Two-Way Parikh Automata with a Visibly Pushdown Stack

Jean-Marc Talbot [2]
Joint Work L. Dartois [1], E. Filiot [1]

[1]Université Libre de Bruxelles

[2]Université d'Aix-Marseille

Delta - Mar 27 2018

Outline

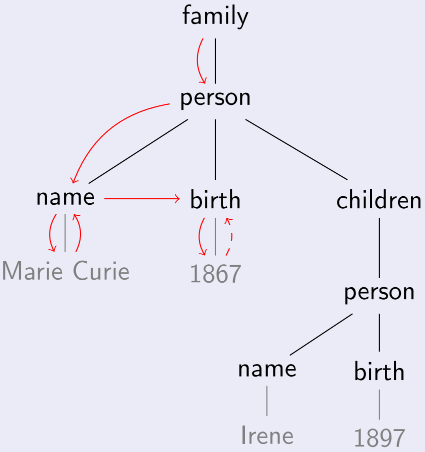
Outline

0 - Motivations

Well-nested words

Encoding

Well-nested words ≡ linearizations of (unranked) trees



⇒

< family >
< person >
< name > MarieCurie *< \name >*
< birth > 1867 *< \birth >*
< children >
< name > Irene *< \name >*
< birth > 1897 *< \birth >*
< \children >
< \person >
< \family >

Well-nested words

Definition (Structured Alphabet)

A *structured alphabet*, Σ, is a set $\Sigma = \Sigma_c \uplus \Sigma_r$ (*call* and *return* symbols respectively).

A word is *well-nested* if there is no pending call c_i nor return symbols r_i (Σ_{wn}^*)

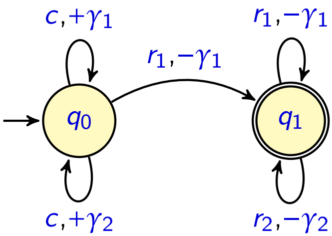
$c_1 \; c_2 \; r_2 \; r_1 c_1 r_1$ ✓ $\qquad$ $c_1 \; c_2 \; r_2 \; r_1 \underline{r_3} c_1 r_1 \underline{c_2}$ ✗

Well-nested words are generated by the grammar $u, v \in \Sigma_{\mathrm{wn}}^* \; : \; \epsilon \mid u.v \mid c u r$.

Visibly Pushdown Automata (VPAs) [Alur,Madhusudan,04]

VPAs = Pushdown Automata on *structured* alphabet $\Sigma = \Sigma_c \uplus \Sigma_r$:

- push **one** stack symbol on **call** symbols Σ_c
- pop **one** stack symbol on **return** symbols Σ_r
- accept on final state



$$L(A) = \{c^n \cdot r_1 \cdot (r_1 + r_2)^{n-1} \mid n > 0\}.$$

- The height of the stack at some position in the word does not depend on the considered computation, only on the position

Two-way VPA (2VPA) [Madhusudan,Viswanathan,09]

Main features :

- the head is placed between symbols
- directed states $Q \times \{\leftarrow, \rightarrow\}$ ($\rightarrow$: read the symbol on the right, ...)
- Behaviour :
  - Forward ($\rightarrow$): just as VPA
  - Backward ($\leftarrow$): dually, **pop** on call and **push** on return

- The height of the stack at some position in the word does not depend on the considered computation, only on the position

Two-way VPA (2VPA) [Madhusudan,Viswanathan,09]

Main features :

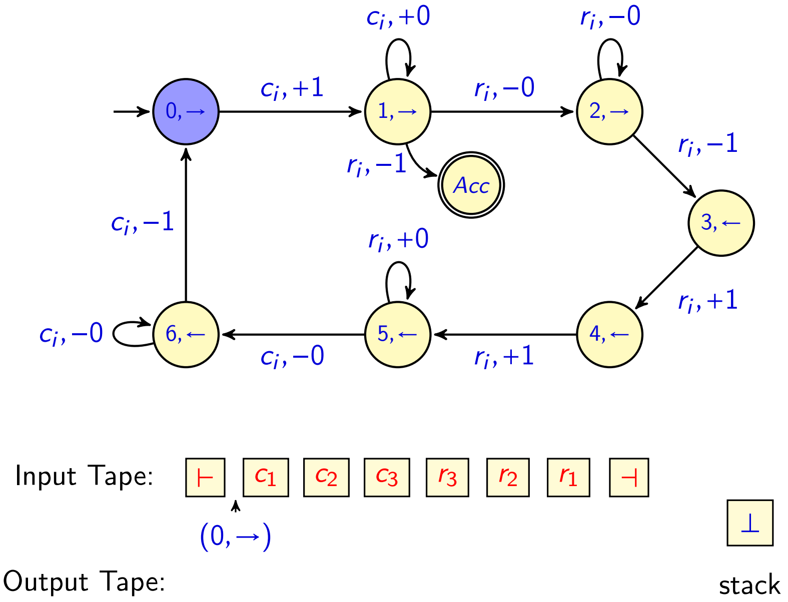
- the head is placed between symbols
- directed states $Q \times \{\leftarrow, \rightarrow\}$ ($\rightarrow$: read the symbol on the right, ...)
- Behaviour :
  - Forward ($\rightarrow$): just as VPA
  - Backward ($\leftarrow$): dually, **pop** on call and **push** on return

- The height of the stack at some position in the word does not depend on the considered computation, only on the position

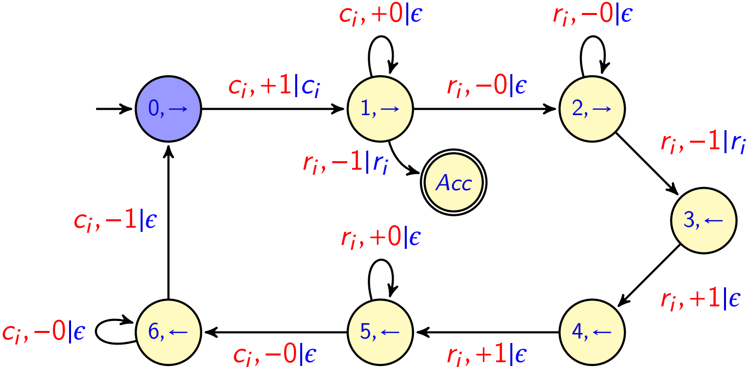
Theorem

2VPA are as expressive as VPA (reading from left to right).

2-way Visibly Pushdown Transducers (2VPT)



2-way Visibly Pushdown Transducers (2VPT)



Input Tape: | ⊢ | c_1 | c_2 | c_3 | r_3 | r_2 | r_1 | ⊣ |

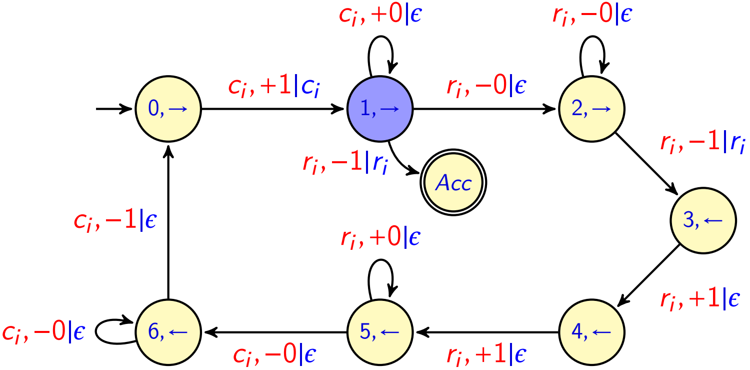
(0, →)

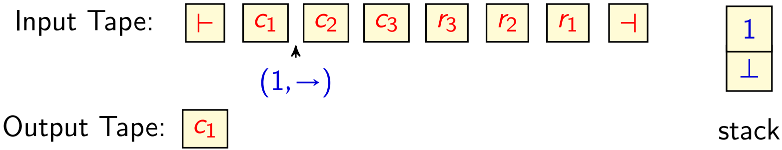
Output Tape:

⊥

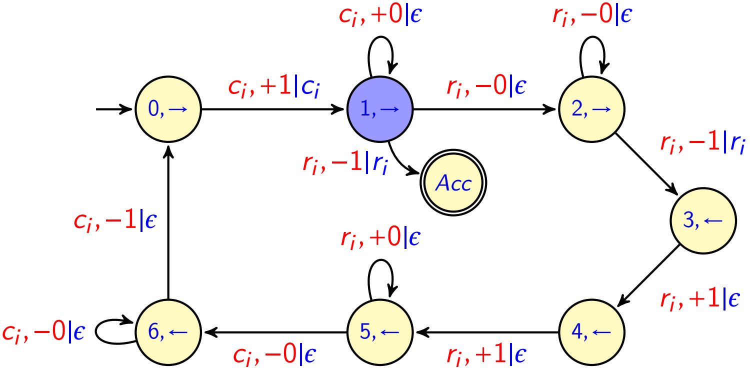
stack

2-way Visibly Pushdown Transducers (2VPT)





2-way Visibly Pushdown Transducers (2VPT)



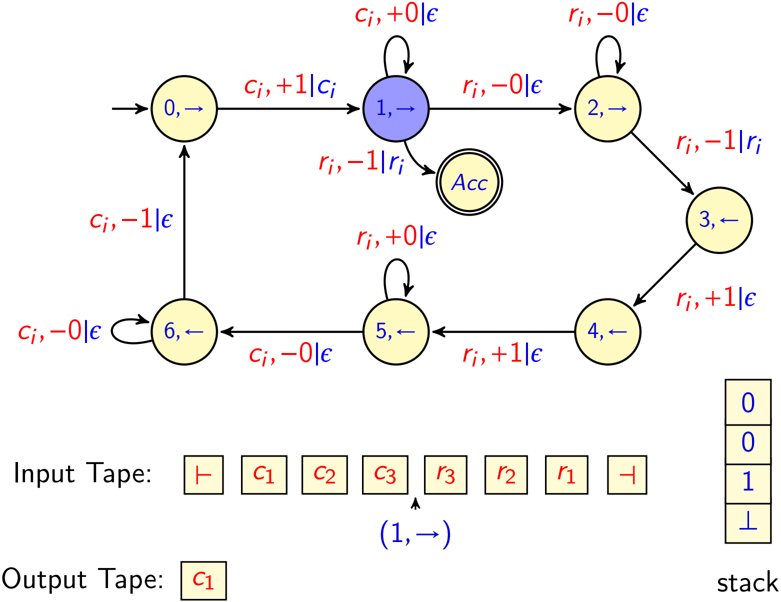
Input Tape: $\vdash$ c_1 c_2 c_3 r_3 r_2 r_1 $\dashv$

$(1, \rightarrow)$

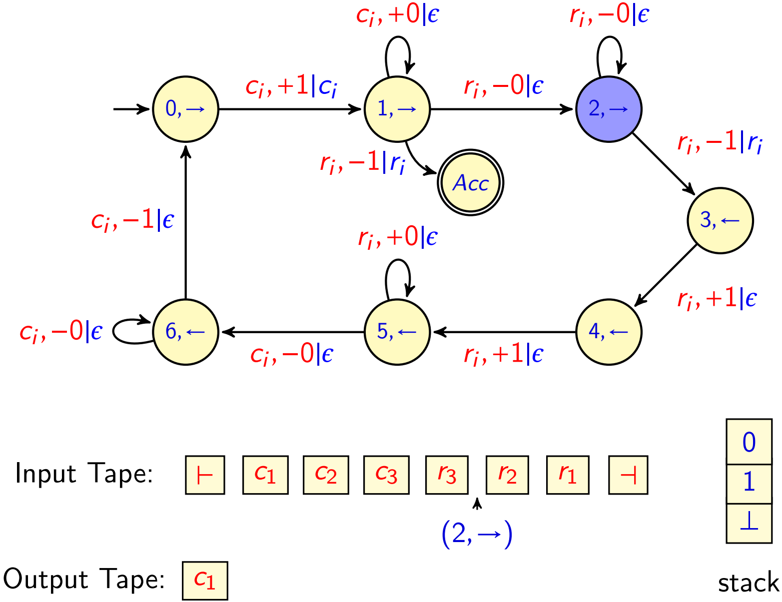
Output Tape: c_1

stack: 0, 1, $\perp$

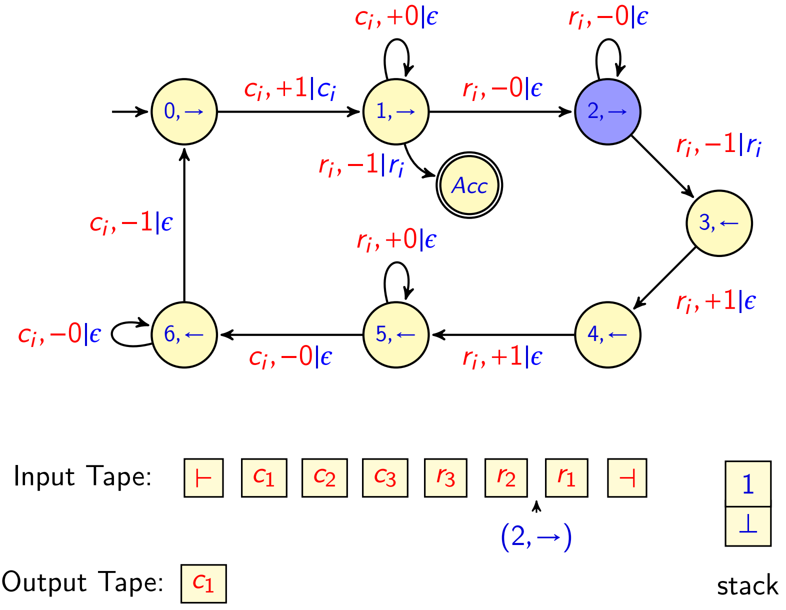
2-way Visibly Pushdown Transducers (2VPT)



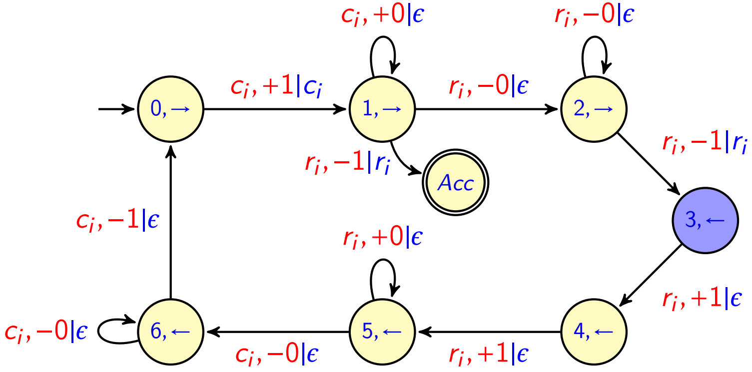
2-way Visibly Pushdown Transducers (2VPT)



2-way Visibly Pushdown Transducers (2VPT)



2-way Visibly Pushdown Transducers (2VPT)



Input Tape: $\vdash$ c_1 c_2 c_3 r_3 r_2 r_1 $\dashv$

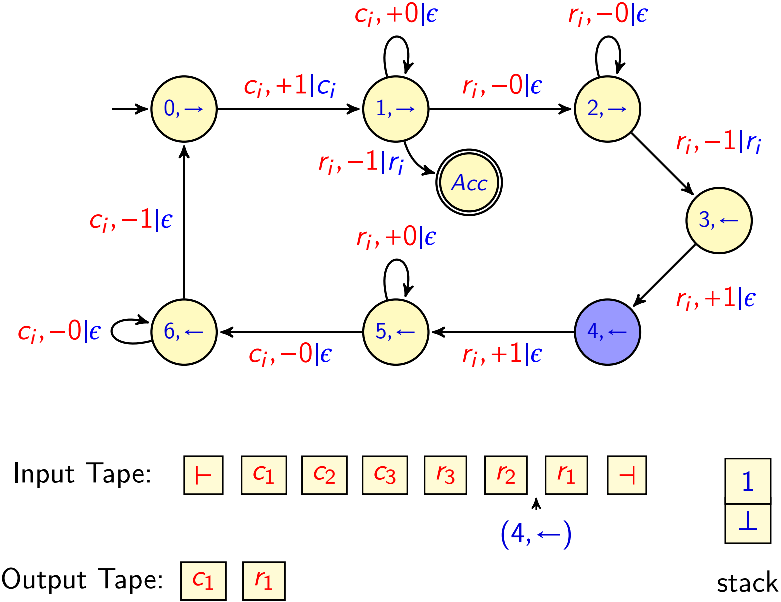
$(3, \leftarrow)$

$\perp$

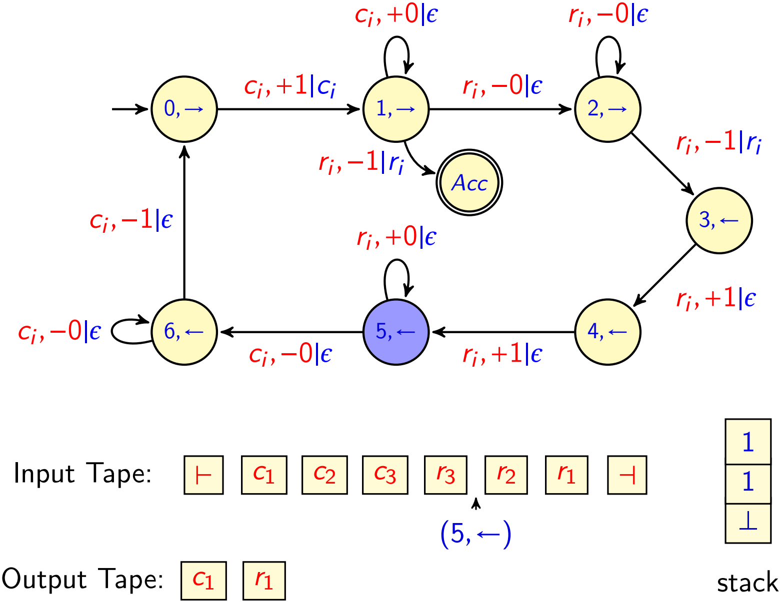
Output Tape: c_1 r_1

stack

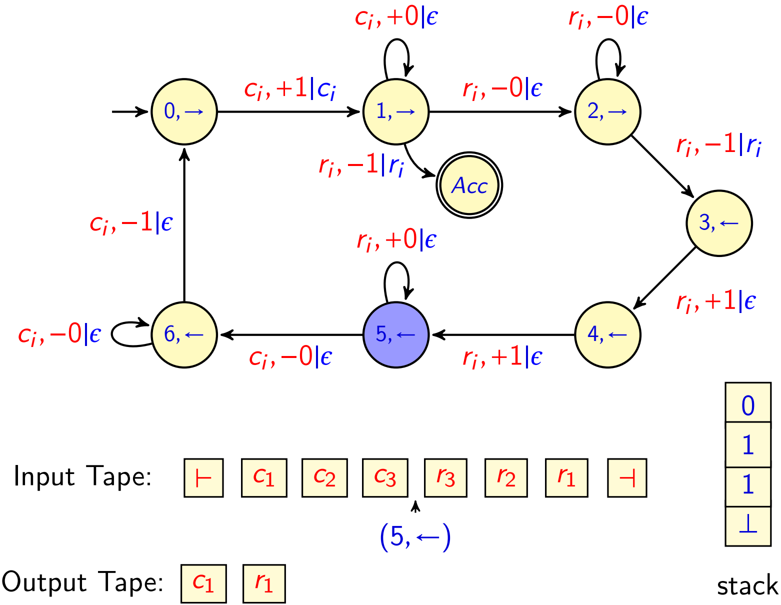
2-way Visibly Pushdown Transducers (2VPT)



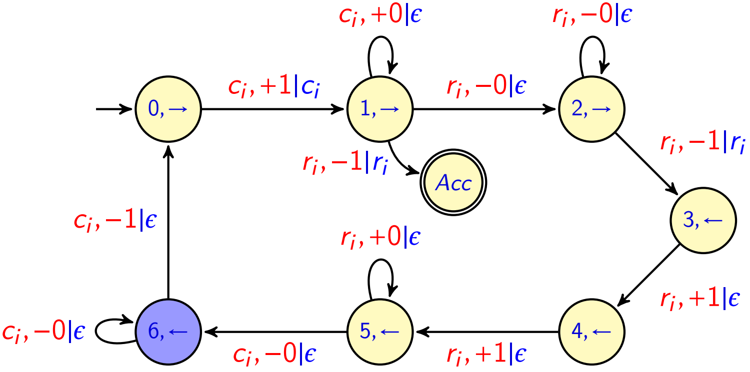
2-way Visibly Pushdown Transducers (2VPT)

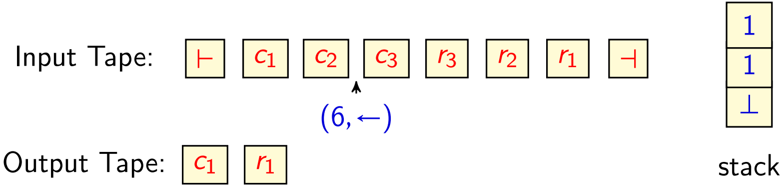


2-way Visibly Pushdown Transducers (2VPT)

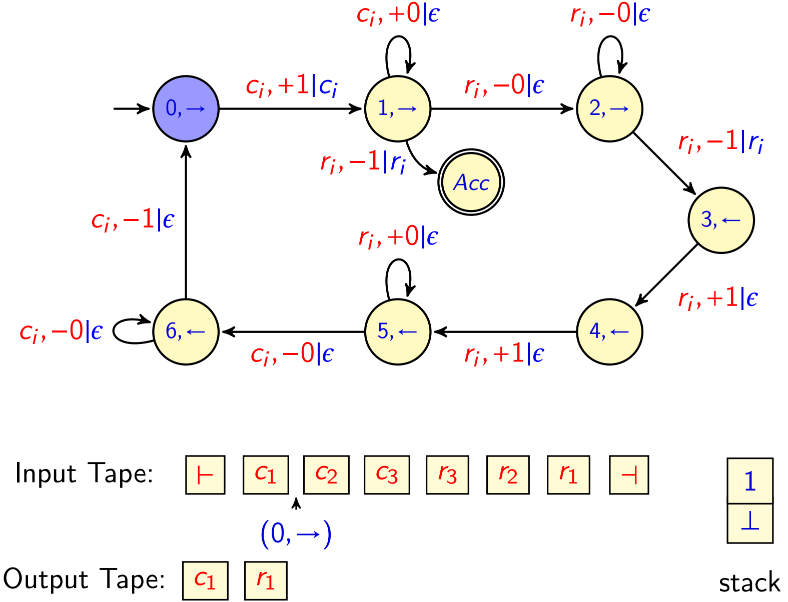


2-way Visibly Pushdown Transducers (2VPT)

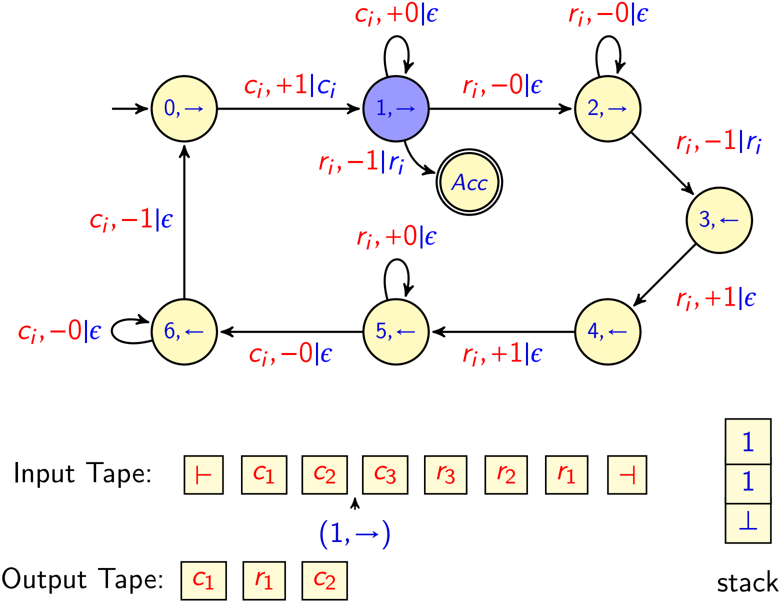




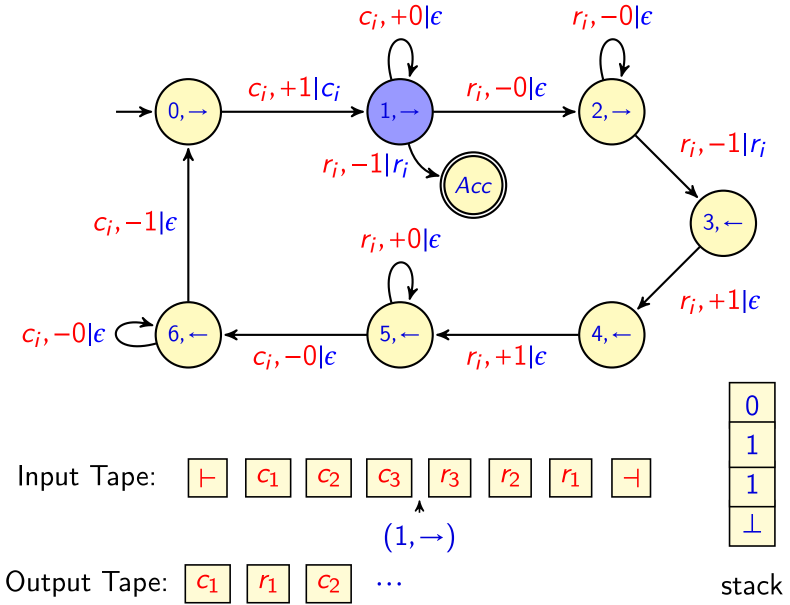
2-way Visibly Pushdown Transducers (2VPT)



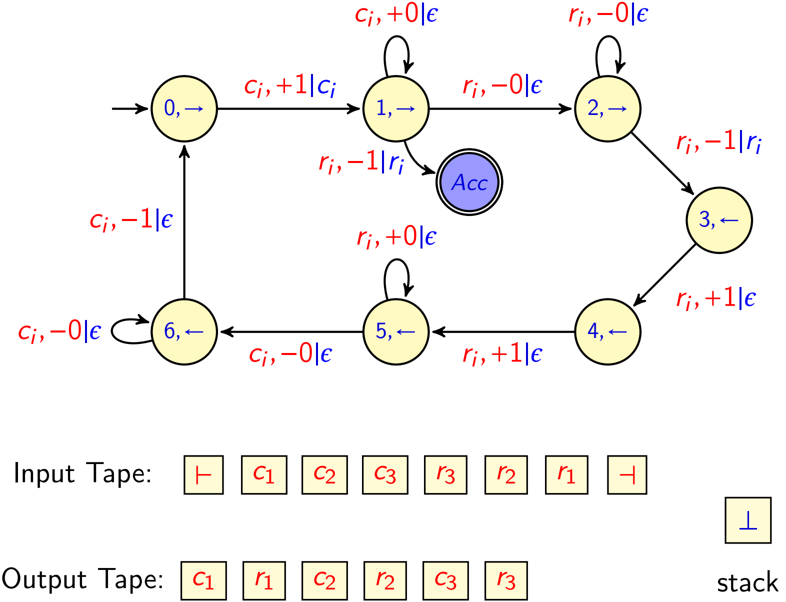
2-way Visibly Pushdown Transducers (2VPT)



2-way Visibly Pushdown Transducers (2VPT)

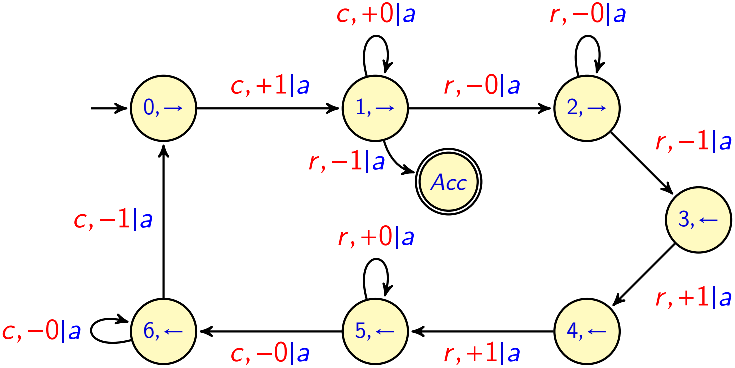


2-way Visibly Pushdown Transducers (2VPT)



Output Growth

D2VPT are not linear size increase ($\nexists k \in \mathbb{N}, \forall w \in \Sigma_{wn}^*, |[\![A]\!](w)| \le k * |w|$).



$$|T(c^n r^n)| = O(n^2)$$

Single-use Property

Single-use restriction

A D2VPT is *single-use* (D2VPT$_{su}$) if in any accepting run, any producing transition occur at most once at a given position.

Decision

Given a D2VPT, deciding if it satisfies the single-use property is Exptime-complete.

Implication

D2VPT$_{su}$ are linear size increase.

Some Results [Dartois, Filiot, Reynier, T. 16]

Expressiveness

- D2VPT_{su} are as expressive as MSO[nw2w].
- Order-preserving MSO[nw2w] is equivalent to fVPT.

Decision results

- Equivalence problem for D2VPT is decidable (reduction to DTOP2S^{LA} equiv),
- Type-checking problem against a regular language is Exptime-complete.

Some Questions for $\mathrm{D2VPT_{su}}$

- functionality/k-valuedness problem
- well-nested output (when structured output alphabet) $[\![A]\!] \subseteq \Sigma_{wn}^*$

Some Questions for $\mathsf{D2VPT_{su}}$

- functionality/k-valuedness problem
- well-nested output (when structured output alphabet) $[\![A]\!] \subseteq \Sigma^*_{wn}$

Several questions but a single tool :

Two-way Visibly Pushdown Parikh Automata (2VPPA)

Outline

2 - (Two-way) Visibly Parikh Automata (2VPPA)

Parikh Automata

Definition

A Parikh automaton $P = (A, dim, \lambda, S)$ where $A = (\Sigma, Q, I, F, \Delta)$ is an NFA, dim is a natural number, $\lambda : \Delta \mapsto \mathbb{N}^{dim}$ and S is a semi-linear subset of $\mathbb{N}^{dim}$.

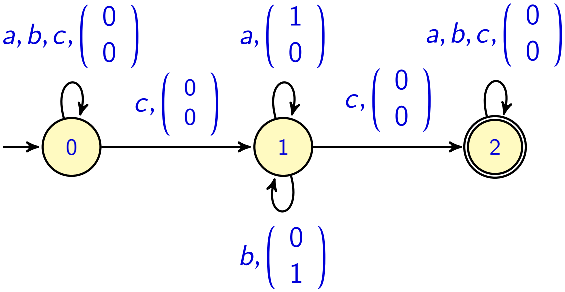
Here, S as a Presburger formula with dim free variables $x_1, \ldots, x_{dim}$

Parikh Automata

Definition

A Parikh automaton $P = (A, dim, \lambda, S)$ where $A = (\Sigma, Q, I, F, \Delta)$ is an NFA, dim is a natural number, $\lambda : \Delta \mapsto \mathbb{N}^{dim}$ and S is a semi-linear subset of $\mathbb{N}^{dim}$.

Here, S as a Presburger formula with dim free variables $x_1, \ldots, x_{dim}$



$$\varphi_S(x_1, x_2) = \ (x_1 = 2 * x_2 + 1)$$

Words containing a factor cwc with $w \in (a+b)^*$ and $|w|_a = 2|w|_b + 1$

Parikh Automata

Definition

A Parikh automaton $P = (A, dim, \lambda, S)$ where $A = (\Sigma, Q, I, F, \Delta)$ is an NFA, dim is a natural number, $\lambda : \Delta \mapsto \mathbb{N}^{dim}$ and S is a semi-linear subset of $\mathbb{N}^{dim}$.

Here, S as a Presburger formula with dim free variables $x_1, \ldots, x_{dim}$

Equi-expressive to (non-deterministic) reversal-bounded counter machines [Ibarra] (weaker in the deterministic case).

In Parikh automata, counter values (and thus, updates) do not influence the control state evolution.

Two-way Visibly Pushdown ParikhParikh Automata

Definition

A two-way Visibly Pushdown Parikh automaton $P = (A, dim, \lambda, S)$ where $A = (\Sigma, Q, I, F, \Gamma, \Delta)$ is an 2VPA, dim is a natural number, $\lambda : \Delta \mapsto \mathbb{N}^{dim}$ and S is a semi-linear subset of $\mathbb{N}^{dim}$.

Two-way Visibly Pushdown ParikhParikh Automata

Definition

A two-way Visibly Pushdown Parikh automaton $P = (A, dim, \lambda, S)$ where $A = (\Sigma, Q, I, F, \Gamma, \Delta)$ is an 2VPA, dim is a natural number, $\lambda : \Delta \mapsto \mathbb{N}^{dim}$ and S is a semi-linear subset of $\mathbb{N}^{dim}$.

Reducing (D)2VPT problems to emptiness of 2VPPA

Well-nestedness of (D)2VPT

Globally as many calls as returns. Locally always more calls then returns.

$dim = 4$

$$q_1 \xrightarrow{\alpha,\pm\gamma|w} q_2 \ : \quad \begin{cases} q_1 \xrightarrow{\alpha,\pm\gamma,(|w|_{\Sigma_c},|w|_{\Sigma_r},|w|_{\Sigma_c},|w|_{\Sigma_r})} q_2 \\ q_1 \xrightarrow{\alpha,\pm\gamma,(|w|_{\Sigma_c},|w|_{\Sigma_r},|w|_{\Sigma_c},|w|_{\Sigma_r})} q_2' \\ q_1' \xrightarrow{\alpha,\pm\gamma,(0,0,|w|_{\Sigma_c},|w|_{\Sigma_r})} q_2' \end{cases}$$

$$\varphi = \ x_3 \neq x_4 \lor x_1 < x_2$$

Not well-nested $\Leftrightarrow$ Not empty

Functionality of 2VPT

On the same input word, there exists an output position where two different letters are output in two different computations (essentially).

$dim = 2$

- Compute a position in the output

$$q_1 \xrightarrow{\alpha,\pm\gamma|w} q_2 \quad \Rightarrow q_1 \xrightarrow{\alpha,\pm\gamma,(|w|,0)} q_2$$

- Guess the (first half of the) mismatched

$$q_1 \xrightarrow{\alpha,\pm\gamma|w} q_2 \quad \Rightarrow q_1 \xrightarrow{\alpha,\pm\gamma,(|w_1|,0)} q_2^a \qquad w = w_1 a w_2$$

- Rewind the input word and start reading again

$$q_1 \xrightarrow{\alpha,\pm\gamma|w} q_2 \quad \Rightarrow q_1^a \xrightarrow{\alpha,\pm\gamma,(0,|w|)} q_2^a$$

- Guess the (second half of the) mismatched

$$q_1 \xrightarrow{\alpha,\pm\gamma|w} q_2 \quad \Rightarrow q_1^a \xrightarrow{\alpha,\pm\gamma,(0,|w_1|)} q_F \qquad w = w_1 b w_2 \ , \ a \neq b$$

$$\varphi = \ x_1 = x_2$$

Not functional ⇔ Not empty

Outline

3 - Emptiness for 2VPPA

Some Existing Results

Emptiness for :

- deterministic two-way counter machines with 2 counters and k (fixed) counter-reversals is undecidable

Some Existing Results

Emptiness for :

- deterministic two-way counter machines with 2 counters and k (fixed) counter-reversals is undecidable
- (non-deterministic) two-way reversal-bounded counter machines with finite crossing is decidable

Some Existing Results

Emptiness for :

- deterministic two-way counter machines with 2 counters and k (fixed) counter-reversals is undecidable
- (non-deterministic) two-way reversal-bounded counter machines with finite crossing is decidable
- (non-deterministic) two-way pushdown automata with finite crossing is undecidable

Some Bad News

Theorem

Emptiness for deterministic two-way visibly pushdown Parikh automata is undecidable.

Proof Idea :

Reduction of solvability of Diophantine Eq. : $P = Q$ with $P, Q \in \mathbb{N}[\mathscr{X}]$, eg $2xy + zz = 4x + 2xz + 6$

The automaton part encodes only of monomials; sums and equality test encoded in Presburger accepting formula, a dimension for each monomial : $2x_1 + x_2 = 4x_3 + 2x_4 + 6$.

Input nested words represent valuations of variables

$$\begin{array}{ll} \text{Encode}(x * y * 1, [x \mapsto 3, y \mapsto 2]) = & x_c c^3 \, \text{Encode}(y * 1, [x \mapsto 3, y \mapsto 2]) \, r^3 x_r \\ \text{Encode}(1, [x \mapsto 3, y \mapsto 2]) = & \mathbb{1} \end{array}$$

The automaton checks well-formedness

- visibly pushdown "regular" sequence of monomials
- one dimension for each variable occurence in P and Q : test via the Presburger accepting formula that there are valuated the same way.

and evaluates monomials via counters :

The dimension updated as

$$[\![\text{Encode}(x_c c^3 \, \text{Encode}(y * 1, [x \mapsto 3, y \mapsto 2]) \, r^3 x_r)]\!] = 3 * [\![\text{Encode}(y * 1, [x \mapsto 3, y \mapsto 2])]\!]$$

Some (Modest) Good News

Theorem

The non-emptiness problem for (one-way) VPPA *and* PPA *is* NP*-complete.*

Some (Modest) Good News

Theorem

The non-emptiness problem for (one-way) VPPA and PPA is NP-complete.

Upperbound : let V be the values used for update in the PPA A

New alphabet $\Sigma' = \Sigma \cup (u_v^i)$ (update counter i by adding value v ($v \in V$)).

Some (Modest) Good News

Theorem

The non-emptiness problem for (one-way) VPPA and PPA is NP-complete.

Upperbound : let V be the values used for update in the PPA A

New alphabet $\Sigma' = \Sigma \cup (u^i_v)$ (update counter i by adding value v ($v \in V$)).

- Step 1 : Define a NPA B accepting words of the form $\alpha_1 u^1_0 u^2_1 \alpha_2 u^1_3 u^2_2 \alpha_3 u^1_0 u^2_0 \alpha_4 u^1_0 u^2_1 \alpha_5 \ldots$ such that
  - $\alpha_1 \alpha_2 \alpha_3 \alpha_4 \alpha_5 \ldots$ has a run in A from an initial to a final configuration
  - each $u^1_{v_1} u^2_{v_2}$ corresponds to the update performed by the run at that position : $(v_1, v_2) \Rightarrow u^1_{v_1} u^2_{v_2}$
  - B is of polynomial size in A

Some (Modest) Good News

Theorem

The non-emptiness problem for (one-way) VPPA and PPA is NP-complete.

Upperbound : let V be the values used for update in the PPA A

New alphabet $\Sigma' = \Sigma \cup (u_v^i)$ (update counter i by adding value v ($v \in V$)).

- Step 1 : Define a NPA B accepting words of the form $\alpha_1 u_0^1 u_1^2 \alpha_2 u_3^1 u_2^2 \alpha_3 u_0^1 u_0^2 \alpha_4 u_0^1 u_1^2 \alpha_5 \ldots$ such that
  - $\alpha_1 \alpha_2 \alpha_3 \alpha_4 \alpha_5 \ldots$ has a run in A from an initial to a final configuration
  - each $u_{v_1}^1 u_{v_2}^2$ corresponds to the update performed by the run at that position : $(v_1, v_2) \Rightarrow u_{v_1}^1 u_{v_2}^2$
  - B is of polynomial size in A
- Step 2 : There is an existential Presburger formula $\psi((y_\sigma)_{\sigma \in \Sigma'})$ of size polynomial in B defining the Parikh image of $L(B)$. [Verma, et al. 05]

Some (Modest) Good News

Theorem

The non-emptiness problem for (one-way) VPPA and PPA is NP-complete.

Upperbound : let V be the values used for update in the PPA A

New alphabet $\Sigma' = \Sigma \cup (u_v^i)$ (update counter i by adding value v ($v \in V$)).

- Step 1 : Define a NPA B accepting words of the form $\alpha_1 u_0^1 u_1^2 \alpha_2 u_3^1 u_2^2 \alpha_3 u_0^1 u_0^2 \alpha_4 u_0^1 u_1^2 \alpha_5 \ldots$ such that
  - $\alpha_1 \alpha_2 \alpha_3 \alpha_4 \alpha_5 \ldots$ has a run in A from an initial to a final configuration
  - each $u_{v_1}^1 u_{v_2}^2$ corresponds to the update performed by the run at that position : $(v_1, v_2) \Rightarrow u_{v_1}^1 u_{v_2}^2$
  - B is of polynomial size in A
- Step 2 : There is an existential Presburger formula $\psi((y_\sigma)_{\sigma \in \Sigma'})$ of size polynomial in B defining the Parikh image of $L(B)$. [Verma, et al. 05]
- Step 3 : In NP [Scarpellini84], satisfiability test of

$$\psi((y_\sigma)_{\sigma \in \Sigma'}) \wedge \phi(x_1, \ldots, x_n) \wedge \bigwedge_{1 \leq i \leq dim} x_i = \sum_{v \in V} v.y_{u_v^i}$$

Some (Modest) Good News

Lower bound : (hardness does not rely on the acceptance set, eg Presburger satisfiability).

Some (Modest) Good News

Lower bound : (hardness does not rely on the acceptance set, eg Presburger satisfiability).

Reduction of the 2-partition problem :

- I a finite set of natural numbers represented in binary.
- Does there exists $J \subseteq I$ such that

$$\sum_{e \in J} e = \sum_{e \notin J} e$$

Some (Modest) Good News

Lower bound : (hardness does not rely on the acceptance set, eg Presburger satisfiability).

Reduction of the 2-partition problem :

- I a finite set of natural numbers represented in binary.
- Does there exists $J \subseteq I$ such that
$$\sum_{e \in J} e = \sum_{e \notin J} e$$

One dimension for J and one for $I \setminus J$.

Input words : $T_1 v_1 L_2 v_2 T_3 v_3 T_4 v_4 L_5 v_5 \ldots$

- v_k : representation of the unary encoding of the kth natural.
- T_i : take the ith natural (in J) - L_j : leave the jth natural

v_k is encoded as $u_n u_{n-1} \ldots u_1 u_0$ where

- $u_j = \epsilon$ if the jth bit of v_k is 0 and w_j otherwise
- recursively, $w_0 = 1$ and $w_l = c_l w_{l-1} r_l c_l w_{l-1} r_l$

and can be recognized by a small VPA.

Some (Modest) Good News

Lower bound : (hardness does not rely on the acceptance set, eg Presburger satisfiability).

Reduction of the 2-partition problem :

- I a finite set of natural numbers represented in binary.
- Does there exists $J \subseteq I$ such that

$$\sum_{e \in J} e = \sum_{e \notin J} e$$

One dimension for J and one for $I \setminus J$.

Input words : $T_1 v_1 L_2 v_2 T_3 v_3 T_4 v_4 L_5 v_5 \dots$

- v_k : representation of the unary encoding of the kth natural.
- T_i : take the ith natural (in J) - L_j : leave the jth natural

v_k is encoded as $u_n u_{n-1} \dots u_1 u_0$ where

- $u_j = \epsilon$ if the jth bit of v_k is 0 and w_j otherwise
- recursively, $w_0 = 1$ and $w_l = c_l w_{l-1} r_l c_l w_{l-1} r_l$

and can be recognized by a small VPA.

Thus, holds even if the automata are deterministic, with a fixed dimension 2, tuples of values in $\{0,1\}^2$ and with a fixed Presburger formula $(x_1, x_2) = \ x_1 = x_2$

Emptiness for single-use 2VPPA

Single-use 2VPPA : a position is visited only once per counter modifying transition.

Emptiness for single-use 2VPPA

Single-use 2VPPA : a position is visited only once per counter modifying transition.

Proposition

For any single-use 2VPPA, there exists an equivalent VPPA of at most exponential size.

Emptiness for single-use 2VPPA

Single-use 2VPPA : a position is visited only once per counter modifying transition.

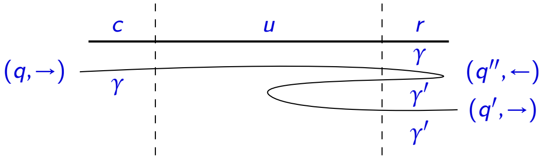
Proposition

For any single-use 2VPPA, there exists an equivalent VPPA of at most exponential size.

Proof Idea

- Extends [Dartois, Filiot, Reynier, T. 16] following Sherpherson's ideas (for FSA) on traversals T.

$((q,d)(q',d')) \in T(u)$ iff some run enters u by (q,d) and leaves it by (q',d')



Emptiness for single-use 2VPPA

Single-use 2VPPA : a position is visited only once per counter modifying transition.

Proposition

For any single-use 2VPPA, there exists an equivalent VPPA of at most exponential size.

Proof Idea

- Extends [Dartois, Filiot, Reynier, T. 16] following Sherpherson's ideas (for FSA) on traversals T.

  $((q,d)(q',d')) \in T(u)$ iff some run enters u by (q,d) and leaves it by (q',d')

- Productive and Non-productive traversals form a finite algebra.
- Consider possible decompositions of traversals $((q,\rightarrow)(q',\leftarrow))$ on $c_1 r w_2$ into productive traversals (finitely many as single-use) reachable from each other by non-productive traversals.
- Using commutativity of addition over $\mathbb{N}^{dim}$, extract a VPA and a mapping λ

Emptiness for single-use 2VPPA

Single-use 2VPPA : a position is visited only once per counter modifying transition.

Proposition

For any single-use 2VPPA, there exists an equivalent VPPA of at most exponential size.

Corollary

Emptiness for single-use 2VPPA is in NEXPtime.

Proposition

Emptiness for single-use 2VPPA is in NEXPtime-hard.

Implied Results

For well-nestedness, functionality and k-valuedness, single-use 2VPT yields single-use 2VPPA

Proposition

For single-use 2VPT, well-nestedness, functionality and k-valuedness are in NEXPtime.

Both known as EXPtime-hard.

Thank You !