First-order tree functions

October 31, 2018

1 Ranked sets and operations on them

Ranked sets. Fix a set of variables V.

Definition 1.1 (Ranked set) *A* ranked set *consists of a set, called the* underlying set*, together with a mapping which assigns to each element of the underlying set an* arity*, which is a finite subset of the variables* V*.*

When talking about elements of a ranked set, we mean elements of the underlying set. For a ranked set A and a finite set of variables $X \subseteq \mathsf{V}$, we write $(A)_X$ for the elements of A that have arity X.

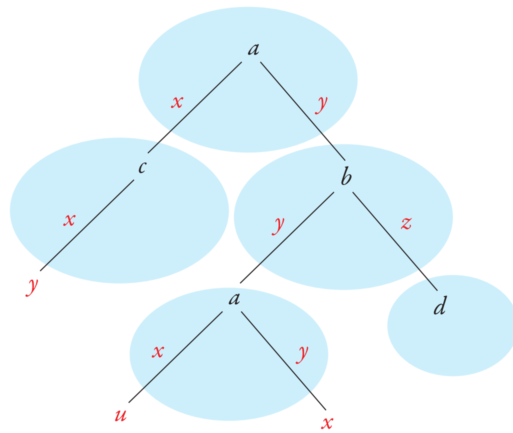
Type constructors. Consider the following operations, which are used to define new ranked sets based on existing ones.

1. **Disjoint union** $A + B$**.** An element of $A + B$ is either an element of A or an element of B, with the arities inherited from A, B.
2. **Cartesian[1] product** $A \times B$**.** An arity X element of $A \otimes B$ is a pair (a, b) such $a \in A$ and $b \in B$ are elements whose arities are both X.
3. **Tensor product** $A \otimes B$**.** An arity X element of $A \otimes B$ is a pair (a, b) such $a \in A$ and $b \in B$ are elements whose arities form a partition of X.
4. **Trees.** For a ranked set A, define a *term* over A of arity X to be a tree, where nodes are labelled by A or X, and edges are labelled by variables, such that:
   - labels from X appear only in the leaves, and each label from X appears exactly once;
   - if a node has a label $a \in A$ of arity Y, then outgoing edges of the node are labelled by variables from Y in one-to-one fashion.

   Define $\mathsf{T}A$ to be the ranked set of terms over A. Here is a picture:

[1] This is the Cartesian product in the category where objects are ranked sets, and morphisms are arity preserving functions.

A term in TA of rank $\{y, u, x\}$



An alternative definition of $\mathsf{T}\sigma$ is that it is the least set which satisfies the following recursion

$$\mathsf{T}\sigma = \mathsf{V} + \coprod_{a\in\sigma} \underbrace{\mathsf{T}\sigma \otimes \cdots \otimes \mathsf{T}\sigma}_{\text{arity of } a \text{ times}}$$

2 First-order tree functions

Definition 2.1 (Types) *The* atomic types *are:*

- *every ranked set with finitely many elements;*
- *a ranked set, call it* 1*, which has one element on every arity;*

A type is any ranked set obtained from atomic types and applying the constructors $+$, $\times$, $\otimes$ *and* T.

Definition 2.2 (Atomic functions) *Let* τ, σ, τ_0 *and* τ_1 *be ranked sets. The following functions are called* atomic functions.

1. *The unique function* $! : \sigma \to 1$
2. *Every arity-preserving function with finite domain*

$$f : \tau \to \sigma$$

3. *Projection and co-projection*

$$\pi_i : \tau_0 \times \tau_1 \to \tau_i \qquad \iota_i : \tau_i \to \tau_0 + \tau_1$$

4. *Distribute*

$$(\tau_0 + \tau_1) \times \sigma \to (\tau_0 \times \sigma) + (\tau_1 \times \sigma)$$

5. *Tree construction: for every σ and every $a \in \sigma$ a function*

$$\underbrace{\mathsf{T}\sigma \otimes \cdots \otimes \mathsf{T}\sigma}_{\textit{arity of } a \textit{ times}} \to \mathsf{T}\sigma$$

6. *Tree deconstruction: for every σ and every $a \in \sigma$ a function*

$$\mathsf{T}\sigma \to 1 + \underbrace{\mathsf{T}\sigma \otimes \cdots \otimes \mathsf{T}\sigma}_{\textit{arity of } a \textit{ times}}$$

7. *The port-order function*

$$\mathsf{T}(\sigma + \tau) \to \mathsf{T}(\sigma + \tau)$$

8. *The block function*

$$\mathsf{T}(\sigma + \tau) \to \mathsf{T}(\mathsf{T}\sigma + \mathsf{T}\tau)$$

9. *For every variables $x, y \in \mathsf{V}$ (mabye just $x = 1$ and $y = 2$) the function*

$$swap : \mathsf{T}\tau \to \mathsf{T}\tau$$

10. *The yield function*

$$\mathsf{T}\sigma \to \mathsf{T}1$$

11. *Some kind of swapping (maybe not needed)*
12. *Let $\circ$ be a ranked set with one element of arity $\emptyset$.*

$$\mathsf{T}(\sigma + \tau) \to \mathsf{T}(\sigma + \tau + \circ) \otimes \mathsf{T}(\sigma + \tau + \circ)$$

Definition 2.3 (Combinators)

1. *Function composition*

$$\frac{f : \tau \to \sigma \quad g : \sigma \to \theta}{g \circ f : \tau \to \theta}$$

2. *Lifting functions to trees*

$$\frac{f : \tau \to \sigma}{\mathsf{T}f : \mathsf{T}\tau \to \mathsf{T}\sigma}$$

3. *Cases*

$$\frac{f_0 : \tau_0 \to \sigma \quad f_1 : \tau_1 \to \sigma}{[f_0, f_1] : \tau_0 + \tau_1 \to \sigma}$$

4. *Pairing functions*

$$\frac{f_0 : \tau \to \sigma_0 \quad f_1 : \tau \to \sigma_1}{(f_0, f_1) : \tau \to \sigma_0 \times \sigma_1}$$

5. *Tensor product of functions*

$$\frac{f_0 : \tau_0 \to \sigma_0 \quad f_1 : \tau_1 \to \sigma_1}{\langle f_0, f_1 \rangle : \tau_0 \otimes \tau_1 \to \sigma_0 \otimes \sigma_1}$$

Definition 2.4 (First-order tree functions) *The class of* first-order tree functions *is the smallest class of functions which contains the atomic functions from Definition 2.2 and is closed under the combinators from Definition 2.3.*

We are now ready to state the main result of this paper.

Theorem 2.5 *The following classes of functions are equal*

- *First-order tree-to-tree transductions;*
- *Restrictions to arity $\emptyset$ of first-order tree functions.*

3 First-order rational functions

Definition 3.1 (Fo-rational functions)

1. Characteristic functions. *Let $\varphi(x)$ be a formula of first-order logic, over vocabulary σ. Define the* characteristic function *of $\varphi(x)$ to be the function*

$$f : \mathsf{T}_\emptyset \sigma \to \mathsf{T}_\emptyset(2 \otimes \sigma)$$

   which adds 0 or 1 to the label of each node depending on whether the node satisfies $\varphi(x)$.

2. Fo-rational. *An* fo-rational *function is any finite composition of charactersitic functions as defined in the previous item, and tree-to-tree homomorphisms.*

Definition 3.2 (2-ctl) *Define 2-ctl to be the least set of unary queries which contains queries of the form "node x has label a", and which is closed under the following connectives*

1. Boolean. *Boolean combinations of unary queries, including negation;*

2. Next. *If φ is a unary query and $i \in \mathbb{N}$, then $\mathsf{X}_i\varphi$ is also a unary query, which is true in nodes whose i-th child satisfies φ.*

3. Until. *If φ, ψ are unary queries, then $\varphi \mathsf{U} \psi$ is also a unary query, which is true in a node x if there exists some $y > x$ such that: (a) ψ is true in y; (b) φ is true in all nodes z with $x < z < y$;*

4. Since. *If φ, ψ are unary queries, then $\varphi \mathsf{S} \psi$ is also a unary query, which is true in a node x if there exists some $y < x$ such that: (a) ψ is true in y; (b) φ is true in all nodes z with $y < z < x$.*

Theorem 3.3 *[1, Theorem 2.6] The unary queries in 2-ctl are exactly the unary queries that are definable in first-order logic.*

Lemma 3.4 *Let τ, σ be finite ranked sets. Then for every fo-rational*

$$f : \mathsf{T}_{\emptyset}\tau \to \mathsf{T}_{\emptyset}\sigma$$

there is a function in the class which agrees with f on elements of arity $\emptyset$.

Proof
Using the 2-ctl theorem. □

Lemma 3.5 *The unfold function for*

References

[1] Bernd-Holger Schlingloff. Expressive completeness of temporal logic of trees. *Journal of Applied Non-Classical Logics*, 2(2):157–180, 1992.