Undecidable Problems in Algebra

From Turing Machines to Algebras

Matthew Moore

The University of Colorado at Boulder

March 15, 2011

We describe a method by which each Turing machine, $\mathcal{T}$, is encoded in a finite algebra, $\mathrm{A}(\mathcal{T})$. The algebra $\mathrm{A}(\mathcal{T})$ will be such that $\mathrm{A}(\mathcal{T})$ possesses certain properties if and only if $\mathcal{T}$ halts, thus showing that these properties are undecidable in general.

Last time...

- Turing machines: A theoretical machine consisting of a tape, a reading head, and a program consisting of 5-tuples of the form (α, r, w, D, β) where α, β are states, $r, w \in \{0, 1\}$, and $D \in \{\mathrm{L}, \mathrm{R}\}$. Meant to be interpreted as "if in state α reading r, write w, move D, and enter state β."
- The Church-Turing Thesis: Any effectively calculable function is a computable function.
- The halting problem: given a Turing machine $\mathcal{T}$ and an input tape, n, decide if $\mathcal{T}(n)$ halts. This problem is uncomputable (undecidable).
- Thus, if we show that

$$\forall \mathcal{T}\,[\mathrm{A}(\mathcal{T}) \text{ has } P \Leftrightarrow \mathcal{T} \text{ halts}]\,,$$

then we have shown that P is undecidable.

Configurations

Definition

Given a Turing machine, $\mathcal{T}$, a *configuration* of $\mathcal{T}$ is a triple $\mathcal{Q} = (t, n, \gamma)$, where $t : \mathbb{Z} \to \{0, 1\}$ is a tape, $n \in \mathbb{Z}$ is the position of the reading head on the tape, and γ is a state of $\mathcal{T}$. If the line $(\gamma, t(n), w, D, \gamma')$ appears in the program of $\mathcal{T}$, we write $\mathcal{T}(\mathcal{Q}) = (t', n \pm 1, \gamma')$, where t' is the modified tape and $n \pm 1$ is determined by D.

$\cdots$	$t(n-1)$	$t(n)$	$t(n+1)$	$t(n+2)$	$t(n+3)$	$\cdots$

$$\boxed{(\gamma, t(n), w, R, \gamma')}$$

$\cdots$	$t(n-1)$	w	$t(n+1)$	$t(n+2)$	$t(n+3)$	$\cdots$

$$\boxed{(\gamma', t(n+1), \ldots)}$$

$$\mathcal{T}(t, n, \gamma) = (t', n+1, \gamma')$$

The Configuration Algebra

- The set of all possible configurations of $\mathcal{T}$ together with the unary partial operation $\mathcal{T}(\cdot)$ is called the *configuration algebra* of $\mathcal{T}$.

- The configuration algebra is finite if and only if $\mathcal{T}$ halts.

- We will define $\mathrm{A}(\mathcal{T})$ such that $\mathbf{B} \leq \mathrm{A}(\mathcal{T})^X$ encodes the configuration algebra as certain subsets of $\mathbf{B}$.

The underlying set of A($\mathcal{T}$)

- Let $U = \{1, 2, H\}$ and $W = \{C, D, \overline{C}, \overline{D}\}$. Let
$$A = \{0\} \cup U \cup W.$$

- Let $\mu_0, \ldots, \mu_k$ be the states of $\mathcal{T}$, with μ_0 the halting state and μ_1 the starting state.

- Let
$$V_{ir}^{w} = \{C_{ir}^{w}, D_{ir}^{w}, M_i^r, \overline{C_{ir}^{w}}, \overline{D_{ir}^{w}}, \overline{M_i^r}\}$$
for $0 \leq i \leq k$ and $r, w \in \{0, 1\}$.

- Let $V_{ir} = V_{ir}^0 \cup V_{ir}^1$, $V_i = V_{i0} \cup V_{i1}$, and $V = \bigcup_i V_i$.

- $\mathrm{A}(\mathcal{T}) = A \cup V$.

Marking the Tape

Define a non-transitive relation $\prec$ on $U = \{1, 2, H\}$ by

$$\begin{tikzcd} 2 \arrow[loop left] \arrow[r, dash, "\prec"] & H \arrow[r, dash, "\prec"] & 1 \arrow[loop right] \end{tikzcd}$$

Extend the relation pointwise to U^X: $f \prec g$ iff $f(x) \prec g(x)$ for all $x \in X$.

Definition

For $\mathbf{B} \leq \mathrm{A}(\mathcal{T})^X$ and $F \subseteq B \cap U^X$, we say that F is *sequentiable* if $f^{-1}(\{H\}) \neq \emptyset$ for all $f \in F$ and there is some ordering, $F = \{f_n \mid n \in N\}$ and $N = [a, b] \cap \mathbb{Z}$, such that $f_n \prec f_{n+1}$.

f_{n-1}	1	$\cdots$	1	H	2	2	2	$\cdots$	2
f_n	1	$\cdots$	1	1	H	2	2	$\cdots$	2
f_{n+1}	1	$\cdots$	1	1	1	H	2	$\cdots$	2

Marking the Tape

Let $X_n = f_n^{-1}(\{H\}) \neq \emptyset$,

$$X_L = \bigcap_{f \in F} f^{-1}(\{1\}) \quad \text{and} \quad X_R = \bigcap_{f \in F} f^{-1}(\{2\}).$$

Then $X = X_L \cup X_R \cup \bigcup X_n$ is a partitioning of X.

	X_L	$\cdots$	X_{n-2}	X_{n-1}	X_n	X_{n+1}	X_{n+2}	$\cdots$	X_R
f_{n-1}	1	$\cdots$	1	H	2	2	2	$\cdots$	2
f_n	1	$\cdots$	1	1	H	2	2	$\cdots$	2
f_{n+1}	1	$\cdots$	1	1	1	H	2	$\cdots$	2

Encoding the Configurations

Let $n \in N$, $\mathcal{Q} = (t, n, \mu_i)$ be a configuration, and $\eta \in \{0,1\}^X$ any function. Define an element $\beta = \beta(\mathcal{Q}) \in \mathbf{B} \leq \mathsf{A}(\mathcal{T})^X$ by

$$\beta(x) = \begin{cases} C_{it(n)}^{\eta(x)} & \text{when } x \in X_L \\ C_{it(n)}^{t(j)} & \text{when } x \in X_j, j < n \\ M_i^{t(n)} & \text{when } x \in X_n \\ D_{it(n)}^{t(j)} & \text{when } x \in X_j, j > n \\ D_{it(n)}^{\eta(x)} & \text{when } x \in X_R \end{cases}$$

Note that $\beta(\mathcal{Q})$ encodes t (restricted to N), μ_i, $t(n)$, and n as $\beta(x) = M_i^{t(n)}$ when $x \in X_n$.

A Picture

		$n-2$	$n-1$	n	$n+1$	$n+2$	
t	$\cdots$	0	1	0	1	1	$\cdots$

	X_L		X_{n-2}	X_{n-1}	X_n	X_{n+1}	X_{n+2}		X_R
f_{n-1}	1	$\cdots$	1	H	2	2	2	$\cdots$	2
f_n	1	$\cdots$	1	1	H	2	2	$\cdots$	2
f_{n+1}	1	$\cdots$	1	1	1	H	2	$\cdots$	2

β	C_{i0}^{η}	$\cdots$	C_{i0}^{0}	C_{i0}^{1}	M_i^0	D_{i0}^{1}	D_{i0}^{1}	$\cdots$	D_{i0}^{η}

$$\beta = \beta(\mathcal{Q}) \quad \text{and} \quad \mathcal{Q} = (t, n, \mu_i).$$

Encoding the Initial Input

Define the unary operation I on $\mathrm{A}(\mathcal{T})$ by

$$I(x) = \begin{cases} C_{10}^0 & \text{if } x = 1 \\ M_1^0 & \text{if } x = H \\ D_{10}^0 & \text{if } x = 2 \\ 0 & \text{otherwise} \end{cases}$$

Then

$$I(1, \dots, 1, H, 2, \dots, 2) = (C_{10}^0, \dots, C_{10}^0, M_1^0, D_{10}^0, \dots D_{10}^0) = \beta(\overline{0}, n, \mu_1).$$

Encoding the Turing Program (or $\mathcal{T}(\cdot)$)

For each instruction (μ_i, r, w, L, μ_j) in the program of $\mathcal{T}$, and for each $s \in \{0, 1\}$, define the 3-ary operation L_{irs} on $\mathrm{A}(\mathcal{T})$ by

$$L_{irs}(x, y, z) = \begin{cases} C_{js}^{w'} & \text{if } x = y = 1, z = C_{js}^{w'} \text{ for some } w' \\ M_j^s & \text{if } x = H, y = 1, z = C_{ir}^{w} \\ D_{js}^{w} & \text{if } x = 2, y = H, z = M_i^r \\ D_{js}^{w'} & \text{if } x = y = 2, z = D_{ir}^{w'} \text{ for some } w' \\ \overline{v} & \text{if } z \in V \text{ and } L_{irs}(x, y, \overline{z}) = v \in V \\ 0 & \text{otherwise} \end{cases}$$

This emulates the operation of $\mathcal{T}$ when it is in state μ_i reading r and the square to the left of the head contains an s.

Encoding the Turing Program (or $\mathcal{T}(\cdot)$)

For each instruction (μ_i, r, w, R, μ_j) in the program of $\mathcal{T}$, and for each $s \in \{0, 1\}$, define the 3-ary operation R_{irs} on $\mathrm{A}(\mathcal{T})$ by

$$R_{irs}(x, y, z) = \begin{cases} C_{js}^{w'} & \text{if } x = y = 1, z = C_{js}^{w'} \text{ for some } w' \\ C_{js}^{w} & \text{if } x = H, y = 1, z = M_i^r \\ M_j^s & \text{if } x = 2, y = H, z = D_{ir}^w \\ D_{js}^{w'} & \text{if } x = y = 2, z = D_{ir}^{w'} \text{ for some } w' \\ \overline{v} & \text{if } z \in V \text{ and } R_{irs}(x, y, \overline{z}) = v \in V \\ 0 & \text{otherwise} \end{cases}$$

This emulates the operation of $\mathcal{T}$ when it is in state μ_i reading r and the square to the right of the head contains an s.

Another Picture

$\beta = \beta(t, n, \mu_i)$, $(\mu_i, 0, 1, R, \mu_j) \in \mathcal{T}$,
$\mathcal{T}(t, n, \mu_i) = (t', n+1, \mu_j)$, $\beta' = \beta(t', n+1, \mu_j)$

		$n-2$	$n-1$	n	$n+1$	$n+2$	
t	$\cdots$	0	1	0	1	1	$\cdots$

	X_L		X_{n-2}	X_{n-1}	X_n	X_{n+1}	X_{n+2}		X_R
f_{n-1}	1	$\cdots$	1	H	2	2	2	$\cdots$	2
f_n	1	$\cdots$	1	1	H	2	2	$\cdots$	2
f_{n+1}	1	$\cdots$	1	1	1	H	2	$\cdots$	2
β	C^η_{i0}	$\cdots$	C^0_{i0}	C^1_{i0}	M^0_i	D^1_{i0}	D^1_{i0}	$\cdots$	D^η_{i0}
β'	C^η_{j1}	$\cdots$	C^0_{j1}	C^1_{j1}	C^1_{j1}	M^1_j	D^1_{j1}	$\cdots$	D^η_{j1}

$$R_{i01}(f_n, f_{n+1}, \beta) = \beta'$$

How Did We Do?

Let $B_0 = \{f \in B \mid 0 \notin f(X)\}$.

Lemma

Let $j, j', n \in N$, $\mathcal{Q} = (t, n, \mu_i)$, and $t(n) = r$.

- *Suppose that $(\mu_{i'}, r', w', L, \gamma) \in \mathcal{T}$ and $\varepsilon \in \{0, 1\}$. Then $L_{i'r'\varepsilon}(f_j, f_{j'}, \beta(\mathcal{Q})) \in B - B_0$ iff $i' = i$, $r' = r$, $j' = n$, $j = n - 1$, and $\varepsilon = t(n-1)$. In this case, $L_{i'r'\varepsilon}(f_j, f_{j'}, \beta(\mathcal{Q})) = \beta(\mathcal{T}(\mathcal{Q}))$.*
- *Suppose that $(\mu_{i'}, r', w', R, \gamma) \in \mathcal{T}$ and $\varepsilon \in \{0, 1\}$. Then $R_{i'r'\varepsilon}(f_j, f_{j'}, \beta(\mathcal{Q})) \in B - B_0$ iff $i' = i$, $r' = r$, $j' = n + 1$, $j = n$, and $\varepsilon = t(n+1)$. In this case, $R_{i'r'\varepsilon}(f_j, f_{j'}, \beta(\mathcal{Q})) = \beta(\mathcal{T}(\mathcal{Q}))$.*

Thus, we can produce $\beta(\mathcal{T}(\mathcal{Q}))$ from $\beta(\mathcal{Q})$ by applying $L_{ir\varepsilon}$ or $R_{ir\varepsilon}$, which have nonzero coordinates ($\notin B_0$) precisely when the correct one has been applied and ε is the correct value of $t(n-1)$ (for L) or $t(n+1)$ (for R).

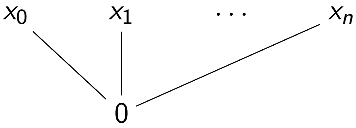
Other Operations on $\mathrm{A}(\mathcal{T})$

- The operations I, L_{irs}, and R_{irs} encode the configuration algebra of $\mathcal{T}$.
- We want $\mathrm{A}(\mathcal{T})$ to not only model the configuration algebra of $\mathcal{T}$, but to also have certain algebraic properties. Thus, we need more structure (i.e. more operations).
- Define a "multiplication": $x \cdot y = 0$ unless

$$\begin{aligned} 2 \cdot D = H \cdot C &= D, & 1 \cdot C &= C \\ 2 \cdot \overline{D} = H \cdot \overline{C} &= \overline{D}, & 1 \cdot \overline{C} &= \overline{C}. \end{aligned}$$

- $\langle \mathrm{A}(\mathcal{T}); \wedge \rangle$ is a height 1 meet semilattice with 0 at the bottom:

$$\begin{aligned} x \wedge y &= 0 \text{ if } x \neq y \\ x \wedge x &= x \end{aligned}$$



Other Operations on $\mathrm{A}(\mathcal{T})$

$$J(x,y,z) = \begin{cases} x & \text{if } x = y \\ x \wedge z & \text{if } x = \overline{y} \\ 0 & \text{otherwise} \end{cases} \qquad S_0(u,x,y,z) = \begin{cases} 0 & \text{if } u \notin V_0 \\ (x \wedge y) \vee (x \wedge z) & \end{cases}$$

$$J'(x,y,z) = \begin{cases} x \wedge z & \text{if } x = y \\ x & \text{if } x = \overline{y} \\ 0 & \text{otherwise} \end{cases} \qquad S_1(u,x,y,z) = \begin{cases} 0 & \text{if } u \notin \{0,1\} \\ (x \wedge y) \vee (x \wedge z) & \end{cases}$$

$$S_2(u,v,x,y,z) = \begin{cases} 0 & \text{if } u \neq \overline{v} \\ (x \wedge y) \vee (x \wedge z) & \end{cases}$$

$$T(x,y,z,u) = \begin{cases} 0 & \text{unless } x \cdot y = z \cdot u \neq 0 \\ x \cdot y & \text{if } x \cdot y, x = z, y = u \\ \overline{x \cdot y} & \text{if } x \cdot y = z \cdot u, [x \neq z \text{ or } y \neq u] \end{cases}$$

Summary

- $\mathrm{A}(\mathcal{T})$ has underlying set

$$\{0\} \cup \{1, H, 2\} \cup \{C, D, \overline{C}, \overline{D}\} \\ \cup \left\{ C_{ir}^{w}, D_{ir}^{w}, M_{i}^{r}, \overline{C_{ir}^{w}}, \overline{D_{ir}^{w}}, \overline{M_{i}^{r}} \mid r, w \in \{0, 1\}, 1 \leq i \leq k \right\},$$

with operations

$$\{I, L_{irs}, R_{irs} \mid r, s \in \{0, 1\}, 1 \leq i \leq k\} \cup \{\cdot, \wedge, J, J', S_0, S_1, S_2, T\}$$

- Certain tuples from $\{1, H, 2\}$ allow for a "marking" of the tape.
- Certain tuples from the last set encode the configurations of $\mathcal{T}$.
- I produces an empty tape with a head marker in the initial state.
- L_{irs} and R_{irs} emulate the action of $\mathcal{T}$ on a configuration.
- The Turing machine "computations" in this encoding aren't represented in $\mathrm{A}(\mathcal{T})$, but in certain subalgebras of powers of $\mathrm{A}(\mathcal{T})$. These are elements of $\mathcal{V}(\mathrm{A}(\mathcal{T}))$.

Thank you.