

# Problem równoważności transdukcji: algorytmy bazujące na algebrze przemiennej

Autoreferat

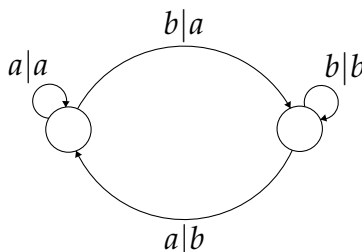
Janusz Schmude

W niniejszej rozprawie koncentrujemy się na problemie rozstrzygalności funkcyjności oraz równoważności funkcyjnych transduktorów rejestrowych (register transducers), których dane wejściowe są drzewami. Głównym motywem rozprawy jest Metoda Hilberta, która w zarysie polega na redukcji obiektów kombinatorycznych, takich jak słowa bądź drzewa, do obiektów algebraicznych, takich jak liczby czy wielomiany, w taki sposób, że operacje tych pierwszych mogą być zredukowane do operacji tych drugich. Szczegóły Metody Hilberta opisujemy później; teraz przejdźmy do transduktorów.

W zarysie, *transduktor drzewiasty* jest to formalny model maszynowy zbudowany na automacie drzewiastym, który nie tylko akceptuje drzewa, ale i zwraca wyjście, które może być na przykład innym drzewem bądź liczbą. W konsekwencji, definiuje on nie tylko język, lecz także transformację. Definicję transduktora rejestrowego podajemy dalej w tekście. Transduktor nazywamy *funkcyjnym*, jeżeli zdefiniowana przez niego transformacja jest funkcją (tj. ma co najwyżej jedno wyjście dla każdego wejścia), a dwa transduktory nazywamy *równoważnymi*, jeżeli definiują tę samą transformację. Procedura decyzyjna rozstrzygająca problem równoważności transduktorów może zostać wykorzystana na przykład do zweryfikowania, czy dany transduktor definiuje oczekiwaną przez nas transformację. W tym celu można wziąć inny transduktor, który definiuje interesującą nas transformację w sposób niewątpliwy (choć być może nieoptymalny), a następnie sprawdzić, czy oba transduktory są równoważne.

Motywacja do rozważania transduktorów drzewiastych pochodzi między innymi z teorii transformacji dokumentów<sup>1</sup>, teorii kompilatorów<sup>2</sup> oraz przetwarzania języka naturalnego<sup>3</sup>. Po więcej informacji i referencji dotyczących transduktorów drzewiastych i ich zastosowań odsyłamy do podręcznika Fülöpa i Voglera<sup>4</sup>.

W dziedzinie transduktorów można znaleźć całą gamę modeli. Pozwólmy sobie podać kilka przykładów. Rozpocznijmy od transduktora słownego – jest to szczególny przypadek transduktora drzewiastego, ponieważ słowa mogą być reprezentowane jako drzewa, w których każdy wierzchołek ma co najwyżej jedno dziecko. Najstarszy i najprostszy model transduktora słownego to Maszyna Mealy’ego (Mealy Machine), wprowadzona w 1955 roku przez Mealy’ego<sup>5</sup>. Przechodzi ona przez słowo wejściowe od lewej do prawej tak jak automat skończony (DFA) oraz w każdym kroku zwraca jedną literę, w zależności od aktualnego stanu i litery wejściowej; stany maszyny służą wyłącznie do determinowania liter wyjściowych – maszyna zwraca wyjście dla każdego słowa wejściowego.



Rysunek 1: Maszyna Mealy’ego. Na przykład, dla słowa wejściowego *abbba* zwraca słowo *aabbb*.

Nieco ogólniejszym modelem transduktorowym jest deterministyczny transduktor skończenie stanowy (deterministic finite state transducer). Można

<sup>1</sup>Hosoya, 2010.

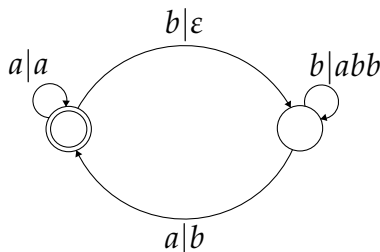
<sup>2</sup>Na przykład Thirunarayan, 2009. Gramatyki atrybutowe (attribute grammars) można zinterpretować jako transduktory drzewiaste: gramatyki atrybutowe oraz drzewiaste transduktory atrybutowe (attributed tree transducers) mają tę samą siłę ekspresji (Maneth, 1998).

<sup>3</sup>Maletti, 2015.

<sup>4</sup>Fülöp and Vogler, 1998.

<sup>5</sup>Mealy, 1955.

go krótko opisać jako automat skończony z wyjściem: w każdym kroku zwraca słowo (które, inaczej niż w przypadku Maszyny Mealy'ego, może zawierać więcej niż jedną literę) oraz akceptuje lub odrzuca słowo wejściowe w zależności od stanu osiągniętego po przeczytaniu go.



Rysunek 2: Deterministyczny transduktor skończenie stanowy. Zwraca wyjście dla słów postaci  $(a + b)^*a$ ; na przykład, dla słowa wejściowego *abbba* zwraca słowo *aabbabbb*.

Analogicznie, można skonstruować modele transduktorowe posługując się innymi modelami automatowymi, otrzymując co następuje: niedeterministyczny transduktor skończenie stanowy (nondeterministic finite state transducer), transduktor ze stosem (push-down transducer), dwukierunkowy (nie) deterministyczny transduktor skończony (two-way (non)deterministic finite transducer), transduktor kamykowy (pebble transducer)<sup>6</sup> oraz transduktor marmurowy (marble transducer)<sup>7</sup> (nie twierdzimy bynajmniej, że przedstawiona lista jest wyczerpująca). Co więcej, można zastosować analogiczną konstrukcję do automatów drzewiastych, otrzymując (nie)deterministyczny góra-dół (dół-góra) transduktor ((non)deterministic top-down (bottom-up) transducer) czy też drzewiaste transduktory kamykowe (pebble tree transducers). Istnieją również modele transduktorowe innych rodzajów, wspomnijmy o drzewiastym transduktorze „macro” (macro tree transducer)<sup>8</sup>, transduktorze atrybutowym (attribute transducer), otrzymanym z gramatyki atrybutowej (attribute grammar)<sup>9</sup>, transduktorze ze stosem wyższego

<sup>6</sup>Automaty kamykowe (pebble automata) zostały wprowadzone w pracy Globberman and Harel, 1996.

<sup>7</sup>Automaty marmurowe (marble automata) zostały wprowadzone, dla drzew, w pracy Engelfriet, Hoogeboom, and Van Best, 1999.

<sup>8</sup>Engelfriet and Vogler, 1985.

<sup>9</sup>Fülöp, 1981.

rzędu (Higher Order deterministic top-down transducer)<sup>10</sup> oraz drzewiastym transduktorze uziemionym (ground tree transducer)<sup>11</sup>.

**Modele transformacji używane w tej rozprawie.** Przejdźmy do modeli transformacji, których używamy w niniejszej rozprawie. Pierwszy jest zdefiniowany z użyciem monadycznej logiki drugiego rzędu (MSO); jest to MSO *transdukcja*. MSO jest rozszerzeniem logiki pierwszego rzędu, w którym kwantyfikacja przebiega po zbiorach elementów struktury. Związki MSO i teorii automatów pojawiły się w latach sześćdziesiątych ubiegłego wieku kiedy to sławne twierdzenie Büchiego, Elgota oraz Trakhtenbrota stwierdziło równoważność DFA oraz MSO na słowach z relacją następnika<sup>12</sup>; podobny wynik został otrzymany zarówno dla drzew<sup>13</sup> jak i dla grafów<sup>14</sup>. Związek ten został rozszerzony z automatów do transduktorów: Engelfriet i Hoogeboom udowodnili równoważność dwukierunkowych deterministycznych automatów na słowach oraz MSO transdukcji słów z relacją następnika<sup>15</sup>, a Engelfriet i Maneth udowodnili równoważność drzewiastych transduktorów „macro” z regularnym wglądem naprzód (macro tree transducers with regular look-ahead) z restrykcją jednokrotnego wykorzystania (single-use restriction) oraz MSO transdukcji drzew<sup>16</sup>; co więcej, Alur et al. wprowadzili inny równoważny model (transduktor rejestrowy, por. następny akapit) zarówno dla słów<sup>17</sup> oraz drzew<sup>18</sup>. MSO transdukcje były badane również dla grafów, patrz przegląd Courcelle’a<sup>19</sup>, aczkolwiek bez związków z drzewiastymi transduktorami. (W rozdziale 3 rozważamy MSO transdukcje grafów (o ograniczonej szerokości drzewiastej), wprowadzając związki z transduktorami rejestrowymi (wykorzystując wspomniane wyniki Alura et al.).)

Drugim i najważniejszym modelem transformacji którego używamy

---

<sup>10</sup>Gallot, Lemay, and Salvati, 2020.

<sup>11</sup>Dauchet et al., 1990.

<sup>12</sup>Büchi, 1960, Elgot, 1961, and Trakhtenbrot, 1961.

<sup>13</sup>Dowód jest analogiczny jak w przypadku słów.

<sup>14</sup>Courcelle and Engelfriet, 2012.

<sup>15</sup>Engelfriet and Hoogeboom, 2001.

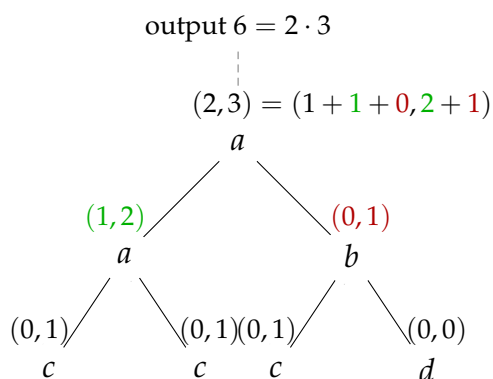
<sup>16</sup>Engelfriet and Maneth, 1999.

<sup>17</sup>Alur and Černý, 2010; Alur and Černý, 2011.

<sup>18</sup>Alur and D’Antoni, 2012.

<sup>19</sup>Courcelle, 1994; patrz również powiązane komentarze we wstępie w pracy Bojańczyk and Pilipczuk, 2016.

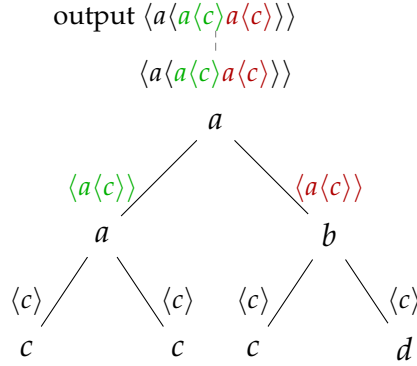
jest, podążając za pracami Engelfrieta, Manetha, Alura, Černýego i D'Antoni'ego<sup>20</sup>, transduktor drzewiasty wykorzystujący rejestry; nazywamy go *transduktorem rejestrowym*, za Bojańczykiem<sup>21</sup>. Zwraca on elementy pewnej ustalonej algebry (na przykład monoidu wolnego lub ciała liczb wymiernych), oznaczonej jako algebra wyjściowa. (Transduktor rejestrowy z algebrą wyjściową  $\mathbb{A}$  nazywamy również transduktorem rejestrowym z wyjściem w  $\mathbb{A}$ .) Transduktor rejestrowy bazuje na automacie drzewiastym dół-góra i jest wzbogacony o krotkę zmiennych, zwanych rejestrami, przechowujących elementy algebry wyjściowej. Krotka rejestrów jest inicjalizowana w każdym liściu drzewa wejściowego, następnie w każdym wierzchołku nie będącym liściem jest ona syntetyzowana przez zastosowanie wielomianowej operacji algebry  $\mathbb{A}$  (w sensie algebry uniwersalnej) na rejestrach jego dzieci, a w końcu zwracane wyjście jest ewaluacją wielomianowej operacji  $A$  na rejestrach korzenia. Algebra wyjściowa może być dowolna, choć zwykle jest to albo algebra "kombinatoryczna", jak na przykład monoid wolny lub algebra drzew urangowanych (ranked trees), albo pierścień, na przykład liczb całkowitych lub wielomianów.



Rysunek 3: Bieg transduktora rejestrowego z wyjściem w pierścieniu liczb całkowitych, który oblicza iloczyn liczb wierzchołków drzewa wejściowego etykietowanych literami  $a$  i  $c$ .

<sup>20</sup>Engelfriet and Maneth, 1999; Alur and Černý, 2010; Alur and Černý, 2011; Alur and D'Antoni, 2012.

<sup>21</sup>Bojańczyk, 2019.



Rysunek 4: Bieg transduktora rejestrowego z wyjściem w monoidzie wolnym, który oblicza reprezentację słowną drzewa wejściowego.

Transduktory rejestrowe są ekspresywne: dla algebry wyjściowej będącej monoidem wolnym oraz alfabetu wejściowego składającego się z symboli unarnych i zero-arnych ściśle zawierają MSO transdukcje słów, a dla algebry wyjściowej będącej wolną algebrą leśną (forest algebra, wprowadzona przez Bojańczyka i Walukiewicza<sup>22</sup>) ściśle zawierają MSO transdukcje drzew. Zostały one wprowadzone, w tzw. wariancie niekopiującym (copyless restriction), przez Alura i Černý'ego dla słów<sup>23</sup> oraz przez Alura i D'Antoniego dla drzew<sup>24</sup> (we wspomnianym wariancie niekopiującym w obu przypadkach transduktory rejestrowe i odpowiadające im transdukcje MSO są jednakowo ekspresywne<sup>25</sup>). Transduktory rejestrowe były również używane przez Seidla et al.<sup>26</sup>, przy czym algebrą wyjściową było ciało liczb wymiernych, dla których udowodnili oni rozstrzygalność równoważności<sup>27</sup>.

**Metoda Hilberta.** Schemat dowodowy, którego używamy w tej pracy, jest

<sup>22</sup>Bojańczyk and Walukiewicz, 2008.

<sup>23</sup>Alur and Černý, 2010.

<sup>24</sup>Alur and D'Antoni, 2012.

<sup>25</sup>Alur and Černý, 2011; Alur and D'Antoni, 2012.

<sup>26</sup>Seidl, Maneth, and Kemper, 2018.

<sup>27</sup>Dowód jest przeprowadzony dla transduktorów deterministycznych; przypadek funkcyjny wynika z dość standardowej redukcji do przypadku deterministycznego, por. dowód Lematu 2.9 tej rozprawy.

analogiczny do tego, którego użyli Seidl et al. Symulowali oni podklasę transduktorów rejestrowych z wyjściem w monoidzie wolnym (deterministyczne transduktory góra-dół z drzew do słów (deterministic top-down tree-to-string transducers)) przez transduktory rejestrowe z wyjściem w pierścieniu liczb całkowitych. Ta symulacja sprowadza się do symulacji struktur wyjściowych: monoid wolny jest symulowany przez pierścień liczb całkowitych, poprzez zanurzenie go w multiplikatywnym monoidzie pierścienia macierzy, patrz poniższy rysunek<sup>28</sup>.

$$\begin{aligned} a_1 &\xrightarrow{\phi} \begin{bmatrix} 3 & 1 \\ 0 & 1 \end{bmatrix} \\ a_2 &\xrightarrow{\phi} \begin{bmatrix} 3 & 2 \\ 0 & 1 \end{bmatrix} \\ w \cdot v &\xrightarrow{\phi} \phi(w) \cdot \phi(v) \quad \text{dla } w, v \in \{a_1, a_2\}^*. \end{aligned}$$

Rysunek 5: Symulacja monoidu wolnego  $\{a_1, a_2\}^*$  przez pierścień liczb całkowitych, z wykorzystaniem homomorfizmu do multiplikatywnego monoidu macierzy; słowo  $w = a_{i_1} a_{i_2} \dots a_{i_n}$  jest odwzorowane na  $\begin{bmatrix} 3^n & w_3 \\ 0 & 1 \end{bmatrix}$  gdzie  $w_3$  jest liczbą reprezentowaną przez słowo  $i_1 i_2 \dots i_n \in \{1, 2\}^*$  w systemie trójkowym.

Seidl et al. uzyskali rozstrzygalność równoważności dla pierwszego modelu transduktorowego poprzez udowodnienie rozstrzygalności równoważności dla drugiego modelu, używając ideałów w pierścieniu wielomianów. Analogiczny schemat dowodowy, wywodzący się z teorii reprezentacji, był już wcześniej stosowany w teorii języków formalnych<sup>29</sup>, a ostatnio został nazwany *Metodą Hilberta* przez Bojańczyka<sup>30</sup>. Wykorzystujemy go i rozwijamy w każdym z następnych rozdziałów – za każdym razem z algebrą inną niż monoid wolny i pierścieniem innym niż pierścień liczb całkowitych – i uzyskujemy wyniki rozstrzygalności dla innych klas przekształceń.

<sup>28</sup>Ten konkretny homomorfizm injektywny pochodzi z Seidl, Maneth, and Kemper, 2018, Example 3.2; w literaturze można znaleźć analogiczne homomorfizmy, patrz na przykład klasyczna liniowa reprezentacja Sanova grupy wolnej.

<sup>29</sup>Albert and Lawrence, 1985; Honkala, 2000; Benedikt et al., 2017.

<sup>30</sup>Bojańczyk, 2019.

**Organizacja rozprawy.** W rozdziale 1 podajemy preliminaria, dotyczące pierścieni, ideałów w pierścieniu wielomianów oraz baz Gröbnera.

W rozdziale 2 przedstawiamy dowód rozstrzygalności funkcyjności oraz równoważności funkcyjnych transduktorów rejestrowych z wyjściem w obliczalnym pierścieniu przemiennym bez dzielników zera (lub, bez straty ogólności, z wyjściem w ciele obliczalnym<sup>31</sup>) (Twierdzenie 5)<sup>32</sup>. Następnie symulujemy pewną algebrę lasów nieuporządkowanych przez pierścień wielomianów jednej zmiennej, uzyskując w ten sposób rozstrzygalność dla transduktorów rejestrowych z wyjściem w tej algebrze. Formalizujemy również pojęcie symulacji algebr oraz zastosowany schemat dowodowy.

W rozdziale 3 podajemy model transduktora rejestrowego, który pozwala wyrazić transdukcję MSO grafów o ograniczonej szerokości drzewiastej. Następnie symulujemy grafy o ograniczonej szerokości drzewiastej – z dokładnością do pewnego osłabienia relacji izomorfizmu, które nazywamy *równoważnością spacerową* – przez ciało funkcji wymiernych jednej zmiennej wzbogacone o operację dzielenia. Rozszerzamy zakres Twierdzenia 5 do transduktorów rejestrowych wykorzystujących operację dzielenia, dzięki czemu nasza symulacja pozwala otrzymać rozstrzygalność funkcyjności oraz równoważności funkcyjnych MSO transdukcji grafów o ograniczonej szerokości drzewiastej, w których grafy wyjściowe są identyfikowane z dokładnością do spacerowej równoważności.

W rozdziale 4 rozważamy transduktory rejestrowe z wyjściem w pierścieniu wielomianów wzbogaconym o operację podstawiania. Najpierw otrzymujemy nierozstrzygalność równoważności dla tego modelu. Następnie, otrzymujemy dwa pozytywne wyniki, wykorzystując ograniczone warianty tego modelu. Pierwszy wynik mówi o rozstrzygalności spełnialności (potencjalnie nieskończonego) układu równań na słowach przez każdą krotkę słów z danego języka, w przypadku jeśli zarówno układ, jak i język są generowane przez transduktor rejestrowy (zauważmy, że równanie o zmiennych ze zbioru  $X$  może być reprezentowane jako para

<sup>31</sup>Transduktory rejestrowe z wyjściem w ciałach mają taką samą moc ekspresji jak transduktory rejestrowe z wyjściem w pierścieniach bez dzielników zera (por. Twierdzenie 5 rozprawy). Jest tak dlatego, że każde ciało jest pierścieniem bez dzielników zera, a każdy pierścień przemienny bez dzielników zera jest zawarty w jakimś ciele (np. w swoim ciele ułamków).

<sup>32</sup>Dowód powtarza argumenty Seidla et al. (w równoważnym formalizmie gramatyk wielomianowych), ale przy słabszych założeniach na pierścieniach wyjściowych.



słów nad alfabetem zawierającym  $X$ ). Drugi wynik mówi o rozstrzygalności funkcyjności oraz równoważności funkcyjnych transduktorów rejestrowych z wyjściem w monoidzie wolnym wzbogaconym o homomorfizmy słów, które spełniają następujące ograniczenia: (1) homomorfizmy muszą być zaaplikowane jednocześnie do wszystkich rejestrów, oraz (2) każdy z tych homomorfizmów, po zastosowaniu do niego naszej symulacji monoidu wolnego przez pierścień wielomianów, indukuje automorfizm pewnego ciała zawierającego wyjściowy pierścień wielomianów (pokazujemy, że własność tę można łatwo rozstrzygnąć).

W rozdziale 5 proponujemy schemat dowodowy na rozstrzygalność funkcyjności oraz równoważności funkcyjnych transduktorów rejestrowych z wyjściem w danym skończeniu przedstawionym monoidzie (klasycznym przykładem takiego monoidu w informatyce jest monoid śladów (trace monoid)). Schemat ten stosujemy z powodzeniem dla przykładowego monoidu ( $\mathbb{M} = \langle a, b, c \mid abc = cba \rangle$ ), choć nie pokazujemy, że jest on bardziej użyteczny niż prostsze metody dowodowe. Wreszcie, zauważamy, że nasz schemat dowodowy pozwala otrzymać kryterium *równaniowej Noetherowskości* (equational Noetherianity) (zwanej też *własnością zwartości* (compactness property)) danego monoidu skończenie przedstawionego, czyli własności mówiącej, że każdy nieskończony układ równań posiada skończony równoważny podukład.

## Referencje

- Albert, M. H. and J. Lawrence (Dec. 1985). “A Proof of Ehrenfeucht’s Conjecture”. In: *Theor. Comput. Sci.* 41.1, pp. 121–123. ISSN: 0304-3975.
- Alur, R. and P. Černý (2010). “Expressiveness of streaming string transducers”. In: *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2010)*. Ed. by Kamal Lodaya and Meena Mahajan. Vol. 8. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, pp. 1–12. ISBN: 978-3-939897-23-1. DOI: 10.4230/LIPIcs.FSTTCS.2010.1. URL: <http://drops.dagstuhl.de/opus/volltexte/2010/2853>.
- (2011). “Streaming Transducers for Algorithmic Verification of Single-Pass List-Processing Programs”. In: *Proceedings of the 38th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*.

- POPL '11. Austin, Texas, USA: Association for Computing Machinery, pp. 599–610. ISBN: 9781450304900. DOI: 10 . 1145 / 1926385 . 1926454. URL: <https://doi.org/10.1145/1926385.1926454>.
- Alur, R. and L. D’Antoni (2012). “Streaming Tree Transducers”. In: *Automata, Languages, and Programming*. Ed. by Artur Czumaj, Kurt Mehlhorn, Andrew Pitts, and Roger Wattenhofer. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 42–53. ISBN: 978-3-642-31585-5.
- Benedikt, M., T. Duff, A. Sharad, and J. Worrell (2017). “Polynomial automata: Zeroness and applications”. In: *2017 32nd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pp. 1–12. DOI: 10 . 1109 / LICS . 2017 . 8005101.
- Bojańczyk, M. (Feb. 2019). “The Hilbert Method for Transducer Equivalence”. In: *ACM SIGLOG News* 6.1, pp. 5–17. DOI: 10 . 1145 / 3313909 . 3313911. URL: <https://doi.org/10.1145/3313909.3313911>.
- Bojańczyk, M. and Mi. Pilipczuk (2016). “Definability Equals Recognizability for Graphs of Bounded Treewidth”. In: *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science*. LICS ’16. New York, NY, USA: Association for Computing Machinery, pp. 407–416. ISBN: 9781450343916. DOI: 10 . 1145 / 2933575 . 2934508. URL: <https://doi.org/10.1145/2933575.2934508>.
- Bojańczyk, M. and I. Walukiewicz (2008). “Forest algebras”. In: *Logic and Automata: History and Perspectives [in Honor of Wolfgang Thomas]*. Ed. by Jörg Flum, Erich Grädel, and Thomas Wilke. Vol. 2. Texts in Logic and Games. Amsterdam University Press, pp. 107–132.
- Büchi, J. (1960). “Weak Second-Order Arithmetic and Finite Automata”. In: *Mathematical Logic Quarterly* 6, pp. 66–92.
- Courcelle, B. (1994). “Monadic second-order definable graph transductions: a survey”. In: *Theoretical Computer Science* 126.1, pp. 53–75. ISSN: 0304-3975. DOI: [https://doi.org/10.1016/0304-3975\(94\)90268-2](https://doi.org/10.1016/0304-3975(94)90268-2). URL: <https://www.sciencedirect.com/science/article/pii/0304397594902682>.
- Courcelle, B. and J. Engelfriet (2012). *Graph Structure and Monadic Second-Order Logic: A Language-Theoretic Approach*. Encyclopedia of Mathematics and its Applications. Cambridge University Press. DOI: 10 . 1017 / CB09780511977619.
- Dauchet, M., T. Heuillard, P. Lescanne, and S. Tison (1990). “Decidability of the confluence of finite ground term rewrite systems and of other related term rewrite systems”. In: *Information and Computation*

- 88.2, pp. 187–201. ISSN: 0890-5401. DOI: [https://doi.org/10.1016/0890-5401\(90\)90015-A](https://doi.org/10.1016/0890-5401(90)90015-A). URL: <https://www.sciencedirect.com/science/article/pii/089054019090015A>.
- Elgot, C. C. (1961). “Decision Problems of Finite Automata Design and Related Arithmetics”. In: *Transactions of the American Mathematical Society* 98.1, pp. 21–51. ISSN: 00029947. URL: <http://www.jstor.org/stable/1993511>.
- Engelfriet, J. and H. J. Hoogeboom (Apr. 2001). “MSO Definable String Transductions and Two-Way Finite-State Transducers”. In: *ACM Trans. Comput. Logic* 2.2, pp. 216–254. ISSN: 1529-3785. DOI: 10.1145/371316.371512. URL: <https://doi.org/10.1145/371316.371512>.
- Engelfriet, J., H. J. Hoogeboom, and J.-P. Van Best (1999). “Trips on Trees”. In: *ACTA CYBERNETICA* 14, pp. 51–64.
- Engelfriet, J. and S. Maneth (1999). “Macro Tree Transducers, Attribute Grammars, and MSO Definable Tree Translations”. In: *Information and Computation* 154.1, pp. 34–91. ISSN: 0890-5401. DOI: 10.1006/inco.1999.2807. URL: <https://www.sciencedirect.com/science/article/pii/S0890540199928079>.
- Engelfriet, J. and H. Vogler (1985). “Macro tree transducers”. In: *Journal of Computer and System Sciences* 31.1, pp. 71–146. ISSN: 0022-0000. DOI: 10.1016/0022-0000(85)90066-2. URL: <https://www.sciencedirect.com/science/article/pii/0022000085900662>.
- Fülöp, Z. (Jan. 1981). “On attributed tree transducers”. In: *Acta Cybern.* 5, pp. 261–279. DOI: 10.1007/978-3-642-72248-6\_5.
- Fülöp, Z. and H. Vogler (1998). *Syntax-Directed Semantics: Formal Models Based on Tree Transducers*. Monographs in Theoretical Computer Science An EATCS Series. Springer, Berlin, Heidelberg. DOI: 10.1007/978-3-642-72248-6.
- Gallot, P.D., A. Lemay, and S. Salvati (2020). “Linear High-Order Deterministic Tree Transducers with Regular Look-Ahead”. In: *45th International Symposium on Mathematical Foundations of Computer Science (MFCS 2020)*. Ed. by Javier Esparza and Daniel Král’. Vol. 170. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 38:1–38:13. ISBN: 978-3-95977-159-7. DOI: 10.4230/LIPIcs.MFCS.2020.38. URL: <https://drops.dagstuhl.de/opus/volltexte/2020/12705>.

- Globerman, N. and D. Harel (1996). "Complexity Results for Two-Way and Multi-Pebble Automata and their Logics". In: *Theoretical Computer Science* 169, pp. 161–184.
- Honkala, J. (2000). "A short solution for the HDTOL sequence equivalence problem". In: *Theoretical Computer Science* 244.1, pp. 267–270. ISSN: 0304-3975. DOI: 10.1016/S0304-3975(00)00158-4. URL: <https://www.sciencedirect.com/science/article/pii/S0304397500001584>.
- Hosoya, H. (2010). "Tree transducers". In: *Foundations of XML Processing: The Tree-Automata Approach*. Cambridge University Press, pp. 127–137. DOI: 10.1017/CB09780511762093.011.
- Maletti, A. (2015). "Extended Tree Transducers in Natural Language Processing". In: *Proceedings of the 12th International Conference on Finite-State Methods and Natural Language Processing 2015 (FSMNLP 2015 Düsseldorf)*. Association for Computational Linguistics. URL: <https://aclanthology.org/W15-4801>.
- Maneth, S. (Dec. 1998). "The Generating Power of Total Deterministic Tree Transducer". In: *Inf. Comput.* 147.2, pp. 111–144. ISSN: 0890-5401. DOI: 10.1006/inco.1998.2736. URL: <https://doi.org/10.1006/inco.1998.2736>.
- Mealy, George H. (1955). "A method for synthesizing sequential circuits". In: *The Bell System Technical Journal* 34.5, pp. 1045–1079. DOI: 10.1002/j.1538-7305.1955.tb03788.x.
- Seidl, H., S. Maneth, and G. Kemper (Apr. 2018). "Equivalence of Deterministic Top-Down Tree-to-String Transducers Is Decidable". In: *J. ACM* 65.4. ISSN: 0004-5411. DOI: 10.1145/3182653. URL: <https://doi.org/10.1145/3182653>.
- Thirunarayan, Krishnaprasad (Jan. 2009). "Attribute Grammars and Their Applications". In: DOI: 10.4018/978-1-60566-026-4.ch046.
- Trakhtenbrot, B. A. (1961). "Finite automata and logic of monadic predicates". In: *Doklady Akademii Nauk SSSR*, 149:326–329.