal}(x, n), \mathsf{g}(x), \mathsf{g}(x, \mathsf{pred}(n)))
\end{array}
$$

$$
\begin{array}{rc}
 & \mathsf{f}(\mathsf{either}(0, \mathsf{either}(2)), 1) \\
\to & \mathsf{ifelse}(\mathsf{equal}(\mathsf{either}(0, \mathsf{either}(2)), 1), \mathsf{g}(\dots), \mathsf{f}(\dots, \mathsf{pred}(1))) \\
\to & \mathsf{ifelse}(\mathsf{equal}(2, 1), \mathsf{g}(\dots), \mathsf{f}(\dots, \mathsf{pred}(1))) \\
\to^* & \mathsf{f}(\mathsf{either}(0, \mathsf{either}(2)), \mathsf{pred}(1)) \\
\to & \mathsf{f}(\mathsf{either}(0, \mathsf{either}(2)), 0) \\
\to & \mathsf{ifelse}(\mathsf{equal}(\mathsf{either}(0, \mathsf{either}(2)), 0), \mathsf{g}(\dots), \mathsf{f}(\dots, \mathsf{pred}(0))) \\
\to & \mathsf{ifelse}(\mathsf{equal}(0, 0), \mathsf{g}(\dots), \mathsf{f}(\dots, \mathsf{pred}(0))) \\
\to & \mathsf{g}(\mathsf{either}(0, \mathsf{either}(2)))
\end{array}
$$

Definition: a rule $f(\ell_1, \ldots, \ell_n) \to r$ is semi-linear if:
for each ℓ_i:
if ℓ_i is a variable then ℓ_i occurs at most once in r

What does this prohibit?

$$\begin{array}{rcl} \mathsf{either}(x, y) & \to & x \\ \mathsf{either}(x, y) & \to & y \\ \mathsf{f}(x, n) & \to & \mathsf{ifelse}(\mathsf{equal}(x, n), \mathsf{g}(x), \mathsf{g}(x, \mathsf{pred}(n))) \end{array}$$

$$\begin{array}{ll} & \mathsf{f}(\mathsf{either}(0, \mathsf{either}(2)), 1) \\ \to & \mathsf{ifelse}(\mathsf{equal}(\mathsf{either}(0, \mathsf{either}(2)), 1), \mathsf{g}(\ldots), \mathsf{f}(\ldots, \mathsf{pred}(1))) \\ \to & \mathsf{ifelse}(\mathsf{equal}(2, 1), \mathsf{g}(\ldots), \mathsf{f}(\ldots, \mathsf{pred}(1))) \\ \to^* & \mathsf{f}(\mathsf{either}(0, \mathsf{either}(2)), \mathsf{pred}(1)) \\ \to & \mathsf{f}(\mathsf{either}(0, \mathsf{either}(2)), 0) \\ \to & \mathsf{ifelse}(\mathsf{equal}(\mathsf{either}(0, \mathsf{either}(2)), 0), \mathsf{g}(\ldots), \mathsf{f}(\ldots, \mathsf{pred}(0))) \\ \to & \mathsf{ifelse}(\mathsf{equal}(0, 0), \mathsf{g}(\ldots), \mathsf{f}(\ldots, \mathsf{pred}(0))) \\ \to & \mathsf{g}(\mathsf{either}(0, \mathsf{either}(2))) \end{array}$$

Definition:

a rule $f(\ell_1, \dots, \ell_n) \to r$ is semi-linear if:
for each ℓ_i:
if ℓ_i is a variable then ℓ_i occurs at most once in r

What does this allow?

Definition:

a rule $f(\ell_1, \ldots, \ell_n) \to r$ is semi-linear if:

for each ℓ_i:

if ℓ_i is a variable then ℓ_i occurs at most once in r

What does this allow?

- duplication of variables which must be instantiated by irreducible terms

Definition:

a rule $f(\ell_1, \dots, \ell_n) \to r$ is semi-linear if:

for each ℓ_i:

if ℓ_i is a variable then ℓ_i occurs at most once in r

What does this allow?

- duplication of variables which must be instantiated by irreducible terms (due to cons-freeness)

Definition:

a rule $f(\ell_1, \dots, \ell_n) \to r$ is semi-linear if:
for each ℓ_i:
if ℓ_i is a variable then ℓ_i occurs at most once in r

What does this allow?

- duplication of variables which must be instantiated by irreducible terms (due to cons-freeness)
- passing on reducible terms but never copying

Definition:

a rule $f(\ell_1, \ldots, \ell_n) \to r$ is semi-linear if:

for each ℓ_i:

if ℓ_i is a variable then ℓ_i occurs at most once in r

What does this allow?

- duplication of variables which must be instantiated by irreducible terms (due to cons-freeness)
- passing on reducible terms but never copying (so in $\mathsf{f}(\mathsf{either}(0, \mathsf{either}(1, 2)))$ we cannot *both* test reducts *and* pass on the term)

Definition:

a rule $f(\ell_1, \dots, \ell_n) \to r$ is semi-linear if:
for each ℓ_i:
if ℓ_i is a variable then ℓ_i occurs at most once in r

What does this allow?

- duplication of variables which must be instantiated by irreducible terms (due to cons-freeness)
- passing on reducible terms but never copying (so in $\mathsf{f}(\mathsf{either}(0, \mathsf{either}(1, 2)))$ we cannot *both* test reducts *and* pass on the term)

$\Longrightarrow$ we might as well reduce subterms immediately

Definition:

a rule $f(\ell_1, \ldots, \ell_n) \to r$ is semi-linear if:
for each ℓ_i:
if ℓ_i is a variable then ℓ_i occurs at most once in r

What does this allow?

- duplication of variables which must be instantiated by irreducible terms (due to cons-freeness)
- passing on reducible terms but never copying (so in $\mathsf{f}(\mathsf{either}(0, \mathsf{either}(1, 2)))$ we cannot *both* test reducts *and* pass on the term)

$\Longrightarrow$ we might as well reduce subterms immediately

Caution:

$$\begin{array}{rcl} \mathsf{f}(x) & \to & 0 \\ \mathsf{a} & \to & \mathsf{a} \end{array}$$

Definition:

a rule $f(\ell_1, \dots, \ell_n) \to r$ is semi-linear if:
for each ℓ_i:
if ℓ_i is a variable then ℓ_i occurs at most once in r

What does this allow?

- duplication of variables which must be instantiated by irreducible terms (due to cons-freeness)
- passing on reducible terms but never copying (so in $\mathsf{f}(\mathsf{either}(0, \mathsf{either}(1, 2)))$ we cannot *both* test reducts *and* pass on the term)

$\Longrightarrow$ we might as well reduce subterms immediately

Caution:

$$\begin{array}{rcl} \mathsf{f}(x) & \to & 0 \\ \mathsf{a} & \to & \mathsf{a} \end{array}$$

$\mathsf{f}(\mathsf{a})$

Semi-Linear Term Rewriting

Definition:

a rule $f(\ell_1, \dots, \ell_n) \to r$ is semi-linear if:
for each ℓ_i:
if ℓ_i is a variable then ℓ_i occurs at most once in r

Claim:

rewriting with semi-linear rules characterises PTIME

Proof:

- same algorithm as for innermost can be expressed semi-linear
- if all rules in $\mathcal{R}$ are semi-linear
  then exists $\mathcal{R}'$ such that
  $s \to_{\mathcal{R}}^* t$ with t data if and only if $s \to_{\mathcal{R}',\text{call-by-value}}^* t$

Semi-Linear Term Rewriting

Definition:

a rule $f(\ell_1, \ldots, \ell_n) \to r$ is semi-linear if:
for each ℓ_i:
if ℓ_i is a variable then ℓ_i occurs at most once in r

Claim:

rewriting with semi-linear rules characterises PTIME

Proof:

- same algorithm as for innermost can be expressed semi-linear
- if all rules in $\mathcal{R}$ are semi-linear
  then let $\mathcal{R}' = \mathcal{R} \cup \{f(x_1, \ldots, x_n) \to \bot \mid f \text{ a function symbol}\}$
  then $s \to_{\mathcal{R}}^* t$ with t data if and only if $s \to_{\mathcal{R}',\text{call-by-value}}^* t$

Semi-Linear Term Rewriting

Definition:

a rule $f(\ell_1, \ldots, \ell_n) \to r$ is semi-linear if:
for each ℓ_i:
if ℓ_i is a variable then ℓ_i occurs at most once in r

Claim:

rewriting with confluent rules characterises PTIME

Proof:

- same algorithm as for innermost is confluent
- if all rules in $\mathcal{R}$ are confluent
then let $\mathcal{R}' = \mathcal{R} \cup \{f(x_1, \ldots, x_n) \to \bot \mid f \text{ a function symbol}\}$
then $s \to_{\mathcal{R}}^* t$ with t data if and only if $s \to_{\mathcal{R}',\text{call-by-value}}^* t$

“Constrained” Term Rewriting [Carvalho and Simonsen '14]

"Constrained" Term Rewriting [Carvalho and Simonsen '14]

Definition: there is $\mathcal{A} \subseteq \mathcal{F}$ such that:

- all rules $f(\ell_1, \ldots, \ell_n) \to r$ with $f \in \mathcal{A}$ are semi-linear
- for all rules $f(\ell_1, \ldots, \ell_n) \to r$ with $f \notin \mathcal{A}$: if ℓ_i is a variable, and $r = C[g(t_1, \ldots, D[\ell_i], \ldots, t_m)]$ then $g \in \mathcal{A}$

"Constrained" Term Rewriting [Carvalho and Simonsen '14]

Definition: there is $\mathcal{A} \subseteq \mathcal{F}$ such that:

- all rules $f(\ell_1, \ldots, \ell_n) \to r$ with $f \in \mathcal{A}$ are semi-linear
- for all rules $f(\ell_1, \ldots, \ell_n) \to r$ with $f \notin \mathcal{A}$: if ℓ_i is a variable, and $r = C[g(t_1, \ldots, D[\ell_i], \ldots, t_m)]$ then $g \in \mathcal{A}$

$\Longrightarrow$ easily adapted to a system with only semi-linear rules

"Constrained" Term Rewriting [Carvalho and Simonsen '14]

Definition: there is $\mathcal{A} \subseteq \mathcal{F}$ such that:

- all rules $f(\ell_1, \dots, \ell_n) \to r$ with $f \in \mathcal{A}$ are semi-linear
- for all rules $f(\ell_1, \dots, \ell_n) \to r$ with $f \notin \mathcal{A}$: if ℓ_i is a variable, and $r = C[g(t_1, \dots, D[\ell_i], \dots, t_m)]$ then $g \in \mathcal{A}$

$\Longrightarrow$ easily adapted to a system with only semi-linear rules

$\Longrightarrow$ easily adapted to a system on which call-by-value rewriting can be imposed

Overview

Results:

Results:

- we transposed the characterisation of PTIME in [Jones '01] to call-by-value term rewriting

Results:

- we transposed the characterisation of PTIME in [Jones '01] to call-by-value term rewriting (including non-determinism)

Results:

- we transposed the characterisation of PTIME in [Jones '01] to call-by-value term rewriting (including non-determinism)
- we showed that this characterisation can be used as a tool:

Results:

- we transposed the characterisation of PTIME in [Jones '01] to call-by-value term rewriting (including non-determinism)
- we showed that this characterisation can be used as a tool:
  - simplified proof of [Carvalho and Simonsen '14] through a transformation to call-by-value term rewriting

Results:

- we transposed the characterisation of PTIME in [Jones '01] to call-by-value term rewriting (including non-determinism)
- we showed that this characterisation can be used as a tool:
  - simplified proof of [Carvalho and Simonsen '14] through a transformation to call-by-value term rewriting
  - suitability of other constraints on term rewriting might be proved similarly

Results:

- we transposed the characterisation of PTIME in [Jones '01] to call-by-value term rewriting (including non-determinism)
- we showed that this characterisation can be used as a tool:
  - simplified proof of [Carvalho and Simonsen '14] through a transformation to call-by-value term rewriting
  - suitability of other constraints on term rewriting might be proved similarly

Future work:

Results:

- we transposed the characterisation of PTIME in [Jones '01] to call-by-value term rewriting (including non-determinism)
- we showed that this characterisation can be used as a tool:
  - simplified proof of [Carvalho and Simonsen '14] through a transformation to call-by-value term rewriting
  - suitability of other constraints on term rewriting might be proved similarly

Future work:

- extension to higher-order rewriting (transposing all of [Jones '01])

Results:

- we transposed the characterisation of PTIME in [Jones '01] to call-by-value term rewriting (including non-determinism)
- we showed that this characterisation can be used as a tool:
  - simplified proof of [Carvalho and Simonsen '14] through a transformation to call-by-value term rewriting
  - suitability of other constraints on term rewriting might be proved similarly

Future work:

- extension to higher-order rewriting (transposing all of [Jones '01])
- considering other constraints and no constraints

Results:

- we transposed the characterisation of PTIME in [Jones '01] to call-by-value term rewriting (including non-determinism)
- we showed that this characterisation can be used as a tool:
  - simplified proof of [Carvalho and Simonsen '14] through a transformation to call-by-value term rewriting
  - suitability of other constraints on term rewriting might be proved similarly

Future work:

- extension to higher-order rewriting (transposing all of [Jones '01])
- considering other constraints and no constraints (submitted to FSCD'16)

Results:

- we transposed the characterisation of PTIME in [Jones '01] to call-by-value term rewriting (including non-determinism)
- we showed that this characterisation can be used as a tool:
  - simplified proof of [Carvalho and Simonsen '14] through a transformation to call-by-value term rewriting
  - suitability of other constraints on term rewriting might be proved similarly

Future work:

- extension to higher-order rewriting (transposing all of [Jones '01])
- considering other constraints and no constraints (submitted to FSCD'16)
- finding constraints which can not be transposed to call-by-value rewriting

Results:

- we transposed the characterisation of PTIME in [Jones '01] to call-by-value term rewriting (including non-determinism)
- we showed that this characterisation can be used as a tool:
  - simplified proof of [Carvalho and Simonsen '14] through a transformation to call-by-value term rewriting
  - suitability of other constraints on term rewriting might be proved similarly

Future work:

- extension to higher-order rewriting (transposing all of [Jones '01])
- considering other constraints and no constraints (submitted to FSCD'16)
- finding constraints which can not be transposed to call-by-value rewriting

Questions?