An automata characterisation for multiple context-free languages

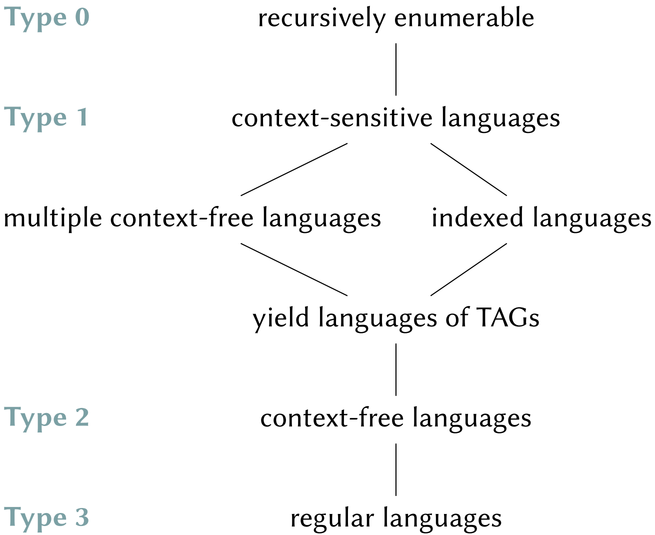
Tobias Denkinger

tobias.denkinger@tu-dresden.de

Institute of Theoretical Computer Science
Faculty of Computer Science
Technische Universität Dresden

DLT, Montréal, 2016-07-27

A set diagram of some language classes



A set diagram of some language classes

Type 0 recursively enumerable
↪Turing machines

Type 1 context-sensitive languages
↪linear bounded automata

multiple context-free languages

indexed languages
↪nested stack automata

yield languages of TAGs
↪embedded pushdown automata

Type 2 context-free languages
↪pushdown automata

Type 3 regular languages
↪finite state automata

A set diagram of some language classes

Type 0 recursively enumerable
↪Turing machines

Type 1 context-sensitive languages
↪linear bounded automata

multiple context-free languages
↪thread automata
+ automata with storage

indexed languages
↪nested stack automata

yield languages of TAGs
↪embedded pushdown automata

Type 2 context-free languages
↪pushdown automata

Type 3 regular languages
↪finite state automata

Outline

1. Multiple context-free grammars
2. Tree stack automata
3. The automata characterisation

Composition functions (example)

$$(\Sigma^*) \times (\Sigma^* \times \Sigma^*) \to (\Sigma^* \times \Sigma^*)$$

Composition functions (example)

$$(\underset{\substack{\big\downarrow\\ x_1}}{\Sigma^*}) \times (\underset{\substack{\big\downarrow\\ y_1}}{\Sigma^*} \times \underset{\substack{\big\downarrow\\ y_2}}{\Sigma^*}) \to (\Sigma^* \times \Sigma^*)$$

Composition functions (example)

$$[x_1 y_2, \mathrm{b} y_1] \colon (\underset{\substack{\downarrow \\ x_1}}{\Sigma^*}) \times (\underset{\substack{\downarrow \\ y_1}}{\Sigma^*} \times \underset{\substack{\downarrow \\ y_2}}{\Sigma^*}) \to (\Sigma^* \times \Sigma^*)$$

each variable occurs at most once

Composition functions (example)

$$[x_1 y_2, \mathrm{b} y_1] \colon (\underset{\substack{\downarrow \\ x_1}}{\Sigma^*}) \times (\underset{\substack{\downarrow \\ y_1}}{\Sigma^*} \times \underset{\substack{\downarrow \\ y_2}}{\Sigma^*}) \to (\Sigma^* \times \Sigma^*)$$

$$[x_1 y_2, \mathrm{b} y_1]\big((\alpha\gamma), (\beta, \alpha)\big) = (\alpha\gamma\alpha, \mathrm{b}\beta)$$

each variable occurs at most once

Multiple context-free grammars (MCFGs)

$$G = \big(\underbrace{\{S, A, B\}}_{\text{nonterminals}}, \underbrace{\{\mathsf{a}, \mathsf{b}, \mathsf{c}, \mathsf{d}\}}_{\text{terminals}}, \underbrace{\{S\}}_{\text{initial nts}}, \underbrace{\{\rho_1, ..., \rho_5\}}_{\text{productions}}\big)$$

productions:

$$\rho_1 = S \to [x_1 y_1 x_2 y_2](A, B)$$
$$\rho_2 = A \to [\mathsf{a}x_1, \mathsf{c}x_2](A)$$
$$\rho_3 = B \to [\mathsf{b}x_1, \mathsf{d}x_2](B)$$
$$\rho_4 = A \to [\mathsf{a}, \mathsf{c}]()$$
$$\rho_5 = B \to [\mathsf{b}, \mathsf{d}]()$$

Multiple context-free grammars (MCFGs)

$$G = (\underbrace{\{S, A, B\}}_{\text{nonterminals}}, \underbrace{\{\mathsf{a}, \mathsf{b}, \mathsf{c}, \mathsf{d}\}}_{\text{terminals}}, \underbrace{\{S\}}_{\text{initial nts}}, \underbrace{\{\rho_1, \ldots, \rho_5\}}_{\text{productions}})$$

productions:

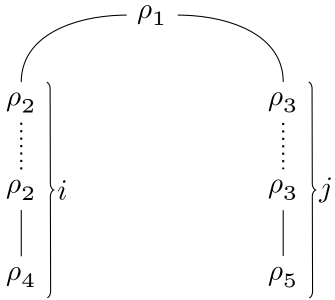
$\rho_1 = S \to [x_1 y_1 x_2 y_2](A, B)$

$\rho_2 = A \to [\mathsf{a}x_1, \mathsf{c}x_2](A)$

$\rho_3 = B \to [\mathsf{b}x_1, \mathsf{d}x_2](B)$

$\rho_4 = A \to [\mathsf{a}, \mathsf{c}]()$

$\rho_5 = B \to [\mathsf{b}, \mathsf{d}]()$



Multiple context-free grammars (MCFGs)

$$G = (\underbrace{\{S, A, B\}}_{\text{nonterminals}}, \underbrace{\{\mathrm{a}, \mathrm{b}, \mathrm{c}, \mathrm{d}\}}_{\text{terminals}}, \underbrace{\{S\}}_{\text{initial nts}}, \underbrace{\{\rho_1, \ldots, \rho_5\}}_{\text{productions}})$$

productions:

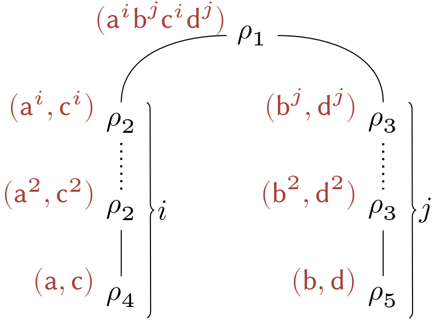
$\rho_1 = S \to [x_1 y_1 x_2 y_2](A, B)$

$\rho_2 = A \to [\mathrm{a}x_1, \mathrm{c}x_2](A)$

$\rho_3 = B \to [\mathrm{b}x_1, \mathrm{d}x_2](B)$

$\rho_4 = A \to [\mathrm{a}, \mathrm{c}]()$

$\rho_5 = B \to [\mathrm{b}, \mathrm{d}]()$



Multiple context-free grammars (MCFGs)

$$G = (\underbrace{\{S, A, B\}}_{\text{nonterminals}}, \underbrace{\{\mathrm{a}, \mathrm{b}, \mathrm{c}, \mathrm{d}\}}_{\text{terminals}}, \underbrace{\{S\}}_{\text{initial nts}}, \underbrace{\{\rho_1, \ldots, \rho_5\}}_{\text{productions}})$$

productions:

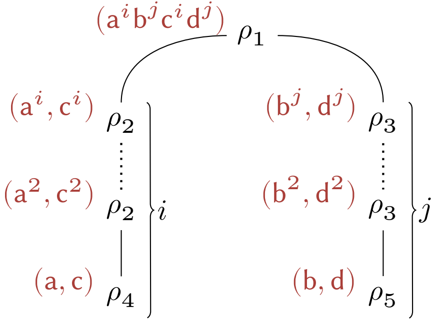
$\rho_1 = S \to [x_1 y_1 x_2 y_2](A, B)$

$\rho_2 = A \to [\mathrm{a}x_1, \mathrm{c}x_2](A)$

$\rho_3 = B \to [\mathrm{b}x_1, \mathrm{d}x_2](B)$

$\rho_4 = A \to [\mathrm{a}, \mathrm{c}]()$

$\rho_5 = B \to [\mathrm{b}, \mathrm{d}]()$



$$L(G) = \{\mathrm{a}^i \mathrm{b}^j \mathrm{c}^i \mathrm{d}^j \mid i, j \geq 1\}$$

Multiple context-free grammars (MCFGs)

$$G = (\underbrace{\{S, A, B\}}_{\text{nonterminals}}, \underbrace{\{\mathrm{a}, \mathrm{b}, \mathrm{c}, \mathrm{d}\}}_{\text{terminals}}, \underbrace{\{S\}}_{\text{initial nts}}, \underbrace{\{\rho_1, \ldots, \rho_5\}}_{\text{productions}})$$

productions:

$\rho_1 = S \to [x_1 y_1 x_2 y_2](A, B)$

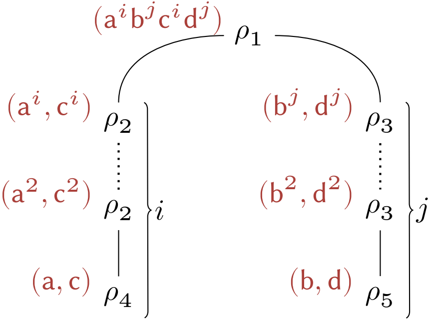
$\rho_2 = A \to [\mathrm{a}x_1, \mathrm{c}x_2](A)$

$\rho_3 = B \to [\mathrm{b}x_1, \mathrm{d}x_2](B)$

$\rho_4 = A \to [\mathrm{a}, \mathrm{c}]()$

$\rho_5 = B \to [\mathrm{b}, \mathrm{d}]()$

fan-out$(G) = 2$



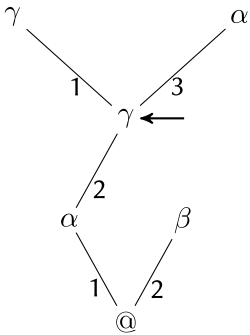
$$L(G) = \{\mathrm{a}^i\mathrm{b}^j\mathrm{c}^i\mathrm{d}^j \mid i, j \geq 1\}$$

The tree stack idea from Villemonte de la Clergerie (2002a,b)

data type $\mathrm{TS}(\Gamma)$

- stack symbols Γ

example



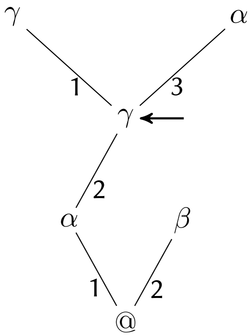
$\in \mathrm{TS}(\Gamma)$

The tree stack idea from Villemonte de la Clergerie (2002a,b)

data type $\mathrm{TS}(\mathit{\Gamma})$

- stack symbols $\mathit{\Gamma}$
- partial function $\xi\colon \mathbb{N}_+^* \to \mathit{\Gamma} \uplus \{@\}$
- stack pointer $p \in \mathbb{N}_+^*$ from the domain of ξ

example

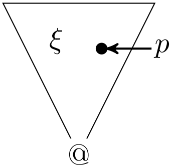


$\in \mathrm{TS}(\mathit{\Gamma})$

The tree stack idea from Villemonte de la Clergerie (2002a,b)

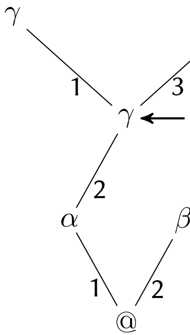
data type $\mathrm{TS}(\Gamma)$

- stack symbols Γ
- partial function
  $\xi\colon \mathbb{N}_+^* \to \Gamma \uplus \{@\}$
- stack pointer $p \in \mathbb{N}_+^*$
  from the domain of ξ
- domain of ξ prefix-closed



- @ exactly at the root

example

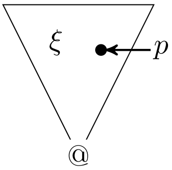


$\in \mathrm{TS}(\Gamma)$

The tree stack idea from Villemonte de la Clergerie (2002a,b)

data type $\mathrm{TS}(\Gamma)$

- stack symbols Γ
- partial function
  $\xi\colon \mathbb{N}_+^* \to \Gamma \uplus \{@\}$
- stack pointer $p \in \mathbb{N}_+^*$
  from the domain of ξ
- domain of ξ prefix-closed



- @ exactly at the root

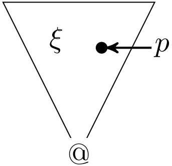
predicates **(unary)**

$$\mathrm{true} = \mathrm{TS}(\Gamma)$$
$$\mathrm{bot} = \{(\xi, p) \in \mathrm{TS}(\Gamma) \mid p = \varepsilon\}$$
$$\mathrm{eq}(\gamma) = \{(\xi, p) \in \mathrm{TS}(\Gamma) \mid \xi(p) = \gamma\}$$

The tree stack idea from Villemonte de la Clergerie (2002a,b)

data type $\mathrm{TS}(\Gamma)$

- stack symbols Γ
- partial function $\xi\colon \mathbb{N}_+^* \to \Gamma \uplus \{@\}$
- stack pointer $p \in \mathbb{N}_+^*$ from the domain of ξ
- domain of ξ prefix-closed



- @ exactly at the root

predicates **(unary)**

$$\mathrm{true} = \mathrm{TS}(\Gamma)$$

$$\mathrm{bot} = \{(\xi, p) \in \mathrm{TS}(\Gamma) \mid p = \varepsilon\}$$

$$\mathrm{eq}(\gamma) = \{(\xi, p) \in \mathrm{TS}(\Gamma) \mid \xi(p) = \gamma\}$$

instructions **(possibly partial)**

id

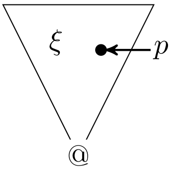
$\mathrm{set}(\gamma)$: $\xi(p) := \gamma$
(only if $p \neq \varepsilon$)

up_i: move stack pointer to i-th child
(only if $\xi(pi)$ is defined)

The tree stack idea from Villemonte de la Clergerie (2002a,b)

data type $\mathrm{TS}(\Gamma)$

- stack symbols Γ
- partial function $\xi\colon \mathbb{N}_+^* \to \Gamma \uplus \{@\}$
- stack pointer $p \in \mathbb{N}_+^*$ from the domain of ξ
- domain of ξ prefix-closed



- @ exactly at the root

predicates **(unary)**

$\mathrm{true} = \mathrm{TS}(\Gamma)$

$\mathrm{bot} = \{(\xi, p) \in \mathrm{TS}(\Gamma) \mid p = \varepsilon\}$

$\mathrm{eq}(\gamma) = \{(\xi, p) \in \mathrm{TS}(\Gamma) \mid \xi(p) = \gamma\}$

instructions **(possibly partial)**

id

$\mathrm{set}(\gamma)$: $\xi(p) := \gamma$
(only if $p \neq \varepsilon$)

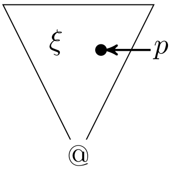
up_i: move stack pointer to i-th child
(only if $\xi(pi)$ is defined)

$\mathrm{push}_i(\gamma)$: push γ to the i-th child
(only if $\xi(pi)$ is undefined)

The tree stack idea from Villemonte de la Clergerie (2002a,b)

data type $\mathrm{TS}(\Gamma)$

- stack symbols Γ
- partial function $\xi\colon \mathbb{N}_+^* \to \Gamma \uplus \{@\}$
- stack pointer $p \in \mathbb{N}_+^*$ from the domain of ξ
- domain of ξ prefix-closed



- @ exactly at the root

***predicates* (unary)**

$\mathrm{true} = \mathrm{TS}(\Gamma)$

$\mathrm{bot} = \{(\xi, p) \in \mathrm{TS}(\Gamma) \mid p = \varepsilon\}$

$\mathrm{eq}(\gamma) = \{(\xi, p) \in \mathrm{TS}(\Gamma) \mid \xi(p) = \gamma\}$

***instructions* (possibly partial)**

id

$\mathrm{set}(\gamma)$: $\xi(p) := \gamma$
(only if $p \neq \varepsilon$)

up_i: move stack pointer to i-th child
(only if $\xi(pi)$ is defined)

$\mathrm{push}_i(\gamma)$: push γ to the i-th child
(only if $\xi(pi)$ is undefined)

down: move stack pointer to parent

Tree-stack automata (TSA) as automata with storage, Scott (1967)

$$
\begin{aligned}
\tau_1 &= (1, \mathrm{a}, && 2) \\
\tau_2 &= (2, \mathrm{a}, && 2) \\
\tau_3 &= (2, \varepsilon, && 3) \\
\tau_4 &= (3, \varepsilon, && 3) \\
\tau_5 &= (3, \mathrm{b}, && 4) \\
\tau_6 &= (4, \mathrm{b}, && 4) \\
\tau_7 &= (4, \varepsilon, && 5) \\
\tau_8 &= (5, \varepsilon, && 5) \\
\tau_9 &= (5, \varepsilon, && 6) \\
\tau_{10} &= (6, \mathrm{c}, && 6) \\
\tau_{11} &= (6, \varepsilon, && 7) \\
\tau_{12} &= (7, \varepsilon, && 7) \\
\tau_{13} &= (7, \varepsilon, && 8) \\
\tau_{14} &= (8, \mathrm{d}, && 8) \\
\tau_{15} &= (8, \varepsilon, && 9)
\end{aligned}
$$

Tree-stack automata (TSA) as automata with storage, Scott (1967)

instructions

$$
\begin{aligned}
\tau_1 &= (1, \mathrm{a}, \boxed{\mathrm{bot}}, \boxed{\mathrm{push}_1(*)}, 2)\\
\tau_2 &= (2, \mathrm{a}, \mathrm{true}, \mathrm{push}_1(*), 2)\\
\tau_3 &= (2, \varepsilon, \mathrm{true}, \mathrm{push}_1(\#), 3)\\
\tau_4 &= (3, \varepsilon, \mathrm{true}, \mathrm{down}, 3)\\
\tau_5 &= (3, \mathrm{b}, \mathrm{bot}, \mathrm{push}_2(*), 4)\\
\tau_6 &= (4, \mathrm{b}, \mathrm{true}, \mathrm{push}_1(*), 4)\\
\tau_7 &= (4, \varepsilon, \mathrm{true}, \mathrm{push}_1(\#), 5)\\
\tau_8 &= (5, \varepsilon, \mathrm{true}, \mathrm{down}, 5)\\
\tau_9 &= (5, \varepsilon, \mathrm{bot}, \mathrm{up}_1, 6)\\
\tau_{10} &= (6, \mathrm{c}, \mathrm{eq}(*), \mathrm{up}_1, 6)\\
\tau_{11} &= (6, \varepsilon, \mathrm{eq}(\#), \mathrm{down}, 7)\\
\tau_{12} &= (7, \varepsilon, \mathrm{eq}(*), \mathrm{down}, 7)\\
\tau_{13} &= (7, \varepsilon, \mathrm{bot}, \mathrm{up}_2, 8)\\
\tau_{14} &= (8, \mathrm{d}, \mathrm{eq}(*), \mathrm{up}_1, 8)\\
\tau_{15} &= (8, \varepsilon, \mathrm{eq}(\#), \mathrm{id}, 9)
\end{aligned}
$$

predicates

Tree-stack automata (TSA) as automata with storage, Scott (1967)

$$
\begin{aligned}
\tau_1 &= (1, \mathrm{a}, \mathrm{bot}, \mathrm{push}_1(*), 2)\\
\tau_2 &= (2, \mathrm{a}, \mathrm{true}, \mathrm{push}_1(*), 2)\\
\tau_3 &= (2, \varepsilon, \mathrm{true}, \mathrm{push}_1(\#), 3)\\
\tau_4 &= (3, \varepsilon, \mathrm{true}, \mathrm{down}, 3)\\
\tau_5 &= (3, \mathrm{b}, \mathrm{bot}, \mathrm{push}_2(*), 4)\\
\tau_6 &= (4, \mathrm{b}, \mathrm{true}, \mathrm{push}_1(*), 4)\\
\tau_7 &= (4, \varepsilon, \mathrm{true}, \mathrm{push}_1(\#), 5)\\
\tau_8 &= (5, \varepsilon, \mathrm{true}, \mathrm{down}, 5)\\
\tau_9 &= (5, \varepsilon, \mathrm{bot}, \mathrm{up}_1, 6)\\
\tau_{10} &= (6, \mathrm{c}, \mathrm{eq}(*), \mathrm{up}_1, 6)\\
\tau_{11} &= (6, \varepsilon, \mathrm{eq}(\#), \mathrm{down}, 7)\\
\tau_{12} &= (7, \varepsilon, \mathrm{eq}(*), \mathrm{down}, 7)\\
\tau_{13} &= (7, \varepsilon, \mathrm{bot}, \mathrm{up}_2, 8)\\
\tau_{14} &= (8, \mathrm{d}, \mathrm{eq}(*), \mathrm{up}_1, 8)\\
\tau_{15} &= (8, \varepsilon, \mathrm{eq}(\#), \mathrm{id}, 9)
\end{aligned}
$$

Tree-stack automata (TSA) as automata with storage, Scott (1967)

$$
\begin{aligned}
\tau_1 &= (1, \mathrm{a}, \mathrm{bot}, \mathrm{push}_1(*), 2)\\
\tau_2 &= (2, \mathrm{a}, \mathrm{true}, \mathrm{push}_1(*), 2)\\
\tau_3 &= (2, \varepsilon, \mathrm{true}, \mathrm{push}_1(\#), 3)\\
\tau_4 &= (3, \varepsilon, \mathrm{true}, \mathrm{down}, 3)\\
\tau_5 &= (3, \mathrm{b}, \mathrm{bot}, \mathrm{push}_2(*), 4)\\
\tau_6 &= (4, \mathrm{b}, \mathrm{true}, \mathrm{push}_1(*), 4)\\
\tau_7 &= (4, \varepsilon, \mathrm{true}, \mathrm{push}_1(\#), 5)\\
\tau_8 &= (5, \varepsilon, \mathrm{true}, \mathrm{down}, 5)\\
\tau_9 &= (5, \varepsilon, \mathrm{bot}, \mathrm{up}_1, 6)\\
\tau_{10} &= (6, \mathrm{c}, \mathrm{eq}(*), \mathrm{up}_1, 6)\\
\tau_{11} &= (6, \varepsilon, \mathrm{eq}(\#), \mathrm{down}, 7)\\
\tau_{12} &= (7, \varepsilon, \mathrm{eq}(*), \mathrm{down}, 7)\\
\tau_{13} &= (7, \varepsilon, \mathrm{bot}, \mathrm{up}_2, 8)\\
\tau_{14} &= (8, \mathrm{d}, \mathrm{eq}(*), \mathrm{up}_1, 8)\\
\tau_{15} &= (8, \varepsilon, \mathrm{eq}(\#), \mathrm{id}, 9)
\end{aligned}
$$

recognises $\{\mathrm{a}^i \mathrm{b}^j \mathrm{c}^i \mathrm{d}^j \mid i, j \leq 1\}$

Tree-stack automata (TSA) as automata with storage, Scott (1967)

$$
\begin{aligned}
\tau_1 &= (1, \mathrm{a}, \mathrm{bot}, \mathrm{push}_1(*), 2)\\
\tau_2 &= (2, \mathrm{a}, \mathrm{true}, \mathrm{push}_1(*), 2)\\
\tau_3 &= (2, \varepsilon, \mathrm{true}, \mathrm{push}_1(\#), 3)\\
\tau_4 &= (3, \varepsilon, \mathrm{true}, \mathrm{down}, 3)\\
\tau_5 &= (3, \mathrm{b}, \mathrm{bot}, \mathrm{push}_2(*), 4)\\
\tau_6 &= (4, \mathrm{b}, \mathrm{true}, \mathrm{push}_1(*), 4)\\
\tau_7 &= (4, \varepsilon, \mathrm{true}, \mathrm{push}_1(\#), 5)\\
\tau_8 &= (5, \varepsilon, \mathrm{true}, \mathrm{down}, 5)\\
\tau_9 &= (5, \varepsilon, \mathrm{bot}, \mathrm{up}_1, 6)\\
\tau_{10} &= (6, \mathrm{c}, \mathrm{eq}(*), \mathrm{up}_1, 6)\\
\tau_{11} &= (6, \varepsilon, \mathrm{eq}(\#), \mathrm{down}, 7)\\
\tau_{12} &= (7, \varepsilon, \mathrm{eq}(*), \mathrm{down}, 7)\\
\tau_{13} &= (7, \varepsilon, \mathrm{bot}, \mathrm{up}_2, 8)\\
\tau_{14} &= (8, \mathrm{d}, \mathrm{eq}(*), \mathrm{up}_1, 8)\\
\tau_{15} &= (8, \varepsilon, \mathrm{eq}(\#), \mathrm{id}, 9)
\end{aligned}
$$

recognises $\{\mathrm{a}^i \mathrm{b}^j \mathrm{c}^i \mathrm{d}^j \mid i, j \leq 1\}$

state: 1 *stack:* @ $\leftarrow$

input tape:

a	a	b	c	c	d

run:

Tree-stack automata (TSA) as automata with storage, Scott (1967)

$$\boxed{\tau_1 = (1, \mathrm{a}, \mathrm{bot}, \mathrm{push}_1(*), 2)}$$
$$\tau_2 = (2, \mathrm{a}, \mathrm{true}, \mathrm{push}_1(*), 2)$$
$$\tau_3 = (2, \varepsilon, \mathrm{true}, \mathrm{push}_1(\#), 3)$$
$$\tau_4 = (3, \varepsilon, \mathrm{true}, \mathrm{down}, 3)$$
$$\tau_5 = (3, \mathrm{b}, \mathrm{bot}, \mathrm{push}_2(*), 4)$$
$$\tau_6 = (4, \mathrm{b}, \mathrm{true}, \mathrm{push}_1(*), 4)$$
$$\tau_7 = (4, \varepsilon, \mathrm{true}, \mathrm{push}_1(\#), 5)$$
$$\tau_8 = (5, \varepsilon, \mathrm{true}, \mathrm{down}, 5)$$
$$\tau_9 = (5, \varepsilon, \mathrm{bot}, \mathrm{up}_1, 6)$$
$$\tau_{10} = (6, \mathrm{c}, \mathrm{eq}(*), \mathrm{up}_1, 6)$$
$$\tau_{11} = (6, \varepsilon, \mathrm{eq}(\#), \mathrm{down}, 7)$$
$$\tau_{12} = (7, \varepsilon, \mathrm{eq}(*), \mathrm{down}, 7)$$
$$\tau_{13} = (7, \varepsilon, \mathrm{bot}, \mathrm{up}_2, 8)$$
$$\tau_{14} = (8, \mathrm{d}, \mathrm{eq}(*), \mathrm{up}_1, 8)$$
$$\tau_{15} = (8, \varepsilon, \mathrm{eq}(\#), \mathrm{id}, 9)$$

recognises $\{\mathrm{a}^i \mathrm{b}^j \mathrm{c}^i \mathrm{d}^j \mid i, j \leq 1\}$

state: 1 *stack:* @ $\leftarrow$

input tape:

a	a	b	c	c	d

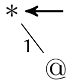
run:

Tree-stack automata (TSA) as automata with storage, Scott (1967)

$$
\begin{aligned}
\tau_1 &= (1, \mathrm{a}, \mathrm{bot}, \mathrm{push}_1(*), 2)\\
\tau_2 &= (2, \mathrm{a}, \mathrm{true}, \mathrm{push}_1(*), 2)\\
\tau_3 &= (2, \varepsilon, \mathrm{true}, \mathrm{push}_1(\#), 3)\\
\tau_4 &= (3, \varepsilon, \mathrm{true}, \mathrm{down}, 3)\\
\tau_5 &= (3, \mathrm{b}, \mathrm{bot}, \mathrm{push}_2(*), 4)\\
\tau_6 &= (4, \mathrm{b}, \mathrm{true}, \mathrm{push}_1(*), 4)\\
\tau_7 &= (4, \varepsilon, \mathrm{true}, \mathrm{push}_1(\#), 5)\\
\tau_8 &= (5, \varepsilon, \mathrm{true}, \mathrm{down}, 5)\\
\tau_9 &= (5, \varepsilon, \mathrm{bot}, \mathrm{up}_1, 6)\\
\tau_{10} &= (6, \mathrm{c}, \mathrm{eq}(*), \mathrm{up}_1, 6)\\
\tau_{11} &= (6, \varepsilon, \mathrm{eq}(\#), \mathrm{down}, 7)\\
\tau_{12} &= (7, \varepsilon, \mathrm{eq}(*), \mathrm{down}, 7)\\
\tau_{13} &= (7, \varepsilon, \mathrm{bot}, \mathrm{up}_2, 8)\\
\tau_{14} &= (8, \mathrm{d}, \mathrm{eq}(*), \mathrm{up}_1, 8)\\
\tau_{15} &= (8, \varepsilon, \mathrm{eq}(\#), \mathrm{id}, 9)
\end{aligned}
$$

recognises $\{\mathrm{a}^i \mathrm{b}^j \mathrm{c}^i \mathrm{d}^j \mid i, j \leq 1\}$

state: 2 *stack:*



input tape:

a	a	b	c	c	d

run:

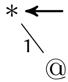
τ_1

Tree-stack automata (TSA) as automata with storage, Scott (1967)

$$
\begin{aligned}
\tau_1 &= (1, \mathrm{a}, \mathrm{bot}, \mathrm{push}_1(*), 2)\\
\tau_2 &= (2, \mathrm{a}, \mathrm{true}, \mathrm{push}_1(*), 2)\\
\tau_3 &= (2, \varepsilon, \mathrm{true}, \mathrm{push}_1(\#), 3)\\
\tau_4 &= (3, \varepsilon, \mathrm{true}, \mathrm{down}, 3)\\
\tau_5 &= (3, \mathrm{b}, \mathrm{bot}, \mathrm{push}_2(*), 4)\\
\tau_6 &= (4, \mathrm{b}, \mathrm{true}, \mathrm{push}_1(*), 4)\\
\tau_7 &= (4, \varepsilon, \mathrm{true}, \mathrm{push}_1(\#), 5)\\
\tau_8 &= (5, \varepsilon, \mathrm{true}, \mathrm{down}, 5)\\
\tau_9 &= (5, \varepsilon, \mathrm{bot}, \mathrm{up}_1, 6)\\
\tau_{10} &= (6, \mathrm{c}, \mathrm{eq}(*), \mathrm{up}_1, 6)\\
\tau_{11} &= (6, \varepsilon, \mathrm{eq}(\#), \mathrm{down}, 7)\\
\tau_{12} &= (7, \varepsilon, \mathrm{eq}(*), \mathrm{down}, 7)\\
\tau_{13} &= (7, \varepsilon, \mathrm{bot}, \mathrm{up}_2, 8)\\
\tau_{14} &= (8, \mathrm{d}, \mathrm{eq}(*), \mathrm{up}_1, 8)\\
\tau_{15} &= (8, \varepsilon, \mathrm{eq}(\#), \mathrm{id}, 9)
\end{aligned}
$$

recognises $\{\mathrm{a}^i \mathrm{b}^j \mathrm{c}^i \mathrm{d}^j \mid i, j \leq 1\}$

state: 2 *stack:*



input tape:

a	a	b	c	c	d

run:

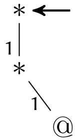
τ_1

Tree-stack automata (TSA) as automata with storage, Scott (1967)

$$
\begin{aligned}
\tau_1 &= (1, \mathrm{a}, \mathrm{bot}, \mathrm{push}_1(*), 2)\\
\tau_2 &= (2, \mathrm{a}, \mathrm{true}, \mathrm{push}_1(*), 2)\\
\tau_3 &= (2, \varepsilon, \mathrm{true}, \mathrm{push}_1(\#), 3)\\
\tau_4 &= (3, \varepsilon, \mathrm{true}, \mathrm{down}, 3)\\
\tau_5 &= (3, \mathrm{b}, \mathrm{bot}, \mathrm{push}_2(*), 4)\\
\tau_6 &= (4, \mathrm{b}, \mathrm{true}, \mathrm{push}_1(*), 4)\\
\tau_7 &= (4, \varepsilon, \mathrm{true}, \mathrm{push}_1(\#), 5)\\
\tau_8 &= (5, \varepsilon, \mathrm{true}, \mathrm{down}, 5)\\
\tau_9 &= (5, \varepsilon, \mathrm{bot}, \mathrm{up}_1, 6)\\
\tau_{10} &= (6, \mathrm{c}, \mathrm{eq}(*), \mathrm{up}_1, 6)\\
\tau_{11} &= (6, \varepsilon, \mathrm{eq}(\#), \mathrm{down}, 7)\\
\tau_{12} &= (7, \varepsilon, \mathrm{eq}(*), \mathrm{down}, 7)\\
\tau_{13} &= (7, \varepsilon, \mathrm{bot}, \mathrm{up}_2, 8)\\
\tau_{14} &= (8, \mathrm{d}, \mathrm{eq}(*), \mathrm{up}_1, 8)\\
\tau_{15} &= (8, \varepsilon, \mathrm{eq}(\#), \mathrm{id}, 9)
\end{aligned}
$$

recognises $\{\mathrm{a}^i \mathrm{b}^j \mathrm{c}^i \mathrm{d}^j \mid i, j \leq 1\}$

state: 2 *stack:*



input tape:

a	a	b	c	c	d

run:

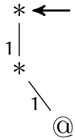
$\tau_1 \tau_2$

Tree-stack automata (TSA) as automata with storage, Scott (1967)

$$
\begin{array}{lll}
\tau_1 &= (1, \mathrm{a}, \mathrm{bot}, \mathrm{push}_1(*), 2) \\
\tau_2 &= (2, \mathrm{a}, \mathrm{true}, \mathrm{push}_1(*), 2) \\
\tau_3 &= (2, \varepsilon, \mathrm{true}, \mathrm{push}_1(\#), 3) \\
\tau_4 &= (3, \varepsilon, \mathrm{true}, \mathrm{down}, 3) \\
\tau_5 &= (3, \mathrm{b}, \mathrm{bot}, \mathrm{push}_2(*), 4) \\
\tau_6 &= (4, \mathrm{b}, \mathrm{true}, \mathrm{push}_1(*), 4) \\
\tau_7 &= (4, \varepsilon, \mathrm{true}, \mathrm{push}_1(\#), 5) \\
\tau_8 &= (5, \varepsilon, \mathrm{true}, \mathrm{down}, 5) \\
\tau_9 &= (5, \varepsilon, \mathrm{bot}, \mathrm{up}_1, 6) \\
\tau_{10} &= (6, \mathrm{c}, \mathrm{eq}(*), \mathrm{up}_1, 6) \\
\tau_{11} &= (6, \varepsilon, \mathrm{eq}(\#), \mathrm{down}, 7) \\
\tau_{12} &= (7, \varepsilon, \mathrm{eq}(*), \mathrm{down}, 7) \\
\tau_{13} &= (7, \varepsilon, \mathrm{bot}, \mathrm{up}_2, 8) \\
\tau_{14} &= (8, \mathrm{d}, \mathrm{eq}(*), \mathrm{up}_1, 8) \\
\tau_{15} &= (8, \varepsilon, \mathrm{eq}(\#), \mathrm{id}, 9)
\end{array}
$$

recognises $\{\mathrm{a}^i \mathrm{b}^j \mathrm{c}^i \mathrm{d}^j \mid i, j \leq 1\}$

state: 2 *stack:*



input tape:

a	a	b	c	c	d

run:

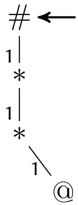
$\tau_1 \tau_2$

Tree-stack automata (TSA) as automata with storage, Scott (1967)

$$
\begin{aligned}
\tau_1 &= (1, \mathrm{a}, \mathrm{bot}, \mathrm{push}_1(*), 2)\\
\tau_2 &= (2, \mathrm{a}, \mathrm{true}, \mathrm{push}_1(*), 2)\\
\tau_3 &= (2, \varepsilon, \mathrm{true}, \mathrm{push}_1(\#), 3)\\
\tau_4 &= (3, \varepsilon, \mathrm{true}, \mathrm{down}, 3)\\
\tau_5 &= (3, \mathrm{b}, \mathrm{bot}, \mathrm{push}_2(*), 4)\\
\tau_6 &= (4, \mathrm{b}, \mathrm{true}, \mathrm{push}_1(*), 4)\\
\tau_7 &= (4, \varepsilon, \mathrm{true}, \mathrm{push}_1(\#), 5)\\
\tau_8 &= (5, \varepsilon, \mathrm{true}, \mathrm{down}, 5)\\
\tau_9 &= (5, \varepsilon, \mathrm{bot}, \mathrm{up}_1, 6)\\
\tau_{10} &= (6, \mathrm{c}, \mathrm{eq}(*), \mathrm{up}_1, 6)\\
\tau_{11} &= (6, \varepsilon, \mathrm{eq}(\#), \mathrm{down}, 7)\\
\tau_{12} &= (7, \varepsilon, \mathrm{eq}(*), \mathrm{down}, 7)\\
\tau_{13} &= (7, \varepsilon, \mathrm{bot}, \mathrm{up}_2, 8)\\
\tau_{14} &= (8, \mathrm{d}, \mathrm{eq}(*), \mathrm{up}_1, 8)\\
\tau_{15} &= (8, \varepsilon, \mathrm{eq}(\#), \mathrm{id}, 9)
\end{aligned}
$$

recognises $\{\mathrm{a}^i \mathrm{b}^j \mathrm{c}^i \mathrm{d}^j \mid i, j \leq 1\}$

state: 3 *stack:*



input tape:

a	a	b	c	c	d

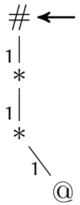
run:

$\tau_1 \tau_2 \tau_3$

Tree-stack automata (TSA) as automata with storage, Scott (1967)

$$
\begin{aligned}
\tau_1 &= (1, \mathrm{a}, \mathrm{bot}, \mathrm{push}_1(*), 2)\\
\tau_2 &= (2, \mathrm{a}, \mathrm{true}, \mathrm{push}_1(*), 2)\\
\tau_3 &= (2, \varepsilon, \mathrm{true}, \mathrm{push}_1(\#), 3)\\
&\boxed{\tau_4 = (3, \varepsilon, \mathrm{true}, \mathrm{down}, 3)}\\
\tau_5 &= (3, \mathrm{b}, \mathrm{bot}, \mathrm{push}_2(*), 4)\\
\tau_6 &= (4, \mathrm{b}, \mathrm{true}, \mathrm{push}_1(*), 4)\\
\tau_7 &= (4, \varepsilon, \mathrm{true}, \mathrm{push}_1(\#), 5)\\
\tau_8 &= (5, \varepsilon, \mathrm{true}, \mathrm{down}, 5)\\
\tau_9 &= (5, \varepsilon, \mathrm{bot}, \mathrm{up}_1, 6)\\
\tau_{10} &= (6, \mathrm{c}, \mathrm{eq}(*), \mathrm{up}_1, 6)\\
\tau_{11} &= (6, \varepsilon, \mathrm{eq}(\#), \mathrm{down}, 7)\\
\tau_{12} &= (7, \varepsilon, \mathrm{eq}(*), \mathrm{down}, 7)\\
\tau_{13} &= (7, \varepsilon, \mathrm{bot}, \mathrm{up}_2, 8)\\
\tau_{14} &= (8, \mathrm{d}, \mathrm{eq}(*), \mathrm{up}_1, 8)\\
\tau_{15} &= (8, \varepsilon, \mathrm{eq}(\#), \mathrm{id}, 9)
\end{aligned}
$$

recognises $\{\mathrm{a}^i \mathrm{b}^j \mathrm{c}^i \mathrm{d}^j \mid i, j \leq 1\}$



state: 3 *stack:*

input tape:

a	a	b	c	c	d

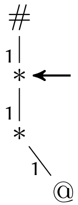
run:

$\tau_1 \tau_2 \tau_3$

Tree-stack automata (TSA) as automata with storage, Scott (1967)

$$
\begin{aligned}
\tau_1 &= (1, \mathrm{a}, \mathrm{bot}, \mathrm{push}_1(*), 2)\\
\tau_2 &= (2, \mathrm{a}, \mathrm{true}, \mathrm{push}_1(*), 2)\\
\tau_3 &= (2, \varepsilon, \mathrm{true}, \mathrm{push}_1(\#), 3)\\
\tau_4 &= (3, \varepsilon, \mathrm{true}, \mathrm{down}, 3)\\
\tau_5 &= (3, \mathrm{b}, \mathrm{bot}, \mathrm{push}_2(*), 4)\\
\tau_6 &= (4, \mathrm{b}, \mathrm{true}, \mathrm{push}_1(*), 4)\\
\tau_7 &= (4, \varepsilon, \mathrm{true}, \mathrm{push}_1(\#), 5)\\
\tau_8 &= (5, \varepsilon, \mathrm{true}, \mathrm{down}, 5)\\
\tau_9 &= (5, \varepsilon, \mathrm{bot}, \mathrm{up}_1, 6)\\
\tau_{10} &= (6, \mathrm{c}, \mathrm{eq}(*), \mathrm{up}_1, 6)\\
\tau_{11} &= (6, \varepsilon, \mathrm{eq}(\#), \mathrm{down}, 7)\\
\tau_{12} &= (7, \varepsilon, \mathrm{eq}(*), \mathrm{down}, 7)\\
\tau_{13} &= (7, \varepsilon, \mathrm{bot}, \mathrm{up}_2, 8)\\
\tau_{14} &= (8, \mathrm{d}, \mathrm{eq}(*), \mathrm{up}_1, 8)\\
\tau_{15} &= (8, \varepsilon, \mathrm{eq}(\#), \mathrm{id}, 9)
\end{aligned}
$$

recognises $\{\mathrm{a}^i \mathrm{b}^j \mathrm{c}^i \mathrm{d}^j \mid i, j \leq 1\}$



state: 3 *stack:*

input tape:

a	a	b	c	c	d

run:

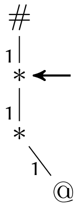
$\tau_1 \tau_2 \tau_3 \tau_4$

Tree-stack automata (TSA) as automata with storage, Scott (1967)

$$
\begin{array}{l}
\tau_1 = (1, \mathrm{a}, \mathrm{bot}, \mathrm{push}_1(*), 2)\\
\tau_2 = (2, \mathrm{a}, \mathrm{true}, \mathrm{push}_1(*), 2)\\
\tau_3 = (2, \varepsilon, \mathrm{true}, \mathrm{push}_1(\#), 3)\\
\boxed{\tau_4 = (3, \varepsilon, \mathrm{true}, \mathrm{down}, 3)}\\
\tau_5 = (3, \mathrm{b}, \mathrm{bot}, \mathrm{push}_2(*), 4)\\
\tau_6 = (4, \mathrm{b}, \mathrm{true}, \mathrm{push}_1(*), 4)\\
\tau_7 = (4, \varepsilon, \mathrm{true}, \mathrm{push}_1(\#), 5)\\
\tau_8 = (5, \varepsilon, \mathrm{true}, \mathrm{down}, 5)\\
\tau_9 = (5, \varepsilon, \mathrm{bot}, \mathrm{up}_1, 6)\\
\tau_{10} = (6, \mathrm{c}, \mathrm{eq}(*), \mathrm{up}_1, 6)\\
\tau_{11} = (6, \varepsilon, \mathrm{eq}(\#), \mathrm{down}, 7)\\
\tau_{12} = (7, \varepsilon, \mathrm{eq}(*), \mathrm{down}, 7)\\
\tau_{13} = (7, \varepsilon, \mathrm{bot}, \mathrm{up}_2, 8)\\
\tau_{14} = (8, \mathrm{d}, \mathrm{eq}(*), \mathrm{up}_1, 8)\\
\tau_{15} = (8, \varepsilon, \mathrm{eq}(\#), \mathrm{id}, 9)
\end{array}
$$

recognises $\{\mathrm{a}^i \mathrm{b}^j \mathrm{c}^i \mathrm{d}^j \mid i, j \leq 1\}$

state: 3 *stack:*



input tape:

a	a	b	c	c	d

run:

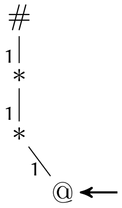
$\tau_1 \tau_2 \tau_3 \tau_4$

Tree-stack automata (TSA) as automata with storage, Scott (1967)

$$
\begin{aligned}
\tau_1 &= (1, \mathrm{a}, \mathrm{bot}, \mathrm{push}_1(*), 2)\\
\tau_2 &= (2, \mathrm{a}, \mathrm{true}, \mathrm{push}_1(*), 2)\\
\tau_3 &= (2, \varepsilon, \mathrm{true}, \mathrm{push}_1(\#), 3)\\
\tau_4 &= (3, \varepsilon, \mathrm{true}, \mathrm{down}, 3)\\
\tau_5 &= (3, \mathrm{b}, \mathrm{bot}, \mathrm{push}_2(*), 4)\\
\tau_6 &= (4, \mathrm{b}, \mathrm{true}, \mathrm{push}_1(*), 4)\\
\tau_7 &= (4, \varepsilon, \mathrm{true}, \mathrm{push}_1(\#), 5)\\
\tau_8 &= (5, \varepsilon, \mathrm{true}, \mathrm{down}, 5)\\
\tau_9 &= (5, \varepsilon, \mathrm{bot}, \mathrm{up}_1, 6)\\
\tau_{10} &= (6, \mathrm{c}, \mathrm{eq}(*), \mathrm{up}_1, 6)\\
\tau_{11} &= (6, \varepsilon, \mathrm{eq}(\#), \mathrm{down}, 7)\\
\tau_{12} &= (7, \varepsilon, \mathrm{eq}(*), \mathrm{down}, 7)\\
\tau_{13} &= (7, \varepsilon, \mathrm{bot}, \mathrm{up}_2, 8)\\
\tau_{14} &= (8, \mathrm{d}, \mathrm{eq}(*), \mathrm{up}_1, 8)\\
\tau_{15} &= (8, \varepsilon, \mathrm{eq}(\#), \mathrm{id}, 9)
\end{aligned}
$$

recognises $\{\mathrm{a}^i \mathrm{b}^j \mathrm{c}^i \mathrm{d}^j \mid i, j \leq 1\}$

state: 3 *stack:*



input tape:

a	a	b	c	c	d

run:

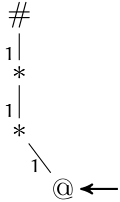
$\tau_1\tau_2\tau_3\tau_4^3$

Tree-stack automata (TSA) as automata with storage, Scott (1967)

$$
\begin{array}{rll}
\tau_1 &= (1, \mathrm{a}, \mathrm{bot} &, \mathrm{push}_1(*) \;, 2) \\
\tau_2 &= (2, \mathrm{a}, \mathrm{true} &, \mathrm{push}_1(*) \;, 2) \\
\tau_3 &= (2, \varepsilon, \mathrm{true} &, \mathrm{push}_1(\#), 3) \\
\tau_4 &= (3, \varepsilon, \mathrm{true} &, \mathrm{down} \;, 3) \\
\boxed{\tau_5} &\boxed{= (3, \mathrm{b}, \mathrm{bot}} &\boxed{, \mathrm{push}_2(*) \;, 4)} \\
\tau_6 &= (4, \mathrm{b}, \mathrm{true} &, \mathrm{push}_1(*) \;, 4) \\
\tau_7 &= (4, \varepsilon, \mathrm{true} &, \mathrm{push}_1(\#), 5) \\
\tau_8 &= (5, \varepsilon, \mathrm{true} &, \mathrm{down} \;, 5) \\
\tau_9 &= (5, \varepsilon, \mathrm{bot} &, \mathrm{up}_1 \;, 6) \\
\tau_{10} &= (6, \mathrm{c}, \mathrm{eq}(*) &, \mathrm{up}_1 \;, 6) \\
\tau_{11} &= (6, \varepsilon, \mathrm{eq}(\#) &, \mathrm{down} \;, 7) \\
\tau_{12} &= (7, \varepsilon, \mathrm{eq}(*) &, \mathrm{down} \;, 7) \\
\tau_{13} &= (7, \varepsilon, \mathrm{bot} &, \mathrm{up}_2 \;, 8) \\
\tau_{14} &= (8, \mathrm{d}, \mathrm{eq}(*) &, \mathrm{up}_1 \;, 8) \\
\tau_{15} &= (8, \varepsilon, \mathrm{eq}(\#) &, \mathrm{id} \;, 9)
\end{array}
$$

recognises $\{\mathrm{a}^i \mathrm{b}^j \mathrm{c}^i \mathrm{d}^j \mid i, j \leq 1\}$

state: 3 *stack:*



input tape:

a	a	b	c	c	d

run:

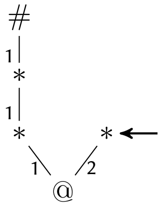
$\tau_1 \tau_2 \tau_3 \tau_4^3$

Tree-stack automata (TSA) as automata with storage, Scott (1967)

$$
\begin{aligned}
\tau_1 &= (1, \mathrm{a}, \mathrm{bot}, \mathrm{push}_1(*), 2)\\
\tau_2 &= (2, \mathrm{a}, \mathrm{true}, \mathrm{push}_1(*), 2)\\
\tau_3 &= (2, \varepsilon, \mathrm{true}, \mathrm{push}_1(\#), 3)\\
\tau_4 &= (3, \varepsilon, \mathrm{true}, \mathrm{down}, 3)\\
\tau_5 &= (3, \mathrm{b}, \mathrm{bot}, \mathrm{push}_2(*), 4)\\
\tau_6 &= (4, \mathrm{b}, \mathrm{true}, \mathrm{push}_1(*), 4)\\
\tau_7 &= (4, \varepsilon, \mathrm{true}, \mathrm{push}_1(\#), 5)\\
\tau_8 &= (5, \varepsilon, \mathrm{true}, \mathrm{down}, 5)\\
\tau_9 &= (5, \varepsilon, \mathrm{bot}, \mathrm{up}_1, 6)\\
\tau_{10} &= (6, \mathrm{c}, \mathrm{eq}(*), \mathrm{up}_1, 6)\\
\tau_{11} &= (6, \varepsilon, \mathrm{eq}(\#), \mathrm{down}, 7)\\
\tau_{12} &= (7, \varepsilon, \mathrm{eq}(*), \mathrm{down}, 7)\\
\tau_{13} &= (7, \varepsilon, \mathrm{bot}, \mathrm{up}_2, 8)\\
\tau_{14} &= (8, \mathrm{d}, \mathrm{eq}(*), \mathrm{up}_1, 8)\\
\tau_{15} &= (8, \varepsilon, \mathrm{eq}(\#), \mathrm{id}, 9)
\end{aligned}
$$

recognises $\{\mathrm{a}^i \mathrm{b}^j \mathrm{c}^i \mathrm{d}^j \mid i, j \leq 1\}$

state: 4 *stack:*



input tape:

a	a	b	c	c	d

run:

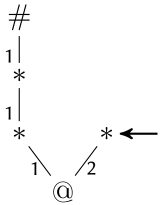
$\tau_1 \tau_2 \tau_3 \tau_4^3 \tau_5$

Tree-stack automata (TSA) as automata with storage, Scott (1967)

$$
\begin{array}{llllll}
\tau_1 &= (1, \mathrm{a}, \mathrm{bot} &, \mathrm{push}_1(*) &, 2) \\
\tau_2 &= (2, \mathrm{a}, \mathrm{true} &, \mathrm{push}_1(*) &, 2) \\
\tau_3 &= (2, \varepsilon, \mathrm{true} &, \mathrm{push}_1(\#) &, 3) \\
\tau_4 &= (3, \varepsilon, \mathrm{true} &, \mathrm{down} &, 3) \\
\tau_5 &= (3, \mathrm{b}, \mathrm{bot} &, \mathrm{push}_2(*) &, 4) \\
\tau_6 &= (4, \mathrm{b}, \mathrm{true} &, \mathrm{push}_1(*) &, 4) \\
\boxed{\tau_7} &\boxed{= (4, \varepsilon, \mathrm{true}} &\boxed{, \mathrm{push}_1(\#)} &\boxed{, 5)} \\
\tau_8 &= (5, \varepsilon, \mathrm{true} &, \mathrm{down} &, 5) \\
\tau_9 &= (5, \varepsilon, \mathrm{bot} &, \mathrm{up}_1 &, 6) \\
\tau_{10} &= (6, \mathrm{c}, \mathrm{eq}(*) &, \mathrm{up}_1 &, 6) \\
\tau_{11} &= (6, \varepsilon, \mathrm{eq}(\#) &, \mathrm{down} &, 7) \\
\tau_{12} &= (7, \varepsilon, \mathrm{eq}(*) &, \mathrm{down} &, 7) \\
\tau_{13} &= (7, \varepsilon, \mathrm{bot} &, \mathrm{up}_2 &, 8) \\
\tau_{14} &= (8, \mathrm{d}, \mathrm{eq}(*) &, \mathrm{up}_1 &, 8) \\
\tau_{15} &= (8, \varepsilon, \mathrm{eq}(\#) &, \mathrm{id} &, 9)
\end{array}
$$

recognises $\{\mathrm{a}^i \mathrm{b}^j \mathrm{c}^i \mathrm{d}^j \mid i, j \leq 1\}$

state: 4 *stack:*



input tape:

a	a	b	c	c	d

run:

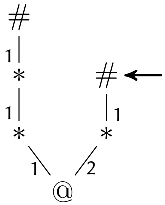
$\tau_1 \tau_2 \tau_3 \tau_4^3 \tau_5$

Tree-stack automata (TSA) as automata with storage, Scott (1967)

$$
\begin{array}{rl}
\tau_1 &= (1, \mathrm{a}, \mathrm{bot}, \mathrm{push}_1(*), 2)\\
\tau_2 &= (2, \mathrm{a}, \mathrm{true}, \mathrm{push}_1(*), 2)\\
\tau_3 &= (2, \varepsilon, \mathrm{true}, \mathrm{push}_1(\#), 3)\\
\tau_4 &= (3, \varepsilon, \mathrm{true}, \mathrm{down}, 3)\\
\tau_5 &= (3, \mathrm{b}, \mathrm{bot}, \mathrm{push}_2(*), 4)\\
\tau_6 &= (4, \mathrm{b}, \mathrm{true}, \mathrm{push}_1(*), 4)\\
\tau_7 &= (4, \varepsilon, \mathrm{true}, \mathrm{push}_1(\#), 5)\\
\tau_8 &= (5, \varepsilon, \mathrm{true}, \mathrm{down}, 5)\\
\tau_9 &= (5, \varepsilon, \mathrm{bot}, \mathrm{up}_1, 6)\\
\tau_{10} &= (6, \mathrm{c}, \mathrm{eq}(*), \mathrm{up}_1, 6)\\
\tau_{11} &= (6, \varepsilon, \mathrm{eq}(\#), \mathrm{down}, 7)\\
\tau_{12} &= (7, \varepsilon, \mathrm{eq}(*), \mathrm{down}, 7)\\
\tau_{13} &= (7, \varepsilon, \mathrm{bot}, \mathrm{up}_2, 8)\\
\tau_{14} &= (8, \mathrm{d}, \mathrm{eq}(*), \mathrm{up}_1, 8)\\
\tau_{15} &= (8, \varepsilon, \mathrm{eq}(\#), \mathrm{id}, 9)
\end{array}
$$

recognises $\{\mathrm{a}^i \mathrm{b}^j \mathrm{c}^i \mathrm{d}^j \mid i, j \leq 1\}$

state: 4 *stack:*



input tape:

a	a	b	c	c	d

run:

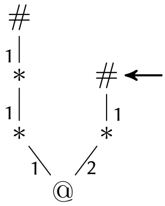
$\tau_1 \tau_2 \tau_3 \tau_4^3 \tau_5 \tau_7$

Tree-stack automata (TSA) as automata with storage, Scott (1967)

$$
\begin{aligned}
\tau_1 &= (1, \mathrm{a}, \mathrm{bot}, \mathrm{push}_1(*), 2)\\
\tau_2 &= (2, \mathrm{a}, \mathrm{true}, \mathrm{push}_1(*), 2)\\
\tau_3 &= (2, \varepsilon, \mathrm{true}, \mathrm{push}_1(\#), 3)\\
\tau_4 &= (3, \varepsilon, \mathrm{true}, \mathrm{down}, 3)\\
\tau_5 &= (3, \mathrm{b}, \mathrm{bot}, \mathrm{push}_2(*), 4)\\
\tau_6 &= (4, \mathrm{b}, \mathrm{true}, \mathrm{push}_1(*), 4)\\
\tau_7 &= (4, \varepsilon, \mathrm{true}, \mathrm{push}_1(\#), 5)\\
\boxed{\tau_8} &\boxed{= (5, \varepsilon, \mathrm{true}, \mathrm{down}, 5)}\\
\tau_9 &= (5, \varepsilon, \mathrm{bot}, \mathrm{up}_1, 6)\\
\tau_{10} &= (6, \mathrm{c}, \mathrm{eq}(*), \mathrm{up}_1, 6)\\
\tau_{11} &= (6, \varepsilon, \mathrm{eq}(\#), \mathrm{down}, 7)\\
\tau_{12} &= (7, \varepsilon, \mathrm{eq}(*), \mathrm{down}, 7)\\
\tau_{13} &= (7, \varepsilon, \mathrm{bot}, \mathrm{up}_2, 8)\\
\tau_{14} &= (8, \mathrm{d}, \mathrm{eq}(*), \mathrm{up}_1, 8)\\
\tau_{15} &= (8, \varepsilon, \mathrm{eq}(\#), \mathrm{id}, 9)
\end{aligned}
$$

recognises $\{\mathrm{a}^i \mathrm{b}^j \mathrm{c}^i \mathrm{d}^j \mid i, j \le 1\}$

state: 4 *stack:*



input tape:

a	a	b	c	c	d

run:

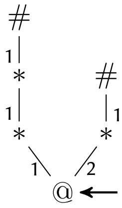
$\tau_1 \tau_2 \tau_3 \tau_4^3 \tau_5 \tau_7$

Tree-stack automata (TSA) as automata with storage, Scott (1967)

$$
\begin{aligned}
\tau_1 &= (1, \mathrm{a}, \mathrm{bot}, \mathrm{push}_1(*), 2)\\
\tau_2 &= (2, \mathrm{a}, \mathrm{true}, \mathrm{push}_1(*), 2)\\
\tau_3 &= (2, \varepsilon, \mathrm{true}, \mathrm{push}_1(\#), 3)\\
\tau_4 &= (3, \varepsilon, \mathrm{true}, \mathrm{down}, 3)\\
\tau_5 &= (3, \mathrm{b}, \mathrm{bot}, \mathrm{push}_2(*), 4)\\
\tau_6 &= (4, \mathrm{b}, \mathrm{true}, \mathrm{push}_1(*), 4)\\
\tau_7 &= (4, \varepsilon, \mathrm{true}, \mathrm{push}_1(\#), 5)\\
\tau_8 &= (5, \varepsilon, \mathrm{true}, \mathrm{down}, 5)\\
\tau_9 &= (5, \varepsilon, \mathrm{bot}, \mathrm{up}_1, 6)\\
\tau_{10} &= (6, \mathrm{c}, \mathrm{eq}(*), \mathrm{up}_1, 6)\\
\tau_{11} &= (6, \varepsilon, \mathrm{eq}(\#), \mathrm{down}, 7)\\
\tau_{12} &= (7, \varepsilon, \mathrm{eq}(*), \mathrm{down}, 7)\\
\tau_{13} &= (7, \varepsilon, \mathrm{bot}, \mathrm{up}_2, 8)\\
\tau_{14} &= (8, \mathrm{d}, \mathrm{eq}(*), \mathrm{up}_1, 8)\\
\tau_{15} &= (8, \varepsilon, \mathrm{eq}(\#), \mathrm{id}, 9)
\end{aligned}
$$

recognises $\{\mathrm{a}^i \mathrm{b}^j \mathrm{c}^i \mathrm{d}^j \mid i, j \leq 1\}$

state: 5 *stack:*



input tape:

a	a	b	c	c	d

run:

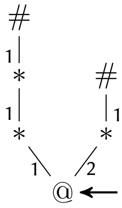
$\tau_1 \tau_2 \tau_3 \tau_4^3 \tau_5 \tau_7 \tau_8^2$

Tree-stack automata (TSA) as automata with storage, Scott (1967)

$$
\begin{array}{rl}
\tau_1 &= (1, \mathrm{a}, \mathrm{bot}, \mathrm{push}_1(*), 2)\\
\tau_2 &= (2, \mathrm{a}, \mathrm{true}, \mathrm{push}_1(*), 2)\\
\tau_3 &= (2, \varepsilon, \mathrm{true}, \mathrm{push}_1(\#), 3)\\
\tau_4 &= (3, \varepsilon, \mathrm{true}, \mathrm{down}, 3)\\
\tau_5 &= (3, \mathrm{b}, \mathrm{bot}, \mathrm{push}_2(*), 4)\\
\tau_6 &= (4, \mathrm{b}, \mathrm{true}, \mathrm{push}_1(*), 4)\\
\tau_7 &= (4, \varepsilon, \mathrm{true}, \mathrm{push}_1(\#), 5)\\
\tau_8 &= (5, \varepsilon, \mathrm{true}, \mathrm{down}, 5)\\
\tau_9 &= (5, \varepsilon, \mathrm{bot}, \mathrm{up}_1, 6)\\
\tau_{10} &= (6, \mathrm{c}, \mathrm{eq}(*), \mathrm{up}_1, 6)\\
\tau_{11} &= (6, \varepsilon, \mathrm{eq}(\#), \mathrm{down}, 7)\\
\tau_{12} &= (7, \varepsilon, \mathrm{eq}(*), \mathrm{down}, 7)\\
\tau_{13} &= (7, \varepsilon, \mathrm{bot}, \mathrm{up}_2, 8)\\
\tau_{14} &= (8, \mathrm{d}, \mathrm{eq}(*), \mathrm{up}_1, 8)\\
\tau_{15} &= (8, \varepsilon, \mathrm{eq}(\#), \mathrm{id}, 9)
\end{array}
$$

recognises $\{\mathrm{a}^i \mathrm{b}^j \mathrm{c}^i \mathrm{d}^j \mid i, j \leq 1\}$

state: 5 *stack:*



input tape:

a	a	b	c	c	d

run:

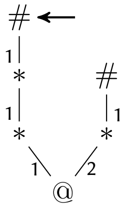
$\tau_1 \tau_2 \tau_3 \tau_4^3 \tau_5 \tau_7 \tau_8^2$

Tree-stack automata (TSA) as automata with storage, Scott (1967)

$$
\begin{aligned}
\tau_1 &= (1, \mathrm{a}, \mathrm{bot}, \mathrm{push}_1(*), 2)\\
\tau_2 &= (2, \mathrm{a}, \mathrm{true}, \mathrm{push}_1(*), 2)\\
\tau_3 &= (2, \varepsilon, \mathrm{true}, \mathrm{push}_1(\#), 3)\\
\tau_4 &= (3, \varepsilon, \mathrm{true}, \mathrm{down}, 3)\\
\tau_5 &= (3, \mathrm{b}, \mathrm{bot}, \mathrm{push}_2(*), 4)\\
\tau_6 &= (4, \mathrm{b}, \mathrm{true}, \mathrm{push}_1(*), 4)\\
\tau_7 &= (4, \varepsilon, \mathrm{true}, \mathrm{push}_1(\#), 5)\\
\tau_8 &= (5, \varepsilon, \mathrm{true}, \mathrm{down}, 5)\\
\tau_9 &= (5, \varepsilon, \mathrm{bot}, \mathrm{up}_1, 6)\\
\tau_{10} &= (6, \mathrm{c}, \mathrm{eq}(*), \mathrm{up}_1, 6)\\
\tau_{11} &= (6, \varepsilon, \mathrm{eq}(\#), \mathrm{down}, 7)\\
\tau_{12} &= (7, \varepsilon, \mathrm{eq}(*), \mathrm{down}, 7)\\
\tau_{13} &= (7, \varepsilon, \mathrm{bot}, \mathrm{up}_2, 8)\\
\tau_{14} &= (8, \mathrm{d}, \mathrm{eq}(*), \mathrm{up}_1, 8)\\
\tau_{15} &= (8, \varepsilon, \mathrm{eq}(\#), \mathrm{id}, 9)
\end{aligned}
$$

recognises $\{\mathrm{a}^i \mathrm{b}^j \mathrm{c}^i \mathrm{d}^j \mid i, j \leq 1\}$

state: 6 *stack:*



input tape:

a	a	b	c	c	d

run:

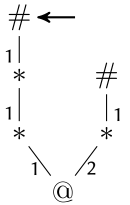
$\tau_1 \tau_2 \tau_3 \tau_4^3 \tau_5 \tau_7 \tau_8^2 \tau_9 \tau_{10}^2$

Tree-stack automata (TSA) as automata with storage, Scott (1967)

$$
\begin{array}{rl}
\tau_1 &= (1, \mathrm{a}, \mathrm{bot}, \mathrm{push}_1(*), 2)\\
\tau_2 &= (2, \mathrm{a}, \mathrm{true}, \mathrm{push}_1(*), 2)\\
\tau_3 &= (2, \varepsilon, \mathrm{true}, \mathrm{push}_1(\#), 3)\\
\tau_4 &= (3, \varepsilon, \mathrm{true}, \mathrm{down}, 3)\\
\tau_5 &= (3, \mathrm{b}, \mathrm{bot}, \mathrm{push}_2(*), 4)\\
\tau_6 &= (4, \mathrm{b}, \mathrm{true}, \mathrm{push}_1(*), 4)\\
\tau_7 &= (4, \varepsilon, \mathrm{true}, \mathrm{push}_1(\#), 5)\\
\tau_8 &= (5, \varepsilon, \mathrm{true}, \mathrm{down}, 5)\\
\tau_9 &= (5, \varepsilon, \mathrm{bot}, \mathrm{up}_1, 6)\\
\tau_{10} &= (6, \mathrm{c}, \mathrm{eq}(*), \mathrm{up}_1, 6)\\
\boxed{\begin{array}{rl}\tau_{11} &= (6, \varepsilon, \mathrm{eq}(\#), \mathrm{down}, 7)\\
\tau_{12} &= (7, \varepsilon, \mathrm{eq}(*), \mathrm{down}, 7)\end{array}}\\
\tau_{13} &= (7, \varepsilon, \mathrm{bot}, \mathrm{up}_2, 8)\\
\tau_{14} &= (8, \mathrm{d}, \mathrm{eq}(*), \mathrm{up}_1, 8)\\
\tau_{15} &= (8, \varepsilon, \mathrm{eq}(\#), \mathrm{id}, 9)
\end{array}
$$

recognises $\{\mathrm{a}^i \mathrm{b}^j \mathrm{c}^i \mathrm{d}^j \mid i, j \leq 1\}$

state: 6 *stack:*



input tape:

a	a	b	c	c	d

run:

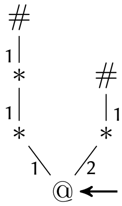
$\tau_1 \tau_2 \tau_3 \tau_4^3 \tau_5 \tau_7 \tau_8^2 \tau_9 \tau_{10}^2$

Tree-stack automata (TSA) as automata with storage, Scott (1967)

$$
\begin{array}{rl}
\tau_1 &= (1, \mathrm{a}, \mathrm{bot}, \mathrm{push}_1(*), 2)\\
\tau_2 &= (2, \mathrm{a}, \mathrm{true}, \mathrm{push}_1(*), 2)\\
\tau_3 &= (2, \varepsilon, \mathrm{true}, \mathrm{push}_1(\#), 3)\\
\tau_4 &= (3, \varepsilon, \mathrm{true}, \mathrm{down}, 3)\\
\tau_5 &= (3, \mathrm{b}, \mathrm{bot}, \mathrm{push}_2(*), 4)\\
\tau_6 &= (4, \mathrm{b}, \mathrm{true}, \mathrm{push}_1(*), 4)\\
\tau_7 &= (4, \varepsilon, \mathrm{true}, \mathrm{push}_1(\#), 5)\\
\tau_8 &= (5, \varepsilon, \mathrm{true}, \mathrm{down}, 5)\\
\tau_9 &= (5, \varepsilon, \mathrm{bot}, \mathrm{up}_1, 6)\\
\tau_{10} &= (6, \mathrm{c}, \mathrm{eq}(*), \mathrm{up}_1, 6)\\
\tau_{11} &= (6, \varepsilon, \mathrm{eq}(\#), \mathrm{down}, 7)\\
\tau_{12} &= (7, \varepsilon, \mathrm{eq}(*), \mathrm{down}, 7)\\
\tau_{13} &= (7, \varepsilon, \mathrm{bot}, \mathrm{up}_2, 8)\\
\tau_{14} &= (8, \mathrm{d}, \mathrm{eq}(*), \mathrm{up}_1, 8)\\
\tau_{15} &= (8, \varepsilon, \mathrm{eq}(\#), \mathrm{id}, 9)
\end{array}
$$

recognises $\{\mathrm{a}^i\mathrm{b}^j\mathrm{c}^i\mathrm{d}^j \mid i, j \leq 1\}$

state: 7 *stack:*



input tape:

a	a	b	c	c	d

run:

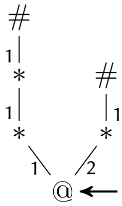
$\tau_1\tau_2\tau_3\tau_4^3\tau_5\tau_7\tau_8^2\tau_9\tau_{10}^2\tau_{11}\tau_{12}^2$

Tree-stack automata (TSA) as automata with storage, Scott (1967)

$$
\begin{aligned}
\tau_1 &= (1, \mathrm{a}, \mathrm{bot}, \mathrm{push}_1(*), 2)\\
\tau_2 &= (2, \mathrm{a}, \mathrm{true}, \mathrm{push}_1(*), 2)\\
\tau_3 &= (2, \varepsilon, \mathrm{true}, \mathrm{push}_1(\#), 3)\\
\tau_4 &= (3, \varepsilon, \mathrm{true}, \mathrm{down}, 3)\\
\tau_5 &= (3, \mathrm{b}, \mathrm{bot}, \mathrm{push}_2(*), 4)\\
\tau_6 &= (4, \mathrm{b}, \mathrm{true}, \mathrm{push}_1(*), 4)\\
\tau_7 &= (4, \varepsilon, \mathrm{true}, \mathrm{push}_1(\#), 5)\\
\tau_8 &= (5, \varepsilon, \mathrm{true}, \mathrm{down}, 5)\\
\tau_9 &= (5, \varepsilon, \mathrm{bot}, \mathrm{up}_1, 6)\\
\tau_{10} &= (6, \mathrm{c}, \mathrm{eq}(*), \mathrm{up}_1, 6)\\
\tau_{11} &= (6, \varepsilon, \mathrm{eq}(\#), \mathrm{down}, 7)\\
\tau_{12} &= (7, \varepsilon, \mathrm{eq}(*), \mathrm{down}, 7)\\
\tau_{13} &= (7, \varepsilon, \mathrm{bot}, \mathrm{up}_2, 8)\\
\tau_{14} &= (8, \mathrm{d}, \mathrm{eq}(*), \mathrm{up}_1, 8)\\
\tau_{15} &= (8, \varepsilon, \mathrm{eq}(\#), \mathrm{id}, 9)
\end{aligned}
$$

recognises $\{\mathrm{a}^i \mathrm{b}^j \mathrm{c}^i \mathrm{d}^j \mid i, j \leq 1\}$

state: 7 *stack:*



input tape:

a	a	b	c	c	d

run:

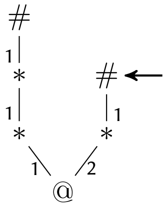
$\tau_1\tau_2\tau_3\tau_4^3\tau_5\tau_7\tau_8^2\tau_9\tau_{10}^2\tau_{11}\tau_{12}^2$

Tree-stack automata (TSA) as automata with storage, Scott (1967)

$$
\begin{aligned}
\tau_1 &= (1, \mathrm{a}, \mathrm{bot}, \mathrm{push}_1(*), 2)\\
\tau_2 &= (2, \mathrm{a}, \mathrm{true}, \mathrm{push}_1(*), 2)\\
\tau_3 &= (2, \varepsilon, \mathrm{true}, \mathrm{push}_1(\#), 3)\\
\tau_4 &= (3, \varepsilon, \mathrm{true}, \mathrm{down}, 3)\\
\tau_5 &= (3, \mathrm{b}, \mathrm{bot}, \mathrm{push}_2(*), 4)\\
\tau_6 &= (4, \mathrm{b}, \mathrm{true}, \mathrm{push}_1(*), 4)\\
\tau_7 &= (4, \varepsilon, \mathrm{true}, \mathrm{push}_1(\#), 5)\\
\tau_8 &= (5, \varepsilon, \mathrm{true}, \mathrm{down}, 5)\\
\tau_9 &= (5, \varepsilon, \mathrm{bot}, \mathrm{up}_1, 6)\\
\tau_{10} &= (6, \mathrm{c}, \mathrm{eq}(*), \mathrm{up}_1, 6)\\
\tau_{11} &= (6, \varepsilon, \mathrm{eq}(\#), \mathrm{down}, 7)\\
\tau_{12} &= (7, \varepsilon, \mathrm{eq}(*), \mathrm{down}, 7)\\
\tau_{13} &= (7, \varepsilon, \mathrm{bot}, \mathrm{up}_2, 8)\\
\tau_{14} &= (8, \mathrm{d}, \mathrm{eq}(*), \mathrm{up}_1, 8)\\
\tau_{15} &= (8, \varepsilon, \mathrm{eq}(\#), \mathrm{id}, 9)
\end{aligned}
$$

recognises $\{\mathrm{a}^i \mathrm{b}^j \mathrm{c}^i \mathrm{d}^j \mid i, j \leq 1\}$

state: 9 *stack:*



input tape:

a	a	b	c	c	d

run:

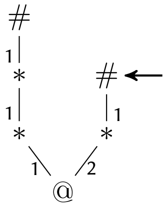
$\tau_1 \tau_2 \tau_3 \tau_4^3 \tau_5 \tau_7 \tau_8^2 \tau_9 \tau_{10}^2 \tau_{11} \tau_{12}^2 \tau_{13} \tau_{14} \tau_{15}$

Tree-stack automata (TSA) as automata with storage, Scott (1967)

$$
\begin{aligned}
\tau_1 &= (1, \mathrm{a}, \mathrm{bot}, \mathrm{push}_1(*), 2)\\
\tau_2 &= (2, \mathrm{a}, \mathrm{true}, \mathrm{push}_1(*), 2)\\
\tau_3 &= (2, \varepsilon, \mathrm{true}, \mathrm{push}_1(\#), 3)\\
\tau_4 &= (3, \varepsilon, \mathrm{true}, \mathrm{down}, 3)\\
\tau_5 &= (3, \mathrm{b}, \mathrm{bot}, \mathrm{push}_2(*), 4)\\
\tau_6 &= (4, \mathrm{b}, \mathrm{true}, \mathrm{push}_1(*), 4)\\
\tau_7 &= (4, \varepsilon, \mathrm{true}, \mathrm{push}_1(\#), 5)\\
\tau_8 &= (5, \varepsilon, \mathrm{true}, \mathrm{down}, 5)\\
\tau_9 &= (5, \varepsilon, \mathrm{bot}, \mathrm{up}_1, 6)\\
\tau_{10} &= (6, \mathrm{c}, \mathrm{eq}(*), \mathrm{up}_1, 6)\\
\tau_{11} &= (6, \varepsilon, \mathrm{eq}(\#), \mathrm{down}, 7)\\
\tau_{12} &= (7, \varepsilon, \mathrm{eq}(*), \mathrm{down}, 7)\\
\tau_{13} &= (7, \varepsilon, \mathrm{bot}, \mathrm{up}_2, 8)\\
\tau_{14} &= (8, \mathrm{d}, \mathrm{eq}(*), \mathrm{up}_1, 8)\\
\tau_{15} &= (8, \varepsilon, \mathrm{eq}(\#), \mathrm{id}, 9)
\end{aligned}
$$

recognises $\{\mathrm{a}^i \mathrm{b}^j \mathrm{c}^i \mathrm{d}^j \mid i, j \leq 1\}$

state: 9 *stack:*



input tape:

a	a	b	c	c	d

run:

$\tau_1 \tau_2 \tau_3 \tau_4^3 \tau_5 \tau_7 \tau_8^2 \tau_9 \tau_{10}^2 \tau_{11} \tau_{12}^2 \tau_{13} \tau_{14} \tau_{15}$

2-restricted: enters each stack position at most 2 times *from below*

The automata characterisation

Theorem Denkinger (2016)

$$k\text{-MCFL} = k\text{-TSL}_{\mathsf{r}}$$

Proof sketch: Show both set inclusions by construction.

---

k-MCFL: languages generated by MCFGs of fan-out at most k

k-TSL: languages recognised by k-restricted tree stack automata

k-MCFL $\subseteq$ k-TSL$_\mathrm{r}$

Lemma Denkinger (2016)

$$k\text{-MCFL} \subseteq k\text{-TSL}_\mathrm{r}$$

k-MCFL $\subseteq$ k-TSL$_\mathrm{r}$

Lemma Denkinger (2016)

$$k\text{-MCFL} \subseteq k\text{-TSL}_\mathrm{r}$$

Construction idea:

$$\rho = A \to [\quad \mathrm{a} \quad x_1 \quad , \quad \mathrm{c} \quad x_2 \quad](B)$$

return addresses

k-MCFL $\subseteq$ k-TSL$_\mathrm{r}$

Lemma Denkinger (2016)

$$k\text{-MCFL} \subseteq k\text{-TSL}_\mathrm{r}$$

Construction idea:

$$\rho = A \to [\ \underset{\langle \rho, 1, 0\rangle}{\bullet}\ \mathrm{a}\ \overset{\langle \rho, 1, 1\rangle}{\bullet}\ x_1\ \underset{\langle \rho, 1, 2\rangle}{\bullet}\ ,\ \overset{\langle \rho, 2, 0\rangle}{\bullet}\ \mathrm{c}\ \underset{\langle \rho, 2, 1\rangle}{\bullet}\ x_2\ \overset{\langle \rho, 2, 2\rangle}{\bullet}\ ](B)$$

return addresses

k-MCFL $\subseteq k$-TSL$_r$

Lemma — Denkinger (2016)

$$k\text{-MCFL} \subseteq k\text{-TSL}_r$$

Construction idea:

$$\rho = A \to [\ \underset{\substack{\uparrow \\ \langle \rho, 1, 0\rangle}}{\bullet}\ \mathrm{a}\ \overset{\substack{\langle \rho, 1, 1\rangle \\ \downarrow}}{\bullet}\ x_1\ \underset{\substack{\uparrow \\ \langle \rho, 1, 2\rangle}}{\bullet}\ ,\ \overset{\substack{\langle \rho, 2, 0\rangle \\ \downarrow}}{\bullet}\ \mathrm{c}\ \underset{\substack{\uparrow \\ \langle \rho, 2, 1\rangle}}{\bullet}\ x_2\ \overset{\substack{\langle \rho, 2, 2\rangle \\ \downarrow}}{\bullet}\ ](B)$$

return addresses

example transitions:

k-MCFL $\subseteq$ k-TSL$_\mathrm{r}$

Lemma Denkinger (2016)

$$k\text{-MCFL} \subseteq k\text{-TSL}_\mathrm{r}$$

Construction idea: $\langle \rho, 1, 1\rangle$ $\langle \rho, 2, 0\rangle$ $\langle \rho, 2, 2\rangle$

$$\rho = A \to [\, \bullet\ \mathrm{a}\ \bullet\ x_1\ \bullet\ ,\ \bullet\ \mathrm{c}\ \bullet\ x_2\ \bullet\ ](B)$$

return addresses $\langle \rho, 1, 0\rangle$ $\langle \rho, 1, 2\rangle$ $\langle \rho, 2, 1\rangle$

example transitions:

read: $(\langle \rho, 1, 0\rangle, a, \text{true}, \text{id}, \langle \rho, 1, 1\rangle)$

k-MCFL $\subseteq$ k-TSL$_\mathrm{r}$

Lemma Denkinger (2016)

$$k\text{-MCFL} \subseteq k\text{-TSL}_\mathrm{r}$$

Construction idea:

$$\rho = A \to [\underset{\langle\rho,1,0\rangle}{\bullet}\ \mathrm{a}\ \overset{\langle\rho,1,1\rangle}{\bullet}\ x_1\ \underset{\langle\rho,1,2\rangle}{\bullet}\ ,\ \overset{\langle\rho,2,0\rangle}{\bullet}\ \mathrm{c}\ \underset{\langle\rho,2,1\rangle}{\bullet}\ x_2\ \overset{\langle\rho,2,2\rangle}{\bullet}\ ](B)$$

return addresses

example transitions: (ρ' has lhs B and $\bar{\rho}$ has A on rhs)

read: $(\langle\rho, 1, 0\rangle, a, \text{true}, \text{id}, \langle\rho, 1, 1\rangle)$

call: $(\langle\rho, 1, 1\rangle, \varepsilon, \text{true}, \text{push}_1(\langle\rho, 1, 2\rangle), \langle\rho', 1, 0\rangle)$

k-MCFL $\subseteq$ k-TSL$_\mathrm{r}$

Lemma Denkinger (2016)

$$k\text{-MCFL} \subseteq k\text{-TSL}_\mathrm{r}$$

Construction idea:

$$\rho = A \to [\,\underset{\langle\rho,1,0\rangle}{\bullet}\ \mathrm{a}\ \overset{\langle\rho,1,1\rangle}{\bullet}\ x_1\ \underset{\langle\rho,1,2\rangle}{\bullet}\ ,\ \overset{\langle\rho,2,0\rangle}{\bullet}\ \mathrm{c}\ \underset{\langle\rho,2,1\rangle}{\bullet}\ x_2\ \overset{\langle\rho,2,2\rangle}{\bullet}\,](B)$$

return addresses

example transitions: (ρ' has lhs B and $\bar{\rho}$ has A on rhs)

read: $(\langle\rho,1,0\rangle,\ a,\ \text{true},\ \text{id},\ \langle\rho,1,1\rangle)$

call: $(\langle\rho,1,1\rangle,\ \varepsilon,\ \text{true},\ \text{push}_1(\langle\rho,1,2\rangle),\ \langle\rho',1,0\rangle)$

return: $(\langle\rho,1,2\rangle, \varepsilon, \text{eq}(\langle\bar{\rho},i,j\rangle), \text{set}(\rho), \langle\bar{\rho},i,j\rangle_\downarrow)$
$(\langle\bar{\rho},i,j\rangle_\downarrow, \varepsilon, \text{true}, \text{down}, \langle\bar{\rho},i,j\rangle)$

k-MCFL $\subseteq$ k-TSL$_\mathrm{r}$

Lemma Denkinger (2016)

$$k\text{-MCFL} \subseteq k\text{-TSL}_\mathrm{r}$$

Construction idea:

$$\rho = A \to [\,\underset{\uparrow\,\langle\rho,1,0\rangle}{\bullet}\ \mathrm{a}\ \overset{\langle\rho,1,1\rangle\,\downarrow}{\bullet}\ x_1\ \underset{\uparrow\,\langle\rho,1,2\rangle}{\bullet}\ ,\ \overset{\langle\rho,2,0\rangle\,\downarrow}{\bullet}\ \mathrm{c}\ \underset{\uparrow\,\langle\rho,2,1\rangle}{\bullet}\ x_2\ \overset{\langle\rho,2,2\rangle\,\downarrow}{\bullet}\,](B)$$

return addresses

example transitions: (ρ' has lhs B and $\bar\rho$ has A on rhs)

read: $(\langle\rho,1,0\rangle, a, \mathrm{true}, \mathrm{id}, \langle\rho,1,1\rangle)$

call: $(\langle\rho,1,1\rangle, \varepsilon, \mathrm{true}, \mathrm{push}_1(\langle\rho,1,2\rangle), \langle\rho',1,0\rangle)$

return: $(\langle\rho,1,2\rangle, \varepsilon, \mathrm{eq}(\langle\bar\rho,i,j\rangle), \mathrm{set}(\rho), \langle\bar\rho,i,j\rangle_\downarrow)$
$(\langle\bar\rho,i,j\rangle_\downarrow, \varepsilon, \mathrm{true}, \mathrm{down}, \langle\bar\rho,i,j\rangle)$

resume: $(\langle\rho,2,1\rangle, \varepsilon, \mathrm{true}, \mathrm{up}_1, \langle\rho,2,1\rangle_\uparrow)$
$(\langle\rho,2,1\rangle_\uparrow, \varepsilon, \mathrm{eq}(\rho'), \mathrm{set}(\langle\rho,2,2\rangle), \langle\rho',2,0\rangle)$

k-$\mathrm{TSL}_\mathrm{r} \subseteq k$-MCFL

Lemma Denkinger (2016)

$$k\text{-}\mathrm{TSL}_\mathrm{r} \subseteq k\text{-}\mathrm{MCFL}$$

k-$\mathrm{TSL_r} \subseteq k$-MCFL

Lemma Denkinger (2016)

$$k\text{-}\mathrm{TSL_r} \subseteq k\text{-}\mathrm{MCFL}$$

Proof idea:

(1) construct an MCFG that generates the runs

(2) use closure of MCFG under homomorphisms

k-TSL$_\mathrm{r}$ $\subseteq$ k-MCFL

Lemma Denkinger (2016)

$$k\text{-TSL}_\mathrm{r} \subseteq k\text{-MCFL}$$

Proof idea:

(1) construct an MCFG that generates the runs

$$\langle \underbrace{q_1, q_1', \ldots, q_m, q_m'}_{\in Q^{2m}}; \underbrace{\gamma_0, \ldots, \gamma_m}_{\in \Gamma^{m+1}} \rangle \Longrightarrow^* (\theta_1, \ldots, \theta_m)$$

if and only if

- $\theta_1, \ldots, \theta_m$ all return to the stack position they started from and never go below it
- θ_i starts with state q_i and stack symbol γ_{i-1} and ends with q_i' and γ_i (for $1 \leq i \leq m$)

(2) use closure of MCFG under homomorphisms

k-TSL$_\mathrm{r}$ $\subseteq$ k-MCFL (monadic example)

transitions:

$$
\begin{aligned}
\tau_1 &= (1, \mathrm{a}, \mathrm{true}, \mathrm{push}_1(*), 1)\\
\tau_2 &= (1, \varepsilon, \mathrm{true}, \mathrm{push}_1(\#), 2)\\
\tau_3 &= (2, \varepsilon, \mathrm{eq}(\#), \mathrm{down}, 2)\\
\tau_4 &= (2, \mathrm{b}, \mathrm{eq}(*), \mathrm{down}, 2)\\
\tau_5 &= (2, \varepsilon, \mathrm{bot}, \mathrm{up}_1, 3)\\
\tau_6 &= (3, \mathrm{c}, \mathrm{eq}(*), \mathrm{up}_1, 3)\\
\tau_7 &= (3, \varepsilon, \mathrm{eq}(\#), \mathrm{down}, 4)\\
\tau_8 &= (4, \mathrm{d}, \mathrm{eq}(*), \mathrm{down}, 4)\\
\tau_9 &= (4, \varepsilon, \mathrm{bot}, \mathrm{id}, 5)
\end{aligned}
$$

k-TSL$_r$ $\subseteq$ k-MCFL (monadic example)

transitions:

$$
\begin{aligned}
\tau_1 &= (1, \mathrm{a}, \mathrm{true}, \mathrm{push}_1(*), 1)\\
\tau_2 &= (1, \varepsilon, \mathrm{true}, \mathrm{push}_1(\#), 2)\\
\tau_3 &= (2, \varepsilon, \mathrm{eq}(\#), \mathrm{down}, 2)\\
\tau_4 &= (2, \mathrm{b}, \mathrm{eq}(*), \mathrm{down}, 2)\\
\tau_5 &= (2, \varepsilon, \mathrm{bot}, \mathrm{up}_1, 3)\\
\tau_6 &= (3, \mathrm{c}, \mathrm{eq}(*), \mathrm{up}_1, 3)\\
\tau_7 &= (3, \varepsilon, \mathrm{eq}(\#), \mathrm{down}, 4)\\
\tau_8 &= (4, \mathrm{d}, \mathrm{eq}(*), \mathrm{down}, 4)\\
\tau_9 &= (4, \varepsilon, \mathrm{bot}, \mathrm{id}, 5)
\end{aligned}
$$

run for $a^2b^2c^2d^2$ and the stack:

111

11

1

ε $(1, @)$

k-TSL$_\mathrm{r}$ $\subseteq$ k-MCFL (monadic example)

transitions:

$$
\begin{aligned}
\tau_1 &= (1, \mathrm{a}, \mathrm{true}, \mathrm{push}_1(*), 1)\\
\tau_2 &= (1, \varepsilon, \mathrm{true}, \mathrm{push}_1(\#), 2)\\
\tau_3 &= (2, \varepsilon, \mathrm{eq}(\#), \mathrm{down}, 2)\\
\tau_4 &= (2, \mathrm{b}, \mathrm{eq}(*), \mathrm{down}, 2)\\
\tau_5 &= (2, \varepsilon, \mathrm{bot}, \mathrm{up}_1, 3)\\
\tau_6 &= (3, \mathrm{c}, \mathrm{eq}(*), \mathrm{up}_1, 3)\\
\tau_7 &= (3, \varepsilon, \mathrm{eq}(\#), \mathrm{down}, 4)\\
\tau_8 &= (4, \mathrm{d}, \mathrm{eq}(*), \mathrm{down}, 4)\\
\tau_9 &= (4, \varepsilon, \mathrm{bot}, \mathrm{id}, 5)
\end{aligned}
$$

run for $a^2b^2c^2d^2$ and the stack:

$$
\begin{array}{ll}
111 & \\
11 & \\
1 & (1, *) \\
 & \uparrow \tau_1 \\
\varepsilon & (1, @)
\end{array}
$$

k-TSL$_\mathrm{r}$ $\subseteq$ k-MCFL (monadic example)

transitions:

$$\begin{aligned}
\tau_1 &= (1, \mathrm{a}, \mathrm{true}, \mathrm{push}_1(*), 1)\\
\tau_2 &= (1, \varepsilon, \mathrm{true}, \mathrm{push}_1(\#), 2)\\
\tau_3 &= (2, \varepsilon, \mathrm{eq}(\#), \mathrm{down}, 2)\\
\tau_4 &= (2, \mathrm{b}, \mathrm{eq}(*), \mathrm{down}, 2)\\
\tau_5 &= (2, \varepsilon, \mathrm{bot}, \mathrm{up}_1, 3)\\
\tau_6 &= (3, \mathrm{c}, \mathrm{eq}(*), \mathrm{up}_1, 3)\\
\tau_7 &= (3, \varepsilon, \mathrm{eq}(\#), \mathrm{down}, 4)\\
\tau_8 &= (4, \mathrm{d}, \mathrm{eq}(*), \mathrm{down}, 4)\\
\tau_9 &= (4, \varepsilon, \mathrm{bot}, \mathrm{id}, 5)
\end{aligned}$$

run for $a^2b^2c^2d^2$ and the stack:

$$\begin{array}{ll}
111 & \\
11 & (1, *) \\
 & \uparrow \tau_1 \\
1 & (1, *) \\
 & \uparrow \tau_1 \\
\varepsilon & (1, @)
\end{array}$$

k-TSL$_\mathrm{r}$ $\subseteq$ k-MCFL (monadic example)

transitions:

$$
\begin{array}{llllll}
\tau_1 = (1, & \mathrm{a}, & \mathrm{true} & , \mathrm{push}_1(*) & , 1) \\
\tau_2 = (1, & \varepsilon, & \mathrm{true} & , \mathrm{push}_1(\#) & , 2) \\
\tau_3 = (2, & \varepsilon, & \mathrm{eq}(\#) & , \mathrm{down} & , 2) \\
\tau_4 = (2, & \mathrm{b}, & \mathrm{eq}(*) & , \mathrm{down} & , 2) \\
\tau_5 = (2, & \varepsilon, & \mathrm{bot} & , \mathrm{up}_1 & , 3) \\
\tau_6 = (3, & \mathrm{c}, & \mathrm{eq}(*) & , \mathrm{up}_1 & , 3) \\
\tau_7 = (3, & \varepsilon, & \mathrm{eq}(\#) & , \mathrm{down} & , 4) \\
\tau_8 = (4, & \mathrm{d}, & \mathrm{eq}(*) & , \mathrm{down} & , 4) \\
\tau_9 = (4, & \varepsilon, & \mathrm{bot} & , \mathrm{id} & , 5)
\end{array}
$$

run for $a^2b^2c^2d^2$ and the stack:

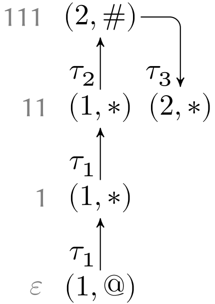
$$
\begin{array}{rc}
111 & (2, \#) \\
 & \uparrow \tau_2 \\
11 & (1, *) \\
 & \uparrow \tau_1 \\
1 & (1, *) \\
 & \uparrow \tau_1 \\
\varepsilon & (1, @)
\end{array}
$$

k-TSL$_\mathrm{r}$ $\subseteq$ k-MCFL (monadic example)

transitions:

$$
\begin{aligned}
\tau_1 &= (1, \mathrm{a}, \mathrm{true}, \mathrm{push}_1(*), 1)\\
\tau_2 &= (1, \varepsilon, \mathrm{true}, \mathrm{push}_1(\#), 2)\\
\tau_3 &= (2, \varepsilon, \mathrm{eq}(\#), \mathrm{down}, 2)\\
\tau_4 &= (2, \mathrm{b}, \mathrm{eq}(*), \mathrm{down}, 2)\\
\tau_5 &= (2, \varepsilon, \mathrm{bot}, \mathrm{up}_1, 3)\\
\tau_6 &= (3, \mathrm{c}, \mathrm{eq}(*), \mathrm{up}_1, 3)\\
\tau_7 &= (3, \varepsilon, \mathrm{eq}(\#), \mathrm{down}, 4)\\
\tau_8 &= (4, \mathrm{d}, \mathrm{eq}(*), \mathrm{down}, 4)\\
\tau_9 &= (4, \varepsilon, \mathrm{bot}, \mathrm{id}, 5)
\end{aligned}
$$

run for $a^2b^2c^2d^2$ and the stack:

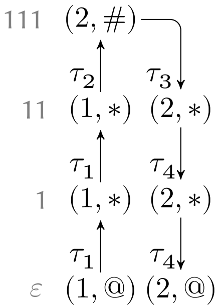


k-TSL$_\mathrm{r}$ $\subseteq$ k-MCFL (monadic example)

transitions:

$$
\begin{aligned}
\tau_1 &= (1, \mathrm{a}, \mathrm{true}, \mathrm{push}_1(*), 1)\\
\tau_2 &= (1, \varepsilon, \mathrm{true}, \mathrm{push}_1(\#), 2)\\
\tau_3 &= (2, \varepsilon, \mathrm{eq}(\#), \mathrm{down}, 2)\\
\tau_4 &= (2, \mathrm{b}, \mathrm{eq}(*), \mathrm{down}, 2)\\
\tau_5 &= (2, \varepsilon, \mathrm{bot}, \mathrm{up}_1, 3)\\
\tau_6 &= (3, \mathrm{c}, \mathrm{eq}(*), \mathrm{up}_1, 3)\\
\tau_7 &= (3, \varepsilon, \mathrm{eq}(\#), \mathrm{down}, 4)\\
\tau_8 &= (4, \mathrm{d}, \mathrm{eq}(*), \mathrm{down}, 4)\\
\tau_9 &= (4, \varepsilon, \mathrm{bot}, \mathrm{id}, 5)
\end{aligned}
$$

run for $a^2b^2c^2d^2$ and the stack:

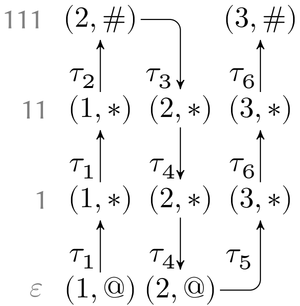


k-TSL$_\mathrm{r}$ $\subseteq$ k-MCFL (monadic example)

transitions:

$$
\begin{aligned}
\tau_1 &= (1, \mathrm{a}, \mathrm{true}, \mathrm{push}_1(*), 1)\\
\tau_2 &= (1, \varepsilon, \mathrm{true}, \mathrm{push}_1(\#), 2)\\
\tau_3 &= (2, \varepsilon, \mathrm{eq}(\#), \mathrm{down}, 2)\\
\tau_4 &= (2, \mathrm{b}, \mathrm{eq}(*), \mathrm{down}, 2)\\
\tau_5 &= (2, \varepsilon, \mathrm{bot}, \mathrm{up}_1, 3)\\
\tau_6 &= (3, \mathrm{c}, \mathrm{eq}(*), \mathrm{up}_1, 3)\\
\tau_7 &= (3, \varepsilon, \mathrm{eq}(\#), \mathrm{down}, 4)\\
\tau_8 &= (4, \mathrm{d}, \mathrm{eq}(*), \mathrm{down}, 4)\\
\tau_9 &= (4, \varepsilon, \mathrm{bot}, \mathrm{id}, 5)
\end{aligned}
$$

run for $a^2b^2c^2d^2$ and the stack:

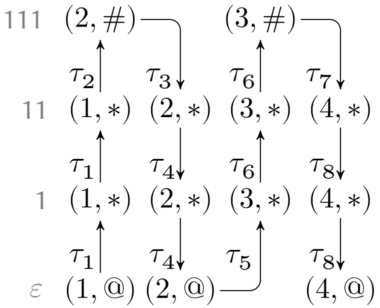


k-TSL$_\mathrm{r}$ $\subseteq$ k-MCFL (monadic example)

transitions:

$$\begin{aligned}
\tau_1 &= (1, \mathrm{a}, \mathrm{true}, \mathrm{push}_1(*), 1)\\
\tau_2 &= (1, \varepsilon, \mathrm{true}, \mathrm{push}_1(\#), 2)\\
\tau_3 &= (2, \varepsilon, \mathrm{eq}(\#), \mathrm{down}, 2)\\
\tau_4 &= (2, \mathrm{b}, \mathrm{eq}(*), \mathrm{down}, 2)\\
\tau_5 &= (2, \varepsilon, \mathrm{bot}, \mathrm{up}_1, 3)\\
\tau_6 &= (3, \mathrm{c}, \mathrm{eq}(*), \mathrm{up}_1, 3)\\
\tau_7 &= (3, \varepsilon, \mathrm{eq}(\#), \mathrm{down}, 4)\\
\tau_8 &= (4, \mathrm{d}, \mathrm{eq}(*), \mathrm{down}, 4)\\
\tau_9 &= (4, \varepsilon, \mathrm{bot}, \mathrm{id}, 5)
\end{aligned}$$

run for $a^2b^2c^2d^2$ and the stack:

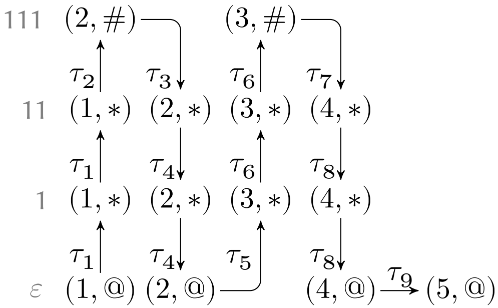


k-TSL$_{\mathrm{r}}$ $\subseteq$ k-MCFL (monadic example)

transitions:

$$
\begin{aligned}
\tau_1 &= (1, \mathrm{a}, \mathrm{true}, \mathrm{push}_1(*), 1)\\
\tau_2 &= (1, \varepsilon, \mathrm{true}, \mathrm{push}_1(\#), 2)\\
\tau_3 &= (2, \varepsilon, \mathrm{eq}(\#), \mathrm{down}, 2)\\
\tau_4 &= (2, \mathrm{b}, \mathrm{eq}(*), \mathrm{down}, 2)\\
\tau_5 &= (2, \varepsilon, \mathrm{bot}, \mathrm{up}_1, 3)\\
\tau_6 &= (3, \mathrm{c}, \mathrm{eq}(*), \mathrm{up}_1, 3)\\
\tau_7 &= (3, \varepsilon, \mathrm{eq}(\#), \mathrm{down}, 4)\\
\tau_8 &= (4, \mathrm{d}, \mathrm{eq}(*), \mathrm{down}, 4)\\
\tau_9 &= (4, \varepsilon, \mathrm{bot}, \mathrm{id}, 5)
\end{aligned}
$$

run for $a^2b^2c^2d^2$ and the stack:

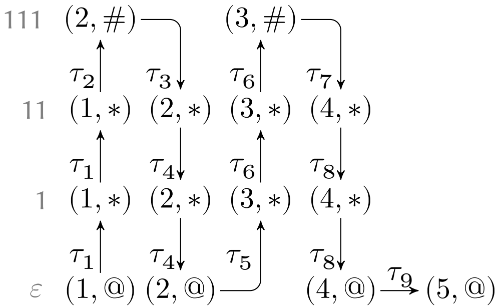


k-$\mathrm{TSL_r} \subseteq k$-MCFL (monadic example)

transitions:

$$\begin{aligned}
\tau_1 &= (1, \mathrm{a}, \mathrm{true}, \mathrm{push}_1(*), 1)\\
\tau_2 &= (1, \varepsilon, \mathrm{true}, \mathrm{push}_1(\#), 2)\\
\tau_3 &= (2, \varepsilon, \mathrm{eq}(\#), \mathrm{down}, 2)\\
\tau_4 &= (2, \mathrm{b}, \mathrm{eq}(*), \mathrm{down}, 2)\\
\tau_5 &= (2, \varepsilon, \mathrm{bot}, \mathrm{up}_1, 3)\\
\tau_6 &= (3, \mathrm{c}, \mathrm{eq}(*), \mathrm{up}_1, 3)\\
\tau_7 &= (3, \varepsilon, \mathrm{eq}(\#), \mathrm{down}, 4)\\
\tau_8 &= (4, \mathrm{d}, \mathrm{eq}(*), \mathrm{down}, 4)\\
\tau_9 &= (4, \varepsilon, \mathrm{bot}, \mathrm{id}, 5)
\end{aligned}$$

run for $a^2b^2c^2d^2$ and the stack:



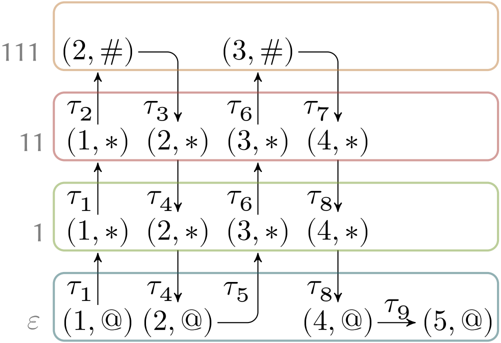
principle of crossing sequences

k-TSL$_{\mathrm{r}} \subseteq k$-MCFL (monadic example)

transitions:

$$
\begin{aligned}
\tau_1 &= (1, \mathrm{a}, \mathrm{true}, \mathrm{push}_1(*), 1)\\
\tau_2 &= (1, \varepsilon, \mathrm{true}, \mathrm{push}_1(\#), 2)\\
\tau_3 &= (2, \varepsilon, \mathrm{eq}(\#), \mathrm{down}, 2)\\
\tau_4 &= (2, \mathrm{b}, \mathrm{eq}(*), \mathrm{down}, 2)\\
\tau_5 &= (2, \varepsilon, \mathrm{bot}, \mathrm{up}_1, 3)\\
\tau_6 &= (3, \mathrm{c}, \mathrm{eq}(*), \mathrm{up}_1, 3)\\
\tau_7 &= (3, \varepsilon, \mathrm{eq}(\#), \mathrm{down}, 4)\\
\tau_8 &= (4, \mathrm{d}, \mathrm{eq}(*), \mathrm{down}, 4)\\
\tau_9 &= (4, \varepsilon, \mathrm{bot}, \mathrm{id}, 5)
\end{aligned}
$$

run for $a^2b^2c^2d^2$ and the stack:



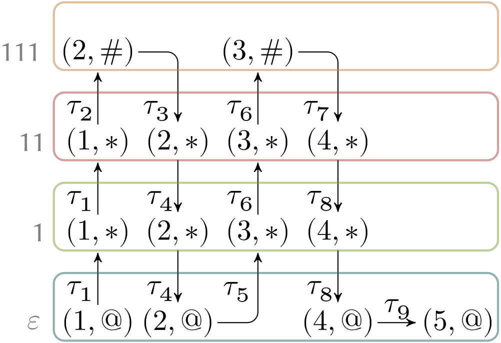
principle of crossing sequences

k-TSL$_\mathrm{r}$ $\subseteq$ k-MCFL (monadic example)

transitions:

$$
\begin{aligned}
\tau_1 &= (1, \mathrm{a}, \mathrm{true}, \mathrm{push}_1(*), 1)\\
\tau_2 &= (1, \varepsilon, \mathrm{true}, \mathrm{push}_1(\#), 2)\\
\tau_3 &= (2, \varepsilon, \mathrm{eq}(\#), \mathrm{down}, 2)\\
\tau_4 &= (2, \mathrm{b}, \mathrm{eq}(*), \mathrm{down}, 2)\\
\tau_5 &= (2, \varepsilon, \mathrm{bot}, \mathrm{up}_1, 3)\\
\tau_6 &= (3, \mathrm{c}, \mathrm{eq}(*), \mathrm{up}_1, 3)\\
\tau_7 &= (3, \varepsilon, \mathrm{eq}(\#), \mathrm{down}, 4)\\
\tau_8 &= (4, \mathrm{d}, \mathrm{eq}(*), \mathrm{down}, 4)\\
\tau_9 &= (4, \varepsilon, \mathrm{bot}, \mathrm{id}, 5)
\end{aligned}
$$

run for $a^2b^2c^2d^2$ and the stack:



some rules:

principle of crossing sequences

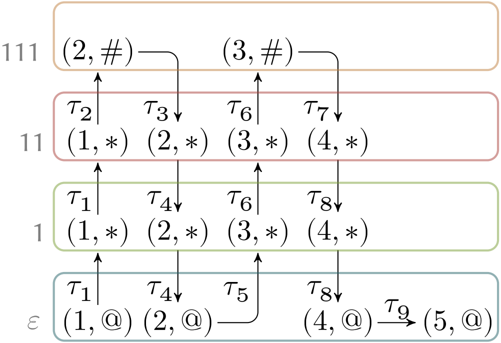
$$\langle 1, 5; @, @\rangle \to [\tau_1 x_1 \tau_4 \tau_5 x_2 \tau_8 \tau_9](\langle 1, 2, 3, 4; *, *, *\rangle)$$

k-TSL$_r$ $\subseteq$ k-MCFL (monadic example)

transitions:

$$\begin{aligned}
\tau_1 &= (1, \mathrm{a}, \mathrm{true}, \mathrm{push}_1(*), 1)\\
\tau_2 &= (1, \varepsilon, \mathrm{true}, \mathrm{push}_1(\#), 2)\\
\tau_3 &= (2, \varepsilon, \mathrm{eq}(\#), \mathrm{down}, 2)\\
\tau_4 &= (2, \mathrm{b}, \mathrm{eq}(*), \mathrm{down}, 2)\\
\tau_5 &= (2, \varepsilon, \mathrm{bot}, \mathrm{up}_1, 3)\\
\tau_6 &= (3, \mathrm{c}, \mathrm{eq}(*), \mathrm{up}_1, 3)\\
\tau_7 &= (3, \varepsilon, \mathrm{eq}(\#), \mathrm{down}, 4)\\
\tau_8 &= (4, \mathrm{d}, \mathrm{eq}(*), \mathrm{down}, 4)\\
\tau_9 &= (4, \varepsilon, \mathrm{bot}, \mathrm{id}, 5)
\end{aligned}$$

run for $a^2b^2c^2d^2$ and the stack:



some rules:

principle of crossing sequences

$$\langle 1,2,3,4; *,*,* \rangle \to [\tau_1 x_1 \tau_4, \tau_6 x_2 \tau_8](\langle 1,2,3,4; *,*,* \rangle)$$

$$\langle 1,5; @,@ \rangle \to [\tau_1 x_1 \tau_4 \tau_5 x_2 \tau_8 \tau_9](\langle 1,2,3,4; *,*,* \rangle)$$

k-TSL$_r$ $\subseteq$ k-MCFL (monadic example)

transitions:

$\tau_1 = (1, \mathrm{a}, \mathrm{true}, \mathrm{push}_1(*), 1)$

$\tau_2 = (1, \varepsilon, \mathrm{true}, \mathrm{push}_1(\#), 2)$

$\tau_3 = (2, \varepsilon, \mathrm{eq}(\#), \mathrm{down}, 2)$

$\tau_4 = (2, \mathrm{b}, \mathrm{eq}(*), \mathrm{down}, 2)$

$\tau_5 = (2, \varepsilon, \mathrm{bot}, \mathrm{up}_1, 3)$

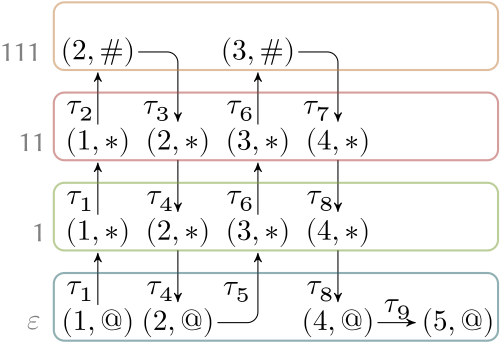
$\tau_6 = (3, \mathrm{c}, \mathrm{eq}(*), \mathrm{up}_1, 3)$

$\tau_7 = (3, \varepsilon, \mathrm{eq}(\#), \mathrm{down}, 4)$

$\tau_8 = (4, \mathrm{d}, \mathrm{eq}(*), \mathrm{down}, 4)$

$\tau_9 = (4, \varepsilon, \mathrm{bot}, \mathrm{id}, 5)$

run for $a^2b^2c^2d^2$ and the stack:



some rules:

principle of crossing sequences

$$\langle 1,2,3,4; *,*,* \rangle \to [\tau_2 x_1 \tau_3, \tau_6 x_2 \tau_7](\langle 2,2,3,3; \#,\#,\# \rangle)$$

$$\langle 1,2,3,4; *,*,* \rangle \to [\tau_1 x_1 \tau_4, \tau_6 x_2 \tau_8](\langle 1,2,3,4; *,*,* \rangle)$$

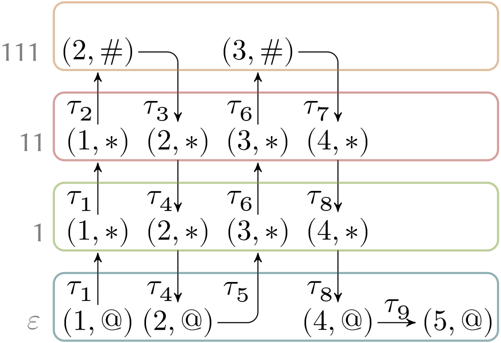
$$\langle 1,5; @,@ \rangle \to [\tau_1 x_1 \tau_4 \tau_5 x_2 \tau_8 \tau_9](\langle 1,2,3,4; *,*,* \rangle)$$

k-TSL$_\mathrm{r}$ $\subseteq$ k-MCFL (monadic example)

transitions:

$$\tau_1 = (1, \mathrm{a}, \mathrm{true}, \mathrm{push}_1(*), 1)$$
$$\tau_2 = (1, \varepsilon, \mathrm{true}, \mathrm{push}_1(\#), 2)$$
$$\tau_3 = (2, \varepsilon, \mathrm{eq}(\#), \mathrm{down}, 2)$$
$$\tau_4 = (2, \mathrm{b}, \mathrm{eq}(*), \mathrm{down}, 2)$$
$$\tau_5 = (2, \varepsilon, \mathrm{bot}, \mathrm{up}_1, 3)$$
$$\tau_6 = (3, \mathrm{c}, \mathrm{eq}(*), \mathrm{up}_1, 3)$$
$$\tau_7 = (3, \varepsilon, \mathrm{eq}(\#), \mathrm{down}, 4)$$
$$\tau_8 = (4, \mathrm{d}, \mathrm{eq}(*), \mathrm{down}, 4)$$
$$\tau_9 = (4, \varepsilon, \mathrm{bot}, \mathrm{id}, 5)$$

run for $a^2b^2c^2d^2$ and the stack:



some rules:

principle of crossing sequences

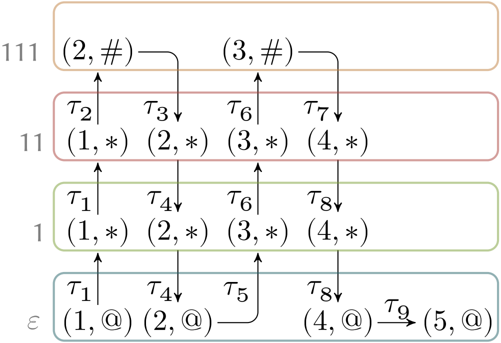
$$\langle 2,2,3,3;\#,\#,\#\rangle \to [\varepsilon,\varepsilon]()$$
$$\langle 1,2,3,4;*,*,*\rangle \to [\tau_2 x_1 \tau_3, \tau_6 x_2 \tau_7](\langle 2,2,3,3;\#,\#,\#\rangle)$$
$$\langle 1,2,3,4;*,*,*\rangle \to [\tau_1 x_1 \tau_4, \tau_6 x_2 \tau_8](\langle 1,2,3,4;*,*,*\rangle)$$
$$\langle 1,5;@,@\rangle \to [\tau_1 x_1 \tau_4 \tau_5 x_2 \tau_8 \tau_9](\langle 1,2,3,4;*,*,*\rangle)$$

k-TSL$_\mathrm{r}$ $\subseteq$ k-MCFL (monadic example)

transitions:

$$
\begin{aligned}
\tau_1 &= (1, \mathrm{a}, \mathrm{true}, \mathrm{push}_1(*), 1)\\
\tau_2 &= (1, \varepsilon, \mathrm{true}, \mathrm{push}_1(\#), 2)\\
\tau_3 &= (2, \varepsilon, \mathrm{eq}(\#), \mathrm{down}, 2)\\
\tau_4 &= (2, \mathrm{b}, \mathrm{eq}(*), \mathrm{down}, 2)\\
\tau_5 &= (2, \varepsilon, \mathrm{bot}, \mathrm{up}_1, 3)\\
\tau_6 &= (3, \mathrm{c}, \mathrm{eq}(*), \mathrm{up}_1, 3)\\
\tau_7 &= (3, \varepsilon, \mathrm{eq}(\#), \mathrm{down}, 4)\\
\tau_8 &= (4, \mathrm{d}, \mathrm{eq}(*), \mathrm{down}, 4)\\
\tau_9 &= (4, \varepsilon, \mathrm{bot}, \mathrm{id}, 5)
\end{aligned}
$$

run for $a^2b^2c^2d^2$ and the stack:



some rules:

principle of crossing sequences

$$\langle 2,2,3,3;\#,\#,\#\rangle \to [\varepsilon,\varepsilon]()$$

$$\langle 1,2,3,4;*,*,*\rangle \to [\; x_1 \;,\; \mathrm{c}\, x_2 \;](\langle 2,2,3,3;\#,\#,\#\rangle)$$

$$\langle 1,2,3,4;*,*,*\rangle \to [\,\mathrm{a}\, x_1\, \mathrm{b} \,,\; \mathrm{c}\, x_2\, \mathrm{d}\,](\langle 1,2,3,4;*,*,*\rangle)$$

$$\langle 1,5;@,@\rangle \to [\,\mathrm{a}\, x_1\, \mathrm{b} \;\; x_2\, \mathrm{d}\,](\langle 1,2,3,4;*,*,*\rangle)$$

k-TSL$_r$ $\subseteq$ k-MCFL (monadic example)

transitions:

$\tau_1 = (1, \mathrm{a}, \mathrm{true}, \mathrm{push}_1(*), 1)$

$\tau_2 = (1, \varepsilon, \mathrm{true}, \mathrm{push}_1(\#), 2)$

$\tau_3 = (2, \varepsilon, \mathrm{eq}(\#), \mathrm{down}, 2)$

$\tau_4 = (2, \mathrm{b}, \mathrm{eq}(*), \mathrm{down}, 2)$

$\tau_5 = (2, \varepsilon, \mathrm{bot}, \mathrm{up}_1, 3)$

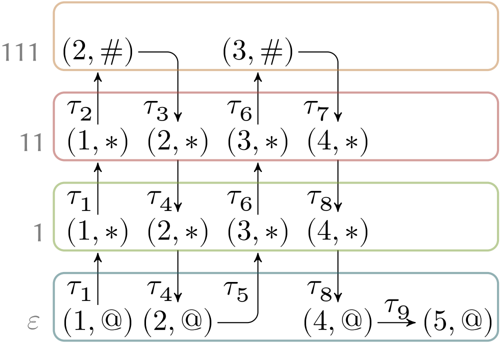
$\tau_6 = (3, \mathrm{c}, \mathrm{eq}(*), \mathrm{up}_1, 3)$

$\tau_7 = (3, \varepsilon, \mathrm{eq}(\#), \mathrm{down}, 4)$

$\tau_8 = (4, \mathrm{d}, \mathrm{eq}(*), \mathrm{down}, 4)$

$\tau_9 = (4, \varepsilon, \mathrm{bot}, \mathrm{id}, 5)$

run for $a^2b^2c^2d^2$ and the stack:



some rules:

principle of crossing sequences

$B \to [\varepsilon, \varepsilon]()$

$A \to [\ x_1\ ,\ \mathrm{c}\, x_2\ ](B)$

$A \to [\,\mathrm{a}\, x_1\, \mathrm{b}\, ,\ \mathrm{c}\, x_2\, \mathrm{d}\,](A)$

$S \to [\,\mathrm{a}\, x_1\, \mathrm{b}\quad x_2\, \mathrm{d}\ ](A)$

References

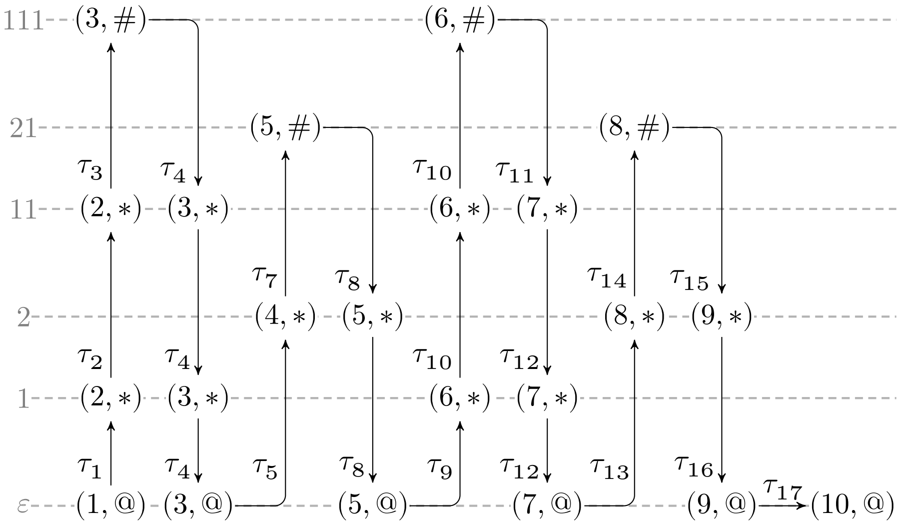
thread automata

- É. Villemonte de la Clergerie. “Parsing Mildly Context-Sensitive Languages with Thread Automata”. 2002.
- É. Villemonte de la Clergerie. “Parsing MCS languages with thread automata”. 2002.

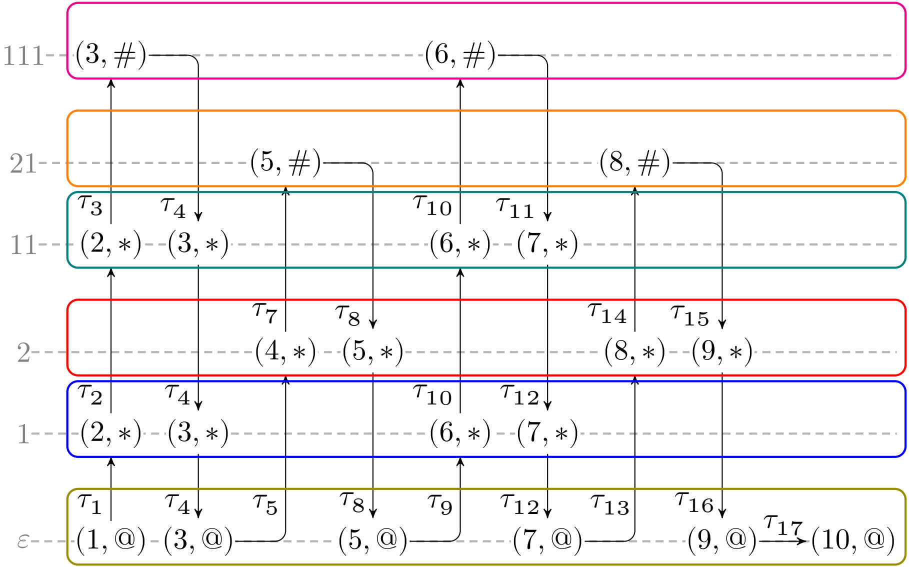
automata with storage

- D. Scott. “Some definitional suggestions for automata theory”. 1967.
- J. Engelfriet. “Context-free grammars with storage”. 2014.
- L. Herrmann and H. Vogler. “A Chomsky-Schützenberger Theorem for Weighted Automata with Storage”. 2015.

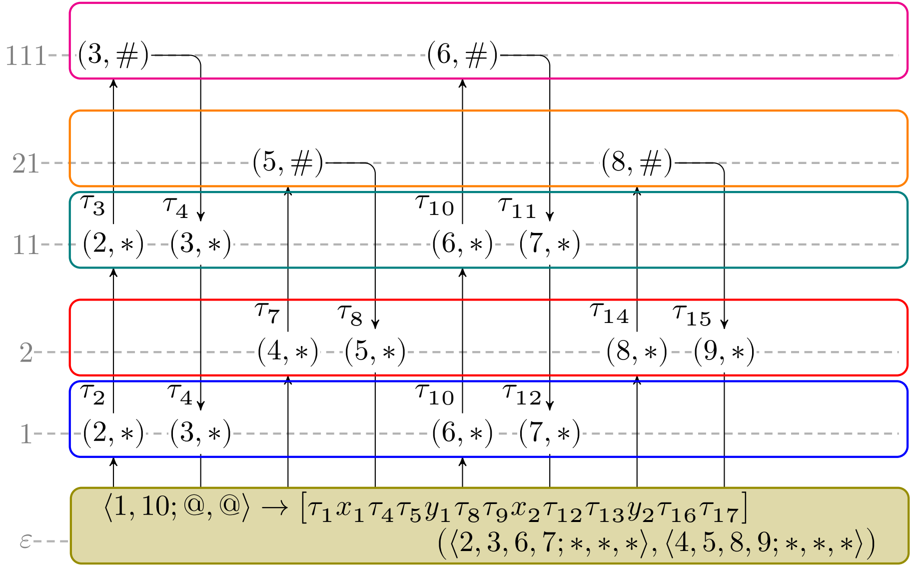
k-TSL$_r$ $\subseteq$ k-MCFL (non-monadic example)



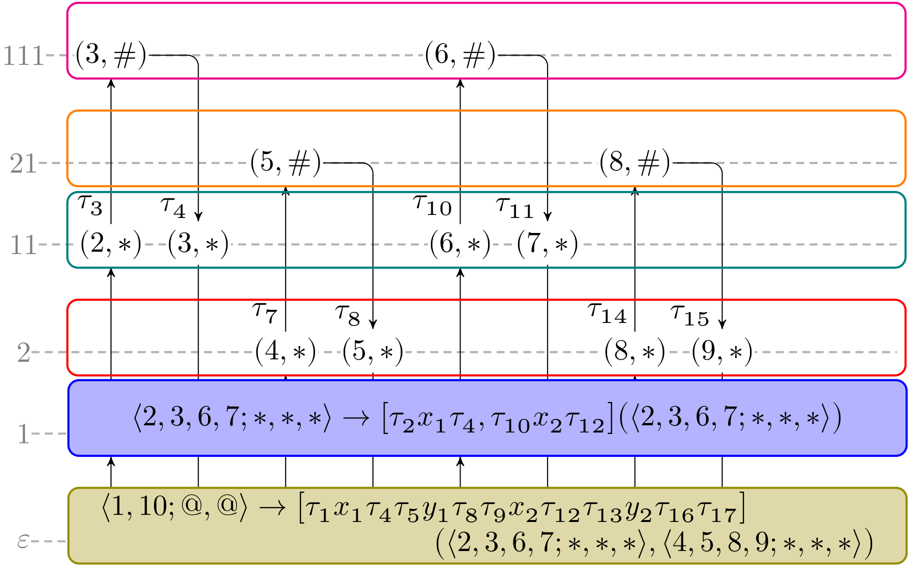
k-$\mathrm{TSL_r} \subseteq k$-MCFL (non-monadic example)



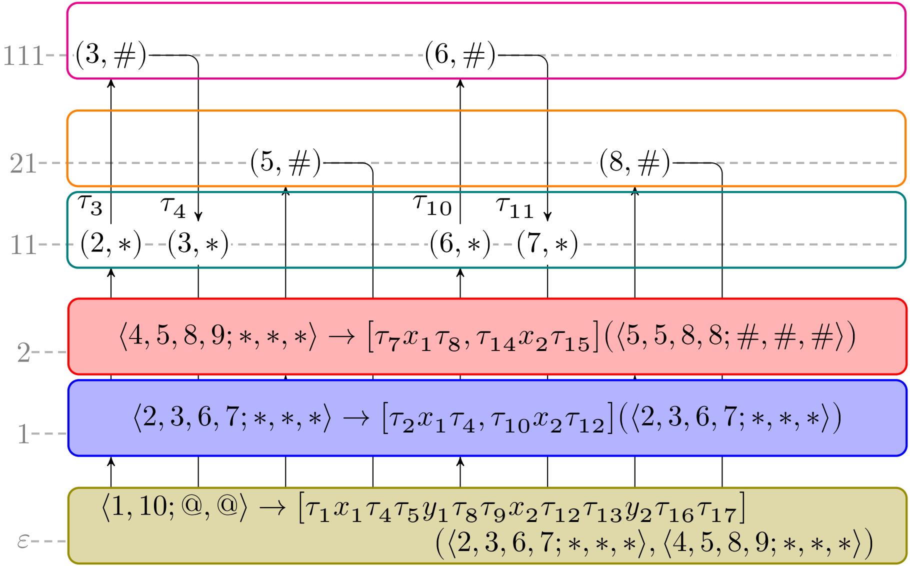
k-TSL$_\mathrm{r}$ $\subseteq$ k-MCFL (non-monadic example)



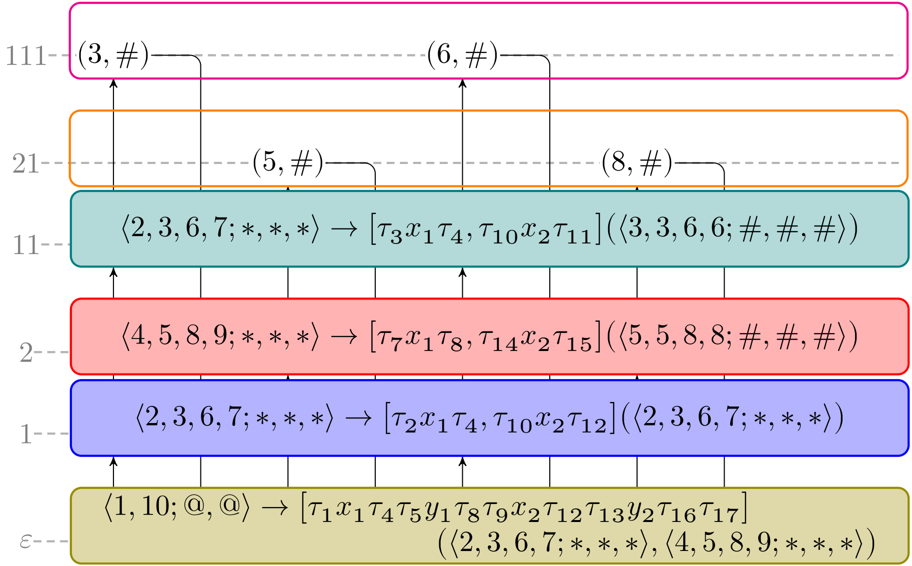
k-TSL$_\mathrm{r}$ $\subseteq$ k-MCFL (non-monadic example)



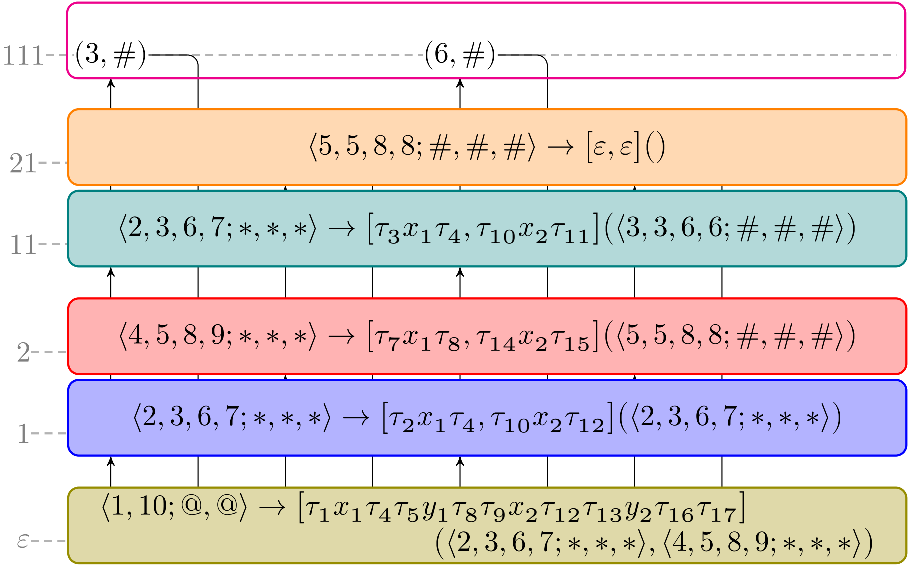
k-TSL$_\mathrm{r}$ $\subseteq$ k-MCFL (non-monadic example)



k-TSL$_\mathrm{r}$ $\subseteq$ k-MCFL (non-monadic example)



k-TSL$_\mathrm{r}$ $\subseteq$ k-MCFL (non-monadic example)



k-TSL$_{\mathrm{r}} \subseteq k$-MCFL (non-monadic example)

111 $\langle 3,3,6,6;\#,\#,\#\rangle \to [\varepsilon,\varepsilon]()$

21 $\langle 5,5,8,8;\#,\#,\#\rangle \to [\varepsilon,\varepsilon]()$

11 $\langle 2,3,6,7;*,*,*\rangle \to [\tau_3 x_1 \tau_4, \tau_{10} x_2 \tau_{11}](\langle 3,3,6,6;\#,\#,\#\rangle)$

2 $\langle 4,5,8,9;*,*,*\rangle \to [\tau_7 x_1 \tau_8, \tau_{14} x_2 \tau_{15}](\langle 5,5,8,8;\#,\#,\#\rangle)$

1 $\langle 2,3,6,7;*,*,*\rangle \to [\tau_2 x_1 \tau_4, \tau_{10} x_2 \tau_{12}](\langle 2,3,6,7;*,*,*\rangle)$

ε $\langle 1,10;@,@\rangle \to [\tau_1 x_1 \tau_4 \tau_5 y_1 \tau_8 \tau_9 x_2 \tau_{12} \tau_{13} y_2 \tau_{16} \tau_{17}]$
$(\langle 2,3,6,7;*,*,*\rangle, \langle 4,5,8,9;*,*,*\rangle)$

k-TSL$_\mathrm{r}$ $\subseteq$ k-MCFL (non-monadic example)