The Logical Approach to Automatic Sequences
Part 3: Proving Claims about Automatic Sequences with Walnut

Jeffrey Shallit
School of Computer Science
University of Waterloo
Waterloo, Ontario N2L 3G1
Canada
shallit@cs.uwaterloo.ca
https://cs.uwaterloo.ca/~shallit
As we will see, the logical theory we discussed yesterday is powerful enough to express many assertions about automatic sequences.

Luckily, Hamoon Mousavi has created a Java prover Walnut that implements the decision procedure discussed yesterday, and it is publicly available.

There is also a software manual available that describes its use.

Today, we’ll look at a variety of properties of automatic sequences and prove them using Walnut.
We can write a formula for ultimate periodicity of a sequence $S$ as follows:

$$\exists n \geq 0 \ \exists p \geq 1 \ \forall j \geq n \ S[j] = S[j + p].$$

When we translate this to Walnut:

– we need not specify $n \geq 0$ explicitly, as this is implicit in the domain $\mathbb{N}$:

– we translate $\exists p \geq 1 \cdots$ to

$$\mathsf{Ep} (p \geq 1) \& \ldots$$

– we translate $\forall j \geq n \cdots$ to

$$\mathsf{Aj} (j \geq n) \Rightarrow \ldots$$
Checking ultimate periodicity of the Thue-Morse sequence

cd Walnut/bin
java Main.prover
eval tmup "En Ep (p>=1) & Aj (j >= n) => T[j] = T[j+p]":

Now go and check the file tmup.txt in the directory Walnut/Result, and it says “false”.

So the Thue-Morse sequence is not ultimately periodic.
Theorem. If a $k$-DFAO of $n$ states generates an ultimately periodic sequence $S$, then the preperiod and period are bounded by $k^3 \cdot 2^{4n^2}$.

Proof. We can make a DFA accepting those $(j, l)_k$ such that $S[j] = S[l]$ in $n^2$ states. We can enforce $(j \geq n) \land (l = j + p)$ using a total of $4n^2$ states. Checking $\forall j$ requires some nondeterminism and another negation, giving $2^{4n^2}$. Finally, checking $p \geq 1$ takes 3 states, so $3 \cdot 2^{4n^2}$ states. Such an automaton, if it accepts anything at all, must accept $p$ and $n$ having at most $3 \cdot 2^{4n^2}$ symbols.

Better results: Honkala, Sakarovitch, etc.
Squares

We can write a formula for the orders of squares in a sequence $S$ as follows:

$$(n > 0) \land \exists i \forall j \ (j < n) \implies S[i + j] = S[i + j + n]$$

In Walnut, for the Thue-Morse sequence, this is done with the command

eval tmsq "(n>0) & Ei Aj (j<n) => T[i+j] = T[i+j+n]":

Then we go and look in the Result directory for tmsq.gv.

Thus there are squares of order $2^n$ and $3 \cdot 2^n$ for all $n \geq 0$ in the Thue-Morse sequence. Where are they?
Squares

We can write a formula for the positions and orders of squares in a sequence $S$ as follows:

$$(n > 0) \land \forall j \ (j < n) \implies S[i + j] = S[i + j + n]$$

In Walnut, for the Thue-Morse sequence, this is

```
eval tmsqp "(n>0) & Aj (j<n) => T[i+j] = T[i+j+n]";
```

Then we go and look in the Result directory for tmsqp.gv.
We can write a formula for the orders and positions of overlaps in a sequence $S$ as follows:

$$(n \geq 1) \land \forall j \ (j \leq n) \implies S[i + j] = S[i + j + n]$$

When we do this in Walnut for the Thue-Morse sequence we type:

eval tmover 

which gives an automaton that accepts nothing.
Arbitrary fractional powers

Fractional powers are generalizations of integer powers.

We say a string $x$ is a $(\ell/p)$-power if it is of length $\ell$ and has period $p$.

For example, ionization is a $(10/7)$-power.

We say a word $w$ avoids $\alpha$ powers, for $\alpha > 1$ a real number, if $w$ has no factor that is a $(\ell/p)$-power for $(\ell/p) \geq \alpha$.

We say a word $w$ avoid $\alpha^+$ powers if $w$ has no factor that is a $(\ell/p)$-power for $(\ell/p) > \alpha$.

Thus, avoiding squares is avoiding 2-powers, and avoiding overlaps is avoiding $2^+$-powers.
Arbitrary fractional powers

We can write a formula for a word $S$ avoiding $\alpha$-powers:

$$ \neg (\exists i \, \exists n \, (n \geq 1) \land \forall j \,(j + n < \alpha n) \implies S[i + j] = S[i + j + n]) $$

or avoiding $\alpha^+$-powers:

$$ \neg (\exists i \, \exists n \, (n \geq 1) \land \forall j \,(j + n \leq \alpha n) \implies S[i + j] = S[i + j + n]) $$

In order for this to be expressible in our logical theory, we must have $\alpha = \ell/p$ for some integers $\ell, p$. Then we rewrite

$$ j + n < \alpha n \quad \text{as} \quad \ell j < (p - \ell)n $$

and

$$ j + n \leq \alpha n \quad \text{as} \quad \ell j \leq (p - \ell)n. $$
Example: the Leech sequence:

0 → 0121021201210;  1 → 120210201201201;  2 → 2010210120102

This sequence avoids \((15/8)^+\) powers and has infinitely many \((15/8)\)-powers. We can create a file named LE.txt in the Word Automata directory that implements this morphism. Then we say eval le158 "?msd.13 Ei (n>1) & Aj (8*j < 7*n) => LE[i+j] = LE[i+j+n]": After a reasonable delay we get the automaton

\[
\begin{array}{c}
(0) \\
\circlearrowright
\end{array}
\quad
\begin{array}{c}
(0) \\
\circlearrowright
\end{array}
\]

\[
\begin{array}{c}
0 \\
\rightarrow (8) \\
\rightarrow 1
\end{array}
\]

which says that there are powers \(x^{15/8}\) for \(|x| = 8 \cdot 13^i\) and \(i \geq 0\).
Antisquares

Antisquares are binary words of the form $x\overline{x}$, where $\overline{x}$ means change 0 to 1 and 1 to 0.

A formula for lengths of antisquares:

$$E_i \ (n \geq 1) \land \forall j \ (j < n) \implies S[i + j] \neq S[i + j + n].$$
Let's compute antisquare orders for the Rudin-Shapiro sequence in Walnut:

\[ E_i \text{ (} n \geq 1 \text{)} \& A_j \text{ (} j < n \text{)} \Rightarrow RS[i+j] \neq RS[i+j+n] \]

This gives the following orders of antisquares: \(2^i\) for \(i \geq 0\) and 3 and 5.
We can write a formula for the positions of palindromes in a sequence $S$ as follows:

$$\exists i \ \forall j \ (j < n) \implies S[i + j] = S[(i + n) - (j + 1)]$$

When we do this for the Rudin-Shapiro sequence we get

So the only palindrome lengths in Rudin-Shapiro are

$$\{0, 1, 2, 3, 4, 5, 6, 7, 8, 10, 12, 14\}.$$
Maximal palindromes

A maximal palindrome in a word $S$ is a palindrome $x$ such that $axa$ does not appear in $S$ for any $a$.

Formula:

$$\exists i \ (\forall j \ (j < n) \implies S[i+j] = S[(i+n)-(j+1)]) \land$$

$$\forall l (((l > 0) \land \forall m \ (m < n) \implies S[l+m] = S[i+m])) \implies (S[l-1] \neq S[l+n]))$$
Maximal palindromes

When we do this for the Thue-Morse sequence, we get only the lengths $3 \cdot 4^i$ for $i \geq 0$.

![Diagram]

**Exercise:** How could you use Walnut to prove that the only maximal palindromes in the Thue-Morse sequence are $\mu^{2n}(010)$ and $\mu^{2n}(101)$ for $n \geq 0$?
Reversal-freeness

Formula:

\[ \forall i \forall j \exists k \ (k < n) \land S[i + k] \neq S[(j + n) - (k + 1)]. \]

This says that for any word of length \( n \) beginning at position \( i \), all other words beginning at all positions \( j \) have their reversal \( S[j \ldots j + n - 1] \) differing at some position \( k \) from \( S[i \ldots i + n - 1] \).

Here is an example: take the morphism defined by

\[
0 \to 0001011; \quad 1 \to 0010111.
\]

This gives a 7-automatic sequence. We encode this in a file RSR.txt. Then we use the Walnut command:
?msd_7 Ai Aj Ek (k<n) & RSR[i+k] != RSR[(j+n)-(k+1)]
and we get the automaton below. So the only lengths for which words and their reversals are both present are 0, 1, 2, 3, 4, 5.
Recurrence

Recall: $x$ is recurrent if every factor that occurs, occurs infinitely often.

Equivalent to: for every factor that occurs, there is another occurrence at a higher index. Formula

$$\forall i \ \forall n \ \exists j \ (j > i) \land (\forall l (l < n) \implies S[i + l] = S[j + l]).$$
When we run
\[ A_i \land A_n \land E_j \land ((j>i) \land (A_l (l<n) \Rightarrow T[i+l] = T[j+l])) \]
in Walnut for the Thue-Morse sequence we get the automaton

which is Walnut’s way to represent an automaton that accepts everything with 0 free variables.
A nonempty word $x$ is bordered if there is a nonempty word $w$ and a possibly empty word $t$ such that $x = wtw$. For example, ionization is bordered with border ion.

Formula for $S[i..i + n - 1]$ being bordered:

$$\exists l \ (0 < l) \land (l < n) \land (\forall j \ (j < l) \implies S[i + j] = S[(i + n + j) - l])$$

In Walnut we can define a macro for this:

def tmbord "El (0<l) & (l<n) & (Aj (j<l) => T[i+j]=T[(i+n+j)-l] )":

and then use it by saying

eval tmborders "Ei $tmbord(i,n)":

or

eval tmunbord "Ei ~$tmbord(i,n)":
When we do this for Thue-Morse we get

for the lengths of bordered factors of Thue-Morse, which shows that there is a bordered factor for all lengths $> 1$. 
Unbordered factors

When we do this for unbordered factors we get

for the lengths of unbordered factors of Thue-Morse.

So we have proved: there is an unbordered factor of length $n$ of the Thue-Morse sequence iff $(n)_2 \not\in 1(01^*0)^*10^*1$. This improves a 2009 result due to Currie and Saari; they proved $t$ has an unbordered factor of length $n$ if $n \not\equiv 1 \pmod{6}$. 
Balanced words

A word $x$ is balanced if $||y|_a − |z|_a| ≤ 1$ for all equal-length factors $y, z$ of $x$ and all letters $a$.

It is not clear how to state this in first-order logic.

Luckily there is an alternative characterization, which is often quoted as

$x$ is unbalanced iff there exists a palindrome $p$ such that both $0p0$ and $1p1$ are both factors of $x$.

But an even simpler characterization is

$x$ is unbalanced iff there exists a word $y$ (not necessarily a palindrome) such that both $0y0$ and $1y1$ are both factors of $x$.

These two characterizations are easily seen to be equivalent.
Balanced words

Here is a formula for unbalanced factors of length $n$:

$$(n \geq 2) \land \exists i \exists j \left( \forall k ((0 < k) \land (k + 1 < n)) \implies (S[i + k] = S[j + k]) \land S[i] = 0 \land S[j] = 1 \land S[i + n - 1] = 0 \land S[j + n - 1] = 1 \right)$$

In Walnut this is

$$(n \geq 2) \land E_i E_j (A_k ((0 < k) \land (k + 1 < n)) \implies S[i + k] = S[j + k]) \land S[i] = 0 \land S[i + n - 1] = 0 \land S[j] = 1 \land S[j + n - 1] = 1$$
We can count the number of distinct palindromes occurring in a word.

For example, the word Mississippi has the following distinct nonempty palindromes in it:

\[M, i, s, p, ss, pp, sis, issi, ippi, ssiss, ississi\]

**Theorem.** Every word of length \(n\) contains, as factors, at most \(n\) distinct palindromes.

**Proof.** For each index \(p\) of a word \(w\), consider the palindromes ending at this index. Suppose at least two palindromes, \(x\) and \(y\) occur for the first time ending at \(p\). Then wlog \(|x| < |y|\). So then \(x\) is a suffix of \(y\), so \(x^R = x\) is a prefix of \(y\), contradicting the claim that \(x\) occurred for the first time ending at \(p\).

So at each position \(p\) at most 1 new palindrome can end.
Rich words

We say that a length-$n$ word is rich if it contains, as factors, exactly $n$ distinct nonempty palindromes.

We can therefore make a formula for the factor $S[i..i+n-1]$ being rich as follows: at each position $p$ there is a palindrome ending at $p$ that doesn’t occur earlier in that factor.

Exercise. Write a predicate for richness and test it on the Thue-Morse sequence. You should find that there are no rich factors of length $> 16$.

Exercise. Find a 2-automatic sequence where all factors are rich, and prove it using Walnut.
A nonempty word \( w \) is *primitive* if it cannot be written as \( x^{e} \) with \( e \geq 2 \). So a primitive word is a non-power.

It’s easy to see that a word \( w \) is a nontrivial power if and only if there is some cyclic shift (by \( 0 < j < |w| \) positions) of \( w \) that is equal to \( w \). So we can write a formula for \( S[i..i + n - 1] \) being a power as follows:

\[
\exists j, \ 0 < j < n, \ ((\forall t < n - j \ S[i + t] = S[i + j + t]) \land
\ (\forall u < j \ S[i + u] = S[i + n + u - j]))
\]

A formula for being primitive is just the negation of this.
The “substitute variables” trick

Recall our formula for primitivity:

\[
\neg \exists j, \ 0 < j < n,\ ((\forall t < n - j\ S[i + t] = S[i + j + t]) \land \ (\forall u < j\ S[i + u] = S[i + n + u - j]))
\]

This formula is correct, but indexing the automatic sequence by four variables (as in \(i + n + u - j\)) could be prohibitively expensive for our algorithm when the underlying automaton has many states.

To reduce the running time, use the substitution of variables \(t' = i + t\) and \(u' = i + u + n\) to get

\[
\neg \exists j, \ 0 < j < n,\ ((\forall t',\ i \leq t' < n + i - j,\ S[t'] = S[t' + j]) \land \ (\forall u',\ n + i \leq u' < n + i + j,\ S[u' - n] = S[u' - j]))
\]

This one is about twice as fast for the Thue-Morse sequence.
A word $x$ is privileged if is of length $\leq 1$, or it has a border $w$ with $|x|_w = 2$ that is itself privileged. For example, abracadabra has a border abra that appears only at the beginning and end. And abra has a border a that occurs only at the beginning and end. Finally, a is privileged, and so is abra and so is abracadabra.

As stated it is not obvious that we can state this property in first-order logic.

However, there is another way to state the property (due to Luke Schaeffer): a word is privileged if for all $n$ with $1 \leq n < |w|$ there exists a word $x$ of length $\leq n$ such $x$ is a border of $w$ and there is exactly one occurrence of $x$ in the first $n$ symbols of $w$ and one occurrence of $x$ in the last $n$ symbols of $w$.

**Exercise:** write a predicate for the privileged property, and run it on the Thue-Morse word.
A word $x$ is called closed if it is of length $\leq 1$, or if it has a border $w$ with $|x|_w = 2$.

For example, alfalfa is a closed word because of the border alfa. On the other hand, although academia is bordered, it is not closed.

**Theorem.** There is a closed factor of the Thue-Morse word $t$ of every length.
Arbitrarily large common factors between two $k$-automatic sequences:

$$\exists i \exists j \forall k \ (k < n) \implies R[i + k] = S[j + k]$$

If two $k$-automatic sequences, generated by automata of $s$ and $t$ states, respectively, have a factor of length $\ell > ...$ in common, then they have arbitrarily long factors in common.
A real number $x$ is said to be $k$-automatic in base-$b$ if its base-$b$ expansion mod 1 is generated by a $k$-DFAO. The set of all such numbers is written $L(k, b)$.

Example: the Thue-Morse real number

$$0.0110100110010110 \cdots$$

is 2-automatic in base 10.

Exercise: how can we show that $L(k, b)$ forms a $\mathbb{Q}$-vector space? The difficulty comes because carries can come from arbitrarily far to the right.
The critical exponent of a word $w$ is the supremum, over all factors $x$ of $w$, of the exponent of $x$.

The critical exponent of the Thue-Morse word $t$ is 2.
Representing rational numbers

- Represent rational number $\alpha = p/q$ by pair of integers $(p, q)$, represented in base $k$; pad shorter with leading zeroes.
- So representations of rationals are over the alphabet $\Sigma_k \times \Sigma_k$.
- For example, if $w = [3, 0][5, 0][2, 4][6, 1]$ then $[w]_{10} = (3526, 41)$.
- Define $\text{quo}_k(x) = [\pi_1(x)]_k/[\pi_2(x)]_k$, where $\pi_i$ is the projection onto the $i$’th coordinate.
- So $\text{quo}_{10}(w) = 3526/41 = 86$.
- Canonical representations lack leading $[0, 0]$’s.
- Every rational has infinitely many canonical representations, e.g., as $(1, 2), (2, 4), (3, 6), \ldots$, etc.
Automatic sets over $\mathbb{Q}_{\geq 0}$

- $\text{quo}_k(L) = \bigcup_{x \in L} \{\text{quo}_k(x)\}$

- $A \subseteq \mathbb{Q}_{\geq 0}$ is a $k$-automatic set of rationals if $A = \text{quo}_k(L)$ for some regular language $L \subseteq (\Sigma_k \times \Sigma_k)^*$.

- *not* the same notion as the automatic reals of Boigelot, Brusten, and Bruylère
Example 1. Let $k = 2$, $B = \{[0, 0], [0, 1], [1, 0], [1, 1]\}$, and consider

$$L_1 := B^* \{[0, 1], [1, 1]\} B^*.$$ 

Then $L_1$ consists of all pairs of integers where the second component has at least one nonzero digit — the point being to avoid division by 0. Then $\text{quo}_2(L) = \mathbb{Q}_{\geq 0}$, the set of all non-negative rational numbers.

Example 2. Consider

$$L_2 = \{w \in (\Sigma_k^2)^* : \pi_1(w) \in 0^* C_k \text{ and } \pi_2(w) \in 0^* 1\}.$$ 

Then $\text{quo}_2(L_2) = \mathbb{N}$. 
Examples

**Example 3.** Let $k = 3$, and consider the language

$$L_3 := [0, 1][[0, 0], [2, 0]]^*.$$

Then $\text{quo}_3(L_3)$ is the $3$-adic Cantor set, the set of all rational numbers in the “middle-thirds” Cantor set with denominators a power of 3.

**Example 4.** Let $k = 2$, and consider

$$L_4 := [0, 1][[0, 0], [0, 1]]^*[[1, 0], [1, 1]].$$

Then the numerator encodes the integer 1, while the denominator encodes all positive integers that start with 1. Hence

$$\text{quo}_2(L_4) = \left\{ \frac{1}{n} : n \geq 1 \right\}.$$
Example 5. Let $k = 4$, and consider

$$S := \{0, 1, 3, 4, 5, 11, 12, 13, \ldots\}$$

of all non-negative integers that can be represented using only the digits 0, 1, −1 in base 4. Consider the language

$$L_5 = \{ (p, q)_4 : p, q \in S \}.$$ 

It is not hard to see that $L_5$ is $(\mathbb{Q}, 4)$-automatic. The main result in Loxton & van der Poorten [1987] can be rephrased as follows: $\text{quo}_4(L_5)$ contains every odd integer. In fact, an integer $t$ is in $\text{quo}_4(L_5)$ if and only if the exponent of the largest power of 2 dividing $t$ is even.
Example 6. Consider

\[ L_6 = \{ w \in (\Sigma_k^2)^* : \pi_2(w) \in 0^*1^+0^* \}. \]

An easy exercise using the Fermat-Euler theorem shows that that \( \text{quo}_2(L_6) = \mathbb{Q}^{\geq 0} \).
Example 7. For a word $x$ and letter $a$ let $|x|_a$ denote the number of occurrences of $a$ in $x$. Consider the regular language

$$L_7 = \{ w \in (\Sigma_2^2) : |\pi_1(w)|_1 \text{ is even and } |\pi_2(w)|_1 \text{ is odd} \}.$$ 

Then it follows from a result of Schmid [1984] that

$$\text{quo}_2(L_7) = \mathbb{Q}^{\geq 0} - \{ 2^n : n \in \mathbb{Z} \}.$$
Basic decidability properties

Given a DFA $M$ accepting a language $L$ representing a set of rationals $S$, can decide

- if $S = \emptyset$
- given $\alpha \in \mathbb{Q}^{\geq 0}$, whether there exists $x \in S$ with $x = \alpha$ (resp., $x < \alpha$, $x \leq \alpha$, $x > \alpha$, $x \geq \alpha$, $x \neq \alpha$, etc.)
- if $|S| = \infty$
- given a finite set $F \subseteq \mathbb{Q}^{\geq 0}$, if $F \subseteq S$ or if $S \subseteq F$
- given $\alpha \in \mathbb{Q}^{\geq 0}$, if $\alpha$ is an accumulation point of $S$
sup $A$ is rational or infinite

Given a DFA $M$ accepting $L \subseteq (\Sigma_k \times \Sigma_k)^*$ representing a set of rationals $A \subseteq \mathbb{Q}_{\geq 0}$, what can we say about sup $A$?

**Theorem.** sup $A$ is rational or infinite, and is computable.

**Proof ideas:** quo$_k(uv^iw)$ forms a monotonic sequence. Defining

$$\gamma(u, v) := \frac{[\pi_1(uv)]_k - [\pi_1(u)]_k}{[\pi_2(uv)]_k - [\pi_2(u)]_k}$$

one of the following three cases must hold:

(i) quo$_k(uw) < $ quo$_k(uvw) < $ quo$_k(uv^2w) < \cdots < U$ ;
(ii) quo$_k(uw) = $ quo$_k(uvw) = $ quo$_k(uv^2w) = \cdots = U$ ;
(iii) quo$_k(uw) > $ quo$_k(uvw) > $ quo$_k(uv^2w) > \cdots > U$ .

Furthermore, $\lim_{i \rightarrow \infty}$ quo$_k(uv^iw) = U$. 
It follows that if \( \sup A \) is finite, and the DFA \( M \) has \( n \) states, then
\[
\sup A = \max T,
\]
where
\[
T = T_1 \cup T_2
\]
and
\[
T_1 = \{ \text{quo}_k(x) : |x| < n \text{ and } x \in L \};
\]
\[
T_2 = \{ \gamma(u, v) : |uv| \leq n, |v| \geq 1, \delta(q_0, u) = \delta(q_0, uv), \text{ and there exists } w \text{ such that } uvw \in L \}.
\]
$\sup A$ is computable

We know that $\sup A$ lies in the finite computable set $T$.

For each of $t \in T$, we can check to see if $t \geq \sup A$ by checking if $A \cap (t, \infty)$ is empty.

Then $\sup A$ is the least such $t$. 
Computing the critical exponent

- Previously known to be computable for fixed points of uniform morphisms (Krieger)

**Theorem.** If \( w \) is a \( k \)-automatic sequence, then its critical exponent is rational or infinite. Furthermore, it is computable from the DFAO \( M \) generating \( w \).

**Proof sketch.** Given \( M \), we can transform it into another automaton \( M' \) accepting

\[ \{(m, n) : \text{there exists } i \geq 0 \text{ such that } w[i..i+m-1] \text{ has period } n\} . \]

We then apply our algorithm for computing sup(quo\(_k\)(L)) to \( L(M') \).
Open Questions

- Extend these ideas to morphic sequences (fixed points of possibly non-uniform morphisms, followed by a coding)
  - Some ideas are extendable to, e.g., the Fibonacci word
  - Carton & Thomas proved that \((\mathbb{N}, <, \text{morphic word})\) is decidable

- Which predicates for automatic sequences (like squarefreeness) are decidable in polynomial time? Leroux has proved it for ultimate periodicity.
More open problems

- Extend these ideas to “infinite state” automata (i.e., fixed points of morphism like $n \rightarrow (an + b, cn + d)$) or prove undecidability.

- Is $\sup\{x/y : (x, y)_k \in L\}$ computable for context-free languages $L$?

- Given a regular language $L \subseteq (\Sigma_k \times \Sigma_k)^*$ representing a set $S \subseteq \mathbb{N} \times \mathbb{N}$ of pairs of natural numbers, is it decidable if $S$ contains a pair $(p, q)$ with $p \mid q$?
  - This is a question of $\exists^1(\mathbb{N}, +, V_k, |)$; of course $\text{Th}(\mathbb{N}, +, |)$ is undecidable and $\exists^1(\mathbb{N}, +, |)$ is decidable (Lipshitz).
More Open Questions

Prove or disprove: if $L$ is a regular language with $\text{quo}_k(L) = \mathbb{Q}^\geq 0$, then $L$ contains infinitely many distinct representations for infinitely many distinct rational numbers.

Which of the following questions is decidable? Given $L$ representing a set of rationals $S$,

- Is there some rational $p/q \in S$ having infinitely many distinct representations in $L$?
- Are there infinitely many distinct rationals $p/q \in S$ having infinitely many distinct representations in $L$?