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**: \(X, Y, Z, \ldots\)
- \(=, \leq, \in, \forall, \exists, \rightarrow, \leftrightarrow\) 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, where \(x, y\) and \(X\) are respectively first and second order variables,

- if \(\phi, \psi\) are formulae, then so are
  - \(\neg \phi\), \((\phi \land \psi), (\phi \lor \psi), (\phi \rightarrow \psi), (\phi \leftrightarrow \psi)\)
  - \(\exists x \phi\), \(\forall x \phi\), \(\exists X \phi\), \(\forall X \phi\)

where \(x, X\) are respectively first and second order variables.
the logic is second order by virtue of including variables for subsets of the domain, and the monadic refers to the fact that we only allow variables for sets and not binary, ternary, ... relations.

Now, given a chain \((C, \subseteq)\) and a sentence \(\phi\), i.e., a formula with free variables, we can recursively define satisfaction \((C, \subseteq) \models \phi\) as in first order logic.

**Example.** For \(X\) a 2nd order variable, let \(\text{Sub}(X)\) be the formula

\[
\exists z (z \in X) \land \forall x \forall y ((x \in X) \land (x \subseteq y) \rightarrow y \in X)
\]

so \(\text{Sub}(X)\) expresses that \(X \neq \emptyset\) and is closed upwards.

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

**Definition.** Let \(\text{SFS}\) be the theory of infinite, non-empty chains. \(\text{SFS}\) is the set of all sentences true in all infinite, non-empty chains.
Suppose \( M = (V, E, s, A, l) \) is a DFA over an alphabet \( \Sigma \). We can suppose that \( \Sigma = \{0, 1\}^\pm \) for some \( m \in \mathbb{N} \).

Now, given a chain \( C = \{c_0, c_1, \ldots, c_k\} \), where \( c_0 < c_1 < \ldots < c_k \), let \( B_1, \ldots, B_{m-k} \in C \) and \( b_{m-k+1}, \ldots, b_m \in C \) be fixed. Then any \( c \in C \) defines a letter \( a = a(c) = (a_1, a_2, \ldots, a_m) \in \Sigma \) by

\[
a_{ij} = \begin{cases} 1 & \text{if } j = m-k+1 \text{ and } i \leq c \in B_j^i, \\ 1 & \text{if } m-k < j \text{ and } c \in B_j, \\ 0 & \text{otherwise}.
\end{cases}
\]

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

\[
a(c_0) a(c_1) \cdots a(c_k) \in \Sigma^{t+1} = \{0, 1\}^{(t+1)m}
\]

We denote this string by

\[
\text{word}(C, B_1, \ldots, B_{m-k}, b_{m-k+1}, \ldots, b_m) = a(c_0) \cdots a(c_k).
\]

For example, let \( C = \{\varepsilon_0, 1, \varepsilon_1, \varepsilon_2\} \), \( B_1 = \varepsilon_1, \varepsilon_2 \), \( B_2 = \varepsilon_0, \varepsilon_1 \), \( b_3 = 1 \), \( b_4 = 6 \). Then

\[
\text{word}(C, B_1, B_2, b_3, b_4) = 010010110000000001100000000016.
\]
In other words, given a string
word \((C, B_1, \ldots, B_{n-1}, b_m, \ldots, b_m)\) completely
determines the tuple \((C, B_1, \ldots, B_{n-1}, b_m, \ldots, b_m)\)
up to isomorphism. For this cardinality of \(C\)
and hence its isomorphism type is read
of from the length of the string.

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

\[ C \models \phi(B_1, \ldots, B_m). \]

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

**Proof.** Wlog, we can suppose that \(V = \{0, 1, \ldots, p\}\)
for some \(p > 0\) and \(s = 0\) is the initial state.
Coding of $V$:

Let $Y_0, \ldots, Y_p$ be set variables and let

$D(Y_0, \ldots, Y_p)$ be the formula

$$\bigwedge_{0 \leq i < j \leq p} \forall x \big( x \notin Y_i \lor x \notin Y_j \big) \land \forall x \forall y \forall z \big( z \leq x \land y \leq z \big).$$

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

Coding of $E$:

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

$S(x,y) : x \leq y \land x = y \lor \forall z \big( z \leq x \land y \leq z \big).$

So $C = S(x,y)$ if and only if $y$ is the immediate successor of $x$ in $C$. Thus, using this we can introduce a function $S : C \rightarrow C$ 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 $X_1, \ldots, X_m$ let $\phi_a(x, X_{1, \ldots, 2X_m})$ be the formula
$x \in a_1 X_1 \land x \in a_2 X_2 \land \ldots \land x \in a_n X_n$, \\
where $a = (a_1, \ldots, a_n)$, $e^0 = \epsilon$, $e^1 = \epsilon$.

Thus for $i = 0, \ldots, p$ and $a \in \Sigma$, set $\phi_{i,a}(x, \overline{X}, \overline{y})$ to be

$$(x \in Y_i \land \phi_a(S(x), \overline{X}) \land x \neq S(x)) \rightarrow S(x) \in Y_i$$

where $i \rightarrow j$ is an edge in $a$.

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

Casting the first step of a computation:

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

$$(\forall y (x \leq y) \land \phi_a(x, \overline{X})) \rightarrow x \in Y_i$$

where $i \rightarrow j$ is an edge in $a$.

Casting at $A$

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

$S(x) = x \rightarrow \bigvee_{i \in A} x \in Y_i$
Thus formula $\Phi(X_1, \ldots, X_n)$:

Let now $\Phi(X_1, \ldots, X_n)$ be the formula

$\exists Y_0, \ldots, Y_p \left( D(Y) \land \forall x \bigwedge_{a \in A} \psi(x, X, Y) \land \right.$

$\forall x \ F(x, Y) \land \forall x \bigwedge_{a \in A} \phi_{x,i}(x, X, Y) \bigg)_{0 \leq i \leq p}.$

We claim that this works, i.e., for any chain

$C = \{ c_0, c_1, \ldots, c_m \}, \ c_0 < c_1 < \ldots < c_m,$

and subsets

$B_1, \ldots, B_m \subseteq C,$

we have

$C \models \phi(B_1, \ldots, B_m).$

To see this, let $C, B_1, \ldots, B_m$ be given

and consider

$w = a_1 a_2 \ldots a_t = \text{word} \left( C, B_1, \ldots, B_m \right).$

Suppose first that $w$ accepts $w$. Then there

is a sequence of states $q_0 q_1 \ldots q_{t-1} q_{t+1}$ in $V$

with $q_0 = 0$ and $q_{t+1} \in A$, while

$(q_j, q_{j+1}) \in E$ with $a_j \in L(q_j, q_{j+1}).$

We define subsets $Y_0, \ldots, Y_p \subseteq C$ as follows:
\[ Y_i = \frac{\varepsilon}{2} \{ y \mid q_{f+1} = i \} \quad \text{So the } Y_i \text{ partition} \]

\[ C \quad \text{and hence} \quad C \models D(\overline{y}). \]

Note that \( c_j \in Y_i \Rightarrow \exists \overline{y} \text{ is in state } i \text{ after having read } a_0 \ldots a_j. \]

Also, suppose \( x \in C \). Then \( \overline{s}(x) = \underline{x} \), we have \( x = c_1 \) \quad \text{whence, as } \overline{s} \text{ is } A \text{, also } \forall x \in Y_i. \quad \text{ie} \quad A \]

So \( C \models \forall x. F(x, \overline{y}) \)

\[ \text{Now, suppose } c_j \in C. \text{ Then } \]

\[ \forall r \quad a_j = (a_j^1, a_j^2, \ldots, a_j^m) = w[1:2] \quad \text{we have } \]

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

\[ \text{So } c_j \in B_1 \land \ldots \land c_j \in B_m. \]

\[ \text{In particular, } C \models \varphi_c(c_j, \overline{B}) \quad \Rightarrow \quad r = a_j \in \Sigma. \]

So suppose \( c_j \in C \text{ and } a \in \Sigma \). Then \( \forall y \quad c_j \models \varphi_a(c_j, \overline{B}), \)

\[ \text{we have } y = 0 \text{ and } a = x_0. \text{ It follows that } \]

\[ c_j = c_{x_0} \in Y_i \text{ is same } i \text{ and } \]

\[ q_0 = 0 \rightarrow \varepsilon. \quad \text{So } C \models \forall x. \varphi_c(x, \overline{B}, \overline{y}). \]
Instead suppose now that \( c_j \in C \), \( a \in \Sigma \) and \( \alpha_i \in \Sigma \) satisfy
\[
eq \Phi \left( S(c_j), \overline{B} \right) = c_j \neq S(c_j) .
\]
Then \( j < k \), \( S(c_j) = c_{j+1} \), so \( a = a_{j+1} \) and
\[a_{j+1} = \bar{i} . \]
It follows that \( q_{j+1} = i \xrightarrow{a} k = q_{j+2} \)
and some edge and thus \( S(c_j) = c_{j+1} \in Y_k \).

So \( C \ni \forall x \land \forall \alpha_i \left( x, \overline{B}, \overline{Y} \right) \).

This shows that
\[
\text{word } (C, \overline{B}, \overline{B}) \in L(\Sigma) \implies C \notin \Phi (\overline{B}) .
\]

For the converse implication, suppose that
\( C \notin \Phi (\overline{B}) \) and let \( Y_0, ..., Y_p \in C \)
be the subsets given by \( \Phi (\overline{B}) \).

We define a sequence of states \( q_0, q_1, ..., q_{p+1} \)
by letting \( q_{j+1} = i \xrightarrow{a} q_j \).

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

Also, as before, we see that
\[
C \models \Phi \left( c_j, \overline{B} \right) \iff a = a_j .
\]

So, by checking the formulas, our assumption
Let \( q_0 \xrightarrow{a_0} q_1 \xrightarrow{a_1} q_2 \rightarrow \cdots \rightarrow q_t \xrightarrow{a_t} q_{t+1} \) be a path from \( q_0 = c \) to an accepting state \( q_{t+1} \in A \). So word \((C, b_1, \ldots, b_m) \in L(\Delta)\).

Remark. Since any word \( w \in \Sigma^+ \) is an \( c \)-run of word \((C, b_1, \ldots, b_m)\), it follows that we can reduce the question of \( w \in L(\Delta) \) to a question of satisfiability of a certain formula in \( \mathcal{C} \).

From formulae to automata:

Fix \( 1 \leq k \leq m \) and let \( x_1, \ldots, x_{m-k} \) and \( x_{m-k+1}, \ldots, x_m \) be resp. second and first order variables. \( \overline{x} = (x_1, \ldots, x_{m-k}) \), \( \overline{z} = (x_{m-k+1}, \ldots, x_m) \).

Definition. Let \( \phi(\overline{x}, \overline{z}) \) be a formula and let \( \Sigma \) be an \( \varepsilon \)-NFA over \( \Sigma = \{0, 1, \varepsilon\} \).

We say that all \( \varepsilon \)-NFA over \( \Sigma \) \( \varepsilon \)-accepts all chains \( C, B_1, \ldots, B_{m-k} \subseteq C \) and \( b_{m-k+1}, \ldots, b_m \in C \) if we have

\[ C \models \phi(\overline{b}, \overline{b}) \iff \text{all \( \varepsilon \)-NFA accept word } (C, B_1, \ldots, B_m). \]
Example \( \phi(x_1, x_2) \) is the formula \( x_2 \in X_1 \).

So \( \Sigma = \{ 0, 1 \}^2 \) and for any chain
\[
C = \langle c_0, \ldots, c_t \rangle,
\]
word \( (C, b_1, b_2) = a_0^t a_0^2 a_1^1 a_2^1 \ldots a_t^1 a_{t+1}^2 \),
where
\[
a_i = \begin{cases} 1 & \text{if } b_1 = c_i^1 \\ 0 & \text{if not} \end{cases}
\]
and
\[
a_i = \begin{cases} 1 & \text{if } c_i^1 \in B_1 \\ 0 & \text{if not} \end{cases}
\]

So let \( \Phi \) be given by

\[
\begin{array}{c}
00, 10 \\
\rightarrow
\end{array}
\begin{array}{c}
11 \\
\end{array}
\]

Then \( \Phi \) represents \( \phi(x_1, x_2) \).

Exercise Find automata representing \( x_1 \leq x_2, \quad x_1 = x_2 \).

Example \( \phi(x_1, x_2) \) is the formula \( x_1 \leq x_2 \).

So \( \Sigma = \{ 0, 1 \}^2 \) and for any chain \( C = \langle c_0, \ldots, c_t \rangle \),
word \( (C, b_1, b_2) = a_0^1 a_0^2 a_1^2 a_1^1 \ldots a_t^1 a_{t+1}^2 \),
where
\[
a_i = \begin{cases} 1 & \text{if } b_1 = c_i^1 \\ 0 & \text{if not} \end{cases}
\]
and
\[
a_i = \begin{cases} 1 & \text{if } c_i^1 \leq b_1 \\ 0 & \text{if not} \end{cases}
\]
so not all be given by

\[ 0 \circ \rightarrow 1 \rightarrow 0 \circ \]

Then all represents \( x_1 = x_2 \).

\underline{Lemma} The NFA representable formulas are closed under the logical connectives and quantification.

\underline{Proof} Suppose \( \phi(x, \bar{x}) \) and \( \psi(z, \bar{z}) \) are represented by DFA \( M_1 \) and \( \mathcal{N} \) on alphabets \( \Sigma = \{a, b, c\} \) and \( \Lambda = \{0, 1\} \), respectively. By changing the alphabets and the automata correspondingly, we can suppose that actually \( \bar{x} = \bar{y} \), \( x = y \), and so the machines are on same alphabet \( \Sigma = \{0, 1\} \), \( Y = (x_1, \ldots, x_m) \), \( \bar{y} = (x_{m+1}, \ldots, x_n) \).

Thus, \( \text{word}(C, \bar{b}, \bar{b}) \mid C \models \phi(\bar{b}, \bar{b}) \land \psi(\bar{b}, \bar{b}) \) \]

\[ = \text{L}(M) \cap \text{L}(\mathcal{N}) \text{ which is regular.} \]
Now consider instead \( \exists X \phi (X, x) \).

Thus

\[
\exists \ w : (C, b_1, b_2, \ldots, b_m) \in (\{0, 1\}^m)^+ \mid C \models \exists X \phi (X, b_2, \ldots, b_m)
\]

\[
= \exists w \in (\{0, 1\}^m)^+ \mid \exists v \in \{0, 1\}^n \mid |w| = |v| \text{ and } w \star v \in \mathcal{L}(\mu)
\]

where \( w \star v \) is obtained from \( w \) by filling in

letting \( \star \) into the appropriate spots in \( w \).

It thus suffices to prove that regular

languages are closed under projection.

\[ \square \]

**Theorem** The theory \( S_F \) of monadic second

order logic is decidable.

\[ \text{If we describe an algorithm that given}
\]

any sentence \( \phi \) decides \( \phi \) is true

in any chain.

Suppose \( \phi \) is given and let \( X \) be any variable.

Set \( \psi(X) = \phi \land X = X \). Then clearly for any

chain \( C \)

\[ C \models \phi \iff C \models \psi(C) \]
From $\Psi$ we compute the DFA $M$ representing $\Psi$, so $\Psi$ is on our alphabet $\Sigma_0,1^* \in E$.  Then

$\phi \in \mathcal{SFA} \iff \exists$ any chain $C, \ C \models \Psi(C)$

$\iff \exists$ any chain $C$, word $(C,C) \in L(M)$

But note that if $C = \{c_0, \ldots, c_k\}$ is a chain,

then word $(C,C) = \underbrace{11 \ldots 1}_{k+1}$

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

But this is a decidable question in the DFA $M$. Namely, to check if $\phi$ is true, all needs to contain a configuration

\[ \begin{array}{c}
0 \\
1 \\
2 \\
\end{array} \]

which can easily be checked.