Monadic Second-order Transductions of words and terms

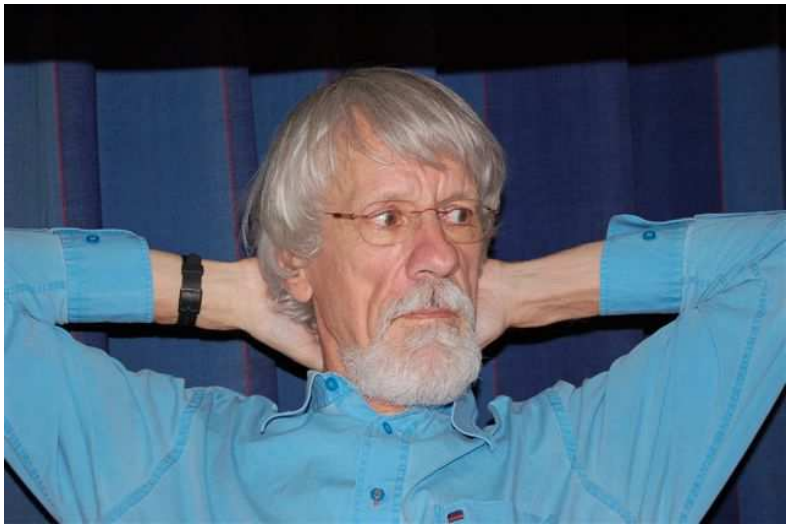
Bruno Courcelle, *LaBRI, Bordeaux*

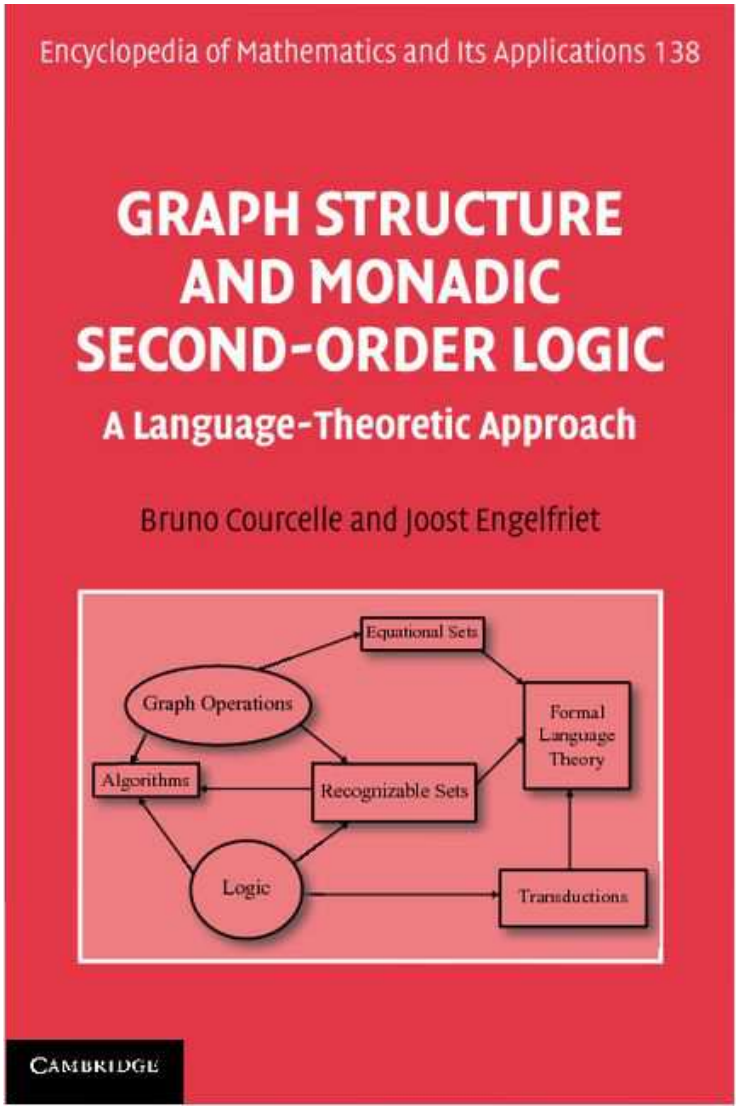
Lecture based on work by

Joost Engelfriet, *LIACS, Leiden*

and many coauthors.

Reference : BC+JE: The Great Book of MSOL and Graphs, *aka*, Graph structure and monadic second-order logic, a language-theoretic approach, *Cambridge University Press*, June 2012, Chapters 7 and 8.




Encyclopedia of Mathematics and Its Applications 138
GRAPH STRUCTURE AND MONADIC SECOND-ORDER LOGIC
A Language-Theoretic Approach
Bruno Courcelle and Joost Engelfriet
Equational Sets
Graph Operations
Formal Language Theory
Algorithms
Recognizable Sets
Logic
Transductions
CAMBRIDGE

Formal Language Theory extends to graphs

1. *Recognizable sets :* an algebraic notion based on finite congruences, well-defined in every algebra.

   They generalize regular languages.

Automata : tools for implementation and theoretical study.

Good for words, terms and trees, *not* for graphs.

Monadic Second-order logic : a specification language for recognizable sets of words, terms, trees and graphs.

MS-definable $\equiv$ Recognizable for words, terms, trees

MS-definable $\subset$ Recognizable for graphs
(two results for two algebras and two MS-definability notions).

2. *Equational sets :* least solutions of systems of recursive set equations, well-defined in every algebra. They generalize context-free languages. For graphs, they have equivalent characterizations by grammars with context-free rewriting rules.

3. *Transductions of structures (words, terms, graphs)* can be specified by :

   automata with outputs (many notions) for words, terms

   Monadic Second-order formulas, for all structures

Main relationships

Recognizable sets of graphs ("generalized regular"),

Monadic second-order definable sets of graphs,

Equational sets of graphs ("generalized context-free") and

Monadic second-order transductions (MST) are related :

$L \cap K \in$ EQUAT if $L \in$ EQUAT and $K \in$ REC

EQUAT = MST(Trees) = MST(EQUAT)

MS-definable $\subset$ REC = MST^{-1}(REC)

$\Rightarrow$ $L \cap K \in$ EQUAT if $L \in$ EQUAT and K is MS-definable.

Monadic Second-order Transductions

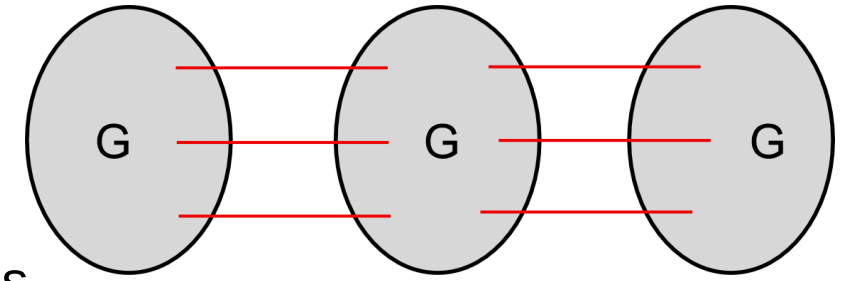
Deterministic (*parameterless*) DetMST:

G : word, term, tree, labelled graph.

G → τ(G) = H

G → $copy_k$(G) → H, specified inside $copy_k$(G) by MS-definable domain restriction and redefinition of edge and labelling relations.

$copy_3$(G)



Red lines relate identical nodes (vertices) in the different copies.

A nondeterministic MST is defined by using an MS-constrained choice ν of auxiliary node (or vertex) labels.

Properties :

1. Linear size increase property : $|\mathrm{Vert}(\tau(G,\nu))| \leq k.|\mathrm{Vert}(G)|$.
2. DetMST and MST are *closed under composition*
   but not under inverse (because of property 1).

These notions extend to relational structures.

We now consider words and terms

Transductions of words and terms have been studied extensively since 1973 by Joost Engelfriet and his coauthors:

Roderick Bloem,

Frank Drewes,

Hendrik Jan Hoogeboom,

Andreas Maletti,

Sebastian Maneth,

Grzegorz Rozenberg,

Vincent van Oostrom,

Heiko Vogler and those I am forgetting.

Words as labelled graphs (directed paths).

Several (essentially equivalent) representations :

Word abbc : * → * → * → * → * (edge labelled graph)

a b b c

or : * → * → * → * (vertex labelled graph)

a b b c

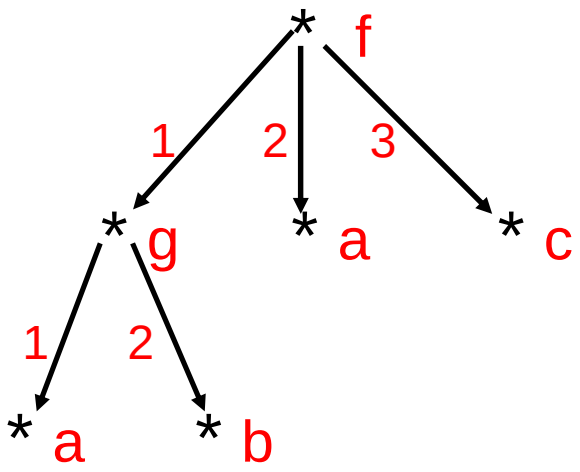
or, with endmarkers : #abbc$ (works for the empty word) :

* → * → * → * → * → * (vertex labelled graph)

a b b c $

Terms as labelled graphs (rooted trees).

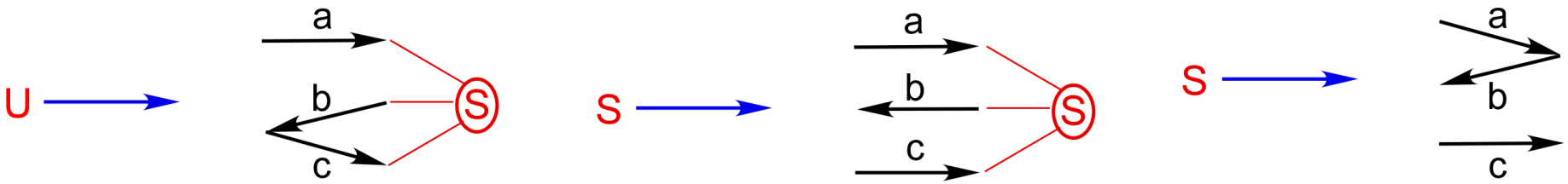
Term f(g(a, b), a, c)



Words (handled as graphs)

Equational sets properly include context-free languages:

$\{a^n b^n c^n \ / \ n \geq 2 \ \}$ is *(linear) equational* but not context-free.



The start symbol is U.

Because equation systems are written with other operations than concatenation. For graphs, these operations yield tree-width and clique-width.

Recognizable sets are the regular languages, as in the free monoid.

Transductions

1. DetMST and MST are incomparable with rational transductions (the *square* mapping u → uu is a DetMST).

2. A rational transduction is an MST ⇔ it has finite images.

3. An MST is a DetMST ⇔ it is a function.

Hence, every DGSM : Deterministic Generalized Sequential Mapping (i.e., DFA with output) is a DetMST.

Theorem (Engelfriet & Hoogeboom, 2001) :

DetMST = 2DGSM : *Deterministic 2-way Generalized Sequential Mappings*

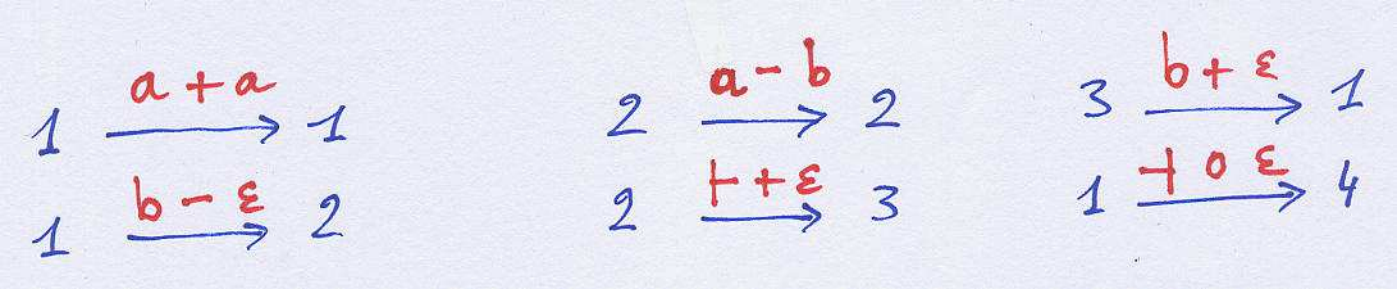
2DGSM recognize regular languages and are closed under composition (Chytil & Jakl, 1977)

Proof : 2DGSM $\subseteq$ DetMST

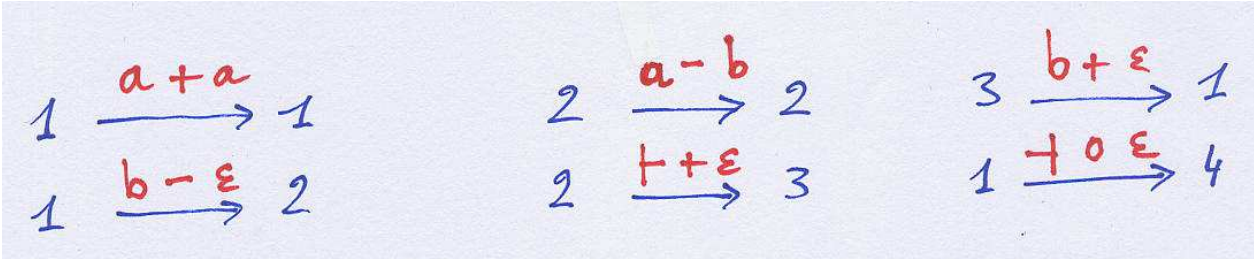
2DGSM $\subseteq$ DetMST. We consider the 2DGSM that transforms

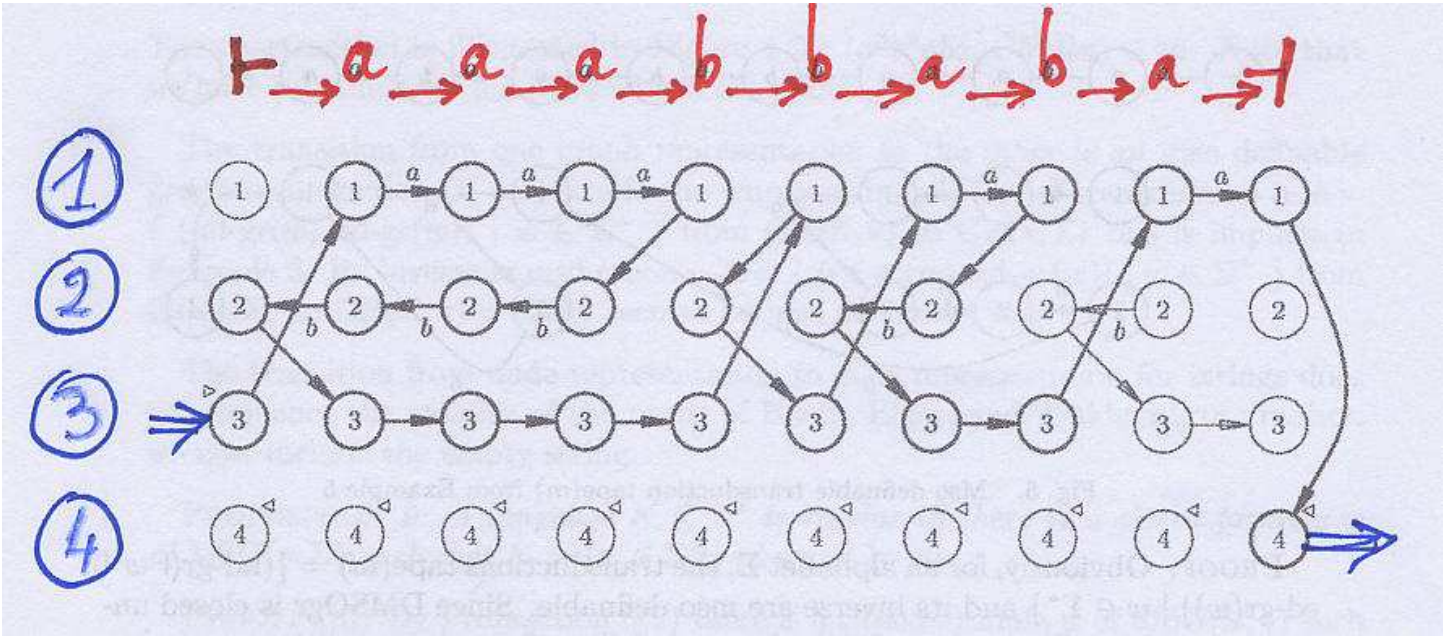
$$a^n b a^m b a^p \ldots \rightarrow a^n b^n a^m b^m a^p \ldots$$

Some rules :



The input word aaabbaba (with end markers in the example below) is transformed into a *computation board* by a first k-copying DetMST (k = number of states):





Another DetMST extracts the *output word* (aaabbbaba) by deleting some parts and contracting the edges with empty output.

We conclude with closure of DetMST under composition.

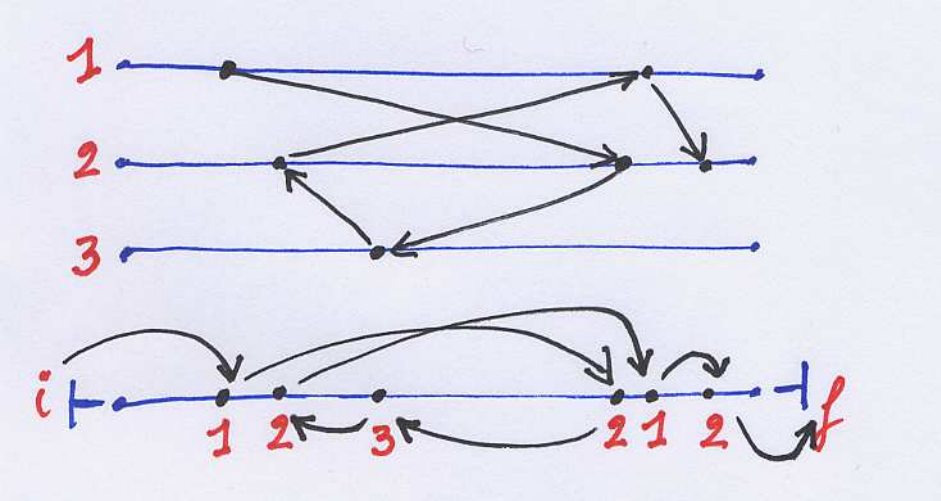
Other direction : DetMST $\subseteq$ 2DGSM^{MS} $\subseteq$ 2DGSM

2DGSM^{MS} : finite state 2-way transducers with global tests and moves (*jumps*) specified by MS formulas $\varphi(u)$ and $\varphi(u_1,u_2)$.

A k-copying DetMST is easily translated into a 2DGSM^{MS} (with k+2 states).

Example : k = 3

i, f new initial and final states.



Last (more technical) step : $2DGSM^{MS} \subseteq 2DGSM$

Basic tool : MS-definability $\equiv$ finite automata

Theories: Fix integer p. For $w \in A^*$, x and y, positions in w :
Th(w ; x,y) = the set of MS formulas $\varphi(u_1,u_2)$ of quantifier-height $\leq p$ such that $w \models \varphi(x,y)$; this set is finite up to (decidable) equivalence.

Similar notions: Th(w ; x) and Th(w) with $\varphi(u_1)$ and φ.

Lemma 1 : Let $w = w_1\, a\, w_2\, b\, w_3$ and x,y be the positions of a,b. Then :

$Th(w ; x,y) = F_{a,b}(Th(w_1), Th(w_2), Th(w_3))$,

$Th(w ; x) = F'_a(Th(w_1), Th(w_2 b w_3))$,

$Th(w) = F''_a(Th(w_1), Th(w_2\, b w_3))$, for functions $F_{a,b}$, F'_a, F''_a.

Lemma 2 : $Th(w_1a) = F''_a(Th(w_1), Th(\varepsilon))$.

Hence, $Th(w_1)$ is computable by a DFA (as a state).

Definition : *Annotated word* : $w \in A^* \rightarrow$ Ann(w).

Each letter a is replaced by $[a, Th(w_1), Th(w_2)]$ if $w = w_1 a w_2$

(The replacement depends on the position).

Lemma 3 : Ann is a 2DGSM.

Proof : By Lemma 2, a first DGSM computes the middle components $Th(w_1)$. Then, a "right to left" DGSM computes the last components $Th(w_2)$ but produces an output that must be reversed by another "right to left" DGSM. Their composition gives a 2DGSM (2-way because of reversals of computation direction).

We now prove : $2DGSM^{MS} \subseteq 2DGSM$

1. Given w, we compute Ann(w) by a 2DGSM, for large enough p.

The computations of the given $2DGSM^{MS}$ will be simulated by a 2DGSM on Ann(w).

2. *MS tests*. Since $Th(w_1 \, a \, w_2 ; x) = F'_a(Th(w_1), Th(w_2))$

(by Lemma 1), the MS "global tests" by formulas φ(u) of quantifier-height ≤ p to be checked in w can be replaced by "local tests" on the "rich" letters $[a, Th(w_1), Th(w_2)]$.

3. *MS jumps* simulated by walks.

Jumps are deterministic. At any position x, an MS test can check if the *unique* position y where to jump is before or after or equal to x.

If *after*, the jump from x to y can be replaced by a forward walk: $x \rightarrow \ldots z \ldots \rightarrow y$ that maintains the information Th(w;x,z): this is possible by Lemma 1 from the annotation and by computing $Th(w_2)$ where w_2 is the subword of w between x and z. Thus we can find the first (and unique) z that satisfies with x the formula $\varphi(u_1,u_2)$ that specifies the jump from x to y.

If *before*, the jump from x to y is replaced by a backward walk.

If *equal*, no move.

Theorem :

(1) 2DGSM(Regular Languages)

= DetMST(Regular Languages)

= MST({0,1}*)

= *Linear* equational sets of words.

(2) This class is closed under 2DGSM, DetMST and MST.

Linear (cf. the $a^n b^n c^n$ language, slide 11) means that every righthand side of a rule of the context-free *graph* grammar has at most one nonterminal.

Terms

We generalize 2DGSM^{MS} into DTWT^{MS}

Deterministic MS Tree-Walking Transducer $\tau : T(F) \rightarrow T(H)$

- global test at node x specified by MS formula $\varphi(u)$,

- jump from node x to y specified by MS formula $\psi(u_1,u_2)$,

$\tau(t) := \tau(t, q_{init}, root_t) \in T(H)$ (for $t \in T(F)$),

$\tau(t, q, x) := \tau(t, q', y)$ if $t \models \varphi(x) \wedge \psi(x,y)$, (unique q', y)

or

$\tau(t, q, x) := h(\tau(t, q', y_1), \tau(t, q'', y_2))$ if

$t \models \varphi(x) \wedge \psi_1(x, y_1) \wedge \psi_2(x, y_2)$, (unique h, q', y_1, q'', y_2).

Special cases:

- no jumps : only *walking steps* up, down-to-*i*-th-son or stay.
- *local tests* : labelled-by-f ?, is-root ?, is-*i*-th-son ?
- *single-use* : $\tau(t, q, x)$ is never called twice in the computation of $\tau(t)$.

Example : The "homomorphism" defined by f(x) → h(x,x), that transforms f(f(a)) into h(h(a,a),h(a,a)) is a *top-down* DTWT that is *not* single-use. It is *not* a DetMST (not of linear size increase).

Proposition 1 : For terms, DetMST = single-use $DTWT^{MS}$.

As for words.

Proposition 2 : Jumps can be replaced by walking steps (keeping global tests).

As for words, with MS tests instead of annotation.

Theorem (Bojanczyk & Colcombet, 2006) : Not every regular set of terms is recognized by a tree-walking automaton.

Hence, the class of single-use $DTWT^{MS}$ is not included in DTWT.

This shows a difference between the cases of words and terms.

Regular sets are recognized by $DTWT^{PD}$: *with a pushdown.*

Proposition 3 : Global tests can be made local by adding a *pushdown.* We have DetMST $\subset$ $DTWT^{PD}$ and loose single-use.

Pushdown : at node x, its length is the distance of x to the root.

It stores information attached to the ancestors y of x.

In the proof, it stores the theories $Th(t\uparrow y)$, where $t\uparrow y$ is the context of y in t, the part of t outside of the subterm t/y.

Each time $Th(t/x)$ is needed, it is recomputed by a depth-first traversal of t/x, in which the pushdown is also used.

We have $Th(t ; x) = F(Th(t\uparrow x), Th(t/x_1), Th(t/x_2))$

where x_1, x_2 are the two sons of x. Cf. Lemma 1 for words.

Theorem :

A transduction of terms is in DetMST

$\Leftrightarrow$ it is in DTWT^{PD} and of linear size increase.

It is decidable whether a DTWT^{PD} is of linear size increase.

Remark : DTWT^{PD} = DMTT (Deterministic Macro Tree Transducer)

(Engelfriet and Vogler, 1986)

Instead of using a pushdown, the input tree can be annotated with all theories Th(t/x) and Th(t$\uparrow$x) by the composition of a bottom-up and a top-down finite automaton (with output). Then a single-use DTWT can be used, which works in linear time.

Corollary : Let τ be a DetMST expressed as a composition of two automata with output and a single-use DTWT. Then τ(t) is computed in *linear time*.

Alternative proof of a result valid for DetMST on graphs of bounded clique-width or tree-width.

Implementation of a DetMST on graphs of bounded tree-width or clique-width by a $DTWT^{PD}$ on terms.

$F_k = \{ \oplus,\ relab_{i \to j},\ add_{i,j},\ *_i \ /\ 1 \leq i, j \leq k \}$

These operations generate the graphs of clique-width $\leq$ k.

val : $T(F_k) \longrightarrow$ Graphs is a DetMST.

Theorem (The Book,2012): If $\tau : T(H) \longrightarrow$ Graphs is a DetMST, then $\tau = val \circ \sigma$ for some DetMST $\sigma : T(H) \longrightarrow T(F_k)$ and some k.

Corollary :

Given a DetMST μ : Graphs $\rightarrow$ Graphs and p, there exist k, and a DetMST σ such that :

$$\begin{array}{ccc} \text{val} \uparrow & T(F_p) & \xrightarrow[\sigma]{} \quad T(F_k) \uparrow \text{val} \\ & & \end{array} .$$

Hence, if a graph G is given by a term $t \in T(F_p)$,

a term $\sigma(t)$ for $\mu(G)$ can be computed by a $DTWT^{PD}$ (and in *linear time* when annotation is used).

Finding t if the clique-width of G has a given upper-bound can be done in time $O(n^3)$. (val is computable in linear time).

Theorem :

A language of terms is equational $\Leftrightarrow$ it is the image of a regular language of terms under a single-use DTWT.

A language of words is equational $\Leftrightarrow$ it is the image of a regular language of terms under a DTWT (necessarily single-use).

It is *linear* equational $\Leftrightarrow$ it is the image of a regular set of words under a 2DGSM.

Conclusion

The equivalence of MS definability and recognizability by deterministic finite automata on words and terms is extended to deterministic MS transductions.

Other results : equivalence problems for transducers:

Decidable for DetMST on words and terms.

Open for DetMST from terms to graphs.

Open for DTWT.

A problem arising in *Computational Linguistics* :

The language of words w over {a, b, c} that have the same numbers of a, b and c is *equational*: it is definable by a Multiple Context-Free Grammar on words, hence, by a Hyper-edge Replacement Graph Grammar (difficult "geometric" proof by Salvati, 2011).

What about the similar language over 4 letters ?

Can we show it is not by techniques based on MST ?