# On the complexity of infinite computations 

Damian Niwiński, Warsaw University
joint work with Igor Walukiewicz and Filip Murlak
June 2005

Complexity theory gives a mathematical meaning to the concept of difficulty.

- Which of the two problems is more difficult than the other?
- Why are some particular problems difficult ?
- Can we characterize/recognize difficult problems ?
E.g., we can always decide if a regular language is star-free, but not if a context-free language is regular.


## Infinite computations

- Büchi (1960) and Rabin (1969) used the concept of infinite computations of finite automata to establish the decidability results in logic.
- D. Muller(1960) used similar concepts to analyze asynchronous digital circuits.
- Since 1980s, computer scientists study infinite computations in context of verification of computing systems (reactive, concurrent, open, ... ).

Non-termination is an expected behavior.

- Mathematicians have been playing infinite games since the 1930s (Banach-Mazur, later Gale-Stewart, . . .).


## Complexity of finite computations

Finitary decision problem

$$
A \subseteq \omega \approx\{0,1\}^{*}
$$

Classical complexity theory studies only decidable $\left(\Delta_{1}^{0}\right)$ problems, in terms of the computation time and space.

- Regular sets of words are extremely simple
$(\mathcal{O}(1)$ space, $\mathcal{O}(n)$ time on one-tape Turing machine).
- Regular sets of trees are in $L$.


## Complexity of infinite computations

An infinite computation can recognize an infinite string, or an infinite tree.


Such an object can be encoded as $f \in \omega^{\omega}$.
Classical definability theory classifies subsets of $\omega^{\omega}(\subseteq \mathcal{R})$ in terms of arithmetical, Borel, and projective hierarchies.

## From infinite to finite-example

Furst, Saxe, and Sipser 1984 have shown that the regular language PARITY is not in $A C^{0}$, i.e., it cannot be recognized by polynomial-size circuits of constant depth.

The result was first achieved by Sipser in infinite setting:
no (infinite) circuit with countable fan-in and constant depth can recognize inf-PARITY.

Here inf-PARITY is any set such that if $w, w^{\prime} \in\{0,1\}^{\omega}$ differ by one bit then

$$
w \in \inf -P A R I T Y \nLeftarrow w^{\prime} \in \inf -P A R I T Y .
$$



## Classical hierarchies

of relations $r(\alpha ; \beta) \subseteq \omega^{k} \times\left(\omega^{\omega}\right)^{\ell}$.

## Borel hierarchy.

$$
\begin{aligned}
& \left.\boldsymbol{\Sigma}_{0}^{0}=\boldsymbol{\Pi}_{0}^{0}=\{r(\alpha ; \beta, \gamma)\}: r \in \Delta_{0}^{0}, \gamma \in\left(\omega^{\omega}\right)^{m}\right\} \\
& \boldsymbol{\Sigma}_{n+1}^{0}=\left\{\{(\alpha ; \beta): \exists x r(\alpha, x ; \beta)\}: r \in \boldsymbol{\Pi}_{n}^{0}\right\} \\
& \mathbf{\Pi}_{n+1}^{0}=\left\{\{(\alpha ; \beta): \forall x r(\alpha, x ; \beta)\}: r \in \boldsymbol{\Sigma}_{n}^{0}\right\} \\
& \boldsymbol{\Sigma}_{\xi}^{0}, \boldsymbol{\Pi}_{\xi}^{0}, \ldots
\end{aligned}
$$

Projective hierarchy.

$$
\begin{aligned}
& \boldsymbol{\Sigma}_{0}^{1}=\boldsymbol{\Pi}_{0}^{1}=\bigcup_{n} \boldsymbol{\Sigma}_{n}^{0} \\
& \boldsymbol{\Sigma}_{n+1}^{1}=\left\{\{(\alpha ; \beta): \exists f r(\alpha ; \beta, f)\}: r \in \boldsymbol{\Pi}_{n}^{1}\right\} \\
& \boldsymbol{\Pi}_{n+1}^{1}=\left\{\{(\alpha ; \beta): \forall f r(\alpha ; \beta, f)\}: r \in \boldsymbol{\Sigma}_{n}^{1}\right\}
\end{aligned}
$$

## Examples - finitary objects

- The first-order theory of the standard model of arithmetics is in $\Delta_{1}^{1}$, but not in $\bigcup_{n} \Sigma_{n}^{0}$.
- The language
$\{\langle M\rangle: M$ is a non-deterministic Turing machine returning to the initial state infinitely often $\}$ is in $\Sigma_{1}^{1}$, but not in $\Pi_{1}^{1}$.

Finite problems beyond $\Delta_{1}^{0}$ ( beyond $\bigcup_{n} \Sigma_{n}^{0}$ ) are considered as (highly) uncomputable.

Examples - infinitary objects

- An $\omega$-language
$\left\{u \in\{a, b\}^{\omega}\right.$ : there are finitely many $b$ 's $\}$
is in $\boldsymbol{\Sigma}_{2}^{\mathbf{0}}$ but not in $\boldsymbol{\Pi}_{\mathbf{2}}^{\mathbf{0}}$,
- A tree language
$\left\{t \in\{a, b\}^{\{l, r\}^{*}}:\right.$ on each path, there are finitely many $b$ 's $\}$ is in $\boldsymbol{\Pi}_{1}^{\mathbf{1}}$ but not in $\boldsymbol{\Sigma}_{\mathbf{1}}^{\mathbf{1}}$.

Still, finite-state (tree) automata can recognize these sets !

```
\exists ocean, ...... ocean ...... stream ......
\exists ocean, ...... ocean ...... ocean ......
```

Classical hierarchies tell us something about complexity of infinite computations.

However, more subtle complexity measures arise from the fine structure of automata, as well as from modal and temporal logic, e.g., the $\mu$-calculus.

## Büchi automata on infinite words

$$
\mathcal{A}=\left\langle\Sigma, Q, q_{I}, \operatorname{Tr}, F\right\rangle
$$

where $\operatorname{Tr} \subseteq Q \times \Sigma \times Q, F \subseteq Q$.

$a, b$


The second one cannot be recognized by a deterministic automaton.

$$
\xrightarrow{a} \xrightarrow{b} \xrightarrow{a} \xrightarrow{a} \xrightarrow{b} \xrightarrow{a} \xrightarrow{a} \xrightarrow{a} \xrightarrow{b} \xrightarrow{a} \xrightarrow{a} \xrightarrow{a} \xrightarrow{a} \xrightarrow{b} \xrightarrow{a} \ldots \ldots \xrightarrow{a} \ldots \xrightarrow{b} \ldots
$$

## The McNaughton Theorem (1966)

Parity automata

A nondeterministic Büchi automaton can be simulated by a deterministic one with the acceptance condition rank : $Q \rightarrow\{0,1, \ldots, k\}$

$$
\limsup _{i \rightarrow \infty} \operatorname{rank}\left(q_{i}\right) \text { is even }
$$



$$
(a+b)^{*} a^{\omega}
$$

The minimal index $k$ may be arbitrarily high (Wagner 1979), but can be effectively computed
(in polynomial time, if the input automaton is deterministic N \& Walukiewicz 1998, Carton \& Maceiras 1999).

## Parity tree automata

$$
\mathcal{A}=\left\langle\Sigma, Q, q_{I}, \operatorname{Tr}, \text { rank },\right\rangle
$$

where $\operatorname{Tr} \subseteq Q \times \Sigma \times Q \times Q$, rank: $Q \rightarrow\{0,1, \ldots, k\}$.


## Parity tree automata ctd.

A run of $\mathcal{A}$ on a tree $t:\{l, r\}^{*} \rightarrow \Sigma$ is a tree $\rho:\{l, r\}^{*} \rightarrow Q$, such that, for each $w \in \operatorname{dom}(\rho),\langle\rho(w), t(w), \rho(w l), \rho(w r)\rangle \in \operatorname{Tr}$


The run is accepting if, for each path $P=p_{0} p_{1} \ldots \in\{l, r\}^{\omega}$,

$$
\limsup _{k \rightarrow \infty} \operatorname{rank}\left(\rho\left(p_{0} p_{1} \ldots p_{k}\right)\right. \text { is even. }
$$

## Example


$\operatorname{rank}(q)=0$

$\operatorname{rank}(p)=1$
recognizes the set of trees where, on each branch, $b$ appears only finitely often.

## Nondeterminism

For trivial reasons, tree automata cannot be. in general, determinized.


## Rabin's counter-example

In contrast to the automata on words, the Büchi condition alone is not sufficient, even in the presence of nondeterminism ( Rabin 1970).


## Rabin's counter-example ctd.

Descriptive complexity argument :
The Büchi recognizable sets of trees are always in $\Sigma_{1}^{1}$,
while the Rabin counter-example is $\Pi_{1}^{1}$-complete.

The idea can be traced back to the Suslin 1916 counter-example.
The set

$$
\{\langle T, u\rangle: u \text { is a branch of } T \text { with infinitely many } b \text { 's }\}
$$

is Borel, but its projection is $\boldsymbol{\Sigma}_{1}^{1}$-complete .


For the case of infinite words, the question was settled already by Wagner 1979.

For trees, we can determine the exact level of $\mathcal{T}(\mathcal{A})$, provided that $\mathcal{A}$ is a deterministic automaton
(N \& Walukiewicz 2003, Murlak 2005).
Non-deterministic case is completely open.

## From trees to words : path automata

A deterministic tree automaton $\mathcal{A}$ over alphabet $\Sigma$ can be identified with a deterministic word automaton $\mathcal{A}^{\prime}$ over alphabet $\Sigma \times\{l, r\}$,

$\mathcal{A}$ recognizes a tree $t:\{l, r\}^{*} \rightarrow \Sigma$ iff $\mathcal{A}^{\prime}$ recognizes all paths of $t$,

$$
\left(t(\varepsilon), p_{0}\right),\left(t\left(p_{0}\right), p_{1}\right),\left(t\left(p_{0} p_{1}\right), p_{2}\right),\left(t\left(p_{0} p_{1} p_{2}\right), p_{3}\right), \ldots
$$

for $p_{0} p_{1} p_{2} p_{3} \ldots \in\{l, r\}^{\omega}$.

## Example

Deterministic tree automaton :

$\operatorname{rank}(q)=0$

$$
\operatorname{rank}(p)=1
$$

Corresponding path automaton :


## Determinization, whenever possible, is effective

The concept of path automaton allows us to decide (in EXPTIME), if a given non-deterministic tree automaton is equivalent to a deterministic one.

It suffices to verify if

$$
L(\mathcal{A})=\operatorname{Trees}(\operatorname{Paths}(L(\mathcal{A})))
$$

## Criterion—effective dichotomy

If a path automaton $\mathcal{A}^{\prime}$ contains a (productive) pattern

then $\mathcal{T}(\mathcal{A})$ is $\boldsymbol{\Pi}_{1}^{1}$-complete, hence non-Borel; otherwise it is in $\boldsymbol{\Pi}_{3}^{0}$
( $\mathrm{N} \&$ Walukiewicz 2003).
The set of trees, such that on each path, there are only finitely many $b$ 's, is in $\boldsymbol{\Pi}_{1}^{1}$, but not in $\boldsymbol{\Sigma}_{1}^{1}$.

The set of trees, such that on each path $l^{m} r^{\omega}$, there are only finitely many $b$ 's, is in $\boldsymbol{\Pi}_{3}^{0}$, but not in $\boldsymbol{\Sigma}_{3}^{0}$.

## The low Borel classes

F. Murlak 2005 settles the remaining cases : $\boldsymbol{\Pi}_{2}^{0}, \boldsymbol{\Sigma}_{2}^{0}$, and $\boldsymbol{\Delta}_{3}^{0}$.

The $\boldsymbol{\Pi}_{2}^{0}$ level turns out to coincide, for deterministic languages, with deterministic Büchi automata.

The basic cases of open and closed were "folklore".

The algorithm runs in time of solving the non-emptiness problem (NP $\cap$ co-NP).

## The A. W. Mostowski's index hierarchy



Strict for tree automata : deterministic (essentially Wagner 1979), non-deterministic (N 1986), alternating (Bradfield, Arnold 1999).

## The Mostowski index hierarchy ctd.

Languages which witness the strictness of the hierarchy.

For deterministic automata on words :

$$
M_{k}=\left\{u \in\{0,1, \ldots, k\}^{\omega}: \limsup _{i \rightarrow \infty} u_{i} \text { is even }\right\}
$$

For deterministic/non-deterministic automata on trees :

$$
T_{k}=\left\{t \in\{0,1, \ldots, k\}^{\{l, r\}^{*}}: \text { each branch is in } M_{k}\right\}
$$

For alternating tree automata :
$W_{k}=$ the "game version" of the above.

## Game tree languages

Alphabet: $\{\exists, \forall\} \times\{0,1, \ldots, n\}$.

Player Eve :


$$
\exists, i
$$


$\forall, i$


Eve wins an infinite play $\left(x_{0}, i_{0}\right),\left(x_{1}, i_{1}\right),\left(x_{2}, i_{2}\right), \ldots \quad\left(x_{\ell} \in\{\exists, \forall\}\right)$
iff $\lim \sup _{\ell \rightarrow \infty} i_{\ell}$ is even.
The set $W_{n}$ consists of all trees such that Eve has a winning strategy.

## André Arnold's proof of the strictness of the hierarchy

For an alternating automaton $\mathcal{A}$ of index $n$, define the mapping

$$
\begin{aligned}
t & \longmapsto \text { compute }_{\mathcal{A}}(t) \\
t \in \mathcal{T}(\mathcal{A}) & \Longleftrightarrow \text { compute }_{\mathcal{A}}(t) \in W_{n}
\end{aligned}
$$

We can easily make it contracting (in the metric space of trees).
By Banach's Fixpoint Theorem, for some $\Delta$,

$$
\Delta=\text { compute }_{\mathcal{A}}(\Delta)
$$

Hence $\bar{W}_{n}$ cannot be recognized by an automaton of index $n$ (otherwise $\Delta \in \bar{W}_{n} \Leftrightarrow \Delta \in W_{n}$, a contradiction).

## The Mostowski hierarchy — relation to the $\mu$-calculus

The set of trees over alphabet $\{a, b\}$ where, on each branch, $b$ appears only finitely often can be presented by

$$
\mu z . \nu y \cdot a(y, y) \cup b(z, z)
$$

where

- $\mu x . t$ is the least fixed point of $x=t(x)$,
- $\nu x . t$ is the greatest fixed point of $x=t(x)$,
- $\mathbf{f}\left(L_{1}, L_{2}\right)=\left\{\quad \mathbf{f} \quad: t_{1} \in L_{1}, t_{2} \in L_{2}\right\}$.



## The Mostowski hierarchy - relation to the $\mu$-calculus ctd.

$$
\begin{aligned}
T_{n} & =\vartheta x_{n} \ldots \mu x_{2} \ldots \nu x_{1} \cdot \mu x_{0} \cdot \bigcup_{i}\left(x_{i}, x_{i}\right) \\
W_{n} & =\vartheta x_{n} \ldots \mu x_{2} \ldots \nu x_{1} \cdot \mu x_{0} \cdot \bigcup_{i}\left(d_{i}\left(x_{i}, t t\right) \cup d_{i}\left(t t, x_{i}\right) \cup c_{i}\left(x_{i}, x_{i}\right)\right)
\end{aligned}
$$

The index hierarchy of automata coincides with the $\mu$-calculus hierarchy of nesting alternately the least $(\mu)$ and the greatest $(\nu)$ fixed points.

The two hierarchies


## The two hierarchies in two versions

Non-deterministic hierarchy :

$$
x\left|f\left(t_{1}, \ldots, t_{k}\right)\right| t_{1} \vee t_{2}|\mu x . t| \nu x . t \equiv \text { non-deterministic automata }
$$

Alternating hierarchy :

$$
x\left|f\left(t_{1}, \ldots, t_{k}\right)\right| t_{1} \vee t_{2}\left|t_{1} \wedge t_{2}\right| \mu x . t \mid \nu x . t \equiv \text { alternating automata }
$$

We have

$$
\bigcup \text { Non-deterministic hierarchy }=\bigcup \text { Alternating hierarchy }
$$

but neither of the hierarchies refines the other :

- All $T_{n}$ 's are in the level $\mu \nu \equiv(0,1)$ of the alternating hierarchy.
- $T_{n}$ and $W_{n}$ are on the same level in non-deterministic hierarchy, but not in the alternating hierarchy.


## The Mostowski hierarchy — relation to complexity

The non-emptiness problem for non-deterministic parity tree automata is in NP $\cap$ co-NP (even UP $\cap$ co-UP ).

It is polynomial-time equivalent to the model-checking problem for the $\mu$-calculus.

Restricted to the automata $\mathcal{A}$ of index $n$, the problem can be solved in time $|\mathcal{A}|^{\mathcal{O}(n)}$.

Can we decide the level of a recognizable

$(1,4)$

tree language in the Mostowski hierarchy ?


Again we know the answer only if an input automaton is deterministic.

## Remarks

The question is interesting only for the non-deterministic hierarchy, because

- Computing a deterministic index of a deterministic language follows easily from the case of word automata, by reduction to path automata.
- All deterministic languages are in alternating class $(0,1)$.

Note however, that a deterministic automaton can be simulated by a non-deterministic one with a smaller index, e.g.,
$\left\{t:\right.$ the leftmost branch is in $\left.M_{n}\right\}$

## The problem

Given: a deterministic parity tree automaton

Compute: the minimal Mostowski index of a non-deterministic automaton recognizing the same language.

Urbański 2000 solved the case of Büchi,
N \& Walukiewicz 2004 settle the whole hierarchy.

```
Tree automata - forbidden patterns
```

If a (deterministic) path automaton contains

it cannot be simulated by a (1,2)-automaton.
(Essentially the Rabin's pumping argument.)

If it contains

it cannot be simulated by a $(0,1)$-automaton.

## Forbidden patterns ctd.

Indices $(1, n)$ and $(0, n-1)$ are dual,
we note $\overline{(\iota, n)}$ for the dual of $(\iota, n)$.
For each index $(\iota, n)(\iota \in\{0,1\})$, we construct a pattern

$$
P(\iota, n)
$$

which "fools" any deterministic automaton of index $\overline{(\iota, n)}$.


$$
\text { The }(0, n) \text { case, } n \geq 3
$$


fools any $(1, n+1)$ automaton.

The $(1, n)$ case, $n \geq 4$

fools any ( $0, n-1$ ) automaton.

Theorem. Let $\mathcal{A}$ be a deterministic tree automaton.
Then $L(\mathcal{A})$ can be recognized by a non-deterministic tree automaton of index $(\iota, n)$ if and only if the corresponding path automaton does not contain any productive $\overline{(\iota, n)}$ pattern.

An idea of the proof.
$(\Leftarrow)$ Unravel a forbidden pattern into a tree and refine Rabin's argument.
$(\Rightarrow)$ Decompose $\mathcal{A}$ into strongly connected components, and apply inductive arguments to the sub-automata induced this way.

Corollary. Consequently, the index of a deterministic tree language can be computed within the complexity of computing productive states (i.e., NP $\cap$ co-NP ).

Relating the hierarchies:
Do the topological hardness and the automata-theoretic hardness always coincide ?

Skurczyński 1993 showed that there are recognizable tree languages on every finite level of the Borel hierarchy, and we now that there are also some $\boldsymbol{\Sigma}_{1}^{1}$ and $\Pi_{1}^{1}$-complete ones.


Comparing languages: Wadge reducibility
Let $\mathcal{T}$ be a topological space, $A, B \subseteq \mathcal{T}$.
$A \leq_{w} B$ if there is a continuous mapping $h: \mathcal{T} \rightarrow \mathcal{T}$,
such that, $h^{-1}(B)=A$.
Fact (Büchi \& Landweber 1969). It is decidable if $A \leq_{w} B$, for $A, B \subseteq \Sigma^{\omega}$, $\omega$-regular languages of infinite words.

Problem. Can we decide if $A \leq_{w} B$, for recognizable sets of trees, $A$ and $B$ ?
At least for deterministic ones ?
Partial answer. Yes, if both are in $\boldsymbol{\Delta}_{2}^{0}$ (Murlak 2005).
Also, if at least one (say $B$ ) is in $\boldsymbol{\Pi}_{1}^{1}-$ Borel (then $A \leq_{w} B$ ).
Conjecture. A recognizable set of trees $L$ is not on the level $(1,2)$ (Büchi) if and only if $T_{1} \leq_{w} L$ (known to hold for deterministic $L$ ).

```
Wadge reducibility - contrast between words and trees
```

Recall : $\quad M_{k}=\left\{u \in\{0,1, \ldots, k\}^{\omega}: \lim \sup _{i \rightarrow \infty} u_{i}\right.$ is even $\}$

$$
T_{k}=\left\{t \in\{0,1, \ldots, k\}^{\{l, r\}^{*}}: \text { each branch is in } M_{k}\right\}
$$

We have $M_{1}<_{w} M_{2}<_{w} M_{3}<_{w} \ldots$
but $T_{1} \equiv_{w} T_{2} \equiv_{w} T_{3} \equiv_{w} \ldots$ (all $T_{n}$ 's are $\Pi_{1}^{1}$-complete).
Yet still, $W_{1}<_{w} W_{2}<_{w} W_{3}<_{w} \ldots$
Fact (Büchi \& Landweber 1969). For $\omega$-regular word languages, if $A \leq_{w} B$ then there exists a finite-state transducer reducing $A$ to $B\left(A \leq_{s} B\right)$.

This is no more true for trees.
However, for deterministic tree automata $\mathcal{A}, \mathcal{B}$, Murlak 2005 defines a game $\mathcal{G}(\mathcal{A}, \mathcal{B})$ (similar to the Wadge game), such that $\mathcal{T}(\mathcal{A}) \leq_{w} \mathcal{T}(\mathcal{B})$ iff duplicator wins $\mathcal{G}(\mathcal{A}, \mathcal{B})$.

## A complete set for deterministic tree languages

Any deterministically recognizable set of trees is reducible by a transducer to $T_{1}$.

We first show a generic reduction of $\mathcal{T}(A)$ to $T_{2}$.
Let $r$ be a unique run of an automaton $A$ on a tree $t$. For each odd $i \leq n$, let

$$
\begin{gathered}
r_{i}(w)= \begin{cases}0 & \text { if } \operatorname{rank} r(w)<i \\
1 & \text { if } \operatorname{rank} r(w)=i \\
2 & \text { if } \operatorname{rank} r(w)>i\end{cases} \\
t \mapsto\left(r_{1}, 0\left(r_{3}, 0\left(r_{5}, \ldots 0\left(r_{2 \cdot\left\lceil\frac{n}{2}\right\rceil-3}, r_{2 \cdot\left\lceil\frac{n}{2}\right\rceil-1}\right) \ldots\right)\right)\right)
\end{gathered}
$$


$\mapsto$


## Reduction of $T_{2}$ to $T_{1}$.


where $\operatorname{local}\left(t_{i}\right)$ is $t_{i}$ reproduced till first 2 ,
$\operatorname{global}\left(0\left(t_{1}^{\prime}, t_{2}^{\prime}\right)\right)=\operatorname{global}\left(1\left(t_{1}^{\prime}, t_{2}^{\prime}\right)\right)=0\left(\operatorname{global}\left(t_{1}^{\prime}\right), \operatorname{global}\left(t_{2}^{\prime}\right)\right)$,
global $\left(2\left(t_{1}^{\prime}, t_{2}^{\prime}\right)\right)$ as above.

## Related questions and results

Given: a formula of some logic $\mathcal{L}$.
Question : is it equivalent to a formula of some sub-logic $\mathcal{L}^{\prime} \subseteq \mathcal{L}$ ?
In particular
Given: a formula of the $\mu$-calculus.
Question: Determine its level in the $\mu \nu$-hierarchy.
M. Otto 1999 showed how to decide if $\mu$ and $\nu$ can be completely eliminated in a formula.

Walukiewicz 2002 settled the $\mu$ and $\nu$ levels. What about the next levels ?
O. Finkel and J. Duparc studied the topological complexity and Wadge reducibility for (deterministic) $\omega$-context-free languages.

Conclusion. In contrast to the finitary case, finite state automata running over infinite words or trees can recognize highly complex properties of infinite computations (e.g., $\boldsymbol{\Pi}_{1}^{1}$-complete).

Automata also provide fine hierarchies, complementary to the classical Borel/projective hierarchies.

For deterministic automata, we can decide its exact level in the complexity hierarchies.

The non-deterministic case needs new ideas.

## References

D. Niwiński and I. Walukiewicz, Relating hierarchies of word and tree automata, STACS 1998, LNCS 1373, 320-331.
D. Niwiński and I. Walukiewicz, A gap property of deterministic tree languages, TCS, 303, 2003, 215-231.
D. Niwiński and I. Walukiewicz, Deciding Nondeterministic Hierarchy of Deterministic Tree Automata, WoLLIC 2004, Electronic Notes in Theoretical Computer Science, 2005, 195-208.
F. Murlak, On Deciding Topological Classes of Deterministic Tree Languages, CSL 2005, to appear.
F. Murlak, notes, 2005.

