Equivalence of Deterministic Nested-Word to Word Transducers

Sławek Staworko
(joint work with Gregoire Laurence, Aurelien Lemay, and Joachim Niehren)

MOSTRARE

INRIA Lille – Nord Europe

University of Lille

FCT'09
September 4, 2009

Motivation: machine learning for XML transformations

Data on the Web

- Host of standards for data presentation, distribution, exchange
- XML is the *de facto* format (XHTML, RSS, SOAP, etc.)
- XML transformation languages (XSLT, XQuery, etc.) too complicated for *dummies*

MOSTRARE

Combine tree automata techniques with machine learning:

- Tree automata/transducers – good tools to model/reason about XML related tasks
- Machine learning – show on examples what you want the computer to do

Motivation, example

```
<book>
  <person>
    <info>
      <name>Slawek</name>
      <pos>Post-doc</pos>
    </info>
  <contact>
    <email>
      staworko@gmail.com
    </email>
    <addr>
      20 Pl. Louise de ...
    </addr>
  </contact>
  ...
<book>
```

```
<HTML><BODY><TABLE>
   <TR><TH>Name</TH></TR>
       <TH>Email</TH></TR>
   <TR><TD>Slawek</TD>
       <TD>staworko@gmail.com</TD></TR>
   ...
</TABLE></BODY></HTML>
```

Motivation

Towards machine learning of XML transformations

1. Choice of XML transformation model (tree transducer)
   - not too general – learning quickly becomes intractable
   - not too simple – practical transformations still learnable
2. Ultimate goal: grammatical inference methods
3. Grammatical inference $\sim$ Myhill-Nerode theorem
4. Myhill-Nerode theorem $\sim$ Equivalence of transducers
5. Equivalence is a fundamental property

Overview

1. Nested-Word Transducer
   - Nested-Word Automata, Trees, and Nested-Words
   - Nested-Word Transducer
   - Equivalence Problem
2. Morphisms
3. Polynomial reduction of equivalence problems
   - From morphisms to $\mathrm{d}\textsc{n2w}^{\downarrow}$
   - From $\mathrm{d}\textsc{n2w}$ to Morphisms
   - Other Models
4. Conclusions and future work

Outline

Nested-Word Automata

NA Definition

$T = (\Sigma, \text{states}, \text{stack}, \text{rules}, \text{initial}, \text{final})$

Two types of rules ($q, q' \in$ states, $a \in \Sigma$, $\gamma \in$ stack):

- opening transitions $q \xrightarrow{\mathtt{op}\ a:\gamma} q'$
- closing transitions $q \xrightarrow{\mathtt{cl}\ a:\gamma} q'$.

Nested-Word Automata

NA Definition

$T = (\Sigma, \text{states}, \text{stack}, \text{rules}, \text{initial}, \text{final})$

Two types of rules ($q, q' \in$ states, $a \in \Sigma$, $\gamma \in$ stack):

- opening transitions $q \xrightarrow{\texttt{op}\ a:\gamma} q'$
- closing transitions $q \xrightarrow{\texttt{cl}\ a:\gamma} q'$.

Determinism (dNA)

- In a opening transition $q \xrightarrow{\texttt{op}\ a:\gamma} q'$, q and a determines γ and q'.
- In a closing transition $q \xrightarrow{\texttt{cl}\ a:\gamma} q'$, q, a and γ determines q'.

Nested-Word Automata

NA Definition

$T = (\Sigma, \text{states}, \text{stack}, \text{rules}, \text{initial}, \text{final})$

Two types of rules ($q, q' \in$ states, $a \in \Sigma$, $\gamma \in$ stack):

- opening transitions $q \xrightarrow{\text{op}\ a:\gamma} q'$
- closing transitions $q \xrightarrow{\text{cl}\ a:\gamma} q'$.

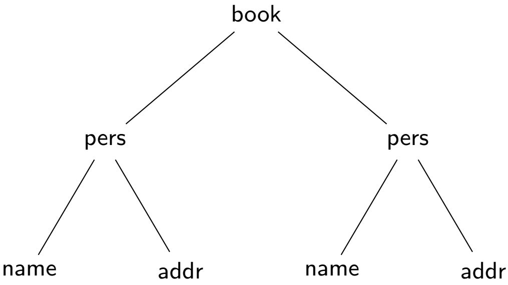
Determinism (dNA)

- In a opening transition $q \xrightarrow{\text{op}\ a:\gamma} q'$, q and a determines γ and q'.
- In a closing transition $q \xrightarrow{\text{cl}\ a:\gamma} q'$, q, a and γ determines q'.

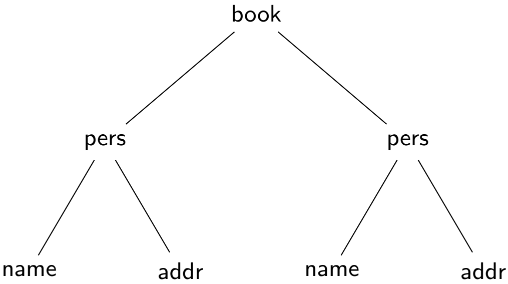
Top-Down ($\text{NA}^{\downarrow}$)

- stack symbols = states
- all closing rules have the form $q \xrightarrow{\text{cl}\ a:q'} q'$

Nested-Word Automata

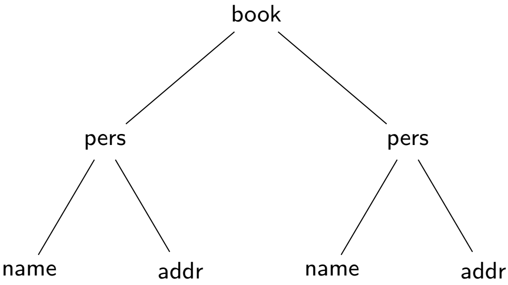


Nested-Word Automata



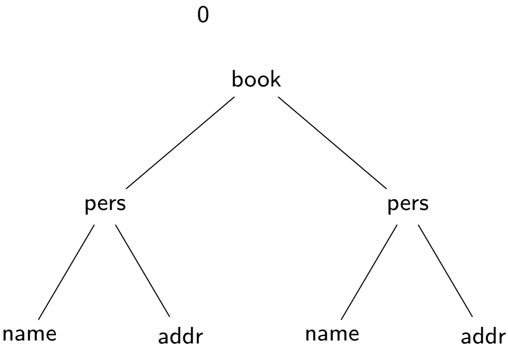

```
<book>
 <pers> <name></name> <addr></addr> </pers>
 <pers> <name></name> <addr></addr> </pers>
</book>
```

Nested-Word Automata



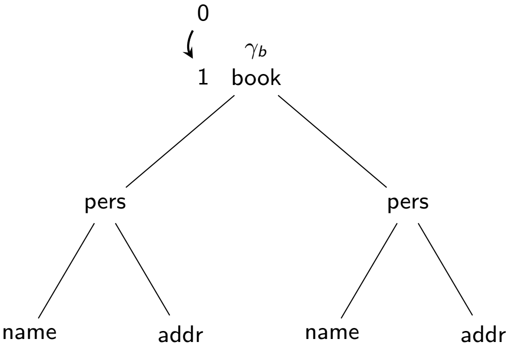
$$0 \xrightarrow{\mathtt{op}\ book:\gamma_b} 1$$
$$1 \xrightarrow{\mathtt{op}\ pers:\gamma_p} 2$$
$$2 \xrightarrow{\mathtt{op}\ name:\gamma_n} 3$$
$$3 \xrightarrow{\mathtt{cl}\ name:\gamma_n} 4$$
$$4 \xrightarrow{\mathtt{op}\ addr:\gamma_a} 3$$
$$3 \xrightarrow{\mathtt{cl}\ addr:\gamma_a} 5$$
$$5 \xrightarrow{\mathtt{cl}\ pers:\gamma_p} 1$$
$$1 \xrightarrow{\mathtt{cl}\ book:\gamma_b} 6$$

Nested-Word Automata



$$0 \xrightarrow{\text{op } book:\gamma_b} 1$$
$$1 \xrightarrow{\text{op } pers:\gamma_p} 2$$
$$2 \xrightarrow{\text{op } name:\gamma_n} 3$$
$$3 \xrightarrow{\text{cl } name:\gamma_n} 4$$
$$4 \xrightarrow{\text{op } addr:\gamma_a} 3$$
$$3 \xrightarrow{\text{cl } addr:\gamma_a} 5$$
$$5 \xrightarrow{\text{cl } pers:\gamma_p} 1$$
$$1 \xrightarrow{\text{cl } book:\gamma_b} 6$$

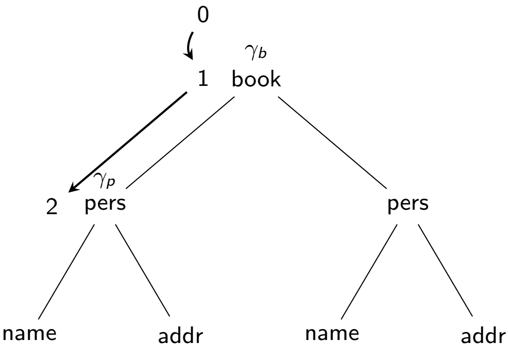
Nested-Word Automata



<book>

$$0 \xrightarrow{\texttt{op}\ book:\gamma_b} 1$$
$$1 \xrightarrow{\texttt{op}\ pers:\gamma_p} 2$$
$$2 \xrightarrow{\texttt{op}\ name:\gamma_n} 3$$
$$3 \xrightarrow{\texttt{cl}\ name:\gamma_n} 4$$
$$4 \xrightarrow{\texttt{op}\ addr:\gamma_a} 3$$
$$3 \xrightarrow{\texttt{cl}\ addr:\gamma_a} 5$$
$$5 \xrightarrow{\texttt{cl}\ pers:\gamma_p} 1$$
$$1 \xrightarrow{\texttt{cl}\ book:\gamma_b} 6$$

Nested-Word Automata



$0 \xrightarrow{\texttt{op}\ book:\gamma_b} 1$

$1 \xrightarrow{\texttt{op}\ pers:\gamma_p} 2$

$2 \xrightarrow{\texttt{op}\ name:\gamma_n} 3$

$3 \xrightarrow{\texttt{cl}\ name:\gamma_n} 4$

$4 \xrightarrow{\texttt{op}\ addr:\gamma_a} 3$

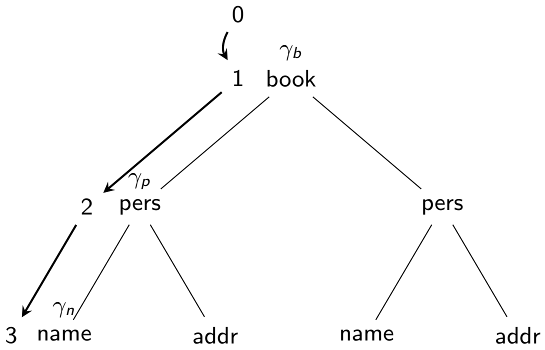
$3 \xrightarrow{\texttt{cl}\ addr:\gamma_a} 5$

$5 \xrightarrow{\texttt{cl}\ pers:\gamma_p} 1$

$1 \xrightarrow{\texttt{cl}\ book:\gamma_b} 6$

```
<book>
 <pers>
```

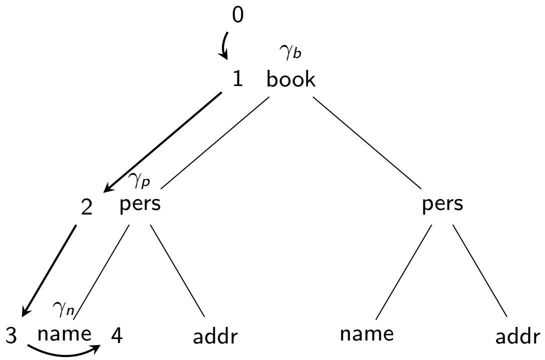
Nested-Word Automata



<book>
<pers> <name>

$$0 \xrightarrow{\texttt{op}\ book:\gamma_b} 1$$
$$1 \xrightarrow{\texttt{op}\ pers:\gamma_p} 2$$
$$2 \xrightarrow{\texttt{op}\ name:\gamma_n} 3$$
$$3 \xrightarrow{\texttt{cl}\ name:\gamma_n} 4$$
$$4 \xrightarrow{\texttt{op}\ addr:\gamma_a} 3$$
$$3 \xrightarrow{\texttt{cl}\ addr:\gamma_a} 5$$
$$5 \xrightarrow{\texttt{cl}\ pers:\gamma_p} 1$$
$$1 \xrightarrow{\texttt{cl}\ book:\gamma_b} 6$$

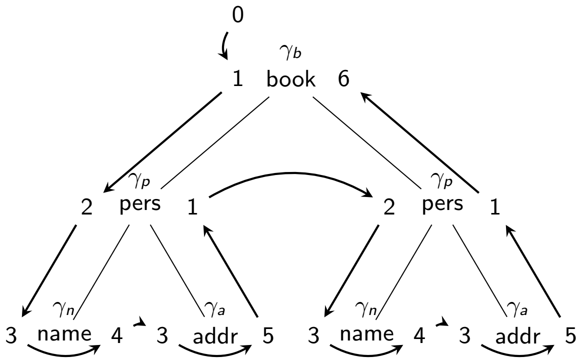
Nested-Word Automata



<book>
<pers> <name></name>

$$0 \xrightarrow{\texttt{op}\ book:\gamma_b} 1$$
$$1 \xrightarrow{\texttt{op}\ pers:\gamma_p} 2$$
$$2 \xrightarrow{\texttt{op}\ name:\gamma_n} 3$$
$$3 \xrightarrow{\texttt{cl}\ name:\gamma_n} 4$$
$$4 \xrightarrow{\texttt{op}\ addr:\gamma_a} 3$$
$$3 \xrightarrow{\texttt{cl}\ addr:\gamma_a} 5$$
$$5 \xrightarrow{\texttt{cl}\ pers:\gamma_p} 1$$
$$1 \xrightarrow{\texttt{cl}\ book:\gamma_b} 6$$

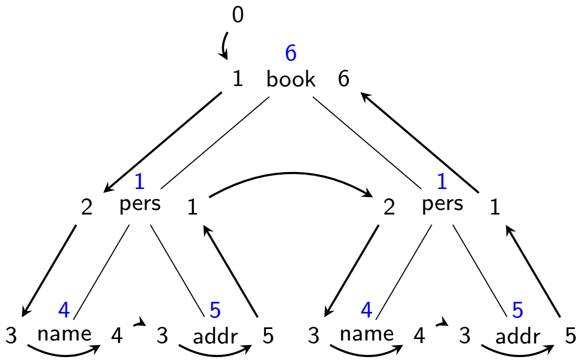
Nested-Word Automata




```
<book>
 <pers> <name></name> <addr></addr> </pers>
 <pers> <name></name> <addr></addr> </pers>
</book>
```

$$0 \xrightarrow{\texttt{op}\ book:\gamma_b} 1$$
$$1 \xrightarrow{\texttt{op}\ pers:\gamma_p} 2$$
$$2 \xrightarrow{\texttt{op}\ name:\gamma_n} 3$$
$$3 \xrightarrow{\texttt{cl}\ name:\gamma_n} 4$$
$$4 \xrightarrow{\texttt{op}\ addr:\gamma_a} 3$$
$$3 \xrightarrow{\texttt{cl}\ addr:\gamma_a} 5$$
$$5 \xrightarrow{\texttt{cl}\ pers:\gamma_p} 1$$
$$1 \xrightarrow{\texttt{cl}\ book:\gamma_b} 6$$

Nested-Word Automata ↓




```
<book>
 <pers> <name></name> <addr></addr> </pers>
 <pers> <name></name> <addr></addr> </pers>
</book>
```

$0 \xrightarrow{\texttt{op}\ \mathit{book}:6} 1$

$1 \xrightarrow{\texttt{op}\ \mathit{pers}:1} 2$

$2 \xrightarrow{\texttt{op}\ \mathit{name}:4} 3$

$3 \xrightarrow{\texttt{cl}\ \mathit{name}:4} 4$

$4 \xrightarrow{\texttt{op}\ \mathit{addr}:5} 3$

$3 \xrightarrow{\texttt{cl}\ \mathit{addr}:5} 5$

$5 \xrightarrow{\texttt{cl}\ \mathit{pers}:1} 1$

$1 \xrightarrow{\texttt{cl}\ \mathit{book}:6} 6$

Nested-Word Transducers

Nested-Word to Word (N2W) Definition

$T = (\Sigma, \Delta, \text{states}, \text{stack}, \text{rules}, \text{initial}, \text{final})$

Rules allow producing output $u \in \Delta^*$ $\quad q \xrightarrow{\texttt{op}\ a/u:\gamma} q' \quad q \xrightarrow{\texttt{cl}\ a/u:\gamma} q'$

$[\![T]\!] \subseteq \mathcal{T}_\Sigma \times \Delta^*$

Visibly Pushdown Transducers [Raskin and Servais, ICALP'08]

- Verification tool (no determinism)
- Input/output synchronization: output always well-nested (testable for N2W in PTIME [Tozawa and Minamide, FOSSACS'07])

Nested-Word Transducers

Nested-Word to Word (N2W) Definition

$T = (\Sigma, \Delta, \text{states}, \text{stack}, \text{rules}, \text{initial}, \text{final})$

Rules allow producing output $u \in \Delta^*$ $\quad q \xrightarrow{\texttt{op}\ a/u:\gamma} q' \quad q \xrightarrow{\texttt{cl}\ a/u:\gamma} q'$

$[\![T]\!] \subseteq \mathcal{T}_\Sigma \times \Delta^*$

Visibly Pushdown Transducers [Raskin and Servais, ICALP'08]

- Verification tool (no determinism)
- Input/output synchronization: output always well-nested (testable for N2W in PTIME [Tozawa and Minamide, FOSSACS'07])

Determinism and Top-Down

Similarly to NA.

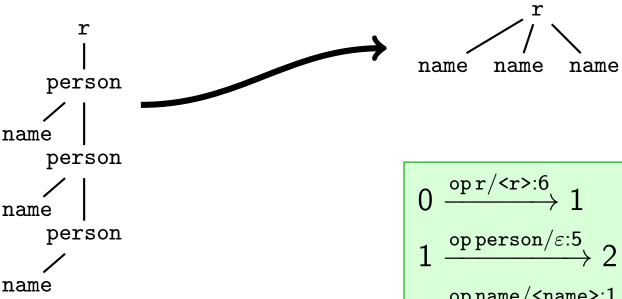
If T is deterministic, then $[\![T]\!]$ is a (possibly partial) function.

Example (dN2WS$^{\downarrow}$ transduction)

```
<book>
  <person>
    <info>
      <name>Slawek</name>
      <pos>Post-doc</pos>
    </info>
  <contact>
    <email>
      staworko@gmail.com
    </email>
    <addr>
      20 Pl. Louise de ...
    </addr>
  </contact>
  ...
<book>
```

```
<HTML><BODY><TABLE>
   <TR><TH>Name</TH></TR>
       <TH>Email</TH></TR>
   <TR><TD>Slawek</TD>
       <TD>staworko@gmail.com</TD></TR>
   ...
</TABLE></BODY></HTML>
```

Another example



$$0 \xrightarrow{\texttt{op}\ \texttt{r}/\texttt{<r>}:6} 1$$

$$1 \xrightarrow{\texttt{op}\ \texttt{person}/\varepsilon:5} 2$$

$$2 \xrightarrow{\texttt{op}\ \texttt{name}/\texttt{<name>}:1} 3$$

$$3 \xrightarrow{\texttt{cl}\ \texttt{name}/\texttt{</name>}:4} 1$$

$$5 \xrightarrow{\texttt{cl}\ \texttt{person}/\varepsilon:5} 5$$

$$5 \xrightarrow{\texttt{cl}\ \texttt{r}/\texttt{</r>}:6} 6$$

Equivalence Problem

Equivalence of N2Ws

Two N2W*s* T_1, T_2 are equivalent iff $[\![T_1]\!] = [\![T_2]\!]$.

Theorem [Griffiths'68]

Testing equivalence of two nondeterministic (word) transducers is undecidable.

Equivalence of dN2Ws

Two *d*N2W*s* T_1, T_2 are equivalent iff $[\![T_1]\!]$ and $[\![T_2]\!]$ have the same domain and $[\![T_1]\!](t) = [\![T_2]\!](t)$ for every t in the domain.

Outline

1. Nested-Word Transducer
   - Nested-Word Automata, Trees, and Nested-Words
   - Nested-Word Transducer
   - Equivalence Problem
2. Morphisms
3. Polynomial reduction of equivalence problems
   - From morphisms to dN2W$^{\downarrow}$
   - From dN2W to Morphisms
   - Other Models
4. Conclusions and future work

Morphism Equivalence on CFG

A (word) *morphism*

- $M : \Sigma \to \Delta^*$
- $M(v_1 \cdot v_2 \cdots v_n) = M(v_1) \cdot M(v_2) \cdots M(v_n)$

Morphism Equivalence on CFG

A (word) *morphism*

- $M : \Sigma \to \Delta^*$
- $M(v_1 \cdot v_2 \cdots v_n) = M(v_1) \cdot M(v_2) \cdots M(v_n)$

Equivalence on CFG

Two morphisms M_1, M_2 are equivalent on a CFG G iff $M_1(w) = M_2(w)$ for all $w \in L(G)$.

Morphism Equivalence on CFG

A (word) *morphism*

- $M : \Sigma \to \Delta^*$
- $M(v_1 \cdot v_2 \cdots v_n) = M(v_1) \cdot M(v_2) \cdots M(v_n)$

Equivalence on CFG

Two morphisms M_1, M_2 are equivalent on a CFG G iff $M_1(w) = M_2(w)$ for all $w \in L(G)$.

Theorem [Plandowski'94]

Testing the equivalence of two morphisms on CFG is in PTIME.

Outline

From Morphisms to dN2W$^\downarrow$

Proposition

Morphism equivalence on CFGs can be reduced in quadratic time to dN2W$^\downarrow$-equivalence.

From Morphisms to dN2W$^{\downarrow}$

Proposition

Morphism equivalence on CFGs can be reduced in quadratic time to dN2W$^{\downarrow}$-equivalence.

Main idea

Given a CFG G and a morphism M, we construct a dN2W$^{\downarrow}$ T:

- input : (extended) parse tree t of $w \in L(G)$
- output : $[\![T]\!](t) = M(w)$

From Morphisms to dN2W$^{\downarrow}$

CFG G

r_1: $S \to RS$
r_2: $S \to R$
r_3: $R \to AB$
r_4: $A \to a$
r_5: $B \to b$

Morphism M

$M(a) = ab$
$M(b) = b$

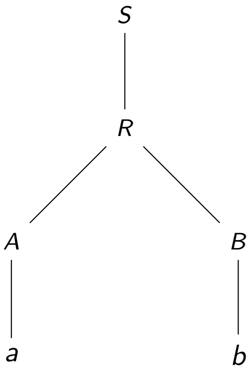
From Morphisms to dN2W$^{\downarrow}$

CFG G

r_1: $S \to RS$
r_2: $S \to R$
r_3: $R \to AB$
r_4: $A \to a$
r_5: $B \to b$

Morphism M

$M(a) = ab$
$M(b) = b$



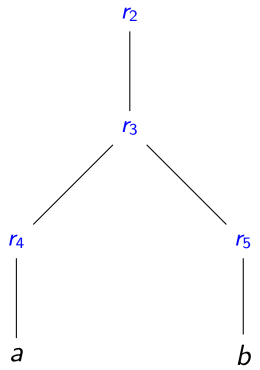
From Morphisms to $\text{dN2W}^{\downarrow}$

CFG G

r_1: $S \to RS$
r_2: $S \to R$
r_3: $R \to AB$
r_4: $A \to a$
r_5: $B \to b$

Morphism M

$M(a) = ab$
$M(b) = b$



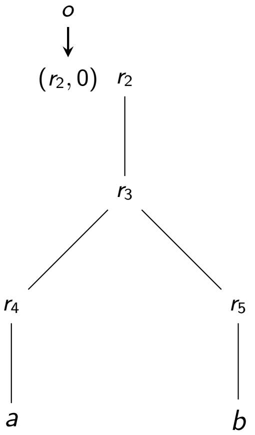
From Morphisms to dN2W$^{\downarrow}$

CFG G

r_1: $S \to RS$
r_2: $S \to R$
r_3: $R \to AB$
r_4: $A \to a$
r_5: $B \to b$

Morphism M

$M(a) = ab$
$M(b) = b$



$$o \xrightarrow{r_2:f} (r_2, 0)$$

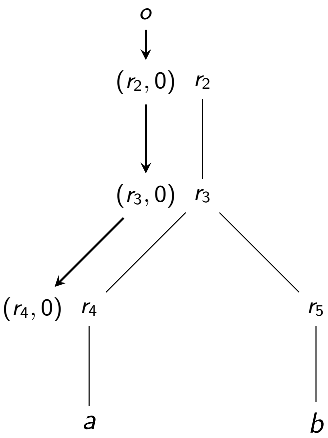
From Morphisms to dN2W$^{\downarrow}$

CFG G

r_1: $S \to RS$
r_2: $S \to R$
r_3: $R \to AB$
r_4: $A \to a$
r_5: $B \to b$

Morphism M

$M(a) = ab$
$M(b) = b$



$(r_2, 0) \xrightarrow{\text{op } r_3:(r_2,1)} (r_3, 0)$ $\quad$ $(r_3, 0) \xrightarrow{\text{op } r_4:(r_3,1)} (r_4, 0)$

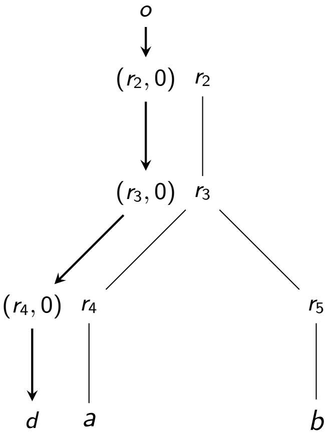
From Morphisms to dN2W$^{\downarrow}$

CFG G

r_1: $S \to RS$
r_2: $S \to R$
r_3: $R \to AB$
r_4: $A \to a$
r_5: $B \to b$

Morphism M

$M(a) = ab$
$M(b) = b$



$$(r_4, 0) \xrightarrow{\text{op } a:(r_4,1)} d$$

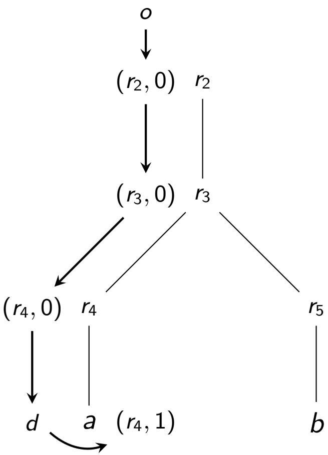
From Morphisms to dN2W$^{\downarrow}$

CFG G

r_1: $S \to RS$
r_2: $S \to R$
r_3: $R \to AB$
r_4: $A \to a$
r_5: $B \to b$

Morphism M

$M(a) = ab$
$M(b) = b$



$$d \xrightarrow{\texttt{cl}\ a:(r_4,1)} (r_4, 1)$$

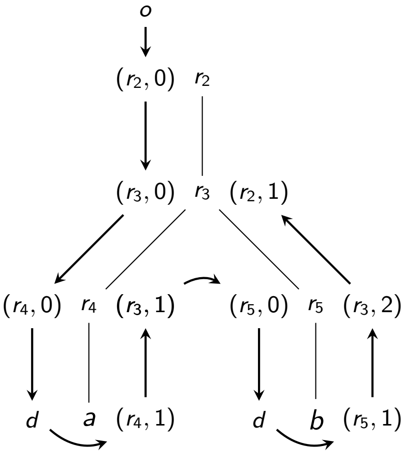
From Morphisms to dN2W$^{\downarrow}$

CFG G

r_1: $S \to RS$
r_2: $S \to R$
r_3: $R \to AB$
r_4: $A \to a$
r_5: $B \to b$

Morphism M

$M(a) = ab$
$M(b) = b$



$(r_3, 1) \xrightarrow{\texttt{op}\ r_5:(r_3,2)} (r_5, 0)$ $\quad$ $(r_5, 1) \xrightarrow{\texttt{cl}\ r_5:(r_3,2)} (r_3, 2)$

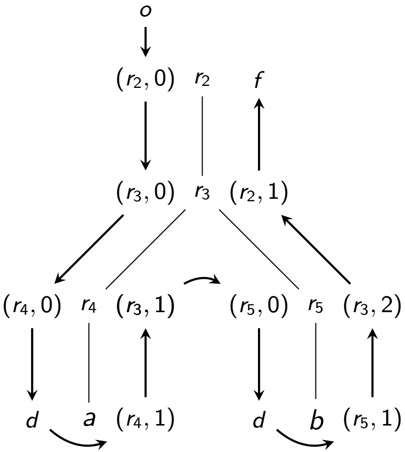
From Morphisms to $\mathrm{dN2W}^{\downarrow}$

CFG G

r_1: $S \to RS$
r_2: $S \to R$
r_3: $R \to AB$
r_4: $A \to a$
r_5: $B \to b$

Morphism M

$M(a) = ab$
$M(b) = b$



$$(r_2, 1) \xrightarrow{\mathtt{cl}\ r_2:f} f$$

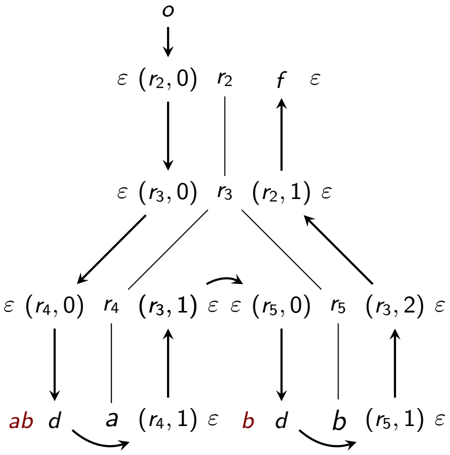
From Morphisms to dN2W$^{\downarrow}$

CFG G

r_1: $S \to RS$
r_2: $S \to R$
r_3: $R \to AB$
r_4: $A \to a$
r_5: $B \to b$

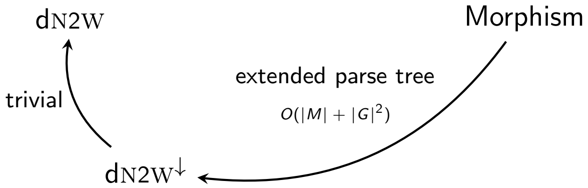
Morphism M

$M(a) = ab$
$M(b) = b$



$(r_4, 0) \xrightarrow{\text{op } a/M(a):(r_4,1)} d$ $\quad$ $(r_5, 0) \xrightarrow{\text{op } b/M(b):(r_5,1)} d$

From Morphisms to dN2W$^{\downarrow}$



From dN2W to Morphisms

Proposition

dN2W-equivalence can be reduced in polynomial time to morphism equivalence on CFGs.

From dN2W to Morphisms

Proposition

dN2W-equivalence can be reduced in polynomial time to morphism equivalence on CFGs.

For dN2W T_1 and T_2, we define :

- a CFG G which recognizes successfull parallel runs, $L(G) \subseteq (\text{rules}_{T_1} \times \text{rules}_{T_2})^*$.
- two morphisms M_1 and M_2 s.t.: For all $s \in L(G)$, $M_1(s) = [\![T_1]\!](t)$ and $M_2(s) = [\![T_2]\!](t)$.

Cf. [Culik and Karhumaki'86] (for synchonizable PDT)

From dN2W to Morphisms

T_1

$(init = \{0\}, fin = \{3\})$

r_1: $0 \xrightarrow{\texttt{op}\ a/\text{<}c\text{>}:\gamma_1} 1$

r_2: $1 \xrightarrow{\texttt{op}\ b/\varepsilon:\gamma_2} 1$

r_3: $1 \xrightarrow{\texttt{cl}\ b/\text{<}b\text{>}\text{<}/b\text{>}:\gamma_2} 2$

r_4: $2 \xrightarrow{\texttt{cl}\ a/\text{<}a\text{>}\text{<}/a\text{>}\text{<}/c\text{>}:\gamma_1} 3$

T_2

$(init = \{0'\}, fin = \{4'\})$

r'_1: $0' \xrightarrow{\texttt{op}\ a/\text{<}c\text{>}:\gamma'_1} 1'$

r'_2: $1' \xrightarrow{\texttt{op}\ b/\text{<}b\text{>}:\gamma'_2} 2'$

r'_3: $3' \xrightarrow{\texttt{cl}\ a/\text{<}/a\text{>}\text{<}/c\text{>}:\gamma'_1} 4'$

r'_4: $2' \xrightarrow{\texttt{cl}\ b/\text{<}/b\text{>}\text{<}a\text{>}:\gamma'_2} 3'$

From dN2W to Morphisms

T_1

$(init = \{0\}, fin = \{3\})$

r_1: $0 \xrightarrow{\texttt{op}\ a/<c>:\gamma_1} 1$

r_2: $1 \xrightarrow{\texttt{op}\ b/\varepsilon:\gamma_2} 1$

r_3: $1 \xrightarrow{\texttt{cl}\ b/:\gamma_2} 2$

r_4: $2 \xrightarrow{\texttt{cl}\ a/<a></c>:\gamma_1} 3$

T_2

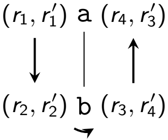
$(init = \{0'\}, fin = \{4'\})$

r'_1: $0' \xrightarrow{\texttt{op}\ a/<c>:\gamma'_1} 1'$

r'_2: $1' \xrightarrow{\texttt{op}\ b/:\gamma'_2} 2'$

r'_3: $3' \xrightarrow{\texttt{cl}\ a/</c>:\gamma'_1} 4'$

r'_4: $2' \xrightarrow{\texttt{cl}\ b/<a>:\gamma'_2} 3'$



From dN2W to Morphisms

T_1

$(init = \{0\}, fin = \{3\})$

r_1: $0 \xrightarrow{\texttt{op}\ a/<c>:\gamma_1} 1$

r_2: $1 \xrightarrow{\texttt{op}\ b/\varepsilon:\gamma_2} 1$

r_3: $1 \xrightarrow{\texttt{cl}\ b/:\gamma_2} 2$

r_4: $2 \xrightarrow{\texttt{cl}\ a/<a></c>:\gamma_1} 3$

T_2

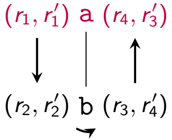
$(init = \{0'\}, fin = \{4'\})$

r'_1: $0' \xrightarrow{\texttt{op}\ a/<c>:\gamma'_1} 1'$

r'_2: $1' \xrightarrow{\texttt{op}\ b/:\gamma'_2} 2'$

r'_3: $3' \xrightarrow{\texttt{cl}\ a/</c>:\gamma'_1} 4'$

r'_4: $2' \xrightarrow{\texttt{cl}\ b/<a>:\gamma'_2} 3'$



$$[\mathbf{S}] \to (r_1, r'_1) \cdot [(\mathbf{1}, \mathbf{2}), (\mathbf{1}', \mathbf{3}')] \cdot (r_4, r'_3)$$

From dN2W to Morphisms

T_1

$(init = \{0\}, fin = \{3\})$

r_1: $0 \xrightarrow{\texttt{op}\ a/<c>:\gamma_1} 1$

r_2: $1 \xrightarrow{\texttt{op}\ b/\varepsilon:\gamma_2} 1$

r_3: $1 \xrightarrow{\texttt{cl}\ b/:\gamma_2} 2$

r_4: $2 \xrightarrow{\texttt{cl}\ a/<a></c>:\gamma_1} 3$

T_2

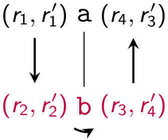
$(init = \{0'\}, fin = \{4'\})$

r'_1: $0' \xrightarrow{\texttt{op}\ a/<c>:\gamma'_1} 1'$

r'_2: $1' \xrightarrow{\texttt{op}\ b/:\gamma'_2} 2'$

r'_3: $3' \xrightarrow{\texttt{cl}\ a/</c>:\gamma'_1} 4'$

r'_4: $2' \xrightarrow{\texttt{cl}\ b/<a>:\gamma'_2} 3'$



$$[\mathbf{S}] \to (r_1, r'_1) \cdot [(\mathbf{1},\mathbf{2}),(\mathbf{1}',\mathbf{3}')] \cdot (r_4, r'_3)$$

$$[(\mathbf{1},\mathbf{2}),(\mathbf{1}',\mathbf{3}')] \to (r_2, r'_2) \cdot [(\mathbf{1},\mathbf{1}),(\mathbf{2}',\mathbf{2}')] \cdot (r_3, r'_4)$$

From dN2W to Morphisms

T_1

$(init = \{0\}, fin = \{3\})$

r_1: $0 \xrightarrow{\texttt{op}\ a/<c>:\gamma_1} 1$

r_2: $1 \xrightarrow{\texttt{op}\ b/\varepsilon:\gamma_2} 1$

r_3: $1 \xrightarrow{\texttt{cl}\ b/:\gamma_2} 2$

r_4: $2 \xrightarrow{\texttt{cl}\ a/<a></c>:\gamma_1} 3$

T_2

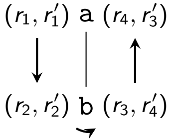
$(init = \{0'\}, fin = \{4'\})$

r'_1: $0' \xrightarrow{\texttt{op}\ a/<c>:\gamma'_1} 1'$

r'_2: $1' \xrightarrow{\texttt{op}\ b/:\gamma'_2} 2'$

r'_3: $3' \xrightarrow{\texttt{cl}\ a/</c>:\gamma'_1} 4'$

r'_4: $2' \xrightarrow{\texttt{cl}\ b/<a>:\gamma'_2} 3'$



$$[\mathbf{S}] \to (r_1, r'_1) \cdot [(\mathbf{1}, \mathbf{2}), (\mathbf{1}', \mathbf{3}')] \cdot (r_4, r'_3)$$
$$[(\mathbf{1}, \mathbf{2}), (\mathbf{1}', \mathbf{3}')] \to (r_2, r'_2) \cdot [(\mathbf{1}, \mathbf{1}), (\mathbf{2}', \mathbf{2}')] \cdot (r_3, r'_4)$$
$$[(\mathbf{1}, \mathbf{1}), (\mathbf{2}', \mathbf{2}')] \to \varepsilon$$

From dN2W to Morphisms

T_1

$(init = \{0\}, fin = \{3\})$

r_1: $0 \xrightarrow{\texttt{op}\ a/<c>:\gamma_1} 1$

r_2: $1 \xrightarrow{\texttt{op}\ b/\varepsilon:\gamma_2} 1$

r_3: $1 \xrightarrow{\texttt{cl}\ b/:\gamma_2} 2$

r_4: $2 \xrightarrow{\texttt{cl}\ a/<a></c>:\gamma_1} 3$

T_2

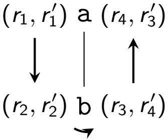
$(init = \{0'\}, fin = \{4'\})$

r'_1: $0' \xrightarrow{\texttt{op}\ a/<c>:\gamma'_1} 1'$

r'_2: $1' \xrightarrow{\texttt{op}\ b/:\gamma'_2} 2'$

r'_3: $3' \xrightarrow{\texttt{cl}\ a/</c>:\gamma'_1} 4'$

r'_4: $2' \xrightarrow{\texttt{cl}\ b/<a>:\gamma'_2} 3'$



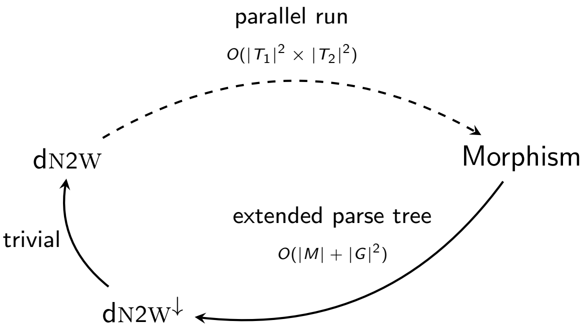
$$[\mathbf{S}] \to (r_1, r'_1) \cdot [(\mathbf{1},\mathbf{2}),(\mathbf{1}',\mathbf{3}')] \cdot (r_4, r'_3)$$

$$[(\mathbf{1},\mathbf{2}),(\mathbf{1}',\mathbf{3}')] \to (r_2, r'_2) \cdot [(\mathbf{1},\mathbf{1}),(\mathbf{2}',\mathbf{2}')] \cdot (r_3, r'_4)$$

$$[(\mathbf{1},\mathbf{1}),(\mathbf{2}',\mathbf{2}')] \to \varepsilon$$

$M_1((r_3, r'_4)) =< b >< /b >$ $\qquad$ $M_2((r_3, r'_4)) =< /b >< a >$

From dN2W to Morphisms



Other Models

Top-Down Ranked Tree to Word ($\mathrm{dR2W}^{\downarrow}$)

$$q(a(\mathtt{x}_1,\ldots,\mathtt{x}_k)) \to u_0 \cdot q_1(\mathtt{x}_1) \cdot u_1 \cdot \ldots \cdot u_{k-1} \cdot q_k(\mathtt{x}_k) \cdot u_k.$$

Generalizaton of the (non-copying and order preserving) top-down ranked tree transducer [TATA; Maneth, PhD Thesis].

Other Models

Top-Down Ranked Tree to Word ($\mathrm{dR2W}^{\downarrow}$)

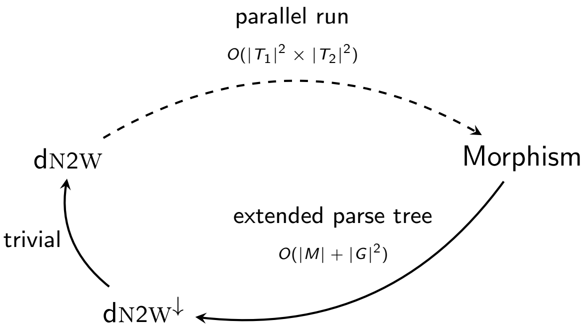
$$q(a(\mathtt{x}_1, \ldots, \mathtt{x}_k)) \to u_0 \cdot q_1(\mathtt{x}_1) \cdot u_1 \cdot \ldots \cdot u_{k-1} \cdot q_k(\mathtt{x}_k) \cdot u_k.$$

Generalizaton of the (non-copying and order preserving) top-down ranked tree transducer [TATA; Maneth, PhD Thesis].

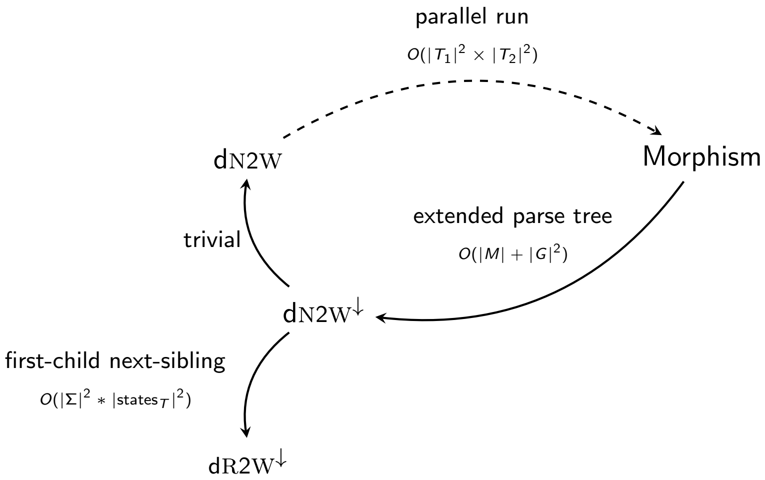
Deterministic Bottom-Up Ranked Tree to Word ($\mathrm{dR2W}^{\uparrow}$)

$$a(q_1(v_1), \ldots, q_k(v_k)) \to q(u_0 \cdot v_1 \cdot u_1 \cdots v_k \cdot u_k).$$

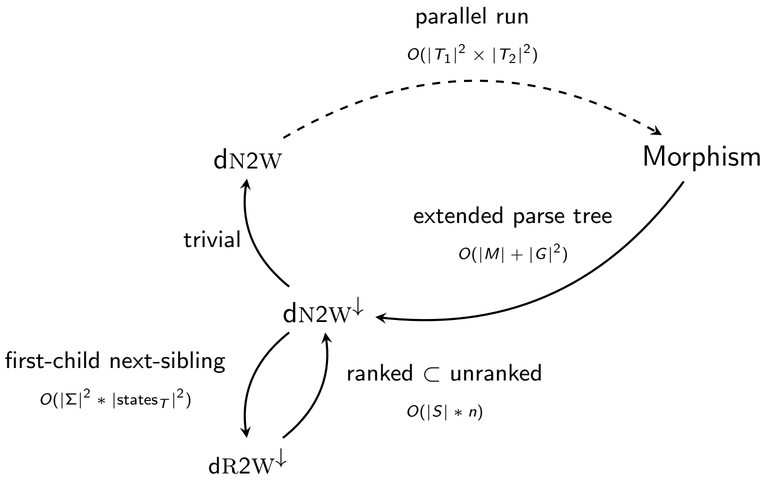
Other Models



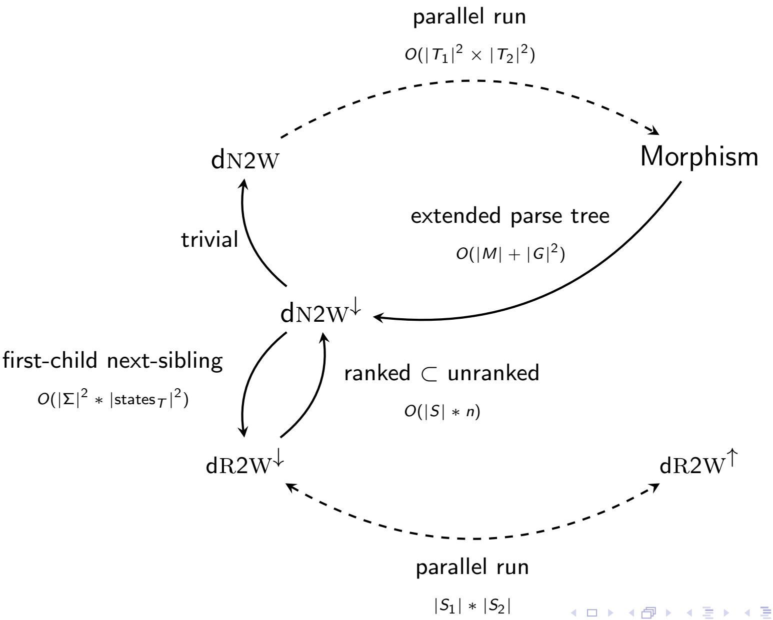
Other Models



Other Models



Other Models



Outline

Conclusion

Summary

- Several new models of tree transducers.
- Study of relationships between equivalence problems.

Future work

- Learning algorithm using grammatical inference.
- Generalization of N2W for copying and reordering (multiple sweeps)
- Uniform top-down tree transducers [Martens and Neven, ICDT'03];
- Study of other problems of *d*N2W (typing, ...)

Thank you!