New Techniques for Universality in Unambiguous Register Automata

Wojciech Czerwiński

Antoine Mottet

Karin Quaas

Plan

Plan

- basic notions

Plan

- basic notions
- motivation and context

Plan

- basic notions
- motivation and context
- results summary

Plan

- basic notions
- motivation and context
- results summary
- illustration of our techniques

Register automata

Register automata

Finite automata with registers

Register automata

Finite automata with registers

Read finite words over infinite domain ($\mathbb{N}$)

Register automata

Finite automata with registers

Read finite words over infinite domain ($\mathbb{N}$)

Register can keep data from $\mathbb{N}$

Register automata

Finite automata with registers

Read finite words over infinite domain ($\mathbb{N}$)

Register can keep data from $\mathbb{N}$

Transition:

Register automata

Finite automata with registers

Read finite words over infinite domain ($\mathbb{N}$)

Register can keep data from $\mathbb{N}$

Transition: changes current location

Register automata

Finite automata with registers

Read finite words over infinite domain ($\mathbb{N}$)

Register can keep data from $\mathbb{N}$

Transition: changes current location

is fired if condition is satisfied

Register automata

Finite automata with registers

Read finite words over infinite domain ($\mathbb{N}$)

Register can keep data from $\mathbb{N}$

Transition: changes current location

is fired if condition is satisfied

equalities/nonequalities
of register data and input data

Register automata

Finite automata with registers

Read finite words over infinite domain ($\mathbb{N}$)

Register can keep data from $\mathbb{N}$

Transition: changes current location

is fired if condition is satisfied

Register automata

Finite automata with registers

Read finite words over infinite domain ($\mathbb{N}$)

Register can keep data from $\mathbb{N}$

Transition:

changes current location

is fired if condition is satisfied

assigns new register values (no guessing)

Register automata

Finite automata with registers

Read finite words over infinite domain ($\mathbb{N}$)

Register can keep data from $\mathbb{N}$

Transition: changes current location

is fired if condition is satisfied

assigns new register values (no guessing)

Example: words with first data equal only to last data

Unambiguity

Unambiguity

for each word there is at most one accepting run

Unambiguity

for each word there is at most one accepting run

many problems become simpler:

Unambiguity

for each word there is at most one accepting run

many problems become simpler:

universality for UFA (PTime)

Unambiguity

for each word there is at most one accepting run

many problems become simpler:

universality for UFA (PTime)

equivalence for UFA (PTime)

Unambiguity

for each word there is at most one accepting run

many problems become simpler:

universality for UFA (PTime)

equivalence for UFA (PTime)

universality for VASS (ExpSpace)

Problems

Problems

Emptiness not easier

Problems

Emptiness not easier

Universality is easier

Problems

Emptiness not easier

Universality is easier

Equivalence is easier

Problems

Emptiness not easier

Universality is easier

Equivalence is easier

Inclusion is easier

Problems

Emptiness not easier

Universality is easier

Equivalence is easier

Inclusion is easier

First step: universality problem

What is known

What is known

Universality is:

What is known

Universality is:

- undecidable for 2-RA [Kaminsky, Francez]

What is known

Universality is:

- undecidable for 2-RA [Kaminsky, Francez]
- Ackermann-hard for 1-RA [Figueira et. al.]

What is known

Universality is:

- undecidable for 2-RA [Kaminsky, Francez]
- Ackermann-hard for 1-RA [Figueira et. al.]

with unambiguity assumption equivalence is:

What is known

Universality is:

- undecidable for 2-RA [Kaminsky, Francez]
- Ackermann-hard for 1-RA [Figueira et. al.]

with unambiguity assumption equivalence is:

- 2ExpSpace for URA [Mottet, Quaas]

What is known

Universality is:

- undecidable for 2-RA [Kaminsky, Francez]
- Ackermann-hard for 1-RA [Figueira et. al.]

with unambiguity assumption equivalence is:

- 2ExpSpace for URA [Mottet, Quaas]
- 2ExpTime for URA [Barloy, Clemente]

What is known

Universality is:

- undecidable for 2-RA [Kaminsky, Francez]
- Ackermann-hard for 1-RA [Figueira et. al.]

with unambiguity assumption equivalence is:

- 2ExpSpace for URA [Mottet, Quaas]
- 2ExpTime for URA [Barloy, Clemente]
- ExpSpace for URA [we]

What is known

Universality is:

- undecidable for 2-RA [Kaminsky, Francez]
- Ackermann-hard for 1-RA [Figueira et. al.]

with unambiguity assumption equivalence is:

- 2ExpSpace for URA [Mottet, Quaas]
- 2ExpTime for URA [Barloy, Clemente]
- ExpSpace for URA [we]
- ExpTime for URA [Bojańczyk, Klin, Moerman]

Results

Results

Theorem

Results

Theorem

1) The inclusion problem for URA is in ExpSpace

Results

Theorem

1) The inclusion problem for URA is in ExpSpace

2) The inclusion problem for k-URA is in PSpace

Results

Theorem

1) The inclusion problem for URA is in ExpSpace

2) The inclusion problem for k-URA is in PSpace

3) The universality problem for 1-URA($\geq$) is in PSpace

Results

Theorem

1) The inclusion problem for URA is in ExpSpace

2) The inclusion problem for k-URA is in PSpace

3) The universality problem for 1-URA($\geq$) is in PSpace

4) The inclusion problem of GRA in 1-GURA is in ExpSpace

Techniques

Techniques

2) The inclusion problem for k-URA is in PSpace

Techniques

2) The inclusion problem for k-URA is in PSpace

2') The universality problem for 1-URA is in PSpace

Techniques

2) The inclusion problem for k-URA is in PSpace

2') The universality problem for 1-URA is in PSpace

Motten, Quaas

ExpSpace:

Techniques

2) The inclusion problem for k-URA is in PSpace

2') The universality problem for 1-URA is in PSpace

Motten, Quaas ExpSpace:

if for some configuration some two data have the same locations then remove one of them

Techniques

2) The inclusion problem for k-URA is in PSpace

2') The universality problem for 1-URA is in PSpace

Motten, Quaas
ExpSpace:

if for some configuration some two data
have the same locations
then remove one of them

we
PSpace:

Techniques

2) The inclusion problem for k-URA is in PSpace

2') The universality problem for 1-URA is in PSpace

Motten, Quaas
ExpSpace:

if for some configuration some two data
have the same locations
then remove one of them

we
PSpace:

in universal 1-URA in each reachable
configuration each location
has at most one datum

Proof

Proof

Lemma

In universal 1-URA in each reachable configuration each location has at most one datum

Proof

Lemma

In universal 1-URA in each reachable configuration each location has at most one datum

Assume that q(1) and q(2) are present in a reachable configuration C

Proof

Lemma

In universal 1-URA in each reachable configuration each location has at most one datum

Assume that q(1) and q(2) are present in a reachable configuration C

We aim at contradiction

Proof

Lemma

In universal 1-URA in each reachable configuration each location has at most one datum

Assume that q(1) and q(2) are present in a reachable configuration C

We aim at contradiction

Both q(1) and q(2) need to accept some words

Proof cont.

Proof cont.

Case 1: $q(1)$ accepts word without 1

Proof cont.

Case 1: q(1) accepts word without 1

Let q(1) accept 2343

Proof cont.

Case 1: q(1) accepts word without 1

Let q(1) accept 2343

Then q(1) accepts 5343

Proof cont.

Case 1: q(1) accepts word without 1

Let q(1) accept 2343

Then q(1) accepts 5343

Then q(2) accepts 5343

Proof cont.

Case 1: q(1) accepts word without 1

Let q(1) accept 2343

Then q(1) accepts 5343

Then q(2) accepts 5343

Contradiction!

Proof cont.

Proof cont.

Case 2: $q(1)$ accepts word with 1

Proof cont.

Case 2: q(1) accepts word with 1

Let q(1) accept 123431

Proof cont.

Case 2: q(1) accepts word with 1

Let q(1) accept 123431

Let 5, 6, 7, 8 be data fresh for C

Proof cont.

Case 2: q(1) accepts word with 1

Let q(1) accept 123431

Let 5, 6, 7, 8 be data fresh for C

Then q(1) accepts 156761

Proof cont.

Case 2: q(1) accepts word with 1

Let q(1) accept 123431

Let 5, 6, 7, 8 be data fresh for C

Then q(1) accepts 156761

Then q(2) accepts 256762

Proof cont.

Case 2: $q(1)$ accepts word with 1

Let $q(1)$ accept 123431

Let 5, 6, 7, 8 be data fresh for C

Then $q(1)$ accepts 156761

Then $q(2)$ accepts 256762

Word 856768 is accepted from $p(d)$

Proof cont.

Case 2: $q(1)$ accepts word with 1

Let $q(1)$ accept 123431

Let 5, 6, 7, 8 be data fresh for C

Then $q(1)$ accepts 156761

Then $q(2)$ accepts 256762

Word 856768 is accepted from $p(d)$

If $d \neq 1$ then $p(d)$ accepts 156761

Proof cont.

Case 2: $q(1)$ accepts word with 1

Let $q(1)$ accept 123431

Let 5, 6, 7, 8 be data fresh for C

Then $q(1)$ accepts 156761

Then $q(2)$ accepts 256762

Word 856768 is accepted from $p(d)$

If $d \neq 1$ then $p(d)$ accepts 156761

If $d \neq 2$ then $p(d)$ accepts 256762

Proof cont.

Case 2: $q(1)$ accepts word with 1

Let $q(1)$ accept 123431

Let 5, 6, 7, 8 be data fresh for C

Then $q(1)$ accepts 156761

Then $q(2)$ accepts 256762

Word 856768 is accepted from $p(d)$

If $d \neq 1$ then $p(d)$ accepts 156761

If $d \neq 2$ then $p(d)$ accepts 256762

Contradiction!

Results

Results

Theorem

Results

Theorem

1) The inclusion problem for URA is in ExpSpace

Results

Theorem

1) The inclusion problem for URA is in ExpSpace

2) The inclusion problem for k-URA is in PSpace

Results

Theorem

1) The inclusion problem for URA is in ExpSpace

2) The inclusion problem for k-URA is in PSpace

3) The universality problem for 1-URA($\geq$) is in PSpace

Results

Theorem

1)The inclusion problem for URA is in ExpSpace

2) The inclusion problem for k-URA is in PSpace

3) The universality problem for 1-URA($\geq$) is in PSpace

4) The inclusion problem of GRA in 1-GURA is in ExpSpace

Thank you!