Weighted Automata and Rewriting Lecture 2

Johannes Waldmann (HTWK Leipzig)

ISR Eindhoven 2017

From strings to trees (in general)

- weighted automaton on strings:
  - final weight: $f_A \in 1 \times Q \to S$,
  - transition: t_A maps letter to linear function (matrix),
  - initial weight: $i_A \in Q \times 1 \to S$.
- weighted automaton on trees:
  - final weight (at top of tree): $f_A \in 1 \times Q \to S$,
  - transition: t_A maps k-ary function symbol g to k-ary multilinear function (tensor)

$$[g](v_1, \dots, v_k) = \sum c_{i_1,\dots,i_j} \otimes v_{1,i_1} \otimes \dots \otimes v_{k,i_k}$$

  - initial weights (at leaves): for each 0-ary symbol, $Q \times 1 \to S$
- semantics (automaton maps term t into S)
  - based on *runs*, where run maps position to state
  - *algebraically*: $f_A \otimes [t]$

From strings to trees (simplified)

- general form of tensor (applied to vectors v_i)
  $T(v_1, \dots, v_k) = \sum c_{i_1,\dots,i_j} \cdot v_{1,i_1} \cdot \dots \cdot v_{k,i_k}$
  Ex. $T(\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}, \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix}) = \begin{pmatrix} 2x_1y_1 + x_2y_3 + 5x_3y_3 \\ x_1y_2 + 3x_2y_3 + 2x_3y_1 \\ 4x_2y_1 \end{pmatrix}$
  not substitution closed ($T(x, x)$ is quadratic)
- restrictions: $x_3 = y_3 = 1$, no mixed monomials
  Ex. $T'(\begin{pmatrix} x_1 \\ x_2 \\ 1 \end{pmatrix}, \begin{pmatrix} y_1 \\ y_2 \\ 1 \end{pmatrix}) = \begin{pmatrix} x_2 + 5 \\ 3x_2 + 2y_1 \\ 1 \end{pmatrix}$
- write as affine functions (T_0 vector; $T_1, \dots$ matrices)
  $T(v_1, \dots, v_k) = T_0 + T_1 \cdot v_i + \dots + T_k \cdot v_k$
  $T'(x, y) = \begin{pmatrix} 5 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 & 5 \\ 0 & 3 \end{pmatrix} x + \begin{pmatrix} 0 & 0 \\ 2 & 0 \end{pmatrix} y$

Matrix Interpretations for Term Rewriting

- use affine functions (T_0 vector; $T_1, \dots$ matrices)
  $T(v_1, \dots, v_k) = T_0 + T_1 \cdot v_i + \dots + T_k \cdot v_k$
- interpret ground term by vector, term with k variables as k-ary affine function
- weight of term t = sum of weights of paths in t
- monotonicity: $\forall 1 \le i \le k : (T_k)_{1,1} \ge 1$.
- order: $S > T$ iff $(S_0)_1 > (T_0)_1$ and $\forall 0 \le i \le k : S_i \ge T_i$ (component-wise)
- local compatibility: $\forall (l, r) \in R : [l] > [r]$
- Exercise: $[f](x, y) = \begin{pmatrix} 0 \\ 1 \end{pmatrix} + \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} x + \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} y$
  is compatible with $\{f(f(x, y), z) \to f(x, f(y, z))\}$

Remark on previous example

- $[f](x, y) = \begin{pmatrix} 0 \\ 1 \end{pmatrix} + \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} x + \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} y$
- can evaluate by CAS (maxima)

```
f(x,y) := matrix([0],[1])
  + matrix([1,1],[0,1]) . x
  + matrix([1,0],[0,1]) . y ;
x : matrix([x1],[x2]) ;
y : matrix([y1],[y2]) ;
z : matrix([z1],[z2]) ;
expand ( [f(f(x,y),z), f(x,f(y,z))] ) ;
```

- Exercise: in $[t] = \begin{pmatrix} t_1 \\ t_2 \end{pmatrix}$, what is the meaning of t_1, t_2 (if there is one)?
- Exercise: does the growth match the derivational complexity (asymptotically)?

Remark on Derivational Complexity

- law of associativity, (AVL) right rotation

$$A = \{f(f(x, y), z) \to f(x, f(y, z))\}$$

- let $L[\cdot] = f(a, \cdot)$; $R[\cdot] = f(\cdot, a)$, then $R[L[y]] \to_A L[R[y]]$
- so A can simulate $RL \to LR$, thus dc_A is at least quadratic
- growth of $[f](x, y) = \begin{pmatrix} 0 \\ 1 \end{pmatrix} + \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix} x + \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} y$
  is $O(n) \cdot \text{growth}\{\begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}\} \in O(n^2)$

Automata with Arctic Weights

- the *arctic* semiring $\mathbb{A} = (\{-\infty\} \cup \mathbb{N}, -\infty, \max, 0, +)$
  this is the opposite of the $(\min, +)$ semiring, named *tropical* in honour of Imre Simon, who lived in Sao Paulo, Brasil
- essential difference to $\mathbb{N}$: monotonicity of $\oplus$
  in $\mathbb{N}$: $x_1 < x_2 \Rightarrow x_1 + y < x_2 + y$
  in $\mathbb{A}$: $x_1 < x_2 \not\Rightarrow x_1 \oplus y < x_2 \oplus y$
  $x_1 < x_2 \wedge y_1 < y_2 \Rightarrow x_1 \oplus y_1 < x_2 \oplus y_2$
- a closed and monotone set of matrices $(M, >)$
  $M = \{A \mid A_{1,1} \ne -\infty\}$
  $A > B \iff \forall i, j : A_{i,j} \gg B_{i,j}$
  where $a \gg b \iff (a > b) \vee (a = -\infty = b)$

Arctic Automata: Examples

- $M = \{A \mid A_{1,1} \ne -\infty\}$
  $A > B \iff \forall i, j : A_{i,j} \gg B_{i,j}$
  where $a \gg b \iff (a > b) \vee (a = -\infty = b)$
- Exercise: check that this is compatible with $\{aa \to aba\}$:

$$t(a) = \begin{pmatrix} 0 & 0 \\ 1 & 1 \end{pmatrix}, t(b) = \begin{pmatrix} 0 & -\infty \\ -\infty & -\infty \end{pmatrix},$$

  compute and compare $t(aa), t(aba)$
- compatible with $\{ab \to ba\}$?

Arctic Automata: Growth

- compatible with $R = \{ab \to ba\}$?
  impossible, since dc_R is quadratic, but...
- Thm: $\forall$ arctic automaton A: growth(A) is linear
- Proof: arctic multiplication = standard addition,
  $\|m_1 \otimes \ldots \otimes m_k\| \le k \cdot \max_i \|m_i\|$
- comments:
  - restricts the power of this termination proof method
  - gives a stronger statement about dc
- *research problem*:
  are there well-founded semirings S with quadratic (or other polynomial) growth (of matrices)?

Arctic Termination for Terms

- arctic affine functions
  $T(v_1, \ldots, v_k) = T_0 \oplus T_1 \otimes v_i \oplus \ldots \oplus T_k \otimes v_k$
  are *not (strictly) monotonic*
- arctic automata "do not work" (for termination) for (≥ 2)-ary symbols.
  they work for unary symbols with T_0 = zero vector
- The *dependency pairs* (DP) transformation
  reduces a termination problem SN(R)
  to a relative top termination problem $\mathrm{SN}(\mathrm{DP}(R)_{\mathrm{top}}/R)$
- for that, arctic affine functions are fine
  - top rewriting $\Rightarrow$ no top context $\Rightarrow$ strict monotonicity not needed
  - relative termination $\Rightarrow$ weak monotonicity is enough

Arctic Top Termination (Example)

- z086 $R = \{a^2 \to bc, b^2 \to ac, c^2 \to ab\}$
- $\mathrm{DP}(R) = \{Aa \to Bc, Aa \to C, Bb \to Ac, Bb \to C, Cc \to Ab, Cc \to B\}$
- remove some rules by counting symbols,
  remaining: $\mathrm{DP}(R)' = \{Aa \to Bc, Bb \to Ac\}$
- $[a] = \begin{pmatrix} 0 & 3 \\ 2 & 1 \end{pmatrix}, [b] = \begin{pmatrix} 3 & 2 \\ 1 & -\infty \end{pmatrix}, [c] = \begin{pmatrix} 0 & 1 \\ 2 & 3 \end{pmatrix}$
  $[A] = [B] = \begin{pmatrix} 0 & -\infty \end{pmatrix}$
- $[Bb] = \begin{pmatrix} 3 & 2 \end{pmatrix} \circleddash [Ac] = \begin{pmatrix} 0 & 1 \end{pmatrix}$
  other rules weakly decreasing
- this works very well (e.g., in termination competitions), also with refinements of DP

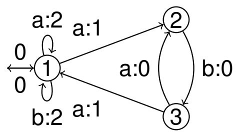
Arctic Top Termination — Remark

- how to find coefficients for arctic matrices?
- constraint system in SMT logic `QF_LIA`
  (linear integer arithmetic = boolean combination of inequalities between linear functions of unknowns)
- Corollary: it is decidable whether finite R has compatible arctic automaton with given size (number of states).
- *challenge problem*: is it also without the size?
  perhaps with a bound on the weights?
- in practice, often use `QF_BV` (bitvectors),
  since we have *a lot* of boolean unknowns
  (one for each $\oplus$, which is max)
  this is not complete (because we fix a bit width in advance)

Fuzzy Weights (Match Bounded Rewriting)

- the fuzzy semiring
  $\mathbb{F} = (\{-\infty, +\infty\} \cup \mathbb{N}, +\infty, \min, -\infty, \max)$
- for any finite M, the set M^* (all products) is finite
  $\Rightarrow$ no $\mathbb{F}$ automaton computes a measure function for termination of rewriting
- but we can transform to a different semiring
  (only used in paper proofs, actual computation is in $\mathbb{F}$)
- historically, this was the first instance (2003) of the matrix termination methods
- the actual motivation was preservation of regularity of languages under rewriting
  (with termination only a side effect)

Fuzzy Weights (Match Bounded Rewriting)

- 

  $[a] = \begin{pmatrix} 2 & 1 & +\infty \\ +\infty & +\infty & +\infty \\ 1 & 0 & +\infty \end{pmatrix}$
  $[b] = \begin{pmatrix} 2 & +\infty & +\infty \\ +\infty & +\infty & 0 \\ +\infty & +\infty & +\infty \end{pmatrix}$

  Exercise: compute $[aa], [aba]$
- embed into semiring $M(\mathbb{F})$
  - domain: $\top \cup$ Multisets over $\mathbb{N}$
  - addition: $\min_{\gg}$ w.r.t. multiset extension $\gg$ of $>$ on $\mathbb{N}$
  - multiplication: multiset union
- $\circleddash_{M(\mathbb{F})}$ is monotone, $\circleddash_{\mathbb{F}}$ implies $\circleddash_{M(\mathbb{F})}$

Decomposition of Match-Bounded Rewriting

- instead of $[l] \circleddash_{\mathbb{F}} [r]$ consider:
  $\mathrm{match}_c(R) :=$ all $(l, r) \in (\Sigma \times \{0, 1, \ldots, c\})^{*2}$
  with max height $l >$ max height $r \wedge$ (base l, base r) $\in R$
  Ex. $(a_2 a_1, a_1 b_0 a_0) \in \mathrm{match}_2\{(aa, aba)\}$
- split rules, using formal left and right inverses:

$$C = \{a_2 \to a_1 b_0 a_0 \overrightarrow{a_1}, \ldots\}, E = \{\overrightarrow{a_1} a_1 \to \epsilon, \ldots\}$$

  $\mathrm{match}_c(R)^* = (C \cup E)^* \cap$ original alphabet
- re-order derivations
  $\mathrm{match}_c(R)^* = (C^* \circ E^*) \cap$ original alphabet
- $\mathrm{match}_c(R)^*$ preserves REG
  (C terminates (!), C^* is substitution, E is inverse monadic)
- R match-bounded (Def: ...) $\Rightarrow R^*$ preserves REG

Constructing Compatible Automata

- that is, w.r.t. local compatibility $A(p, l, q) > A(p, r, q)$
- comes in two flavours: if semiring zero is ...
  - high: uncovered redex $\Rightarrow$ add reduct path
  - low: uncovered reduct $\Rightarrow$ add redex path
- for weights from $\mathbb{F}$, completion actually works:
  - compute closure w.r.t. $(C \cup E)^*$
  - if R is match-bounded, then this stops
- does R have compatible $\mathbb{F}$-weighted automaton with...
  - number of states $\le S$, no bound on weights: decidable
  - weights $\le W$, no bound on states: decidable
  - *challenge problem*: neither bound: decidable?
- *challenge*: give a completion algorithm for $\mathbb{N}$, $\mathbb{A}$
  Example (Dieter Hofbauer): $a^2 b^2 \to b^3 a^3$ over $\mathbb{N}$