Four small time universal Turing machines

Turlough Neary, Damien Woods

NUI MAYNOOTH
Ollscoil na hÉireann Má Nuad

University College Cork
Coláiste na hOllscoile Corcaigh

Department of Computer Science
National University of Ireland, Maynooth
Ireland

Department of Computer Science
University College Cork
Ireland

embarkinitiative
Investing in People and Ideas

Introduction

- Shannon first posed the question of finding the smallest possible universal Turing machine (UTM). (The size of a Turing Machine TM is given by the number of states and symbols).

- In 1962 Minsky created a 7-state, 4-symbol UTM that simulates TMs via 2-tag systems.

- Minsky's technique was more recently used by Rogozhin et al. to create the smallest known UTMs.

- For this talk we consider deterministic single tape TMs with a single tape head.

New universal Turing machines

- Here we present new small polynomial time universal Turing machines with state-symbol pairs of $(5, 5)$, $(6, 4)$, $(9, 3)$ and $(18, 2)$.
- These machines simulate our new variant of tag system, the bi-tag system and are the smallest known universal Turing machines with 5, 4, 3, and 2 symbols respectively.

Bi-tag systems

A bi-tag systems is defined by the tuple (A, E, e_h, P). Each production in P is of one of the following three forms:

$$P(a) = a, \quad P(e, a) = AE, \quad P(e, a) = AAE$$

where $a \in A$ and $e \in E$.

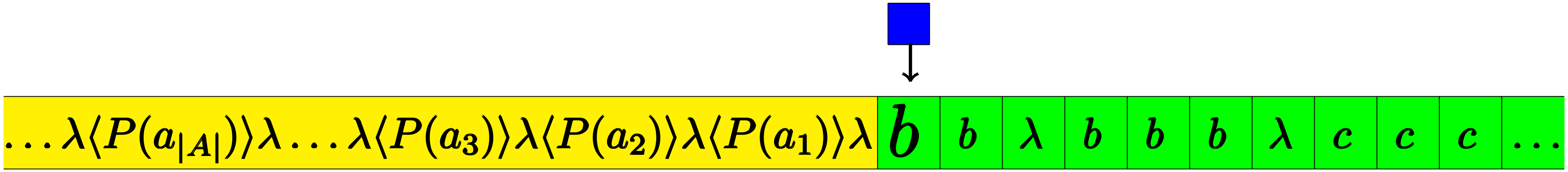
A configuration of a bi-tag system is a word of the form $s = A^*(AE \cup EA)A^*$ called the data word. If a configuration s_2 is obtained from a configuration s_1 via the application of a single production we write $s_1 \vdash s_2$.

A bi-tag systems computation step executes in one of two ways:

(i) If $s = as'$ then $as' \vdash s'P(a)$.

(ii) If $s = eas'$ then $eas' \vdash s'P(e, a)$.

Simulation of a bi-tag system computation step by UTM$(9,3)$



Encoded bi-tag system

Encoded data word

UTM tape head

$\langle X\rangle$ represents the encoding of object X

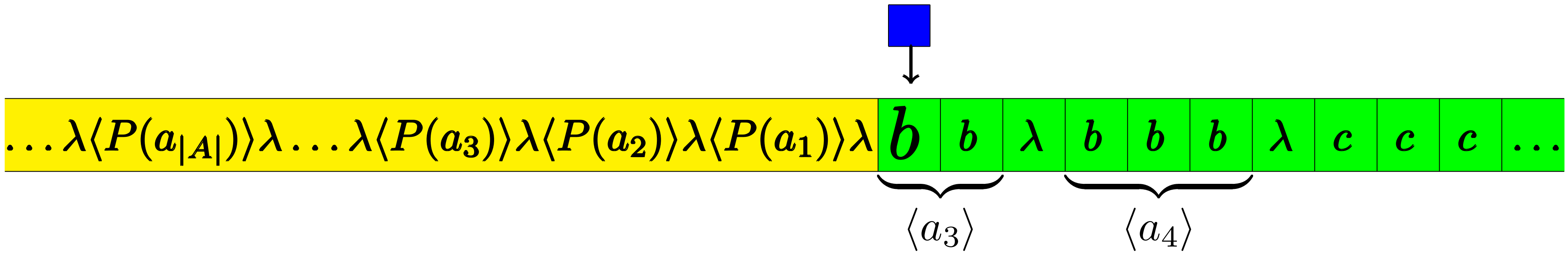
$P(a), P(e,a) \in P$, $a \in A$, $e \in E$

$\langle a\rangle, \langle e\rangle \in b^*$

Simulation of a bi-tag system computation step by UTM$(9,3)$

Simulation of $a_3a_4 \vdash a_4a_3$

Recall $P(a_3) = a_3$



Encoded bi-tag system

Encoded data word

UTM tape head

$\langle X\rangle$ represents the encoding of object X

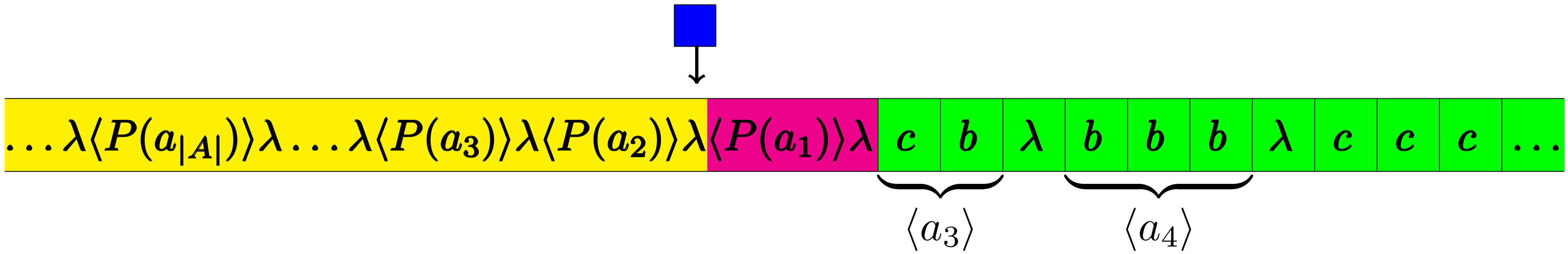
$P(a), P(e,a) \in P$, $a \in A$, $e \in E$

$\langle a\rangle, \langle e\rangle \in b^*$

Simulation of a bi-tag system computation step by UTM$(9,3)$

Simulation of $a_3 a_4 \vdash a_4 a_3$

Recall $P(a_3) = a_3$



Encoded bi-tag system

Encoded data word

UTM tape head

$\langle X\rangle$ represents the encoding of object X

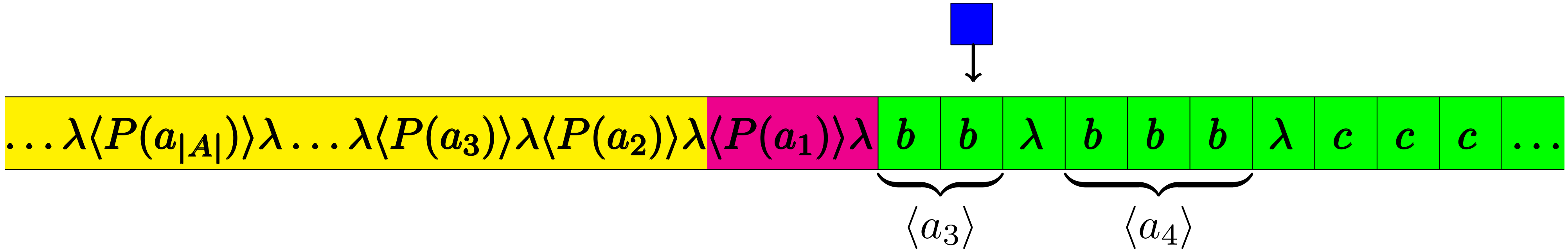
$P(a), P(e,a) \in P$, $a \in A$, $e \in E$

$\langle a\rangle, \langle e\rangle \in b^*$

Simulation of a bi-tag system computation step by UTM$(9,3)$

Simulation of $a_3 a_4 \vdash a_4 a_3$

Recall $P(a_3) = a_3$



Encoded bi-tag system

Encoded data word

UTM tape head

$\langle X\rangle$ represents the encoding of object X

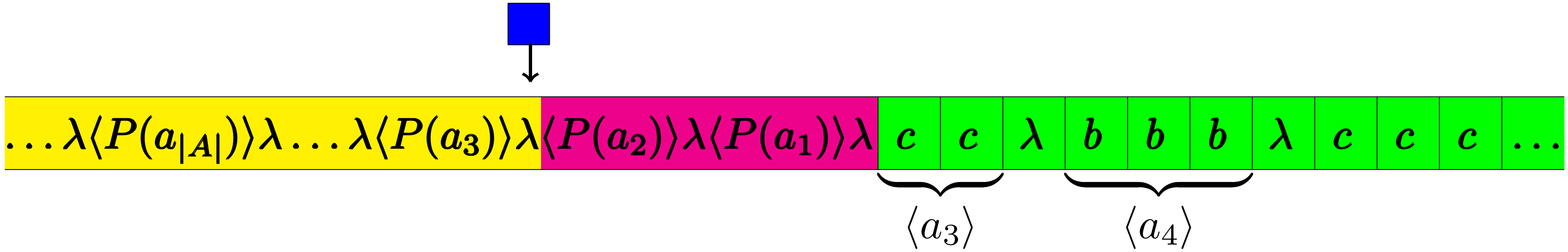
$P(a), P(e,a) \in P$, $a \in A$, $e \in E$

$\langle a\rangle, \langle e\rangle \in b^*$

Simulation of a bi-tag system computation step by UTM$(9, 3)$

Simulation of $a_3 a_4 \vdash a_4 a_3$

Recall $P(a_3) = a_3$



Encoded bi-tag system

Encoded data word

UTM tape head

$\langle X\rangle$ represents the encoding of object X

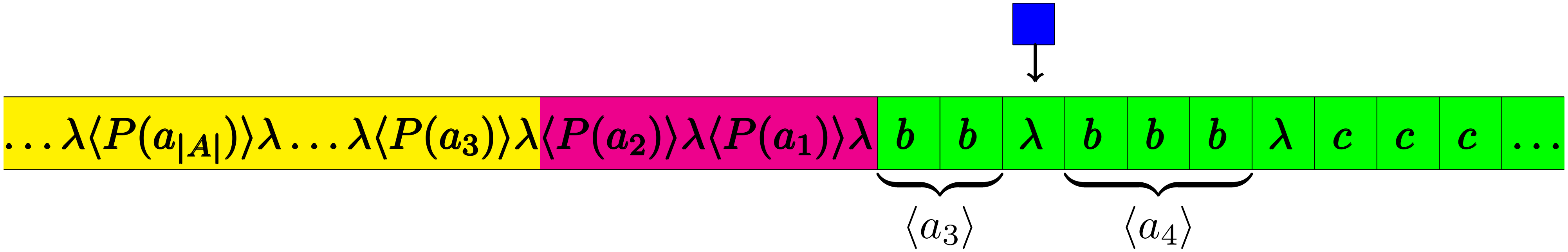
$P(a), P(e, a) \in P$, $a \in A$, $e \in E$

$\langle a\rangle, \langle e\rangle \in b^*$

Simulation of a bi-tag system computation step by UTM$(9, 3)$

Simulation of $a_3 a_4 \vdash a_4 a_3$

Recall $P(a_3) = a_3$



Encoded bi-tag system

Encoded data word

UTM tape head

$\langle X\rangle$ represents the encoding of object X

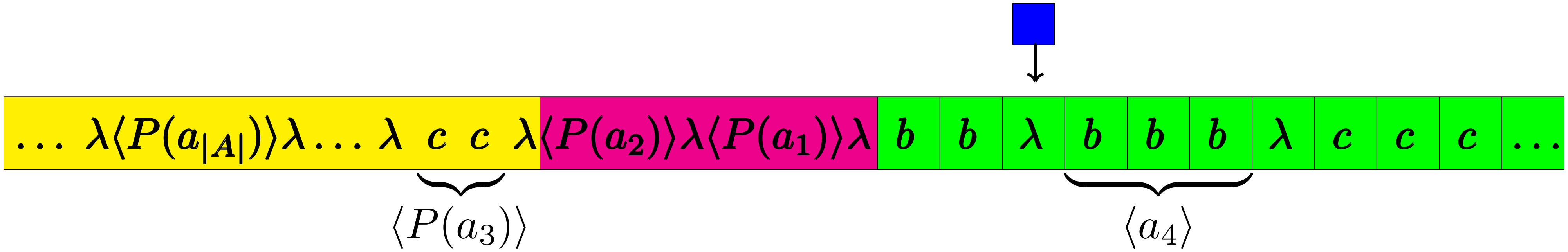
$P(a), P(e, a) \in P$, $a \in A$, $e \in E$

$\langle a\rangle, \langle e\rangle \in b^*$

Simulation of a bi-tag system computation step by UTM$(9, 3)$

Simulation of $a_3 a_4 \vdash a_4 a_3$

Recall $P(a_3) = a_3$



Encoded bi-tag system

Encoded data word

UTM tape head

$\langle X\rangle$ represents the encoding of object X

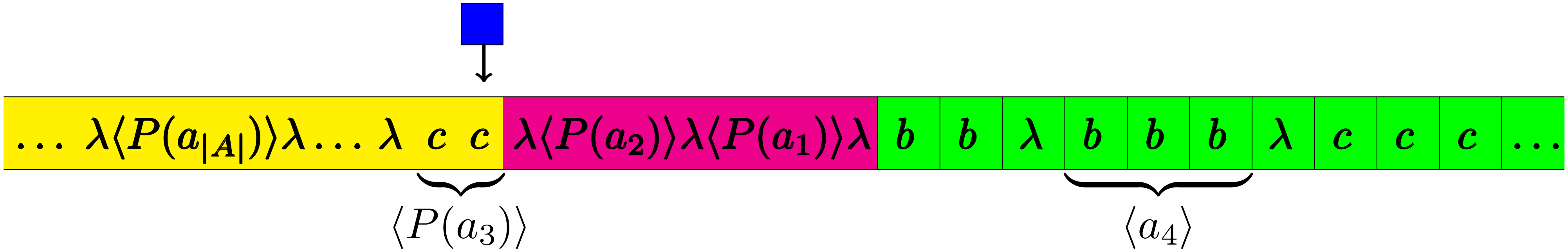
$P(a), P(e, a) \in P$, $a \in A$, $e \in E$

$\langle a\rangle, \langle e\rangle \in b^*$

Simulation of a bi-tag system computation step by UTM$(9,3)$

Simulation of $a_3a_4 \vdash a_4a_3$

Recall $P(a_3) = a_3$



Encoded bi-tag system

Encoded data word

UTM tape head

$\langle X\rangle$ represents the encoding of object X

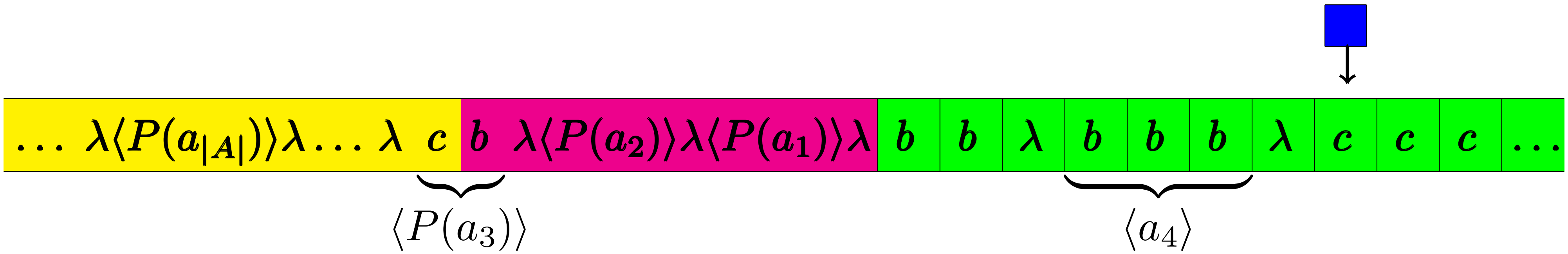
$P(a), P(e,a) \in P$, $a \in A$, $e \in E$

$\langle a\rangle, \langle e\rangle \in b^*$

Simulation of a bi-tag system computation step by UTM$(9,3)$

Simulation of $a_3a_4 \vdash a_4a_3$

Recall $P(a_3) = a_3$



Encoded bi-tag system

Encoded data word

UTM tape head

$\langle X\rangle$ represents the encoding of object X

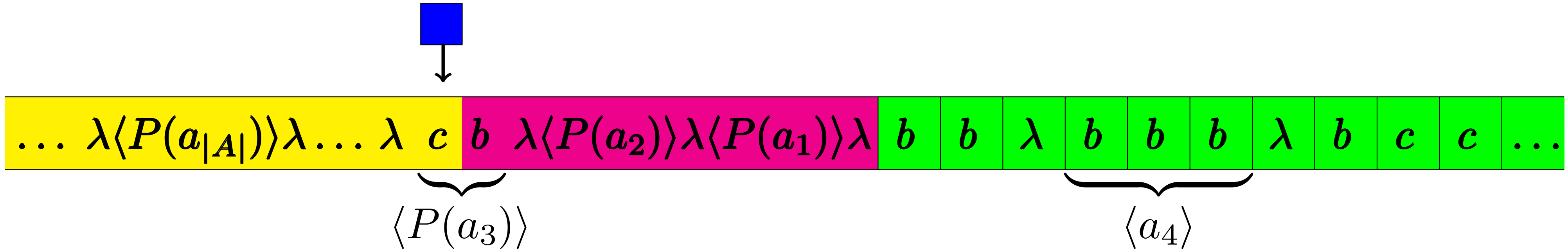
$P(a), P(e,a) \in P$, $a \in A$, $e \in E$

$\langle a\rangle, \langle e\rangle \in b^*$

Simulation of a bi-tag system computation step by UTM$(9,3)$

Simulation of $a_3 a_4 \vdash a_4 a_3$

Recall $P(a_3) = a_3$



Encoded bi-tag system

Encoded data word

UTM tape head

$\langle X\rangle$ represents the encoding of object X

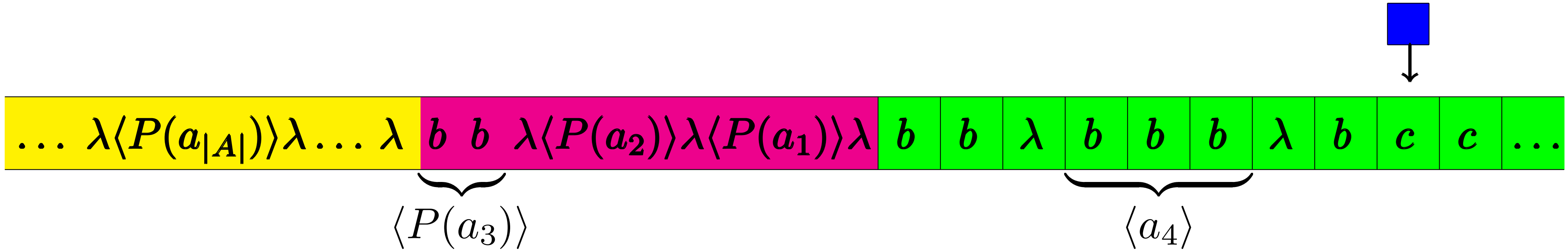
$P(a), P(e,a) \in P$, $a \in A$, $e \in E$

$\langle a\rangle, \langle e\rangle \in b^*$

Simulation of a bi-tag system computation step by UTM$(9,3)$

Simulation of $a_3 a_4 \vdash a_4 a_3$

Recall $P(a_3) = a_3$



Encoded bi-tag system

Encoded data word

UTM tape head

$\langle X\rangle$ represents the encoding of object X

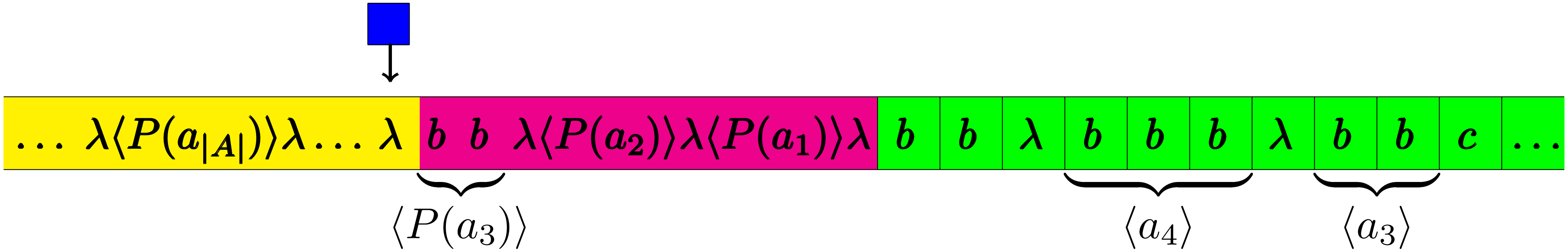
$P(a), P(e,a) \in P$, $a \in A$, $e \in E$

$\langle a\rangle, \langle e\rangle \in b^*$

Simulation of a bi-tag system computation step by UTM$(9,3)$

Simulation of $a_3 a_4 \vdash a_4 a_3$

Recall $P(a_3) = a_3$



Encoded bi-tag system

Encoded data word

UTM tape head

$\langle X\rangle$ represents the encoding of object X

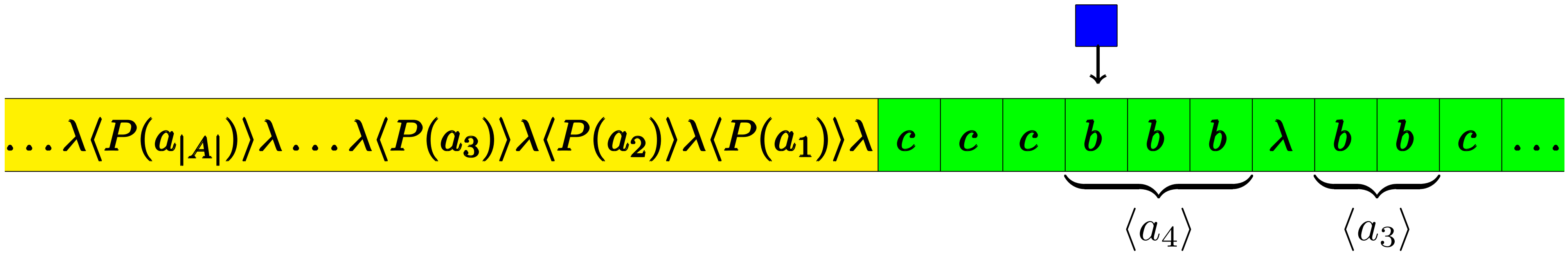
$P(a), P(e,a) \in P$, $a \in A$, $e \in E$

$\langle a\rangle, \langle e\rangle \in b^*$

Simulation of a bi-tag system computation step by UTM$(9,3)$

Simulation of $a_3 a_4 \vdash a_4 a_3$

Recall $P(a_3) = a_3$



Encoded bi-tag system

Encoded data word

UTM tape head

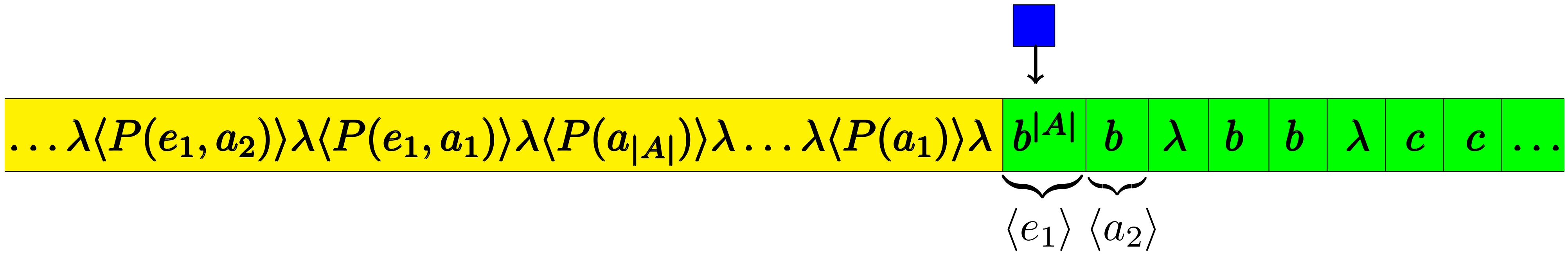
$\langle X\rangle$ represents the encoding of object X

$P(a), P(e,a) \in P$, $a \in A$, $e \in E$

$\langle a\rangle, \langle e\rangle \in b^*$

Simulation of a bi-tag system computation step by UTM$(9,3)$

Indexing a $P(e,a)$ production



Encoded bi-tag system

Encoded data word

UTM tape head

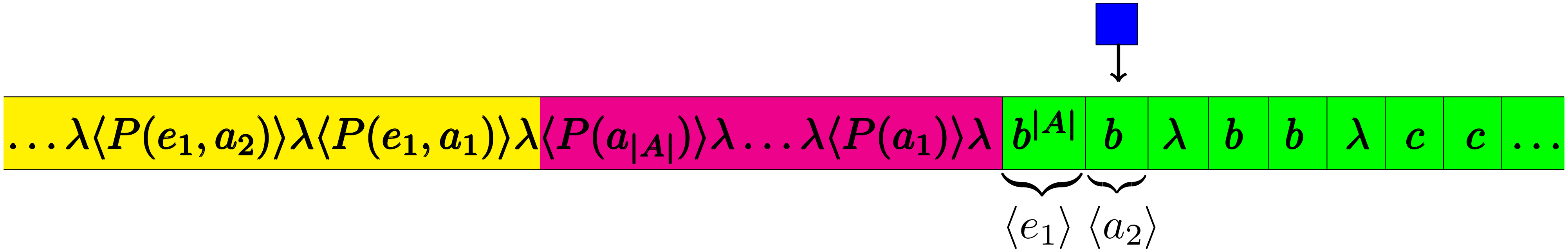
$\langle X\rangle$ represents the encoding of object X

$P(a), P(e,a) \in P$, $a \in A$, $e \in E$

$\langle a\rangle, \langle e\rangle \in b^*$

Simulation of a bi-tag system computation step by UTM$(9,3)$

Indexing a $P(e,a)$ production



Encoded bi-tag system

Encoded data word

UTM tape head

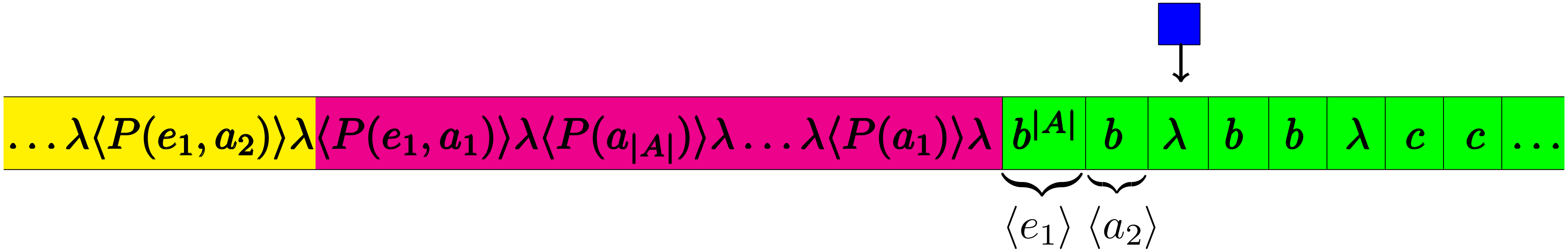
$\langle X\rangle$ represents the encoding of object X

$P(a), P(e,a) \in P$, $a \in A$, $e \in E$

$\langle a\rangle, \langle e\rangle \in b^*$

Simulation of a bi-tag system computation step by UTM$(9, 3)$

Indexing a $P(e, a)$ production



Encoded bi-tag system

Encoded data word

UTM tape head

$\langle X\rangle$ represents the encoding of object X

$P(a), P(e, a) \in P$, $a \in A$, $e \in E$

$\langle a\rangle, \langle e\rangle \in b^*$

Non-halting TMs

- Different definitions of UTM have been given by a number of different authors (Priese 1979, Davis 1956, Davis 1958).

- Non-halting UTM: Instead of the computation ending with a single halting configuration, the computation ends with a finite configuration sequence called the terminal configuration sequence.

- The output of the simulated TM is retrieved by applying the recursive decoding function to this terminal configuration sequence.

Lower bounds for non-halting UTM

- There are many ways to define a terminal configuration sequence. Some examples of possible terminal configurations sequence are:
  - The configurations containing a given sequence of states.
  - A repeated sequence configurations.
  - A single configuration containing a constant word.

- Lower bounds: Given a definition of a terminal configuration sequence we may prove that the terminal sequence problem (will a machine ever execute a terminal configuration sequence) is decidable.

- More general approach is to prove that the terminal sequence problem for all possible terminal sequences, of a machine or set of machines, is decidable.

Future work

- The universality question remains open for 42 state-symbols pair.
- The following are some approaches to find smaller UTMs.
  - Look for other universal systems that require less instructions to simulate.
  - Simplify some existing universal model in order to make it easier to simulate.
  - Find an encoding that allows many different operations to be carried out by the same group of instructions.