Regular Functions and Cost Register Automata

Rajeev Alur   Mukund Raghothaman

University of Pennsylvania

Monday 16th September, 2013

What are we studying?

Regular functions

What are we studying?

Regular functions

Languages, $\Sigma^* \to \mathtt{bool}$

What are we studying?

Regular functions

Languages, $\Sigma^* \rightarrow \texttt{bool}$

DFA

What are we studying?

Regular functions

Languages, $\Sigma^* \rightarrow$ `bool`   DFA
String transductions, $\Sigma^* \rightarrow \Gamma^*$ SST

What are we studying?

Cost register automata

Regular functions from Σ^* to integers $\mathbb{Z}$

Languages, $\Sigma^* \to$ `bool`	DFA
String transductions, $\Sigma^* \to \Gamma^*$	SST
Numerical functions, $\Sigma^* \to \mathbb{Z}$	?

Cost Register Automata
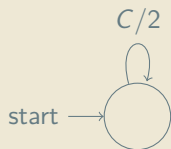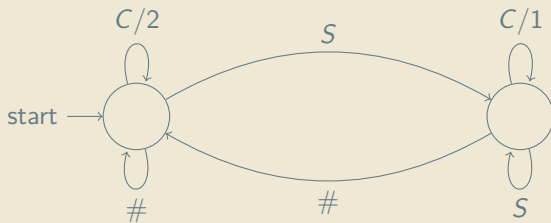
Finite automata with cost labels, a la Mealy machines

Cost Register Automata

Modelling a coffee shop: Attempt 1

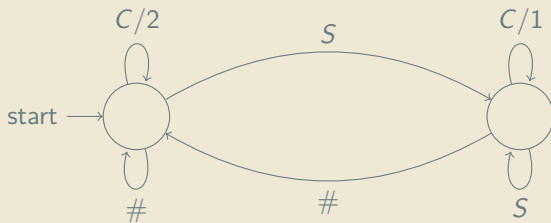## Finite automata with cost labels, a la Mealy machines

Cost Register Automata

Finite automata with cost labels, a la Mealy machines

$C/2$ S $C/1$

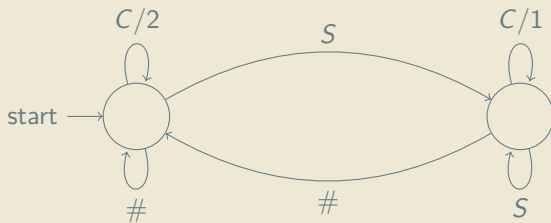start $\longrightarrow$

$\#$ $\#$ S

- Intuitive, analyzable

Cost Register Automata

Modelling a coffee shop: Attempt 1

Finite automata with cost labels, a la Mealy machines

- Intuitive, analyzable
- But not very expressive. . .

Cost Register Automata

What if the survey gives us a discount for coffee already purchased?

- Not possible if costs are paid up front
- Cost of an event cannot be influenced by later events

What if the survey gives us a discount for coffee already purchased?

- Not possible if costs are paid up front
- Cost of an event cannot be influenced by later events

Solution?

Cost Register Automata
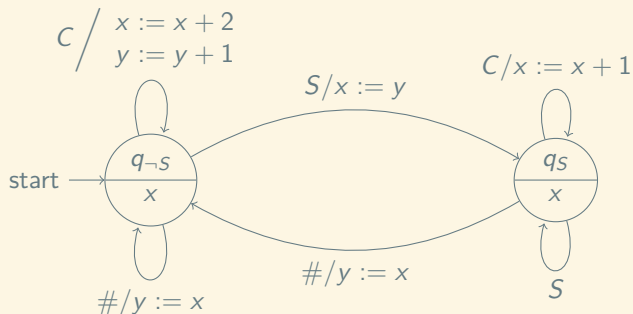
What if the survey gives us a discount for coffee already purchased?

- Not possible if costs are paid up front
- Cost of an event cannot be influenced by later events

Solution? Registers!

Cost Register Automata

Modelling a coffee shop: Attempt 2

Cost Register Automata

Properties

- Closure under linear combination, input reversal, and regular lookahead

- Fast equivalence procedure, decidable containment
- Equivalent to regular string-to-expression-tree transducers

Cost Register Automata

- Closure under linear combination, input reversal, and regular lookahead
  f^{rev} defined as $f^{rev}(\sigma) = f(\sigma^{rev})$ is regular when f is
- Fast equivalence procedure, decidable containment
- Equivalent to regular string-to-expression-tree transducers

Cost Register Automata

Properties

- Closure under linear combination, input reversal, and regular lookahead
  f^{rev} defined as $f^{rev}(\sigma) = f(\sigma^{rev})$ is regular when f is
- Fast equivalence procedure, decidable containment
- Equivalent to regular string-to-expression-tree transducers

What are we studying?

Regular functions from Σ^* to integers $\mathbb{Z}$

Languages, $\Sigma^* \to \texttt{bool}$ DFA

String transductions, $\Sigma^* \to \Gamma^*$ SST

Numerical functions, $\Sigma^* \to \mathbb{Z}$?

What are we studying?

Regular functions from Σ^* to integers $\mathbb{Z}$

Languages, $\Sigma^* \to \texttt{bool}$ DFA

String transductions, $\Sigma^* \to \Gamma^*$ SST

Numerical functions, $\Sigma^* \to \mathbb{Z}$ CRA

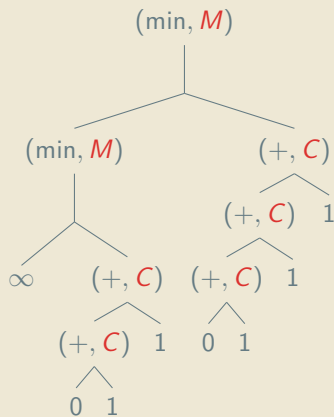# Constructing Expression Trees

Constructing Expression Trees

Regular functions

- Fix a (numerical) domain $\mathbb{D}$, and operations $G = \{+, \min, \times, \ldots\}$
- Recall MSO-definable string-to-tree transductions

Constructing Expression Trees

Fewest C-s between consecutive M-s ($CCMCCCM$)

Constructing Expression Trees

$\text{ACRA} \equiv \text{Regular Functions} (+c) \equiv \text{Regular Functions} (+) \equiv$
$\text{Copyless } \text{CRA} (+)$

$\text{CRA} (\min, +c) \equiv \text{Weighted automata}$

Analysis Problems

Questions to Ask
Analysis problems

Equivalence

- PTIME for ACRAs: Gaussian elimination
- Open for copyless CRA $(\min, +c)$ over $\mathbb{N}$

Containment and min-value

Questions to Ask

Registers

- ▶ Decision problem PSPACE-complete for ACRAs
- ▶ Similar techniques might extend to binary operations (SST)

States

- ▶ Open for ACRAs

Angluin style learning

Given oracles to

- evaluate $f(\sigma)$ given σ, and
- provide a counter-example σ so $f(\sigma) \neq f'(\sigma)$ given a proposal f',

compute a representation of f

Thank you! Questions?

Fin!