Efficiency Analysis for the Elimination of Intermediate Results in Functional Programs by Compositions of Attributed Tree Transducers

Andreas Maletti

Department of Computer Science

Dresden University of Technology

TECHNISCHE UNIVERSITÄT DRESDEN

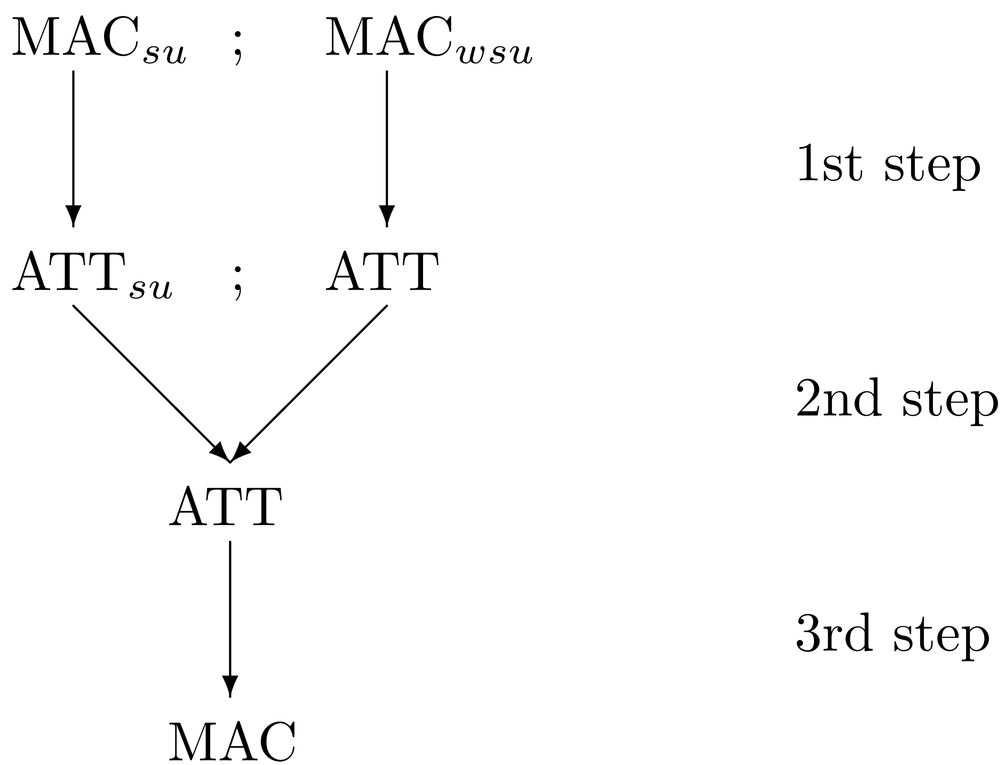
Itinerary

- Motivation and introduction
- Step 1: To Attributed Tree Transducers
- Step 2: Composing Attributed Tree Transducers
- Step 3: Back to Macro Tree Transducers
- Conclusions

Motivation

- Intermediate results are ubiquitous in functional programs.
- Elimination of such results **might** therefore
  - save memory and
  - speed up computation of the final result.
- *Major question:* How can we compose functions symbolically?
- *Minor question:* Can we guarantee a speed-up?

Introduction



Macro Tree Transducers

- Special (restricted) functional programs

Example:

$$M_{\text{pal}} = \left(\{A^{(1)}, B^{(1)}, N^{(0)}\}, \{A^{(1)}, B^{(1)}, N^{(0)}\}, \{s^{(1)}\}, (s\,x_1\,N), R\right)$$

$$\begin{array}{lllll} R = \{ & s\,(A\,x_1)\,y_1 & = & A\left(s\,x_1\,(A\,y_1)\right) & , \\ & s\,(B\,x_1)\,y_1 & = & B\left(s\,x_1\,(B\,y_1)\right) & , \\ & s\,N\,y_1 & = & y_1 & \} \end{array}$$

- Appends the reversed input list to the input list; constructs a palindrome

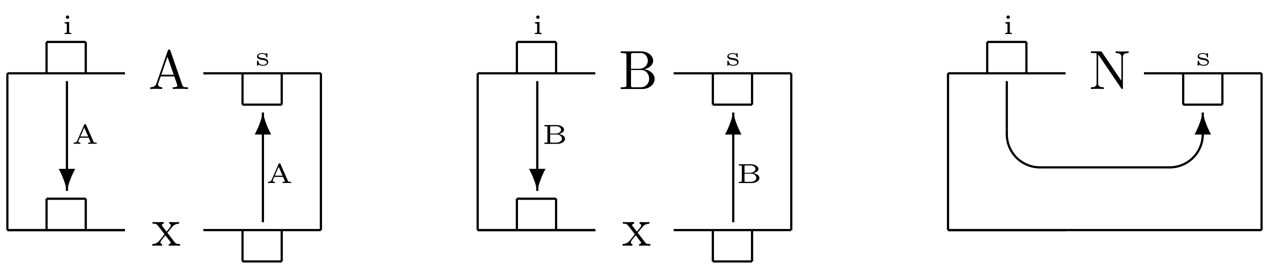
Attributed Tree Transducers

- Special (restricted) attribute grammars

Example:

$$M'_{\mathrm{pal}} = \big(\{A^{(1)}, B^{(1)}, N^{(0)}\}, \{A^{(1)}, B^{(1)}, N^{(0)}\}, \{s\}, \{i\}, \widehat{s}, \widehat{\sigma}, R\big)$$

Depiction of some rules in the rule-set R:



$\mathbf{MAC}_{wsu} \subseteq \mathbf{ATT}$

- Synthesized attributes instead of function symbols
- Simulate context parameters by inherited attributes
- Associate a set of inherited attributes to every synthesized attribute

$\mathbf{MAC}_{wsu} \subseteq$ ATT (cont'd)

- Macro Tree Transducer M,
  Attributed Tree Transducer $M' = C[M]$ ($\tau_{M'} = \tau_M$)
- Established efficiency relation:

$$\text{count}(M) = \text{count}(M') - i - 1$$

- i : number of reduction steps invested to reduce inherited attribute instances

$\Rightarrow$ only count the (non-root) synthesized attribute instances

$\mathbf{MAC}_{wsu} \subseteq$ ATT (Example)

Macro Tree Transducer:

$\underline{s\,(ABN)\,N}$

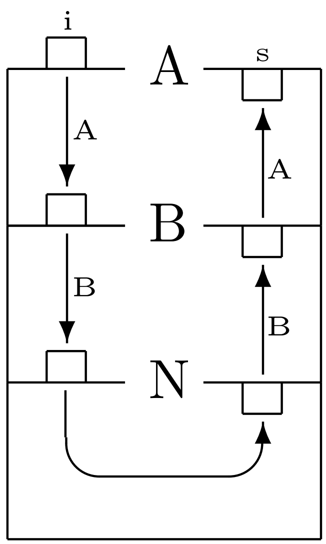
$\Rightarrow A\,(\underline{s\,BN(AN)})$

$\Rightarrow AB\,(\underline{s\,N(BAN)})$

$\Rightarrow ABBAN$

3 reduction steps

Attributed Tree Transducer:



3 synthesized attribute instances

$\mathbf{ATT}_{su}$; ATT $\subseteq$ ATT

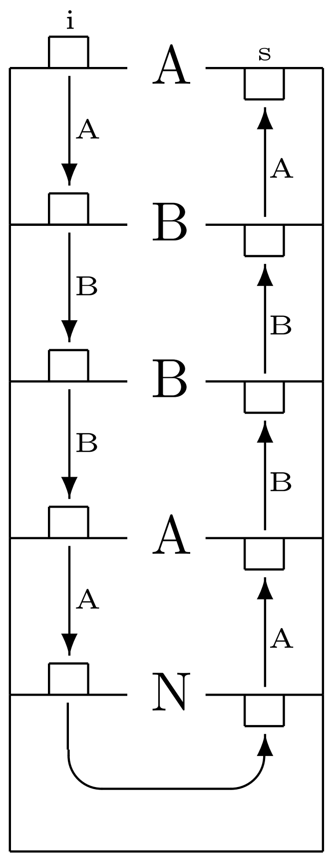
- Core idea of construction: *pairing of attributes*
- Efficiency considerations

M_1, M_2	syntactic single-use atts
F_1, F_2	attribute sets of M_1, M_2
$M = C[M_1, M_2]$	att with $\tau_M = \tau_{M_1} ; \tau_{M_2}$
t_1, t_2	input tree for M_1 and $t_2 = \tau_{M_1}(t_1)$

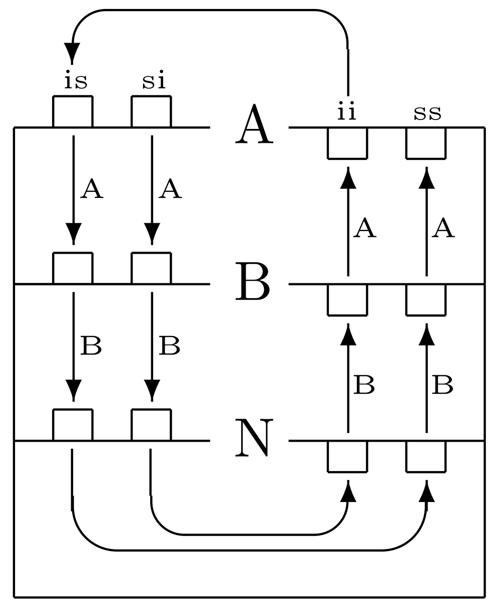
M is more efficient than $(M_1 ; M_2)$, iff

$$\text{size}(t_1)|F_1|\,|F_2| + 1 \;<\; \text{size}(t_1)|F_1| + \text{size}(t_2)|F_2| + 2$$

$\mathbf{ATT}_{su}$; $\mathbf{ATT} \subseteq \mathbf{ATT}$ (Example)



Composition result $(M'_{\mathrm{pal}}; M'_{\mathrm{pal}})$:



$\mathbf{ATT}_{su}$; ATT ⊆ ATT (Example cont'd)

$$\text{size}(t_1)|F_1|\,|F_2| + 1 < \text{size}(t_1)|F_1| + \text{size}(t_2)|F_2| + 2$$

- in our example:

$$3 \cdot 2 \cdot 2 + 1 = 13 \;<\; 18 = 3 \cdot 2 + 5 \cdot 2 + 2$$

- in general for $(M'_{\text{pal}}; M'_{\text{pal}})$:

$$n \cdot 2 \cdot 2 + 1 = 4n + 1 \;<\; 6n = n \cdot 2 + (2n - 1) \cdot 2 + 2$$

$n = \text{size}(t_1) > 0$

ATT $\subset$ MAC

- Function symbols replace synthesized attributes
- Operations on inherited attributes simulated in context parameters
- Every function symbol has as many context parameters as there are inherited attributes

ATT $\subset$ MAC (cont'd)

- single-use Attributed Tree Transducer M',
  Macro Tree Transducer $M = C[M']$ $(\tau_M = \tau_{M'})$
- Established efficiency relation:

$$\text{count}(M) = \text{count}(M') - i - 1$$

- i : number of reduction steps invested to reduce inherited attribute instances

$\Rightarrow$ only count the (non-root) synthesized attribute instances

ATT $\subset$ MAC (Example)

- Running example (let $\Sigma = \{A^{(1)}, B^{(1)}, N^{(0)}\}$):

$$M_{\text{pal;pal}} = \big(\Sigma, \Sigma, \{ss^{(2)}, ii^{(2)}\}, (ss\,x_1\,(ii\,x_1\,N\,N)\,N), R\big)$$

$$
\begin{array}{llcl}
R = \{ & ss\,(A\,x_1)\,y_1\,y_2 & = & A\,\big(ss\,x_1\,(A\,y_1)\,(A\,y_2)\big)\;, \\
& ss\,(B\,x_1)\,y_1\,y_2 & = & B\,\big(ss\,x_1\,(B\,y_1)\,(B\,y_2)\big)\;, \\
& ss\,N\,y_1\,y_2 & = & y_1 \\
& ii\,(A\,x_1)\,y_1\,y_2 & = & A\,\big(ii\,x_1\,(A\,y_1)\,(A\,y_2)\big)\;, \\
& ii\,(B\,x_1)\,y_1\,y_2 & = & B\,\big(ii\,x_1\,(B\,y_1)\,(B\,y_2)\big)\;, \\
& ii\,N\,y_1\,y_2 & = & y_2 \qquad \}
\end{array}
$$

ATT $\subset$ MAC (Example cont'd)

- Reduction:

$$\underline{ss\,(ABN)\,(ii\,(ABN)\,N\,N)\,N}$$
$$\Rightarrow A\,(\underline{ss\,(BN)\,(A\,[ii\,(ABN)\,N\,N])\,(AN)})$$
$$\Rightarrow AB\,(\underline{ss\,N\,(BA\,[ii\,(ABN)\,N\,N])\,(BAN)})$$
$$\Rightarrow ABBA\,[\underline{ii\,(ABN)\,N\,N}]$$
$$\Rightarrow ABBAA\,[\underline{ii\,(BN)\,(AN)\,(AN)}]$$
$$\Rightarrow ABBAAB\,[\underline{ii\,N\,(BAN)\,(BAN)}]$$
$$\Rightarrow ABBAABBAN$$

6 reduction steps

Main theorem

M_1, M_2	syntactic single-use and preserving macs
F_1, F_2	set containing the function symbols of M_1, M_2
$M = C[M_1, M_2]$	mac with $\tau_M = \tau_{M_1}; \tau_{M_2}$
t_1, t_2	input tree for M_1 and $t_2 = \tau_{M_1}(t_1)$

M is more efficient than $(M_1; M_2)$, iff

$$\mathrm{size}(t_1)\Big(|F_1|\,|F_2| + \sum_{f \in F_1} \mathrm{rank}_{F_1}(f) \sum_{f \in F_2} \mathrm{rank}_{F_2}(f)\Big) < \mathrm{size}(t_1)|F_1| + \mathrm{size}(t_2)|F_2|$$

Main theorem (cont'd)

- Running example with $n = \text{size}(t_1)$:

$$n \cdot (1 \cdot 1 + 1 \cdot 1) = 2n \; < \; 3n - 1 = n \cdot 1 + (2n-1) \cdot 1$$

- Some derived results (only additional properties listed):

M_1	M_2	$M = C[M_1, M_2]$ more efficient, if
	tdtt, $\lvert F_2 \rvert = 1$	always
producing		$\lvert F_1 \rvert \; > \; \text{rsum}(F_1) \cdot \text{rsum}(F_2)$

Conclusions

- Composition result seems to suffer heavily from the explosion in the number of attributes.
- From the theorem several small classes can be derived.
- Further studies (especially in connection with further optimization techniques like copy rules elimination) for other composition techniques are necessary.
- Implementations!!