Büchi's logical characterisation of regular languages

Deepak D'Souza
Dept. of Computer Science & Automation
Indian Institute of Science, Bangalore 560012, India.
email: {deepakd}@csa.iisc.ernet.in

October 14, 2006

1 Monadic Second Order Logic

Let Σ be a finite alphabet. The formulas of MSO(Σ) are given by the syntax

$$\varphi ::= Q_a(x) \mid x < y \mid x \in X \mid \neg\varphi \mid (\varphi \vee \varphi) \mid \exists x\varphi \mid \exists X\varphi.$$

Here x ranges over a countable set of first-order variables and X ranges over a countable set of second-order variables. We denote the collection of these two types of variables by Var. For each $a \in \Sigma$, Q_a is a monadic predicate (i.e. a predicate which takes one argument) symbol.

The logic is interpreted over a word w in Σ^* in the following way. Individual variables range over the set of *positions* in w, $pos(w) = \{0, 1, \ldots, |w| - 1\}$. The set variables range over sets of positions in w. The relation '$<$' is the usual ordering of natural numbers. And for each $a \in \Sigma$, Q_a is true precisely at the positions in w which contain the symbol a.

Formally, an interpretation for the logic will be a pair $(w, \mathbb{I})$ where $w \in \Sigma^*$ and $\mathbb{I}$ is an assignment of individual variables to a position in w, and set variables to a set of positions of w. Thus, $\mathbb{I}$ is a map $\mathbb{I} : Var \to pos(w) \cup 2^{pos(w)}$.

Let $\mathbb{I}$ be an assignment for variables with respect to σ. Let $i \in \mathbb{N}$. We will use the notation $\mathbb{I}[i/x]$ to denote the assignment which maps x to i and agrees with $\mathbb{I}$ on all other individual and set variables. Similarly, for a subset S of $\mathbb{N}$, the notation $\mathbb{I}[S/X]$ will denote the interpretation which sends X to S, and agrees with $\mathbb{I}$ on all other variables.

The satisfaction relation $w, \mathbb{I} \models \varphi$ is given below.

$$\begin{array}{lll}
w, \mathbb{I} \models Q_a(x) & \text{iff} & w(\mathbb{I}(x)) = a \\
w, \mathbb{I} \models x < y & \text{iff} & \mathbb{I}(x) < \mathbb{I}(y) \\
w, \mathbb{I} \models x \in X & \text{iff} & \mathbb{I}(x) \in \mathbb{I}(X) \\
w, \mathbb{I} \models w, \mathbb{I} \models \neg\varphi & \text{iff} & w, \mathbb{I} \not\models \varphi \\
w, \mathbb{I} \models \varphi \vee \varphi' & \text{iff} & w, \mathbb{I} \models \varphi \text{ or } w, \mathbb{I} \models \varphi' \\
w, \mathbb{I} \models \exists x \varphi & \text{iff} & \text{there exists } i \in pos(w) \text{ such that } w, \mathbb{I}[i/x] \models \varphi \\
w, \mathbb{I} \models \exists X \varphi & \text{iff} & \text{there exists } S \subseteq pos(w) \text{ such that } w, \mathbb{I}[S/X] \models \varphi
\end{array}$$

We use the standard abbreviations of $\varphi \wedge \psi = \neg(\neg\varphi \wedge \neg\psi)$, $\varphi \Rightarrow \psi = \neg\varphi \vee \psi$, $\forall x \varphi = \neg\exists x \neg\varphi$, etc.

An occurrence of a variable is "free" in a formula if the occurrence is not bound by a quantifier. For example in the formula $\forall x \exists y (x < y \wedge Q_a(z))$, x and y occur bound while z occurs free. We call a formula with no free occurrences of variables a *sentence*. For a sentence, the interpretation of variables plays no role, and hence we can speak of a string satisfying a sentence or not. Each sentence φ in $\mathrm{MSO}(\Sigma)$ thus defines a language of strings in Σ^*, namely the set of strings in Σ^* which satisfy φ. Formally we define $L(\varphi) = \{w \in \Sigma^* \mid w \models \varphi\}$.

For example the sentence $\forall x(\mathit{zero}(x) \Rightarrow Q_a(x))$ defines the language $a(a+b)^*$ over the alphabet $\{a, b\}$. Here $\mathit{zero}(x)$ is an abbreviation for the formula $\neg\exists y(y < x)$.

Theorem 1 *Let $L \subseteq \Sigma^*$. Then L is regular iff L is definable in* $\mathrm{MSO}(\Sigma)$ *(i.e. $L = L(\varphi)$ for some sentence φ in* $\mathrm{MSO}(\Sigma)$*).*

We first show that when L is regular, it is definable by an $\mathrm{MSO}(\Sigma)$ sentence. Let $L = L(\mathcal{A})$ for some DFA $\mathcal{A}$. Let $\mathcal{A} = (Q, s, \delta, F)$, and let $Q = \{q_0, \ldots, q_n\}$ $(n \geq 0)$ with $s = q_0$. Then we can define a sentence $\varphi_{\mathcal{A}}$ which is true over a given word w precisely when $\mathcal{A}$ has an accepting run on w. We can define $\varphi_{\mathcal{A}}$ as follows:

$$\begin{array}{ll}
\exists X_0 \cdots \exists X_n (\forall x (& (\bigwedge_{i \neq j} (x \in X_i \Rightarrow \neg x \in X_j) \wedge \bigvee_i x \in X_i) \wedge \\
& (\mathit{zero}(x) \Rightarrow x \in X_0) \wedge \\
& (\bigwedge_{a \in \Sigma,\ i,j \in \{0,\ldots n\},\ \delta(q_i,a) = q_j} ((x \in X_i \wedge Q_a(x) \wedge \neg \mathit{last}(x)) \Rightarrow \\
& \qquad\qquad \exists y(\mathit{succ}(x,y) \wedge y \in X_j))) \wedge \\
& (\mathit{last}(x) \Rightarrow \bigvee_{a \in \Sigma,\ \delta(q_i,a) \in F} (Q_a(x) \wedge x \in X_i)))).
\end{array}$$

Here $\mathit{last}(x)$ is an abbreviation for $\neg\exists y(x < y)$ and $\mathit{succ}(x,y)$ is an abbreviation for $x < y \wedge \neg\exists z(x < z \wedge z < y)$. The conjuncts above respectively assert that "each position is covered by exactly one X_i", "the zero-th position in the word (if it exists) is in X_0", "each position and the succeeding position respect the transition function δ", and "the last position is labelled by a symbol a and state q_i such that $\delta(q_i, a) \in F$".

It is easy to check that a string w satisfies $\varphi_{\mathcal{A}}$ iff $w \in L(\mathcal{A})$.

In the converse direction, we need to argue that the language defined by any sentence in $\mathrm{MSO}(\Sigma)$ is regular. We do this by induction on the structure of the sentence, and as a result we also need to represent the language of non-sentence formulas. The language associated with a formula is in general a collection of pairs of the form $(w, \mathbb{I})$ where $w \in \Sigma^*$ and $\mathbb{I}$ is an interpretation for variables in $pos(w)$.

Given $w \in \Sigma^*$ and a finite subset T of variables we are interested in, we can represent $(w, \mathbb{I})$ as an element w' of $\Sigma \times \{0,1\}^n$ (where n is the cardinality of T). To do this, we first assume a total ordering on variables in Var. Thus each variable in T is now associated with a fixed "row" j from 0 to $n-1$. Then w' is a valid representation of $(w, \mathbb{I})$, wrt T, iff $|w'| = |w|$ and for each $i \in pos(w)$, $w'(i) = (a, r)$, with $a \in \Sigma$ and $r \in \{0,1\}^n$, with $a = w(i)$ and $r(j) = 1$ iff the variable corresponding to row j is mapped to position i by $\mathbb{I}$ (i.e. if the variable was a first-order variable x, then $\mathbb{I}(x) = j$, and if the variable was a second-order variable X, then $j \in \mathbb{I}(x)$). Now, if φ is a formula whose free variables are in T, then we have the notion of whether $w' \models \varphi$ based on whether the $(w, \mathbb{I})$ encoded by w' satisfies φ or not. Let us denote by $L^T(\varphi)$ the set of extended words w' wrt T, which satisfy φ.

We call these w's *extended* models wrt T. Below is an example of an extended model of $w = aababab$ wrt the set of variables $T = \{x, y, X\}$, with that ordering. The extended word satisfies the formula $x \in X \wedge Q_a(y)$.

Σ	a	a	b	a	b	a	b
x	0	0	1	0	0	0	0
y	0	0	0	0	0	1	0
X	1	0	1	1	0	0	1

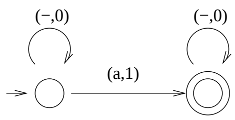
Not every word over $\Sigma \times \{0,1\}^n$ is a valid extended word wrt T. It is valid only if every row corresponding to a first-order variable has exactly one 1. Let us denote the set of valid extended words wrt T by $valid^T(\Sigma)$. The set is clearly regular. Let $\mathcal{A}^T_{val}$ be an automaton which accepts this set.

The claim is now that with every formula φ in MSO(Σ), and any finite set of variables T containing at least the free variables of φ, we can construct an automaton $\mathcal{A}^T_\varphi$ which accepts the language $L^T(\varphi)$.

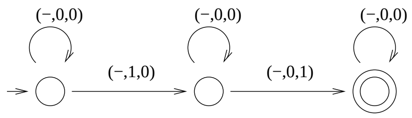
Before we prove the claim, we first note that if we have an automaton $\mathcal{A}$ accepting $L^T(\varphi)$, then for any T' containing T, we also have an automaton $\mathcal{A}'$ accepting $L^{T'}(\varphi)$. We can construct $\mathcal{A}'$ simply by inserting all possible row entries for the new variables in T', in the vectors appearing in the transitions of $\mathcal{A}$. Finally we intersect $\mathcal{A}'$ with $\mathcal{A}^T_{val}$.

We now proceed to prove the claim by induction on the structure of φ. We do the constructions only for T which is exactly the set of free variables in φ, and appeal to the above observation to extend the automaton to larger T'.

Consider the base case of $\varphi = Q_a(x)$. Here $T = \{x\}$. The required automaton $\mathcal{A}^T_\varphi$ is:



For the base case of $\varphi = x < y$, $T = \{x, y\}$ (with variables in that order) and the required automaton $\mathcal{A}_\varphi^T$ is:



The base case $x \in X$ is similar.

Coming to the induction step now, we want to construct an automaton for $\neg\varphi$ wrt T. This is done simply by complementing the automaton $\mathcal{A}_\varphi^T$ and intersecting it with $\mathcal{A}_{val}^T$.

Similarly, for $\varphi \vee \psi$ (wrt T) we take the union of the automata $\mathcal{A}_\varphi^T$ and $\mathcal{A}_\psi^T$ to get the required automaton. Note that this is one place where T may be possibly larger than the exact set of free variables in φ.

For the case of $\exists x\varphi$ and $\exists X\varphi$, we once again use a closure property of regular languages, namely that of closure under "geometric" projection (i.e. where we project each letter (here a vector) to a subset of its dimensions).

This completes the proof of the claim.

The converse direction of Theorem 1 now follows since the claim specialised to sentences gives us the required result.