Synchronizing Data Words for Register Automata

Parvaneh Babari [1], Karin Quaas [1], Mahsa Shirmohammadi [2]

[1] Universität Leipzig, [2] University of Oxford

Highlights of Logic, Games and Automata
Brussels 2016

Data Words and Data Languages

Σ...a finite alphabet
D...an infinite data domain, e.g., $\mathbb{N}$, $\mathbb{R}_{\geq 0}$, `ASCII strings`

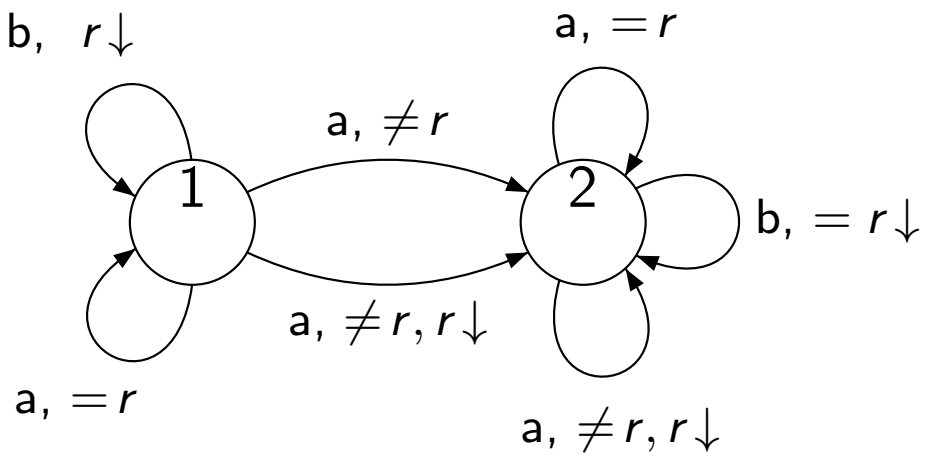
A **data word** is a finite sequence

$$(a_1, d_1)(a_2, d_2) \dots (a_n, d_n)$$

over $\Sigma \times D$.

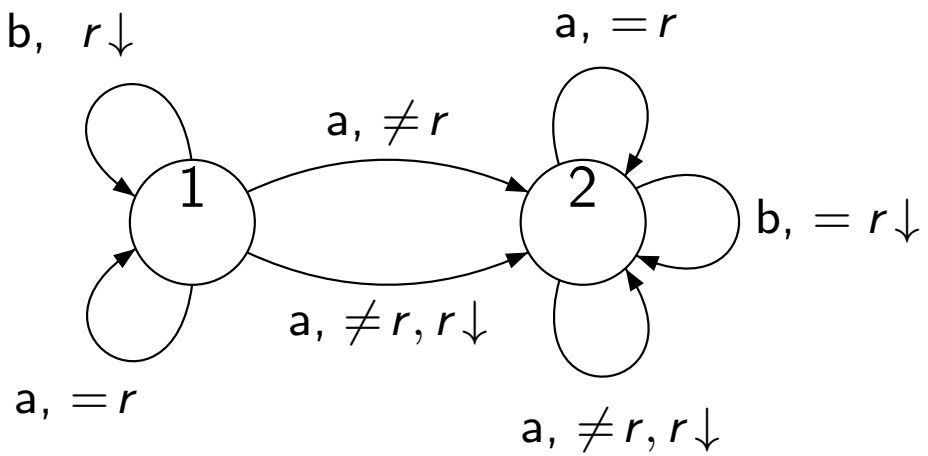
A **data language** is a subset of $(\Sigma \times D)^*$.

Register Automata (RA)



- r register variable
- $r\downarrow$ store current datum into r
- $=r$ is the current datum equal to value stored in r?
- $\neq r$ is the current datum different from value stored in r?

Register Automata (RA)

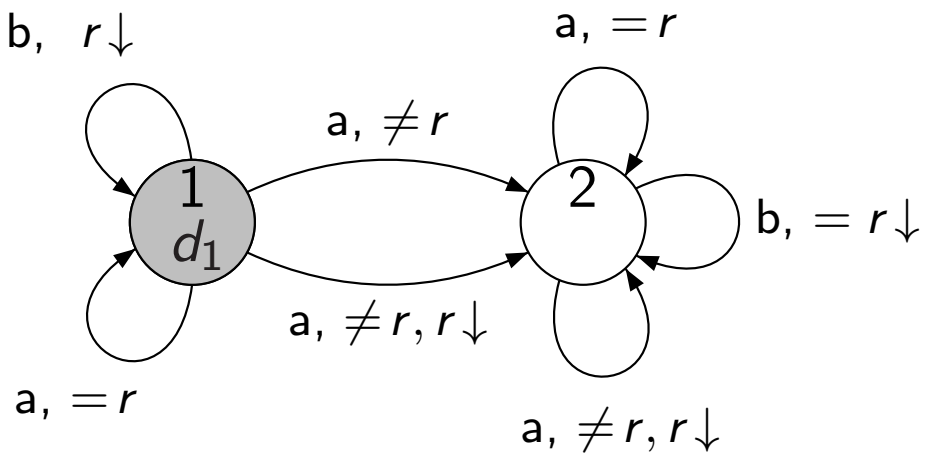


- r register variable
- $r\downarrow$ store current datum into r
- $=r$ is the current datum equal to value stored in r?
- $\neq r$ is the current datum different from value stored in r?

$w = (a, d_1)(a, d_2)(a, d_3)$

$$(1, d_1) \xrightarrow{(a,d_1)} (1, d_1) \xrightarrow{(a,d_2)} (2, d_2) \xrightarrow{(a,d_3)} (2, d_3)$$

Register Automata (RA)

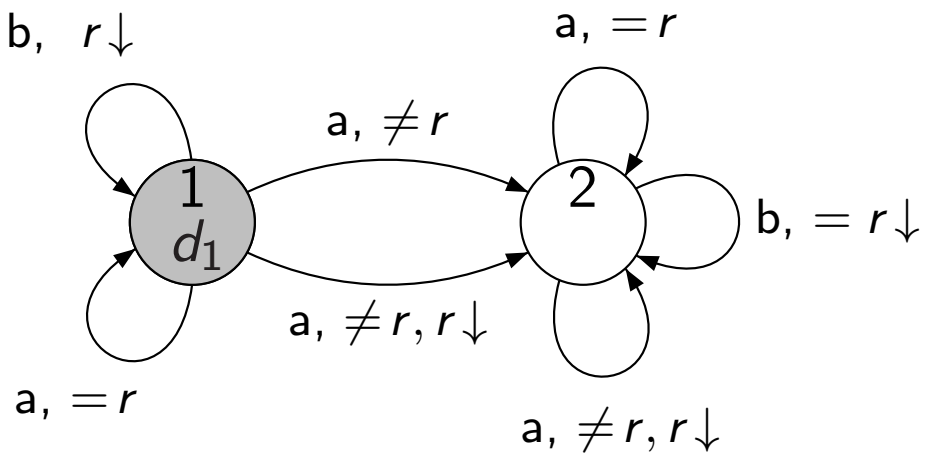


- r register variable
- $r\downarrow$ store current datum into r
- $=r$ is the current datum equal to value stored in r?
- $\neq r$ is the current datum different from value stored in r?

$w = (a, d_1)(a, d_2)(a, d_3)$

$$(1, d_1) \xrightarrow{(a,d_1)} (1, d_1) \xrightarrow{(a,d_2)} (2, d_2) \xrightarrow{(a,d_3)} (2, d_3)$$

Register Automata (RA)

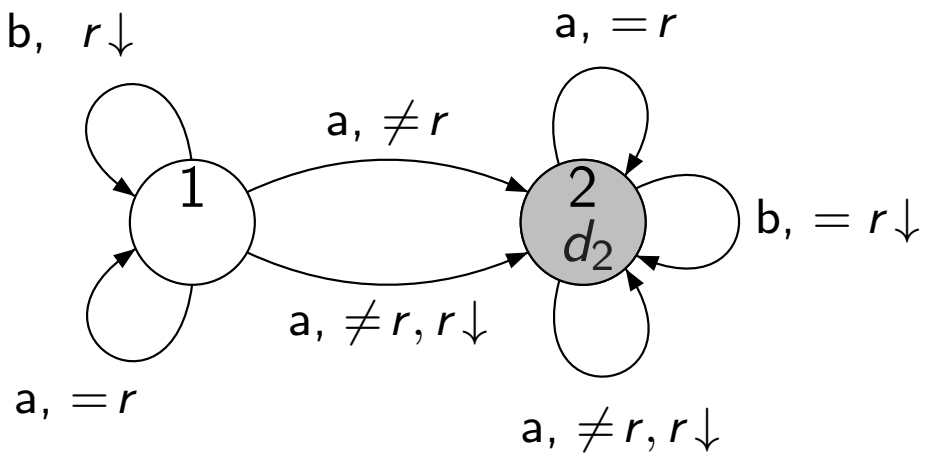


- r register variable
- $r\downarrow$ store current datum into r
- $=r$ is the current datum equal to value stored in r?
- $\neq r$ is the current datum different from value stored in r?

$w = (a, d_1)(a, d_2)(a, d_3)$

$$(1, d_1) \xrightarrow{(a,d_1)} (1, d_1) \xrightarrow{(a,d_2)} (2, d_2) \xrightarrow{(a,d_3)} (2, d_3)$$

Register Automata (RA)

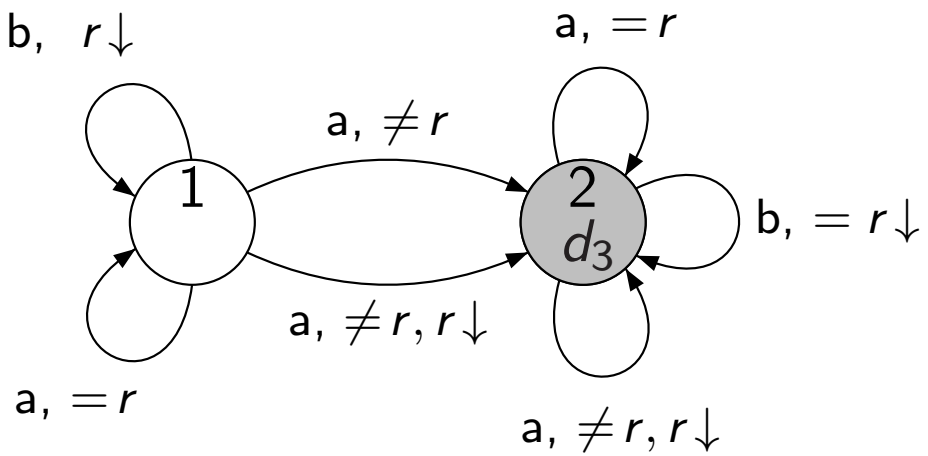


- r register variable
- $r\downarrow$ store current datum into r
- $=r$ is the current datum equal to value stored in r?
- $\neq r$ is the current datum different from value stored in r?

$w = (a, d_1)(a, d_2)(a, d_3)$

$$(1, d_1) \xrightarrow{(a,d_1)} (1, d_1) \xrightarrow{(a,d_2)} (2, d_2) \xrightarrow{(a,d_3)} (2, d_3)$$

Register Automata (RA)



- r register variable
- $r\downarrow$ store current datum into r
- $=r$ is the current datum equal to value stored in r?
- $\neq r$ is the current datum different from value stored in r?

$w = (a, d_1)(a, d_2)(a, d_3)$

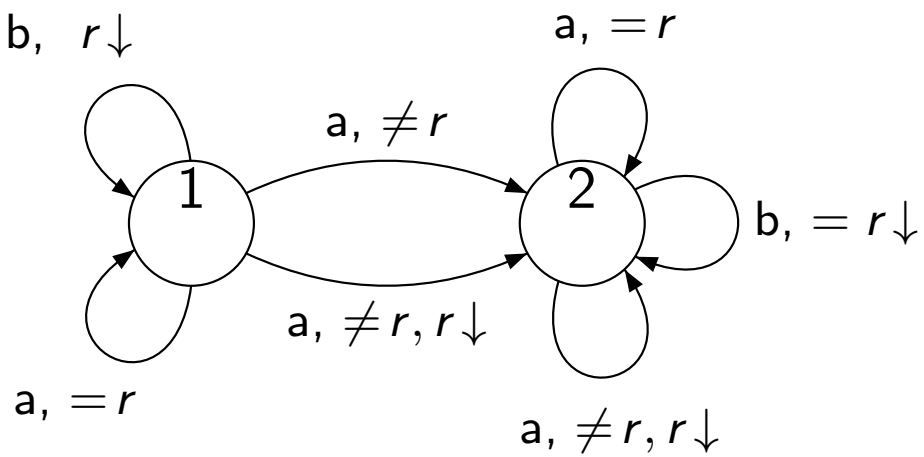
$$(1, d_1) \xrightarrow{(a,d_1)} (1, d_1) \xrightarrow{(a,d_2)} (2, d_2) \xrightarrow{(a,d_3)} (2, d_3)$$

Synchronizing Data Words for Register Automata

A data word **w** is **synchronizing for register automaton** $\mathcal{A}$ if **w** guides all configurations of $\mathcal{A}$ to the same configuration.

Synchronizing Data Words for Register Automata

A data word **w** is **synchronizing for register automaton** $\mathcal{A}$ if **w** guides all configurations of $\mathcal{A}$ to the same configuration.



$$\{1,2\} \times D$$
$$\downarrow (a, d_1)$$
$$\{(1,d_1),(2,d_1)\} \cup (\{2\} \times D \backslash \{d_1\})$$
$$\downarrow (a, d_2)$$
$$\{(2,d_1),(2,d_2)\}$$
$$\downarrow (a, d_3)$$
$$\{(2,d_3)\}$$

The Synchronizing Problem

	Synchronizing Problem
Instance:	Register automaton $\mathcal{A}$
Question:	Does there exist a synchronizing data word for $\mathcal{A}$?

The Synchronizing Problem

	Synchronizing Problem
Instance:	Register automaton $\mathcal{A}$
Question:	Does there exist a synchronizing data word for $\mathcal{A}$?

Results	
NRA	undecidable
1-NRA	Ackermann-complete
DRA	PSPACE-complete
1-DRA	NLOGSPACE-complete

The Synchronizing Problem

	Synchronizing Problem	Non-Universality Problem
Instance:	Register automaton $\mathcal{A}$	Register automaton $\mathcal{A}$
Question:	Does there exist a synchronizing data word for $\mathcal{A}$?	Does there exist some data word that is not accepted by $\mathcal{A}$?

Results	
NRA 1-NRA	undecidable Ackermann-complete
DRA 1-DRA	PSPACE-complete NLOGSPACE-complete

The Synchronizing Problem

	Synchronizing Problem	Non-Universality Problem
Instance:	Register automaton $\mathcal{A}$	Register automaton $\mathcal{A}$
Question:	Does there exist a synchronizing data word for $\mathcal{A}$?	Does there exist some data word that is not accepted by $\mathcal{A}$?

Results		
NRA	undecidable	undecidable [1]
1-NRA	Ackermann-complete	Ackermann-complete [2,3]
DRA	PSPACE-complete	PSPACE-complete [2]
1-DRA	NLOGSPACE-complete	NLOGSPACE-complete

[1] Neven, Schwentick, Vianu: Finite state machines for strings over infinite alphabets, 2004.
[2] Demri, Lazic: LTL with the freeze quantifier and register automata, 2009.
[3] Figueira[2], Schmitz, Schnoebelen: Ackermannian- and primitive recursive bounds with Dickson's Lemma, 2011.

The Synchronizing Problem

	Synchronizing Problem	Non-Universality Problem
Instance:	Register automaton $\mathcal{A}$	Register automaton $\mathcal{A}$
Question:	Does there exist a synchronizing data word for $\mathcal{A}$?	Does there exist some data word that is not accepted by $\mathcal{A}$?

Results		
NRA	undecidable	undecidable [1]
1-NRA	Ackermann-complete	Ackermann-complete [2,3]
DRA	PSPACE-complete	PSPACE-complete [2]
1-DRA	NLOGSPACE-complete	NLOGSPACE-complete

[1] Neven, Schwentick, Vianu: Finite state machines for strings over infinite alphabets, 2004.
[2] Demri, Lazic: LTL with the freeze quantifier and register automata, 2009.
[3] Figueira[2], Schmitz, Schnoebelen: Ackermannian- and primitive recursive bounds with Dickson's Lemma, 2011.

The Bounded Synchronizing Problem

	Bounded Synchronizing Problem	Bounded Non-Universality Problem
Instance:	Register automaton $\mathcal{A}$, $N \in \mathbb{N}$ (in binary)	Register automaton $\mathcal{A}$, $N \in \mathbb{N}$ (in binary)
Question:	Does there exist a synchronizing data word for $\mathcal{A}$ with length bounded by N ?	Does there exist some data word that is not accepted by $\mathcal{A}$ and length bounded by N ?

The Bounded Synchronizing Problem

	Bounded Synchronizing Problem	Bounded Non-Universality Problem
Instance:	Register automaton $\mathcal{A}$, $N \in \mathbb{N}$ (in binary)	Register automaton $\mathcal{A}$, $N \in \mathbb{N}$ (in binary)
Question:	Does there exist a synchronizing data word for $\mathcal{A}$ with length bounded by N ?	Does there exist some data word that is not accepted by $\mathcal{A}$ and length bounded by N ?

Results		
NRA	NEXPTIME-complete	NEXPTIME-complete
1-NRA	NEXPTIME-complete	NEXPTIME-complete

NEXPTIME-completeness of the Bounded Non-Universality Problem

- NEXPTIME-membership: guess a witness for bounded non-universality and check in exponential time

NEXPTIME-completeness of the Bounded Non-Universality Problem

- NEXPTIME-membership: guess a witness for bounded non-universality and check in exponential time
- NEXPTIME-hardness: reduction from the following NEXPTIME-complete [1] problem:

NEXPTIME-completeness of the Bounded Non-Universality Problem

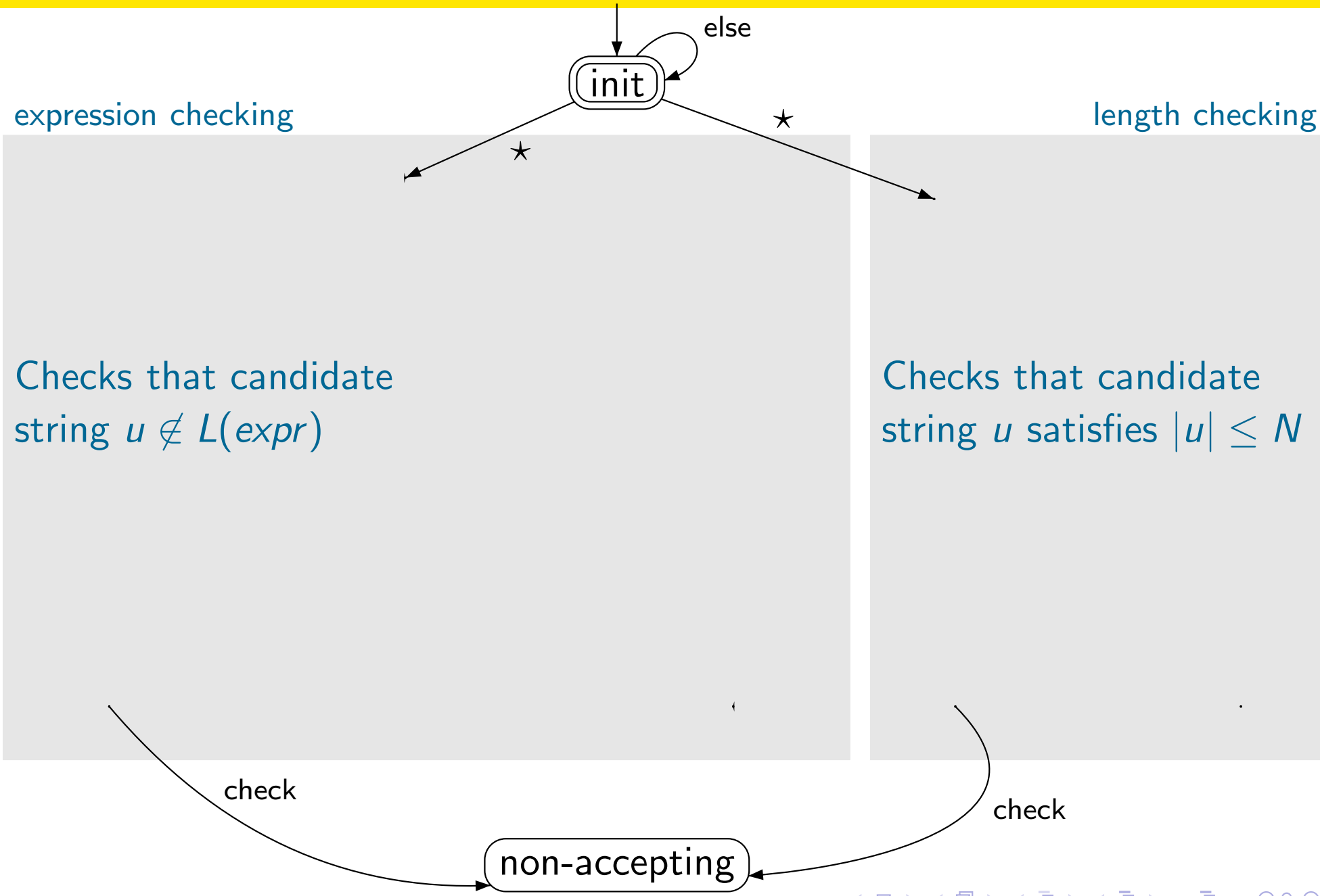
- NEXPTIME-membership: guess a witness for bounded non-universality and check in exponential time
- NEXPTIME-hardness: reduction from the following NEXPTIME-complete [1] problem:

	Bounded Non-Universality Problem for Regular-Like Expressions
Instance:	Regular-like expression expr (built from $\{a, b\}$ and operations $\cdot, +, ^2$), $N \in \mathbf{N}$ (in binary)
Question:	Does there exist some string u with $\|u\| \leq N$ and $u \notin L(expr)$?

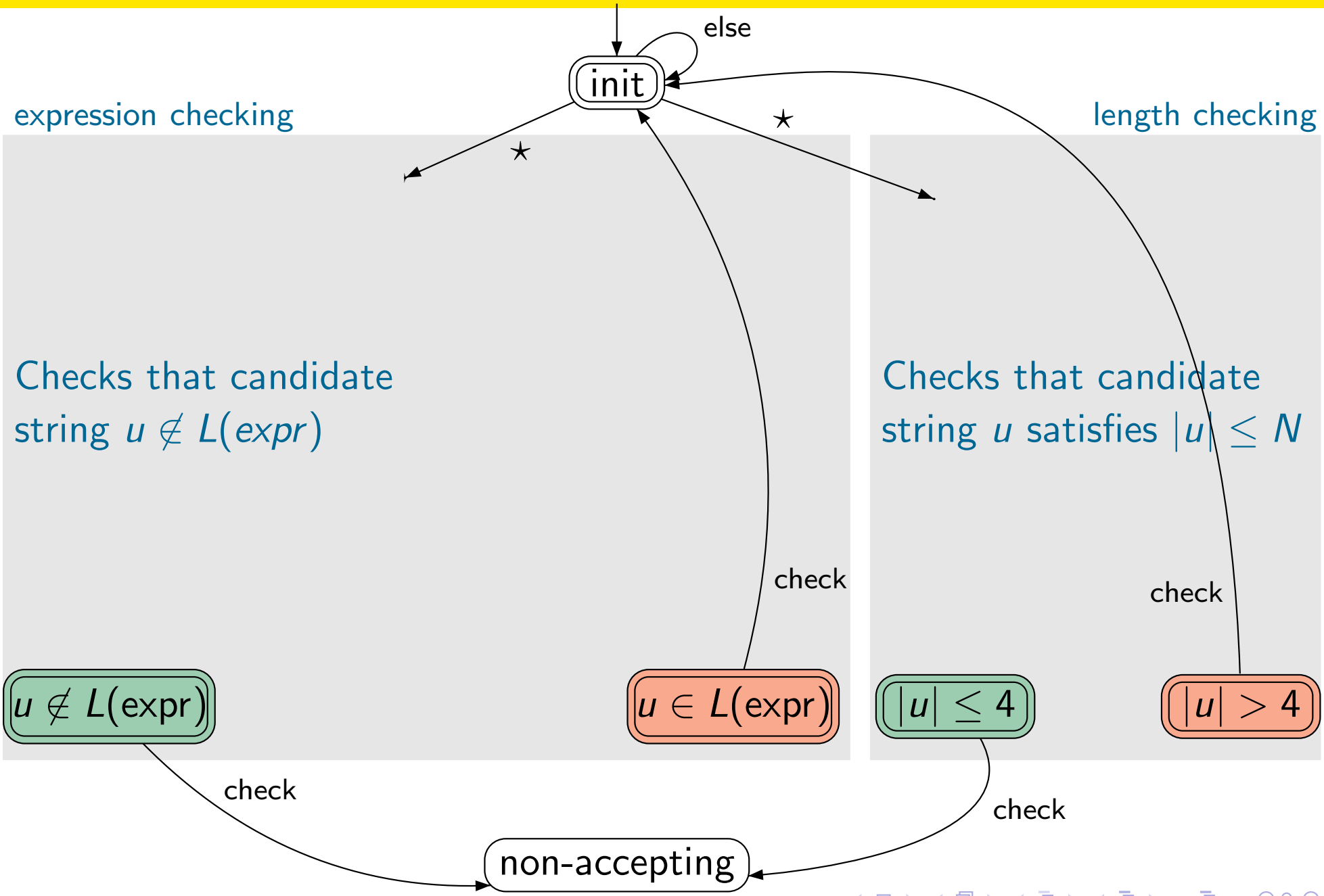
[1] Stockmeyer, Meyer: Word Problems requiring exponential time, 1973.

$\mathrm{expr} = (aa + a)^2, \quad N = 100$

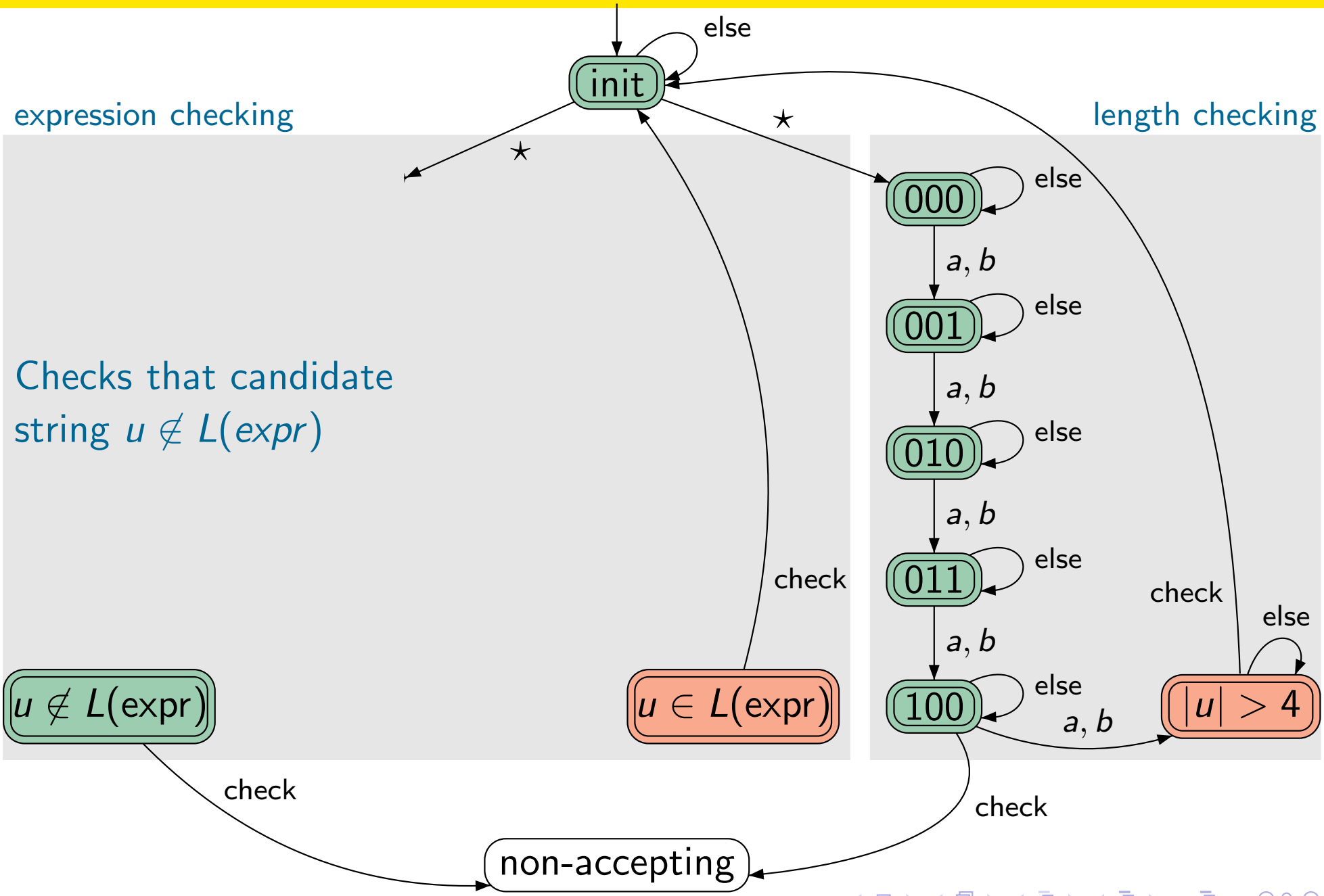
$\mathrm{expr} = (aa + a)^2, \quad N = 100$



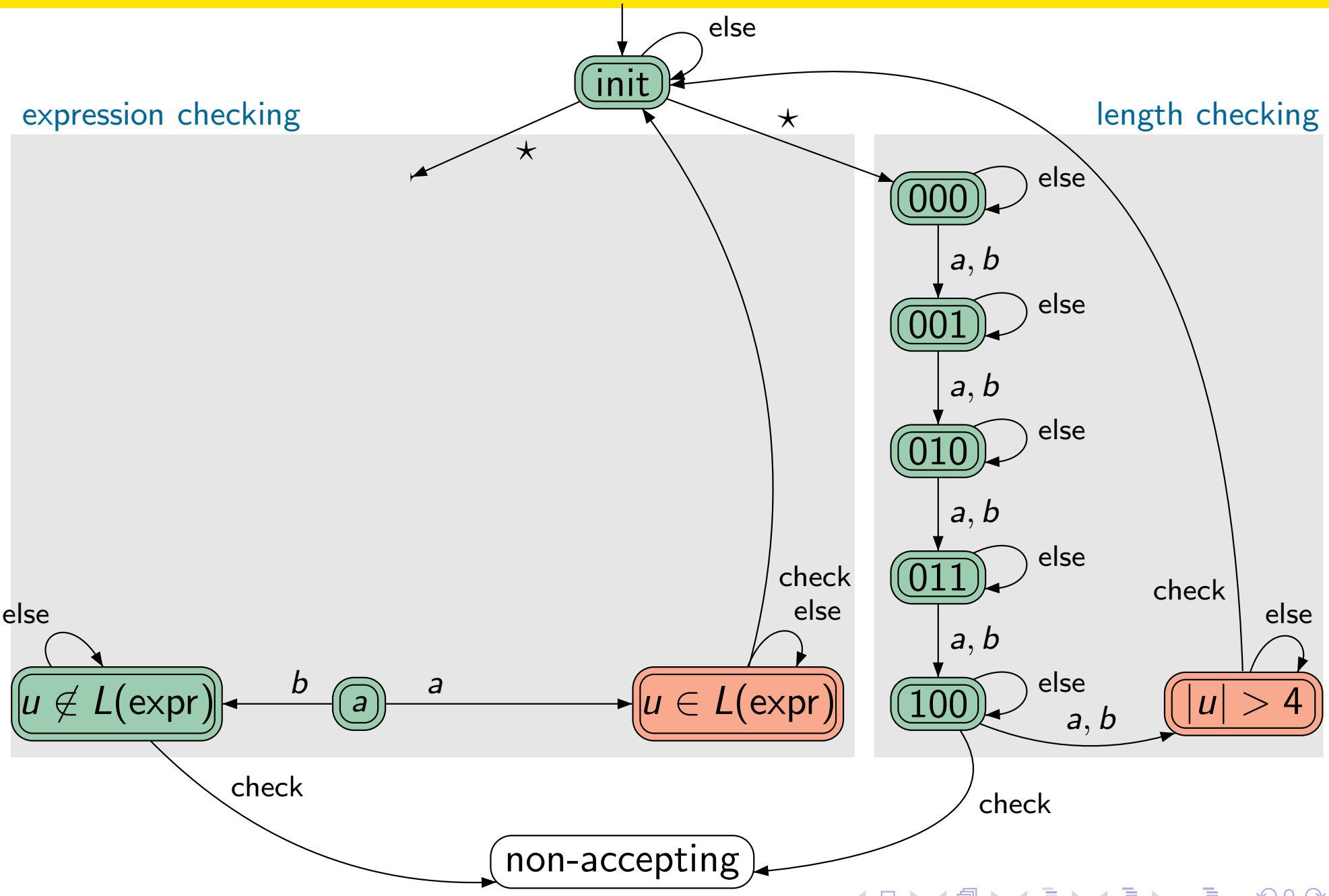
$\text{expr} = (aa + a)^2, \quad N = 100$



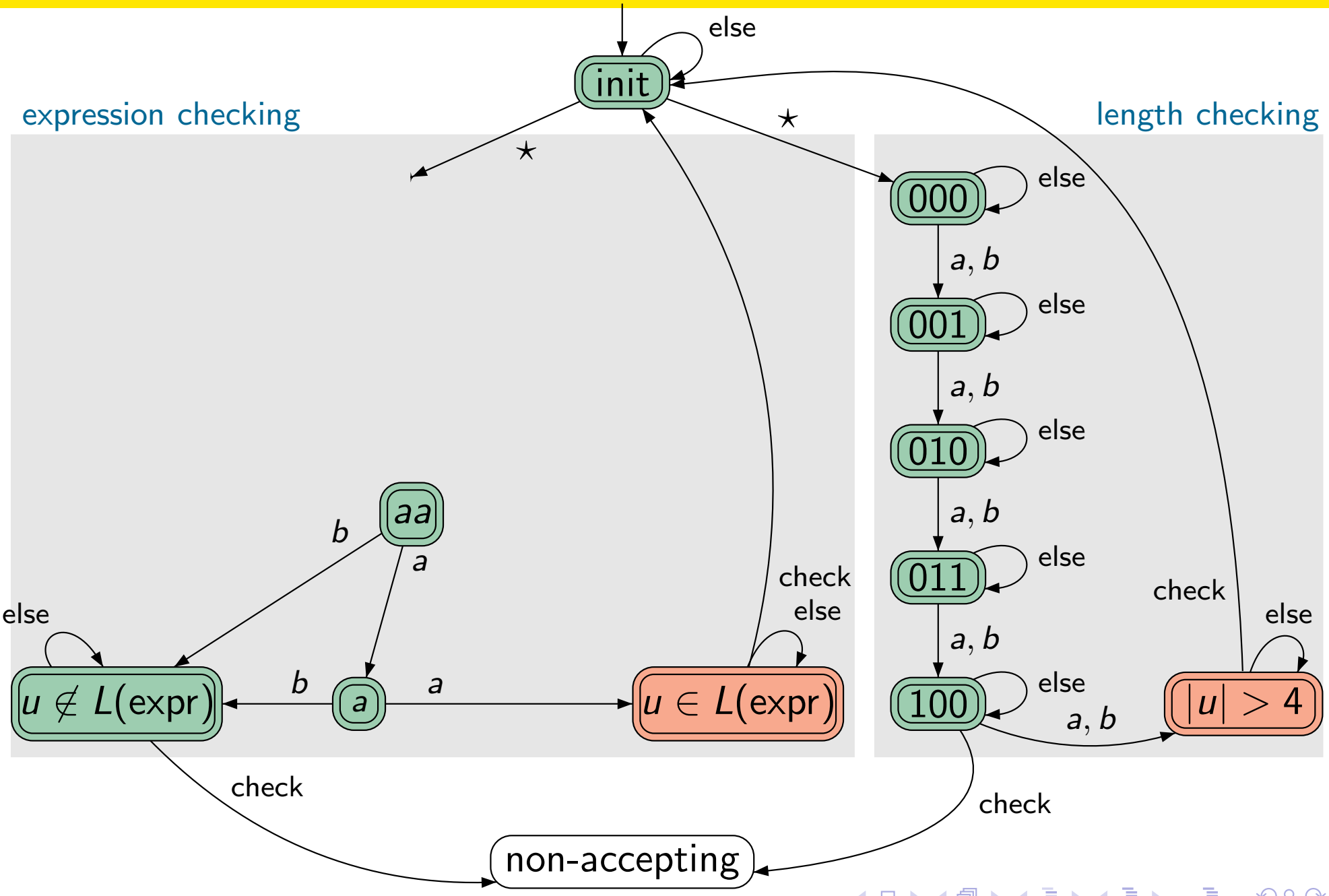
$\text{expr} = (aa + a)^2, \quad N = 100$



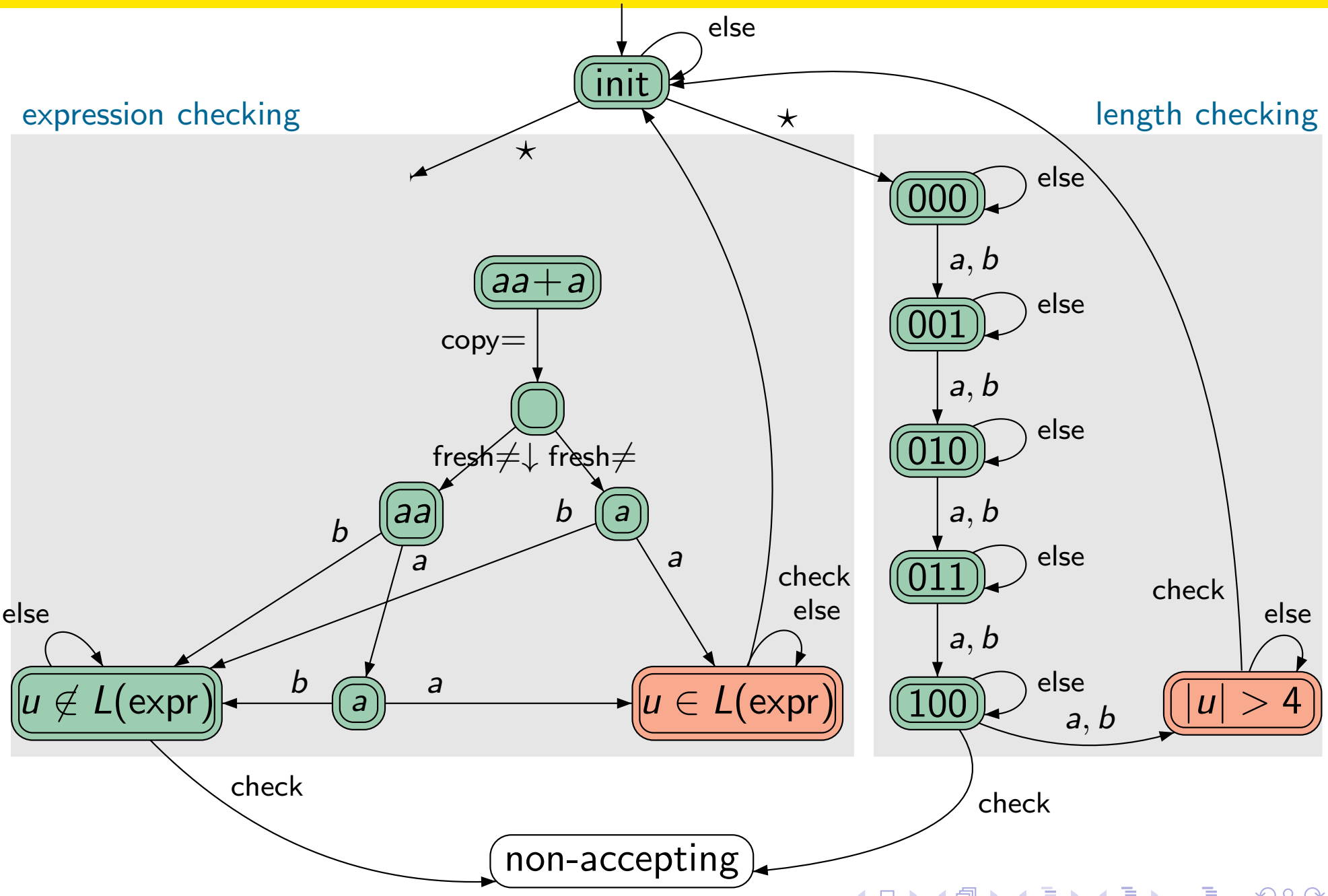
$\mathrm{expr} = (aa + a)^2, \quad N = 100$



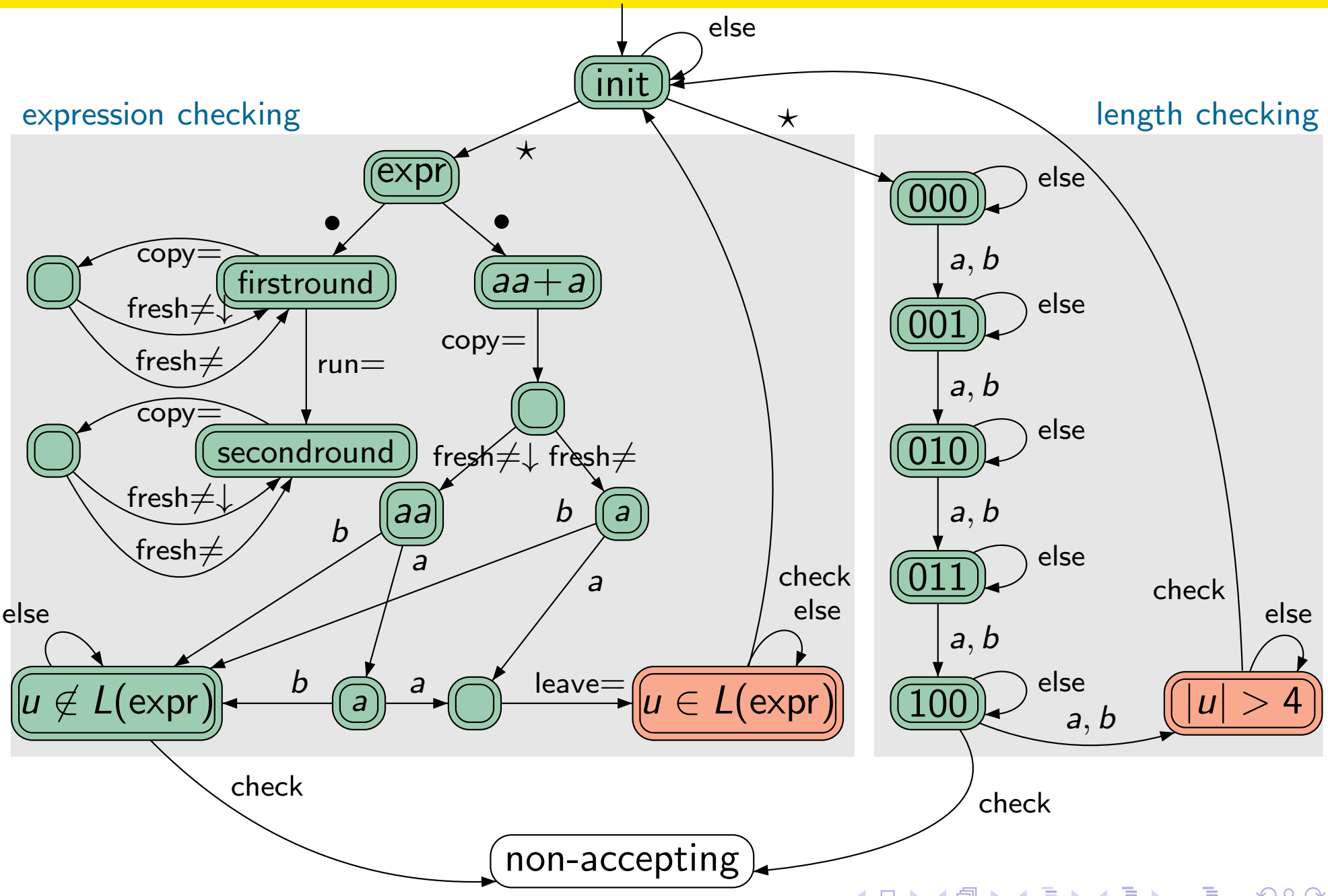
$\text{expr} = (aa + a)^2, \quad N = 100$



$\mathrm{expr} = (aa + a)^2, \quad N = 100$



$\mathrm{expr} = (aa + a)^2, \quad N = 100$



Summary and Open Problems

The synchronizing/non-universality problem is

- PSPACE-complete for DRA
- NLOGSPACE-complete for 1-DRA
- undecidable for NRA
- Ackermann-complete for 1-NRA

Summary and Open Problems

The synchronizing/non-universality problem is

- PSPACE-complete for DRA
- NLOGSPACE-complete for 1-DRA
- undecidable for NRA
- Ackermann-complete for 1-NRA

The bounded synchronizing/non-universality problem for NRA is NEXPTIME-complete.

Summary and Open Problems

The synchronizing/non-universality problem is

- PSPACE-complete for DRA
- NLOGSPACE-complete for 1-DRA
- undecidable for NRA
- Ackermann-complete for 1-NRA

The bounded synchronizing/non-universality problem for NRA is NEXPTIME-complete.

Open: Precise complexity for the bounded synchronizing/non-universality problem for DRA.