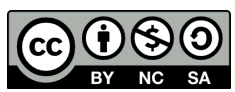
CDM

SOL and Words

Klaus Sutner

Carnegie Mellon University

Fall 2024



Intuitively, FOL allows us to ask "local" queries about a structure.

For example, given a graph $\langle V, E \rangle$ we can describe the local neighborhood of any vertex. We can also assert that there are paths of a certain, fixed length. But we cannot express more global properties such as connectivity: we cannot handle paths of arbitrary length.

This and other limitations makes it natural to try to look for stronger/different logics and hope they might still admit algorithms for model checking.

Gödel noticed that his Completeness Theorem had a somewhat counterintuitive consequence.

Theorem (Compactness Theorem, Gödel 1930)

Γ has a model if, and only if, every finite subset Γ_0 of Γ has a model.

This follows directly be replacing $\models$ by $\vdash$, and the trivial observation that every proof can use only finitely many axioms.

This is positively strange. Think of axioms as specifications that pin down certain requirements. We can use infinitely many specs to describe some (infinite) structure $\mathcal{A}$ in great gory detail, where finitely many would just not provide enough information.

But the Compactness Theorem says that a first-order formula that holds over $\mathcal{A}$ is already forced to hold because of some finite subset of Γ.

For each particular first-order property only finitely many of the axioms matter, not all of them.

We would like (PA) to pin down all the properties of the natural numbers. But, thanks to compactness, here is some very bad news:

> $\mathfrak{N}$ is not the only model of (PA).

To construct an unintended model of (PA), introduce a new constant c and add the following infinitely many new axioms:

$$0 < c, S(0) < c, S^2(0) < c, \ldots, S^n(0) < c, \ldots$$

In other words, we insist that $n < c$ for every "real" natural number n.

Call the new set of axioms (PA^∞).

Claim

(PA^{∞}) *has a model* $\mathfrak{N}^{\star}$ *and this model is not isomorphic to* $\mathfrak{N}$*.*

Proof. To see this, exploit compactness.

Any finite subset Γ_0 of (PA^{∞}) contains only finitely many of the new axioms. So there is a largest n such that $S^n(0) < c \in \Gamma_0$.

But then we can simply interpret c as $n+1$ in the standard model $\mathfrak{N}$.

So let $\mathfrak{N}^{\star}$ be a model of (PA^{∞}). Since $\mathfrak{N}^{\star}$ is in particular a model of (PA) we have a full copy of $\mathfrak{N}$ inside $\mathfrak{N}^{\star}$: there are distinct elements for 0, $S(0)$, $S(S(0))$ and so forth.

But, this structure also contains an "infinitely large" element $c = c^{\mathfrak{N}^\star}$: since c is a constant in the language it is interpreted by some element of $\mathcal{A}$, and it follows from the extra axioms that

$$\mathfrak{N}^\star \models S^n(0) < c$$

for all $n \geq 0$.

The object c is not infinitely large from the perspective of $\mathfrak{N}^\star$, it's just a "natural number" there. As are

$$c \pm 1, c + c, c^2, c^c, \ldots$$

and so on.

Non-standard models of Peano arithmetic like $\mathfrak{N}^\star$ may seem like a mere curiosity, vaguely interesting but essentially useless.

A. Robinson realized in 1960, though, that this type of unintended model is the perfect framework for analysis: one can construct strange models of the reals that contain infinitesimal elements. As a consequence, there is no need for limits in such a model.

In essence, differentiation is just a quotient operation

$$\frac{f(x+h)-f(x)}{h}$$

where $h = 1/c$ is an infinitesimal. Likewise, integrals are just sums.

The drawback is, of course, that someone trying to learn or apply calculus this way must already have some background in logic – perhaps unsurprisingly, non-standard analysis never really took off.

Suppose you are doing ordinary calculus (yes, that still happens) and want to say something along the lines of

> For every differentiable function f, yarglebargle f bargleblargh.

Even if we assume that the underlying structure is $\mathbb{R}$, this is not expressible.

Things get worse when we try to make statements about some class of higher order functions, say, measures on some space.

And, we cannot even pin down a tame structure like $\mathfrak{N}$.

We simply need more expressive power.

In first-order logic quantification takes place only over individuals.

Sets of individuals, and more generally relations and functions on the domain, are given by an appropriate first-order structure but cannot be quantified at the syntactical level.

And, of course, there is no way to quantify over, higher type objects such as families of functions.

This suggests a generalization: how about allowing quantification over all these objects? For example, rewritten in terms of set theory we would like to be able to make assertions

$$\forall\, x \in \mathfrak{P}(A)\ \ldots x \ldots$$

When $A = \mathbb{N}$, this would allow us to talk about sets of reals, a perfectly natural thing to do.

J. Harrison has developed an HOL theorem prover, written in OCaml, that is surprisingly powerful and freely available.

http://www.cl.cam.ac.uk/~jrh13/hol-light/

As we will see in a moment, no such prover can be complete.

However, this one is capable of proving lots of interesting theorems and has had considerable success in the verification of fairly complicated fragments of math.

It has also been used to verify floating-point algorithms for Intel.

1994: Pentium FDIV Bug

$$4195835.0/3145727.0 = 1.33382044913624100\mathbf{25} \quad \text{correct}$$
$$4195835.0/3145727.0 = 1.33373906896890203\mathbf{75894} \quad \text{pentium}$$

Alternatively

$$4195835.0 - 3145727.0 * (4195835.0/3145727.0) = 0 \quad \text{correct}$$
$$4195835.0 - 3145727.0 * (4195835.0/3145727.0) = \mathbf{256} \quad \text{pentium}$$

Discovered in October 1994 by number theorist Thomas R. Nicely, doing research in pure math.

HOL is a huge sledge-hammer. As a first step towards more powerful quantification, consider so-called second-order logic (SOL) where one can only quantify over

- individuals,
- sets of individuals (monadic fragment),
- k-ary relations on individuals.

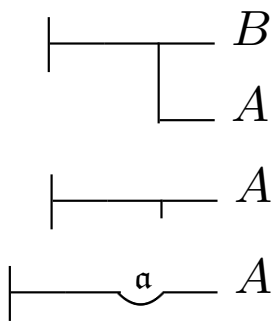
This is called second-order logic (SOL) and turns out to be in essence just as powerful as full HOL.

Incidentally, both FOL and SOL were introduced more or less by Frege in his *Begriffsschrift* in 1879.

Aside: Begriffsschrift

In 1879 Frege published his *Begriffsschrift*, which translates roughly as "concept script." He establishes a language and a (very powerful) logic, the groundwork for the Grundgesetze.

Unfortunately, Frege developed a horrible, two-dimensional notation system.

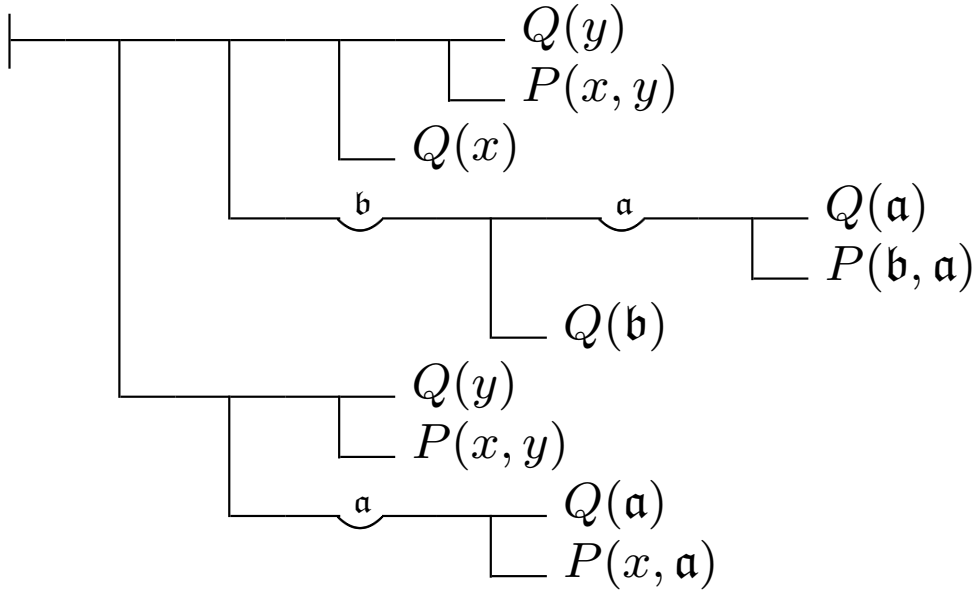


$A \Rightarrow B$

$\neg A$

$\forall\, \mathfrak{a}\, A$

This may seem harmless, but if one constructs larger formulae from these primitives, things start to look quite ominous.


$Q(y)$
$P(x, y)$
$Q(x)$
$\mathfrak{b}$
$\mathfrak{a}$
$Q(\mathfrak{a})$
$P(\mathfrak{b}, \mathfrak{a})$
$Q(\mathfrak{b})$
$Q(y)$
$P(x, y)$
$\mathfrak{a}$
$Q(\mathfrak{a})$
$P(x, \mathfrak{a})$

In modern notation (arguably, a huge improvement):

$$\begin{aligned}[\forall\, a\,(P(x,a) \Rightarrow Q(a)) \Rightarrow (P(x,y) \Rightarrow Q(y))] \Rightarrow \\ [\forall\, b\,(Q(b) \Rightarrow \forall\, a\,(P(b,a) \Rightarrow Q(a)))] \Rightarrow \\ Q(x) \Rightarrow \\ P(x,y) \Rightarrow Q(y)\end{aligned}$$

There are 4 premisses, and altogether they imply $Q(y)$.

Notation matters!

When writing SOL formulae (in modern notation), it is sometimes helpful to indicate the arity of relation and function symbols by a superscript. So, for example,

$$\forall\, R^{(2)}\, (\ldots R(x, y) \ldots)$$

is a statement about all binary relations.

Note that one can get by without function symbols. For example, we can fake unary functions like so. Instead of $\exists\, f^1\ \ldots$ we write

$$\exists\, R^{(2)}\, (\forall\, x\, \exists\, y\, R(x, y) \wedge \forall\, x, y, z\, (R(x, y) \wedge R(x, z) \Rightarrow y = z) \wedge \ldots)$$

We could also get rid of equality on individuals:

$$x = y \iff \forall R^{(1)} \left(R(x) \Leftrightarrow R(y)\right)$$

The importance of this idea was first recognized by Leibniz and is enshrined in his principium identitatis indiscernibilium: if we cannot tell two entities apart, they are identical.

With second-order quantification one can avoid the effects of the compactness theorem for first-order.

For example, suppose we use the standard Peano axioms to describe arithmetic. Then add the following axiom:

$$\forall\, x \forall\, X \left(X(0) \wedge \forall\, z \left(X(z) \Rightarrow X(z+1) \right) \Rightarrow X(x) \right)$$

Here X is a subset of the universe.

In other words, every set containing 0 and closed under successors is already the whole universe: this rules out the non-standard, "infinite" elements.

In fact, SOL can be used to produce a finite axiomatization of arithmetic that has only one model (categoricity).

Aside: Two Uses of Axioms

One standard application of axioms is to describe **all** examples of some particular kind of structure.

For example, the standard (first-order) axioms for groups describe the whole, large and very complex class of all groups.

The other standard application is to construct a set of axioms with the goal of pinning down **one** particular structure precisely.

For example, the Peano axioms form an attempt to give a precise and complete description of the natural numbers. Alas, if first-order this attempt fails, there are non-standard models.

Full SOL is quite powerful, but also quite unwieldy. In particular proof theory breaks down in the following sense: we cannot construct a proof system for SOL in a way that it satisfies the three standard requirements:

- Sound: only valid formulae are provable,
- Complete: all valid formulae are provable,
- Effective: the collection of all proofs is decidable.

This has lead some people like Willard Quine to deny that SOL is "a logic" at all. In their view it's just a branch of set theory.

The reason there can be no adequate notion of proof for SOL is the following: for every sentence φ of arithmetic in FOL, one can effectively construct a sentence $\widehat{\varphi}$ of SOL such that

$$\widehat{\varphi} \text{ valid} \iff \mathfrak{N} \models \varphi.$$

Since proofs are always semidecidable (give a reasonable axiom system), the existence of a deduction system for SOL would immediately imply that the LHS is also semidecidable.

But Gödel and Tarski have shown that the RHS is far from being semidecidable.

In fact, validity of SOL as a decision problem is hideously complicated.

Often it suffices to consider a weak subsystem for full SOL where quantification is restricted to just two kinds:

- individuals, and
- sets of individuals (relations of type $R^{(1)}$).

Notation:

$$\exists X \qquad \forall X$$

$$x \in X \qquad X(x)$$

Example: Least Upper Bounds

Assuming a binary total order $\leq$, we can express the assertion that every bounded set has a least upper bound:

$$\forall\, X \,\big(\exists\, z\, X(z) \wedge \exists\, x \forall\, z\, (X(z) \Rightarrow z \leq x) \Rightarrow$$

$$\exists\, x\, (\forall\, z\, (X(z) \Rightarrow z \leq x) \wedge \forall\, y\, (\forall\, z\, (X(z) \Rightarrow z \leq y) \Rightarrow x \leq y))\big)$$

This is the critical property of the reals and cannot be expressed in FOL.

Example: Well-Order

Again assume a binary total order $\leq$. We can express the assertion that we have a well-order in terms of the least-element principle: every non-empty set has a least element.

$$\forall\, X \left(\exists\, z\, X(z) \Rightarrow \right.$$

$$\left. \exists\, x\, (X(x) \wedge \forall\, z\, (X(z) \Rightarrow x \leq z)\right)$$

This is the critical property of the natural numbers with the standard order, and cannot be expressed in FOL.

Example: Reachability

Lastly, consider a digraph, a single binary edge relation $\rightarrow$.

We can express the assertion that there is a path from s to t as follows:

$$\forall\, X \left(X(s) \wedge \forall\, x, y \, (X(x) \wedge x \rightarrow y \Rightarrow X(y)) \Rightarrow X(t) \right)$$

Again, FOL is not strong enough to express path existence in general (and thus other concepts like connectivity).

Logics are used to describe structures. For example, first-order logic can be used to describe Presburger arithmetic, arithmetic without a multiplication operation.

Wild Idea: Can we think of a single word as a structure?

Just to be clear, we don't want to talk about, say, the structure of all words over some alphabet with concatenation—though that's also interesting and opens the door to algebraic treatment.

We want a structure that consists of a single word.

What on earth might the elements and relations of that structure be?

A priori, W is just a sequence of symbols:

$$W = a_1 a_2 \dots a_{n-1} a_n$$

In order to be able to describe properties of such a word, we need to be able to talk about particular positions in the word.
Thus, positions range over $[n] = \{1, 2, \dots, n\}$ where $n = |W|$.

For example, we want to say "in position 42 there is a letter a."

Or "position x is less than position y."

We will have variables x, y, z, ... that range over positions.

We allow the following basic predicates between variables:

$$x < y \qquad\qquad x = y$$

Of course, we can get, say, $x \geq y$ by Boolean operations.

Most importantly, we write

$$\mathbf{a}(x)$$

for "there is a letter a in position x."

We allow quantification for position variables.

$$\exists\, x\, \varphi \qquad \forall\, x\, \varphi$$

For example, the formula

$$\exists\, x, y\, (x < y \wedge \mathbf{a}(x) \wedge \mathbf{b}(y))$$

intuitively means "somewhere there is an a and somewhere, to the right of it, there is a b."

The formula

$$\forall\, x, y\, (\mathbf{a}(x) \wedge \mathbf{b}(y) \Rightarrow x < y)$$

intuitively means "all the a's come before all the b's."

We also have second-order variables X, Y, Z, ... that range over sets of positions in a word.

$$\exists X \varphi \qquad \forall X \varphi \qquad X(x)$$

Sets of positions are all there is; we do not have variables in our language for, say, binary relations on positions (we do not use full SOL).

This system is called monadic second-order logic (with less-than), written $\mathrm{MSO}[<]$.

We need some notion of satisfaction

$$w \models \varphi$$

where w is a word and φ a sentence in $\mathrm{MSO}[<]$.

We won't give a formal definition, but the basic idea is simple: Let $|w| = n$:

- the first order variables range over $[n] = \{1, 2, \ldots, n\}$,
- the second-order variables range over $\mathfrak{P}([n])$.

The basic predicates $x < y$ and $x = y$ have their obvious meaning. For the $\mathbf{a}(x)$ predicate we let

$$\mathbf{a}(x) \iff w_x = a$$

$$aaacbbb \models \forall\, x\, (\mathbf{a}(x) \vee \mathbf{b}(x) \vee \mathbf{c}(x))$$

$$aaabbb \models \exists\, x, y\, (x < y \wedge \mathbf{a}(x) \wedge \mathbf{b}(y))$$

$$bbbaaa \not\models \exists\, x, y\, (x < y \wedge \mathbf{a}(x) \wedge \mathbf{b}(y))$$

$$aaabbb \models \exists\, x, y\, (x < y \wedge \neg\exists\, z\, (x < z \wedge z < y) \wedge \mathbf{a}(x) \wedge \mathbf{b}(y))$$

$$aaacbbb \not\models \exists\, x, y\, (x < y \wedge \neg\exists\, z\, (x < z \wedge z < y) \wedge \mathbf{a}(x) \wedge \mathbf{b}(y))$$

$$aaacbbb \models \exists\, x\, (\mathbf{c}(x) \Rightarrow \forall\, y\, (x < y \Rightarrow \mathbf{b}(y)))$$

If you are worried about how exactly these structures work, here is the idea. Suppose $\Sigma = \{a, b\}$.

$$\mathcal{A} = \langle [n]; \mathbf{a}, \mathbf{b}, < \rangle$$

where the unary relations $\mathbf{a}$ and $\mathbf{b}$ determine the letters, and $<$ is the usual order on $[n]$.

Note that certain formulae are automatically valid:
$\forall\, x\, (\mathbf{a}(x) \oplus \mathbf{b}(x))$ $\quad$ $\exists\, x \forall\, y\, (x < y \vee x = y)$

For the second example, we have to rule out the empty structure corresponding to the empty word (this is standard in model theory).

In applications, the atomic relation $x < y$ is slightly more useful than $y = x + 1$, but either one would have the same expressiveness.

On the one hand

$$y = x + 1 \iff x < y \wedge \forall\, z\, (x < z \Rightarrow y \leq z)$$

On the other hand write $\mathsf{closed}(X)$ for the formula $\forall\, z\, (X(z) \Rightarrow X(z+1))$. Then

$$x < y \iff x \neq y \wedge \forall\, X\, (X(x) \wedge \mathsf{closed}(X) \Rightarrow X(y))$$

This is sometimes written as $\mathrm{MSO}[<] = \mathrm{MSO}[+1]$.

Example

We can hardwire factors. For example, to obtain a factor abc let

$$\varphi \equiv \exists\, x, y, z\, (y = x + 1 \wedge z = y + 1 \wedge \mathbf{a}(x) \wedge \mathbf{b}(y) \wedge \mathbf{c}(z))$$

Then $w \models \varphi$ iff $w \in \Sigma^\star abc \Sigma^\star$.

Example

Scattered subwords are very similar in this setting:

$$\varphi \equiv \exists\, x, y, z\, (x < y \wedge y < z \wedge \mathbf{a}(x) \wedge \mathbf{b}(y) \wedge \mathbf{c}(z))$$

Then $w \models \varphi$ iff $w \in \Sigma^\star a \Sigma^\star b \Sigma^\star c \Sigma^\star$.

Example

We can split a word into two parts as in

$$\varphi \equiv \exists\, x \forall\, y\, ((y \leq x \Rightarrow \mathbf{a}(y)) \wedge (y > x \Rightarrow \mathbf{b}(y))) \vee \forall\, x\, (\mathbf{b}(x))$$

Then $w \models \varphi$ iff $w \in a^\star b^\star$.

Example

Let $\mathsf{first}(x)$ be shorthand for $\forall\, z\, (x \leq z)$, and $\mathsf{last}(x)$ shorthand for $\forall\, z\, (x \geq z)$. Then

$$\varphi \equiv \exists\, x, y\, (\mathsf{first}(x) \wedge \mathbf{a}(x) \wedge \mathsf{last}(y) \wedge \mathbf{b}(y))$$

Then $w \models \varphi$ iff $w \in a\Sigma^\star b$.

The examples suggest that, for any sentence φ, we should consider the collection of all words that satisfy φ:

$$\mathcal{L}(\varphi) = \{\, w \in \Sigma^\star \mid w \models \varphi \,\}.$$

One cannot fail to notice that, in the examples so far, $\mathcal{L}(\varphi)$ is always regular. Needless to say, this is no coincidence.

Also note that we have not used the second-order part of our language yet.

Example

Write $\mathsf{even}(X)$ to mean that X has even cardinality and consider

$$\varphi \equiv \exists\, X \left(\forall\, x\, (\mathbf{a}(x) \iff X(x)) \wedge \mathsf{even}(X)\right)$$

Then $w \models \varphi$ iff the number of a's in w is even.

We're cheating, of course; we need to show that the predicate $\mathsf{even}(X)$ is definable in our setting. This is tedious but not really hard:

$$\mathsf{even}(X) \iff \exists\, A, B\, (X = A \cup B \wedge \emptyset = A \cap B \wedge \mathsf{alt}(X, A, B))$$

Here $\mathsf{alt}(X, A, B)$ is supposed to express that the elements of A and B strictly alternate as in

$$a_1 < b_1 < a_2 < b_2 < \ldots < a_k < b_k$$

$$
\begin{aligned}
X = A \cup B &\iff \forall\, u \left(X(u) \Leftrightarrow A(u) \oplus B(u) \right) \\
\mathsf{first}_Z(u) &\iff Z(u) \wedge \forall\, z < u \, (\neg Z(z)) \\
\mathsf{last}_Z(u) &\iff Z(u) \wedge \forall\, z > u \, (\neg Z(z)) \\
\mathsf{nxt}_Z(u,v) &\iff Z(u) \wedge Z(v) \wedge \forall\, z \, (u < z < v \Rightarrow \neg Z(z)) \\
\mathsf{alt}(X,A,B) &\iff \exists\, u \, (\mathsf{first}_X(u) \wedge \mathsf{first}_A(u)) \wedge \\
&\qquad \exists\, u \, (\mathsf{last}_X(u) \wedge \mathsf{last}_B(u)) \wedge \\
&\qquad \forall\, u, v \left(A(u) \wedge \mathsf{nxt}_X(u,v) \Rightarrow B(v) \right) \wedge \\
&\qquad \forall\, u, v \left(B(u) \wedge \mathsf{nxt}_X(u,v) \Rightarrow A(v) \right)
\end{aligned}
$$

$$\mathsf{even}(X) \iff \exists\, A, B\, (X = A \uplus B) \wedge \mathsf{alt}(X, A, B)$$

Exercise

As written, the even *formula is not quite right. Fix the bug.*
Can the formula be streamlined?

Exercise

Show that one can check if the number of a's is a multiple of k, for any fixed k.

Definition

A language L is $\mathrm{MSO}[<]$ definable (or simply $\mathrm{MSO}[<]$) if there is some sentence φ such that

$$L = \mathcal{L}(\varphi) = \{\, w \in \Sigma^\star \mid w \models \varphi \,\}.$$

Our examples suggest the following theorem.

Theorem (Buechi 1960, Elgot 1961)

A language is regular if, and only if, it is $\mathrm{MSO}[<]$ *definable.*

The theorem connects complexity with definability: we can recognize a set of strings in constant space if, and only if, the set can be described by a formula in our logic.

Formula to Regular (Sketch)

Obviously, the proof comes in two parts:

- For every regular language L we need to construct a sentence φ such that $L = \mathcal{L}(\varphi)$.
- For every sentence φ we have to show that the language $\mathcal{L}(\varphi)$ is regular.

We should expect part (1) to be harder since there is no good inductive structure to exploit.

Part (2) is by straightforward induction on φ, but there is the usual technical twist: we need to deal not just with sentences but also with free variables. Since we don't have a formal semantics we will not give details of this construction, it is very similar to the argument for automatic structures.

Regular to Formula (Sketch)

We may safely assume that the regular language L is given by a DFA $M = \langle Q, \Sigma, \delta; q_0, F \rangle$.

For simplicity assume $Q = [m]$ and $q_0 = 1$.

We have to construct a formula φ such that $w \models \varphi$ iff M accepts w.

Consider a trace of M on input w of length n (think big)

$$q_0\, w_1\, q_1\, w_2\, q_2 \ldots q_{m-1}\, w_n\, q_n.$$

Here m can be arbitrarily large.

We can think of states as being associated with the letters of the word as in

$$\begin{array}{cccccc} & w_1 & w_2 & w_3 & \ldots & w_n \\ q_0 & q_1 & q_2 & q_3 & \ldots & q_n \end{array}$$

Thus, position $x = 1, \ldots, n$ in the word is associated with state $\delta(q_0, w_1 \ldots w_x)$.

It is convenient to think of the states as colors: so we need to color the letters of the given word by n colors.

This may sound wishy-washy, but it's perfectly precise: we need to partition $[n]$ into m blocks, each representing one color (state). We can do this using m second-order variables:

$$X_1, X_2, \ldots, X_m \qquad \bigcup X_i = [n] \qquad X_i \cap X_j = \emptyset, i < j$$

Of course, the color in position $x + 1$ will depend on the color in position x the symbol w_x as prescribed by the transition function.

$$X_p(x) \iff \delta(q_0, w_1 \dots w_x) = p$$

The critical fact here is that the number of colors is fixed, not matter what the input is. So we only need one formula to pin things down.

Essentially, we are just expressing the transition function in termso of a monadic second-order formula, using m second-order variables.

Technically, this is done by a formula

$$\Phi_{p,a} \equiv \forall\, x \left(X_p(x) \wedge \mathbf{a}(x+1) \Rightarrow X_{\delta(p,a)}(x+1) \right)$$

meaning "if at position x we are in state p and the next letter is an a, then the state in position $x+1$ is $\delta(p,a)$.

Note that this is not quite right, we really need a non-existing position 0 corresponding to state q_0.

Exercise

Figure out how to fix this little glitch. Also figure out how to express "the last state is final."

Now consider the big conjunction of $\Phi_{p,a}$ where $p \in Q$ and $a \in \Sigma$. Add formulae that pin down the first and last state to arrive at a formula of the form

$$\varphi \equiv \exists\, X_1, \ldots, X_n\, \Psi$$

where Ψ is first-order as indicated above. □

Note that in conjunction with the opposite direction of Büchi's theorem, this result has the surprising consequence that every $\mathrm{MSO}[<]$ formula is equivalent to a $\mathrm{MSO}[<]$ formula containing only one block of existential second-order quantifiers. With more work, one can get this down to just a single existential quantifier.

Exercise

Fill in all the details in the last proof. Try to find the reduction to a single quantifier.

How about the opposite direction?

Given a sentence φ in $\mathrm{MSO}[<]$, we need to construct an automaton $\mathcal{A}_\varphi$ that accepts $\mathcal{L}(\varphi)$.

As usual, $\mathcal{A}_\varphi$ is built by induction on the build-up of the formula. One has to confront the problem of free variables: when $\varphi(x_1, \ldots, x_n)$ has n free variables, we have

$$\mathcal{L}(\varphi) \subseteq \Sigma^n$$

In the special case when φ is a sentence we are essentially dealing with a path existence problem in the remaining unlabeled digraph. If you prefer, think of Σ^0 as a one-point set.

Hence, we can solve the model checking problem for words and $\mathrm{MSO}[<]$.

For example, consider the (silly) formula

$$\varphi \equiv \exists\, X, x \left(X(x) \wedge \mathbf{b}(x) \right)$$

We add extra tracks and work over the alphabet $\{a, b\} \times \mathbf{2} \times \mathbf{2}$.

Then all we need is an automaton $\mathcal{A}_0$ that handles the matrix $X(x) \wedge \mathbf{b}(x)$ (we can project away the tracks for the existential quantifiers).

W	a	a	b	a	a	a	b	a	$\ldots$	a	a
x	0	0	1	0	0	0	0	0	$\ldots$	0	0
X	0	0	1	0	1	0	1	0	$\ldots$	0	1

Note that the automaton must enforce a special condition on the binary tracks corresponding to a first-order variable: it must contain exactly one 1, which indicates a unique position in W. In other words, the track must hold a word in $0^\star 1 0^\omega$.

The binary word in a second order track is arbitrary, think of it as the characteristic function of the set.

Note that one can often construct automata that easily check conditions that are difficult to express in terms of formulae, the even/even language from above is a perfect example.

It is natural to ask whether the languages defined by the first-order fragment of $\mathrm{MSO}[<]$ have some natural characterization.

A language $L \subseteq \Sigma^\star$ is star-free iff it can be generated from $\emptyset$ and the singletons $\{a\}$, $a \in \Sigma$, using only operations union, concatenation and complement (but not Kleene star).

Note well: $a^\star b^\star a^\star$ is star-free. More interesting is the following star-free set:

$$(ab)^\star = \left(b\,\emptyset^c + \emptyset^c a + \emptyset^c(aa + bb)\,\emptyset^c\right)^c$$

By contrast, $(aa)^\star$ fails to be star-free.

Let $K, L \subseteq \Sigma^\star$. K splits L if both $L \cap K$ and $L - K$ are infinite.

Lemma

Let $K \subseteq a^\star$ star-free. The K does not split $a^\star$.

Proof.

By induction on the buildup of K.

□

While we're at it: star-free languages are quite interesting since they admit a purely algebraic characterization in terms of their syntactic semigroups.

A semigroup is aperiodic if it contains only trivial subgroups (the idempotents of the semigroup).

Theorem (Schützenberger 1965)

A regular language is star-free if, and only if, its syntactic semigroup is aperiodic.

Theorem

A language $L \subseteq \Sigma^\star$ is $\mathrm{FOL}[<]$ definable if, and only if, L is star-free.

This is no accident: There are lots of other examples in complexity theory where some natural class of languages corresponds exactly to a particular logic.

The Büchi/Elgot theorem establishes a connection between a very low complexity class (constant space) and $\mathrm{MSO}[<]$. In fact, there is the whole area of descriptive complexity that characterizes complexity classes in terms of logic and finite structures:

- $\mathbb{NP}$ corresponds to existential SOL (Fagin 1974).
- PH corresponds to SOL.
- PSPACE corresponds to SOL plus a transitive closure operator.

Exercise

Figure out the details of Fagin's theorem.

Another Wild and Wolly Idea:

Does this also make sense for infinite words?

If we think of an automaton as processing the input by reading it from beginning to end, there is an obvious problem: the infinite word never ends. Surprisingly, this is not really an issue.

Our putative structures are now words

$$W = a_0 a_1 a_2 \dots a_n a_{n+1} \dots$$

and positions correspondingly range over $\mathbb{N}$. Other than that, we copy all the definitions from the finite case.

Simple Example, Again

For example, consider the (silly) formula

$$\varphi \equiv \exists\, X, x \left(X(x) \wedge \mathbf{b}(x) \right)$$

As before, we add extra tracks and work over the alphabet $\{a, b\} \times \mathbf{2} \times \mathbf{2}$.

W	a	a	b	a	a	a	b	a	a	a	$\ldots$
x	0	0	1	0	0	0	0	0	0	0	$\ldots$
X	0	0	1	0	1	0	1	0	0	0	$\ldots$

Note the ellipses have moved to the end

Again, a binary track corresponding to a first-order variable must contain exactly one 1, which indicates a unique position in W. In other words, the track must hold a word in $0^\star 10^\omega$.

The binary word in a second order track is arbitrary in principle, think of it as the characteristic function of the set $X \subseteq \mathbb{N}$.

In fact, it will turn out to be very useful to be able to force the set X to be finite, by requiring the track to be in $\mathbf{2}^\star 0^\omega$.

We need to develop automata that can handle these tasks.