Monadic second order logic of strings

A *chain* is a finite linear order $(C, \leq)$, $C \neq \emptyset$.

We shall develop monadic second order logic of strings and show that this logic is decidable.

We have:

- *First* order variables: $x, y, z, \ldots$
- *Second* order or set variables: $X, Y, Z, \ldots$
- $=, \leq, \in, \forall, \exists, \neg, \vee, \wedge, \rightarrow, \leftrightarrow$ as logical relations, quantifiers and connectives

The set of well formed monadic second order formulae of this language is defined by:

- $(x = y)$, $(X = Y)$, $(x \leq y)$, $(x \in X)$ are *atomic* formulae, whenever x, y and X, Y are resp. first and second order variables,

- if ϕ, ψ are formulae, then so are $\neg\phi$, $(\phi \wedge \psi)$, $(\phi \vee \psi)$, $(\phi \rightarrow \psi)$, $(\phi \leftrightarrow \psi)$, $\exists x \phi$, $\forall x \phi$, $\exists X \phi$, $\forall X \phi$, where x, X are resp. first and second order variables.

then the logic is <u>second order</u> by virtue of including variables for subsets of the domain, and the <u>monadic</u> refers to the fact that we only allow variables for sets and not binary, ternary, ... relations.

Now, given a chain $(C, \leq)$ and a sentence ϕ, i.e., a formula w/o free variables, we can recursively define satisfaction $(C, \leq) \models \phi$ as in first order logic.

<u>Example</u> For X a 2nd order variable, let $\mathrm{Sub}(X)$ be the formula

$$\exists z\,(x \in X) \wedge \forall x \forall y\Big(\big((x \in X) \wedge (x \leq y)\big) \rightarrow y \in X\Big)$$

So $\mathrm{Sub}(X)$ expresses that $X \neq \emptyset$ and is closed upwards.

Henceforth, we shall use the usual conventions for dropping parentheses to increase readability.

<u>Definition</u> Let SFS be the theory of finite, non-empty chains, i.e., SFS is the set of all sentences true in all finite, non-empty chains.

From automata to formulae:

Suppose $\mathcal{M} = (V, E, s, A, \ell)$ is a DFA over an alphabet Σ. Wlog we can suppose that $\Sigma = \{0,1\}^m$ for some $m \geq 1$.

Now, given a chain $C = \{c_0, c_1, \dots, c_t\}$, where $c_0 < c_1 < \dots < c_t$, let $B_1, \dots, B_{m-k} \subseteq C$ and $b_{m-k+1}, \dots, b_m \in C$ be fixed.

Then any $c \in C$ defines a letter $a = a(c) = (a_1 a_2 \cdots a_m) \in \Sigma$ by

$$a_j = \begin{cases} 1 & \text{if } j \leq m-k \text{ \& } c \in B_j \\ 1 & \text{if } m-k < j \text{ \& } c = b_j \\ 0 & \text{otherwise.} \end{cases}$$

Thus, together C and $B_1, \dots, B_{m-k}, b_{m-k+1}, \dots, b_m$ defines the string

$$a(c_0)a(c_1)\dots a(c_t) \in \Sigma^{t+1} = \{0,1\}^{(t+1)m}.$$

We denote this string by

$$\mathrm{word}(C, B_1, \dots, B_{m-k}, b_{m-k+1}, \dots, b_m) = a(c_0) \dots a(c_t).$$

For example, let $C = \{0, 1, \dots, 7\}$, $B_1 = \{1, 4, 7\}$, $B_2 = \{0, 4\}$, $b_3 = 1$, $b_4 = 6$. Then

$$\mathrm{word}(C, B_1, B_2, b_3, b_4) = \underbrace{0100}_{0}\underbrace{1010}_{1}\underbrace{0000}_{2}\underbrace{0000}_{3}\underbrace{1100}_{4}\underbrace{0000}_{5}\underbrace{0001}_{6}\underbrace{1000}_{7}.$$

In other word, given m, the string $\mathrm{word}(C, B_1, \ldots, B_{m-k}, b_{m-k+1}, \ldots, b_m)$ completely determines the tuple $(C, B_1, \ldots, B_{m-k}, b_{m-k+1}, \ldots, b_m)$ up to isomorphism. E.g., the cardinality of C and hence its isomorphism type is read of from the length of the string.

<u>Theorem</u> Let $\mathcal{A} = (V, E, s, A, \ell)$ be an NFA over $\Sigma = \{0,1\}^m$. Then there is a formula $\phi(X_1, \ldots, X_m)$ such that for any chain $(C, \leq)$ and subsets $B_1, \ldots, B_m \subseteq X$:

$$\mathrm{word}(C, B_1, \ldots, B_m) \in L(\mathcal{A})$$
$$\Updownarrow$$
$$C \models \phi(B_1, \ldots, B_m).$$

<u>Remark</u> Note that any word $w \in \Sigma^+$ can be written on the form $w = \mathrm{word}(C, B_1, \ldots, B_m)$ for some chain C and subsets $B_1, \ldots, B_m \subseteq C$.

<u>Proof</u> Wlog, we can suppose that $V = \{0, 1, \ldots, p\}$ for some $p \geq 0$ and $s = 0$ is the initial state.

Coding of V:

Let $Y_0, \dots, Y_p$ be set variables and let $D(Y_0, \dots, Y_p)$ be the formula

$$\bigwedge_{0 \le i < j \le p} \forall x \, (x \notin Y_i \vee x \notin Y_j) \wedge \forall x \bigvee_{0 \le i \le p} x \in Y_i$$

Using $Y_0, \dots, Y_p$ to represent states, $D(Y_0, \dots, Y_p)$ will express that $\mathcal{M}$ will always be in exactly one state.

Coding of E

For this we first define a formula to express the successor function

$$S(x,y) : \quad x \le y \wedge \neg x = y \wedge \forall z \, (z \le x \vee y \le z).$$

So $C \models S(x,y)$ if and only if y is the immediate successor of x in C. Thus, using this we can introduce a function $S : C \to C$ by letting

$$S(x) = \begin{cases} \text{the successor of } x \text{ if } x \text{ is not maximal,} \\ x \text{ otherwise.} \end{cases}$$

Now, given $a \in \Sigma = \{0,1\}^m$ and set variables $X_1, \dots, X_m$ let $\varphi_a(x, X_1, \dots, X_m)$ be the formula :

$$x \in^{a_1} X_1 \wedge x \in^{a_2} X_2 \wedge \dots \wedge x \in^{a_m} X_m ,$$

where $a = (a_1, \dots, a_m)$, $\in^0 = \notin$, $\in^1 = \in$.

Then for $i = 0, \dots, p$ and $a \in \Sigma$, set $\phi_{i,a}(x, \overline{X}, \overline{Y})$ to be

$$\left(x \in Y_i \wedge \phi_a(S(x), \overline{X}) \wedge x \neq S(x)\right) \longrightarrow S(x) \in Y_j$$

where $i \xrightarrow{a} j$ is an edge in $\mathcal{M}$.

So $\phi_{i,a}(x, \overline{X}, \overline{Y})$ should express that if $\mathcal{M}$ is in state i, the next symbol is a and $\mathcal{M}$ has not finished the computation, then the next state is j.

Coding the first step of a computation:

For every $a \in \Sigma$ let $\psi_a(x, \overline{X}, \overline{Y})$ be the formula

$$\left(\forall y\,(x \leq y) \wedge \phi_a(x, \overline{X})\right) \longrightarrow x \in Y_j$$

where $0 \xrightarrow{a} j$ is an edge in $\mathcal{M}$.

Coding of A

Finally, let $F(x, \overline{Y})$ be the formula

$$S(x) = x \longrightarrow \bigvee_{i \in A} x \in Y_i ,$$

The formula $\phi(X_1, \dots, X_m)$:

Let now $\phi(X_1, \dots, X_m)$ be the formula

$$\exists Y_0, \dots, Y_p \Big(D(\bar{Y}) \wedge \forall x \bigwedge_{a \in \Sigma} \psi_a(x, \bar{X}, \bar{Y}) \wedge$$

$$\forall x\, F(x, \bar{Y}) \wedge \forall x \bigwedge_{\substack{a \in \Sigma \\ 0 \le i \le p}} \phi_{a,i}(x, \bar{X}, \bar{Y}) \Big)$$

We claim that this works, i.e., for any chain $C = \{c_0, c_1, \dots, c_t\}$, $c_0 < c_1 < \dots < c_t$, and subsets $B_1, \dots, B_m \subseteq C$, we have

$$\mathrm{word}(C, B_1, \dots, B_m) \in L(\mathcal{M})$$
$$\Updownarrow$$
$$C \models \phi(B_1, \dots, B_m).$$

To see this, let $C, B_1, \dots, B_m$ be given and consider $w = a_0 a_1 a_2 \dots a_t = \mathrm{word}(C, B_1, \dots, B_m)$.

Suppose first that $\mathcal{M}$ accepts w. Then there is a sequence of states $q_0 q_1 \dots q_t q_{t+1}$ in V such that $q_0 = 0$ and $q_{t+1} \in A$, while $(q_j, q_{j+1}) \in E$ with $a_j \in \ell(q_j, q_{j+1})$.

We define subsets $Y_0, \dots, Y_p \subseteq C$ as follows

$Y_i = \{ c_j \mid q_{j+1} = i \}$. So the Y_i partition C and hence $\mathcal{C} \models D(\overline{Y})$.

Note that $c_j \in Y_i \iff \mathcal{M}$ is in state i after having read $a_0 a_1 \ldots a_j$.

Also, suppose $x \in \mathcal{C}$. Then if $S(x) = x$, we have $x = c_t$, whence, as $q_{t+1} \in A$, also $\bigvee_{i \in A} x \in Y_i$.

So $\mathcal{C} \models \forall x \, F(x, \overline{Y})$.

Now, suppose $c_j \in \mathcal{C}$. Then for $a_j = (a_j^1, a_j^2, \ldots, a_j^m) = w[j]$, we have

$$a_j^i = \begin{cases} 1 & \text{if } c_j \in B_i \\ 0 & \text{if } c_j \notin B_i \end{cases}$$

so $c_j \in^{a_j^1} B_1 \wedge \ldots \wedge c_j \in^{a_j^m} B_m$.

In particular, $\mathcal{C} \models \phi_a(c_j, \overline{B}) \iff a = a_j \in \Sigma$.

So suppose $c_j \in \mathcal{C}$ and $a \in \Sigma$. Then if

$$\forall y \, (c_0 \leq y) \wedge \phi_a(c_j, \overline{B}),$$

we have $j = 0$ and $a = a_0$. It follows that $c_j = c_0 \in Y_i$ for some i st:

$q_0 = 0 \xrightarrow{a_0} i$. So $\mathcal{C} \models \forall x \bigwedge_{a \in \Sigma} \psi_a(x, \overline{B}, \overline{Y})$.

Instead suppose now that $c_j \in C$, $a \in \Sigma$ and $0 \le i \le p$ satisfy

$$c_j \in Y_i \wedge \phi_a(S(c_j), \overline{B}) \wedge c_j \neq S(c_j).$$

Then $j < t$, $S(c_j) = c_{j+1}$, so $a = a_{j+1}$ and $q_{j+1} = i$. It follows that $q_{j+1} = i \xrightarrow{a} k = q_{j+2}$ for some edge and thus $S(c_j) = c_{j+1} \in Y_k$.

So $C \models \forall x \bigwedge_{\substack{a \in \Sigma \\ 0 \le i \le p}} \psi_{a,i}(x, \overline{B}, \overline{Y})$.

This shows that

$$\text{word}(C, B_1, \ldots, B_m) \in L(\mathcal{M}) \Rightarrow C \models \phi(\overline{B}).$$

For the converse implication, suppose that $C \models \phi(\overline{B})$ and let $Y_0, \ldots, Y_p \subseteq C$ be the subsets given by $\phi(\overline{B})$.

We define a sequence of states $q_0 q_1 \ldots q_t q_{t+1}$ by letting $q_{j+1} = i$ if $c_j \in Y_i$.

This is well-defined since $C \models D(\overline{Y})$.

Also, as before, we see that

$$C \models \phi_a(c_j, \overline{B}) \iff a = a_j.$$

So, by checking the formula, one sees

that $q_0 \xrightarrow{a_0} q_1 \xrightarrow{a_1} q_2 \longrightarrow \dots \longrightarrow q_t \xrightarrow{a_t} q_{t+1}$ is a path from $q_0 = 0$ to an accepting state $q_{t+1} \in A$. So word $(C, B_1, \dots, B_m) \in L(\mathcal{M})$.

$\square$

<u>Remark</u> Since any word $w \in \Sigma^+$ is in the form $w = \text{word}(C, B_1, \dots, B_m)$, it follows that we can reduce the question of $w \in L(\mathcal{M})$ to a question of satisfiability of a certain formula in C.

<u>From formulae to automata</u>:

Fix $1 \le k \le m$ and let $X_1, \dots, X_{m-k}$ and $x_{m-k+1}, \dots, x_m$ be resp. second and first order variables. $\bar{X} = (X_1, \dots, X_{m-k})$, $\bar{x} = (x_{m-k+1}, \dots, x_m)$

<u>Definition</u> Let $\phi(\bar{X}, \bar{x})$ be a formula and $\mathcal{M}$ be an ε-NFA over $\Sigma = \{0,1\}^m$.
We say that $\mathcal{M}$ <u>represents</u> $\phi(\bar{X}, \bar{x})$ if for all chains C, $B_1, \dots, B_{m-k} \subseteq C$ and $b_{m-k+1}, \dots, b_m \in C$ we have

$$C \models \phi(\bar{B}, \bar{b}) \iff \mathcal{M} \text{ accepts word}(C, \bar{B}, \bar{b}).$$

Example $\phi(X_1, x_2)$ is the formula $x_2 \in X_1$.

So $\Sigma = \{0,1\}^2$ and for any chain

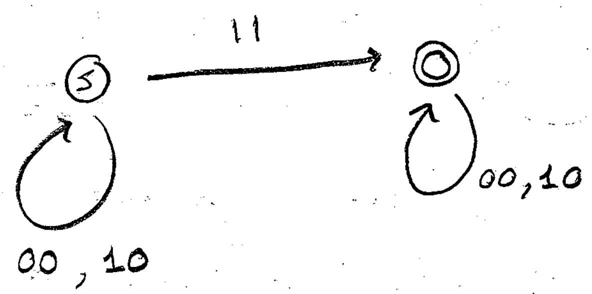
$C = \{c_0, \dots, c_t\}_<$,

$$\text{word}(C, B_1, b_2) = a_0^1 a_0^2 a_1^1 a_1^2 \dots a_t^1 a_t^2 ,$$

where $a_j^2 = \begin{cases} 1 & \text{if } b_2 = c_j \\ 0 & \text{if not} \end{cases}$

and $a_j^1 = \begin{cases} 1 & \text{if } c_j \in B_1 \\ 0 & \text{if not} \end{cases}$.

So let $\mathcal{A}$ be given by



Then $\mathcal{A}$ represents $\phi(X_1, x_2)$.

Exercise Find automata representing $x_1 \le x_2$, $X_1 = X_2$.

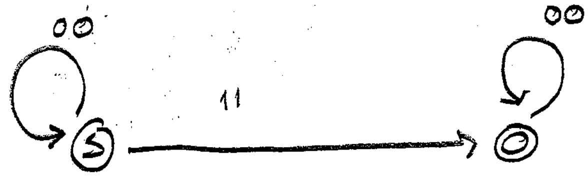
Example Representing $x_1 = x_2$:

Again $\Sigma = \{0,1\}^2$ and for any chain $C = \{c_0, \dots, c_t\}_<$

$$\text{word}(C, b_1, b_2) = a_0^1 a_0^2 a_1^1 a_1^2 \dots a_t^1 a_t^2 , \text{ where}$$

$a_j^i = \begin{cases} 1 & \text{if } b_i = c_j \\ 0 & \text{if not} \end{cases}$

So let $\mathcal{M}$ be given by



Then $\mathcal{M}$ represents "$x_1 = x_2$".

<u>Lemma</u> The NFA representable formulas are closed under the logical connectives and quantification.

<u>Proof</u> Suppose $\phi(\bar{X}, \bar{x})$ and $\psi(\bar{Y}, \bar{y})$ are represented by DFA $\mathcal{M}$ and $\mathcal{N}$ on alphabets $\Sigma = \{0,1\}^{|\bar{X}|+|\bar{x}|}$ and $\Lambda = \{0,1\}^{|\bar{Y}|+|\bar{y}|}$ respectively. By changing the alphabets and the automata correspondingly, we can suppose that actually $\bar{X} = \bar{Y}$, $\bar{x} = \bar{y}$ and so the machines are on the same alphabet

$\Sigma = \{0,1\}^m$, $\bar{X} = (X_1, \dots, X_{m-k})$, $\bar{x} = (x_{m-k+1}, \dots, x_m)$.

Then, e.g.,

$$\{ word(C, \bar{B}, \bar{b}) \mid C \models \phi(\bar{B}, \bar{b}) \wedge \psi(\bar{B}, \bar{b}) \}$$

$$= L(\mathcal{M}) \cap L(\mathcal{N})$$ which is regular.

Same for $\neg$, $\rightarrow$, $\vee$, $\leftrightarrow$.

Now consider instead $\exists X_i \, \phi(\overline{X}, \overline{x})$.

Then

$$\{ \text{word}(C, B_1, \ldots, B_{i-1}, B_{i+1}, \ldots, B_{m-k}, \overline{b}) \in (\{0,1\}^{m-1})^+ \mid C \models \exists X_i \, \phi(B_1, \ldots, B_{i-1}, X_i, B_{i+1}, \ldots, B_{m-k}, \overline{b}) \}$$

$$= \{ w \in (\{0,1\}^{m-1})^+ \mid \exists v \in \{0,1\}^+ \ |w| = |v| \text{ and } w * v \in L(\mathcal{U}) \},$$

where $w * v$ is obtained from w by filling in letters of v into the appropriate spots in w.

It thus suffices to prove that regular languages are closed under projection. $\Box$

<u>Theorem</u> The theory SFS of monadic second order logic for chains is decidable.

<u>Pf</u> We describe an algorithm that given any sentence ϕ decides if ϕ is true in any chain.

Suppose ϕ is given and let X be any variable. Set $\psi(X) = \phi \wedge X = X$. Then clearly for any chain C

$C \models \phi \iff C \models \psi(C)$

From ψ we compute the DFA $\mathcal{M}$ representing ψ. So $\mathcal{M}$ is on the alphabet $\{0,1\} = \Sigma$. Then

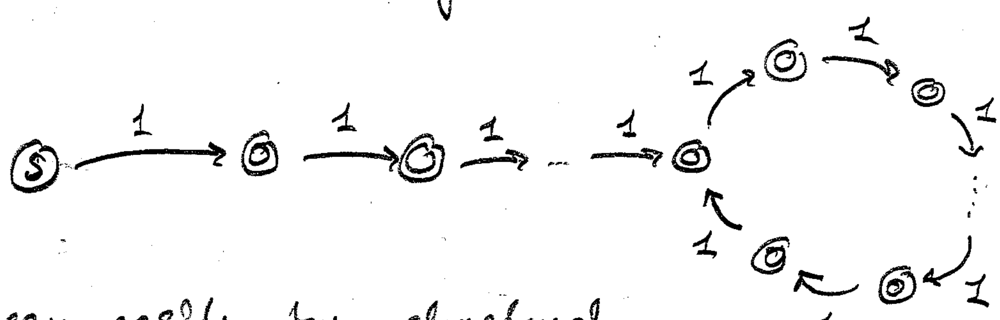
$$\phi \in SFS \iff \text{for any chain } C, \quad C \models \psi(C)$$

$$\iff \text{for any chain } C, \quad \text{word}(C,C) \in L(\mathcal{M}).$$

But note that if $C = \{c_0, \dots, c_t\}_<$ is a chain, then $\text{word}(C,C) = \underbrace{11\dots1}_{t+1}$.

So in other words, $\phi \in SFA \iff 11^* \subseteq L(\mathcal{M})$.

But this is a decidable question in the DFA $\mathcal{M}$. Namely, for this to be true, $\mathcal{M}$ needs to contain a configuration



which can easily be checked. $\square$