Yet more atoms

Mikołaj Bojańczyk (Warsaw)

3 projects I would like to do next year – tell me if they make sense

Two atom questions:

1. model checking alternating automata
2. Mazurkiewicz traces

A tool:

3. Learning transducers

Start with a logical structure $\mathbb{A}$
which we call the atoms, e.g. $\mathbb{A} = (\mathbb{Q}, <)$

Definition. A *definable* set is a set of tuples (of finite dimension) of atoms modulo a definable partial equivalence relation:

$$\mathbb{A}^k{}_{/\sim}$$

such that ~ is defined by a first-order formula

$$\varphi(x_1, \ldots, x_k, y_1, \ldots, y_k)$$

ω-categorical / homogeneous / a Fraïssé limit

Theorem. If the atoms are oligomorphic, then definable sets = orbit-finite equivariant sets.

Example 3-tuples of atoms, modulo same order type

$$\bigwedge_{i,j \in \{1,2,3\}} x_i < x_j \Leftrightarrow y_i < y_j$$

(has thirteen elements)

Example. 2-tuples of atoms, modulo swap

$$(x_1 = y_1 \wedge x_2 = y_2) \vee (x_1 = y_2 \wedge x_2 = y_1)$$

The atom program

λx. definable/orbit-finite x

- nondeterministic automata
- Turing machines
- pushdown automata
- constraint satisfaction programs

1. Alternating automata

Labelled transition system which is orbit-finite

Example

- *states* — pairs of distinct atoms
- *labels* — atoms
- *transitions* — $(a, b) \xrightarrow{c} (b, c) \quad \text{for } a, b, c \text{ distinct}$

On labelled transition system, one can run an alternating automaton.

Example “on some path, a label repeats”

States: $\{\mathrm{start}, \mathrm{end}\} \cup \mathbb{A}$

- all states owned by the existential player
- a run is accepting iff it reaches “end”

Transitions:

$$\mathrm{start} \xrightarrow{a} \mathrm{start} \qquad a \xrightarrow{a} \mathrm{end} \qquad \mathrm{end} \xrightarrow{a} \mathrm{end}$$

$$\mathrm{start} \xrightarrow{a} a \qquad a \xrightarrow{b} a$$

- decidable model checking
- equivalent to μ-calculus
- undecidable emptiness

What can you express using these automata?

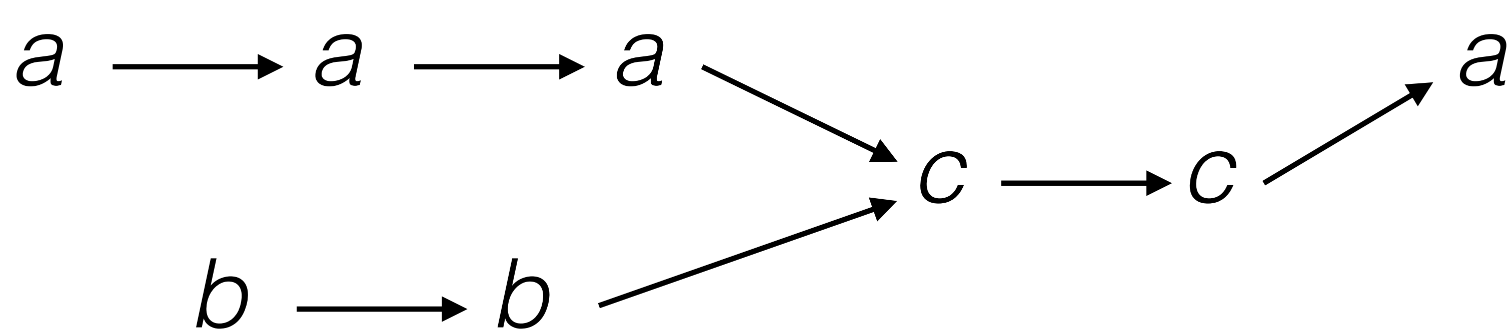
Can you express:
“on some path, infinitely many different labels”?

Project Study alternating automata on orbit-finite lts

2. Mazurkiewicz traces

Example Alphabet is $\{a,b,c\}$ with $ab=ba$

$$abaabcca = bbaaacca$$



Theorem (Zielonka)

For a language, the following conditions are equivalent:

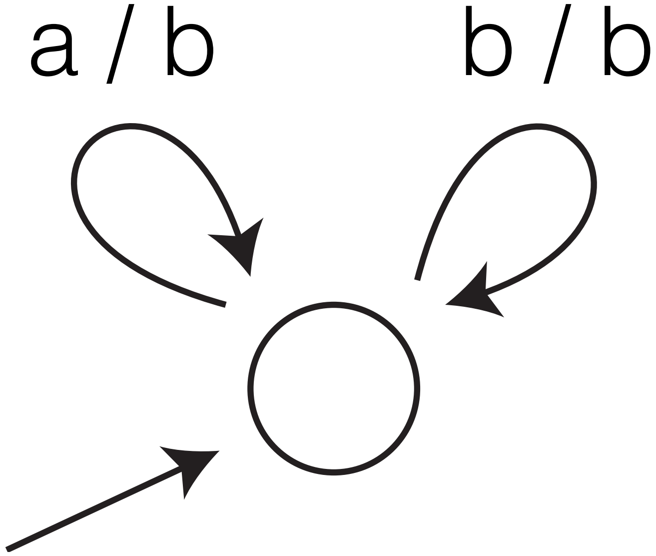
- regular and closed under equivalence
- recognized by a Zielonka automaton

Project Do this for orbit-finite alphabets

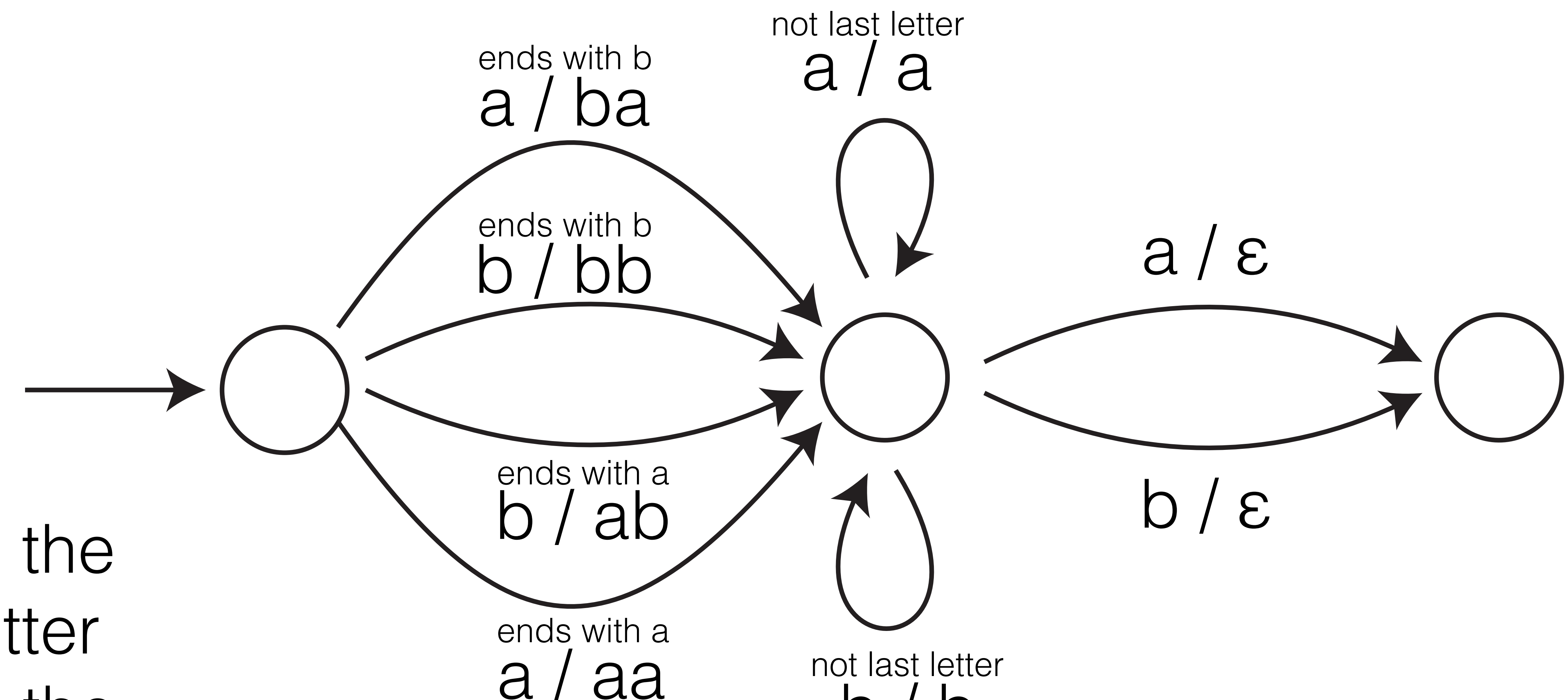
3. Learning Transducers

Gottlob+15

“replace a by b”



“reverse”



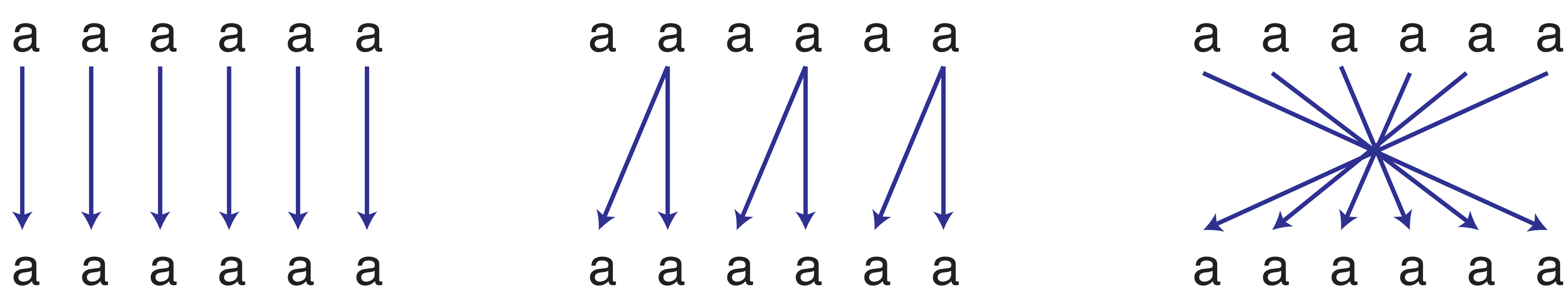
“move the last letter before the first position”

Project Learn transducers

Learning algorithms like to use *minimal/canonical devices*

For most transducer models, no such thing exists

(e.g. the identity function over a one letter alphabet)



Proposed solution origin semantics

A transducer produces:

- an output word
- for each output position, its origin in the input

Origin dogma

Origin is the specification, not the implementation

Who thinks of a text transformation as a set of word pairs?

apart from a psychoanalytic interest, this matters for implementing learning

If you are thinking “replace a by b”, do you:

- retype the text?

or

- use the cursor to replace relevant letters?

Project Build a tool that learns transducers

- use restricted models
- do usability testing
- do trees