CS 228M: Logic in CS

Domain Transformations

Ashwin Abraham

11th October, 2023

Given an FO formula $\eta(x)$ with one free variable x, for any word w, define w_η as the word obtained by only retaining the alphabets in w at positions i satisfying η (ie i is such that w satisfies $\eta(x)$ under the assignment $x \to i$), and deleting the rest. For a language L, we define $L_\eta = \{w : w_\eta \in L\}$[1]. This transformation of a language is known as a domain transformation to the domain of η. We show that if a language L is FO definable, then the transformed language L_η is also FO definable, and we construct an FO formula defining L_η from the FO formula of L.
Recall that the atomic formulae of $FO\,[<]$ are $x = y$, $x < y$ and $Q_a(x)$ where x and y are variables and a is an alphabet. If φ and ψ are FO formulae, then so are $\varphi \wedge \psi$, $\varphi \vee \psi$, $\varphi \implies \psi$, $\neg\varphi$ and $\neg\psi$. If φ contains a free variable x (we write $\varphi(x)$), then $\forall x\ \varphi(x)$ and $\exists x\ \varphi(x)$ are also FO formulae. Note that $x \neq y$ and $S(x, y)$ are shorthands for $\neg(x = y)$ and $x < y \wedge \forall z(x < z \implies (y = z \vee y < z))$ respectively. For each FO formula φ there is a unique sequence of production rules we can apply to generate φ from the atomic formulae. Recall that for any FO sentence (ie formula with no free variables) φ, we define $L(\varphi)$ to be the set of words w satisfying φ.
For any formulae φ and η, define φ_η such that:

- For atomic formulae φ, $\varphi_\eta = \varphi$
- $(\varphi \circ \psi)_\eta = \varphi_\eta \circ \psi_\eta$ for any binary connective $\circ \in \{\wedge, \vee, \implies\}$
- $(\neg\varphi)_\eta = \neg\varphi_\eta$
- $(\forall x\ \varphi(x))_\eta = \forall x(\eta(x) \implies \varphi_\eta(x))$
- $(\exists x\ \varphi(x))_\eta = \exists x(\eta(x) \wedge \varphi_\eta(x))$

Taking the η-transform of an FO formula φ to get φ_η corresponds to modifying the $\forall$ and $\exists$ steps in the sequence of steps generating φ by adding an implication and a conjunction respectively with η, as described above.

Theorem 1. *For any FO sentence φ, $L(\varphi)_\eta = L(\varphi_\eta)$, ie for any word w, $w \vDash \varphi_\eta \iff w_\eta \vDash \varphi$*

Proof. The intuition behind by changing $\forall x\ \varphi(x)$ to $\forall x(\eta(x) \implies \varphi_\eta(x))$ and $\exists x\ \varphi(x)$ to $\exists x(\eta(x) \wedge \varphi_\eta(x))$ we restrict the domain of these variables to that of η and since φ is has no free variables, the domain of every variable is restricted this way.
We can prove the theorem formally by structural induction on φ. Let $\mathcal{D}_\eta(w)$ be the set of positions in w that satisfy η. We strengthen the inductive hypothesis to claim that for any FO formula φ and assignment α of variables to elements of $\mathcal{D}_\eta(w)$, $w \vDash_\alpha \varphi_\eta \iff w_\eta \vDash_{\alpha'} \varphi$. Here α' is an assignment on w_η such that $\alpha'(x)$ is equal to the number of elements in $\mathcal{D}_\eta(w)$ less than $\alpha(x)$, for any variable x.
Let us consider the case where $w_\eta = \varepsilon$, ie $\mathcal{D}_\eta(w)$ is empty separately, as here there are no assignments of variables to elements of $\mathcal{D}_\eta(w)$. Our inductive hypothesis therefore becomes vacuously true. Note that in this case, if φ is a sentence, then $\varepsilon \vDash \varphi$ if and only if assigning all existentially qualified clauses false and all universally qualified true causes the whole sentence to be true. Note that the same condition arises for $w \vDash \varphi_\eta$ to be true, as we have $\forall x\ \neg\eta(x)$, since $\mathcal{D}_\eta(w) = \varnothing$. Therefore, when $w_\eta = \varepsilon$, for any sentence φ, we have $w \vDash \varphi_\eta \iff w_\eta \vDash \varphi$.
When $w_\eta \neq \varepsilon$, ie $\mathcal{D}_\eta(w)$ is not empty, then our inductive hypothesis reduces to $w \vDash \varphi_\eta \iff w_\eta \vDash \varphi$ when φ is a sentence. This is because, for a sentence, satisfaction is independent of the assignment to the variables. Henceforth in our proof, we will assume $\mathcal{D}_\eta(w)$ is non-empty.
The base cases of our structural induction are when φ is an atomic formula such as $x < y$, $x = y$, or $Q_b(x)$ for some alphabet b. In these cases $\varphi_\eta = \varphi$ and it is easy to see why our claim holds. Assuming our claim holds for φ and ψ, it is again easy to see why it holds for $\neg\varphi$ and $\varphi \circ \psi$.
Now, assuming our claim holds for $\varphi(x)$ (where x is a free variable), let us show that it holds $\forall x\ \varphi(x)$ as well. In this case, $(\forall x\ \varphi(x))_\eta = (\forall x(\eta(x) \implies \varphi_\eta(x)))$. If $w_\eta \vDash_{\alpha'} \forall x\ \varphi(x)$, then this means for every $0 \leq i < |\mathcal{D}_\eta(w)|$,

[1] If I had instead defined this as $\{w_\eta : w \in L\}$, would this still always be FO definable?

$w_\eta \vDash_{\alpha'[x\to i]} \varphi(x)$. By the inductive hypothesis, this occurs if and only if $w \vDash_{\alpha_{[x\to j]}} \varphi_\eta(x)$ for every j satisfying η, which occurs if and only if $w \vDash_{\alpha[x\to j]} (\eta(x) \implies \varphi_\eta(x))$ for every j, ie $w \vDash_\alpha \forall x(\eta(x) \implies \varphi_\eta(x))$, ie the claim holds for $\forall x\ \varphi(x)$.
Now let us show that it holds for $\exists x\ \varphi(x)$ too. If $w_\eta \vDash_{\alpha'} \exists x\ \varphi(x)$, then this means there is some $0 \leq i < |\mathcal{D}_\eta(w)|$ such that $w_\eta \vDash_{\alpha'_{[x\to i]}} \varphi(x)$. By the inductive hypothesis, this occurs if and only if $w \vDash_{\alpha_{[x\to j]}} \varphi_\eta(x)$ for some j satisfying η, which itself occurs if and only if there is some j such that $w \vDash_{\alpha_{[x\to j]}} \eta(x) \wedge \varphi_\eta(x)$, ie $w \vDash \exists x(\eta(x) \wedge \varphi_\eta(x))$, ie the claim holds for $\exists x\ \varphi(x)$ too, which completes our induction. $\square$

The concept of domain transformations often proves to be quite useful. For example, we can prove that the concatenation of two FO definable languages $L_1 = L(\varphi)$ and $L_2 = L(\psi)$ is also FO definable. If a word $w \in L(\varphi)L(\psi)$, then there must be some p such that the first p letters of w satisfy φ and the remaining letters satisfy ψ. We will therefore construct two domains, $\{x : x < p\}$ and $\{x : \neg(x < p)\}$. The FO formula defining $L(\varphi)L(\psi)$ then becomes $\exists p(\varphi_{<p} \wedge \psi_{\geq p})$, where $\varphi_{<p} = \varphi_{\{x:x<p\}}$ and so on. This is actually incorrect, as this excludes the possibility that the entire string w is from L_1 and the empty string lies in L_2, so to fix that, we need to modify the equation to be

$$\exists p(\varphi_{<p} \wedge \psi_{\geq p}) \vee \exists p(\varphi_{\leq p} \wedge \psi_{>p})$$

Another use of this concept is in showing that $\texttt{shuffle}(L_1, L_2)$ is FO definable if L_1 is a finite language and L_2 is an FO definable language. Recall that for words $w_1, w_2 \in \Sigma^*$, $\texttt{shuffle}(w_1, w_2)$ was defined as the set $\{x_1y_1 \ldots x_ky_k : x_1 \ldots x_k = w_1, y_1 \ldots y_k = w_2\}$. Note that x_i and y_i here are subwords of w_1 and w_2, not necessarily alphabets. For languages L_1, L_2, $\texttt{shuffle}(L_1, L_2)$ is defined as $\bigcup\limits_{\substack{u\in L_1\\ v\in L_2}} \texttt{shuffle}(u, v)$.
If L_1 is a finite language, say $\{w_1, \ldots w_n\}$, then clearly $\texttt{shuffle}(L_1, L_2) = \bigcup\limits_{1\leq i\leq n} \texttt{shuffle}(w_i, L_2)$. Since the union is finite, showing that $\texttt{shuffle}(w, L)$ is FO definable, when w is a word and L is an FO definable language is sufficient. Let $w = a_1 \ldots a_k$, where a_i are the alphabets. If a word $v \in \texttt{shuffle}(w, L)$, then there must exist positions p_1 to p_k in v corresponding to the insertion of a_1 to a_k, on removal of which we get a word in L. If $L = L(\varphi)$, our formula therefore becomes

$$\exists p_1, \ldots p_k \left[\bigwedge_{1\leq i<k} p_i < p_{i+1} \wedge \bigwedge_{1\leq i\leq k} Q_{a_i}(p_i) \wedge \varphi_\eta\right]$$

where $\eta = \{x : \bigwedge\limits_{1\leq i\leq k} x \neq p_k\}$.