A remark on compression

Andrew Marks (UC Berkeley)

October 19, 2023

Summary

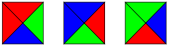
1. The idea to reduce the halting problem to a language by using a recursive compression has appeared in the mathematics literature. E.g. Durand-Romashchenko-Shen used this idea is 2008 to prove an undecidability result about tiling. This analogy was helpful for me to understand the proof of $\mathrm{MIP}^* = \mathrm{RE}$.

2. One can write down a general recursive compression lemma which abstracts the idea. The lemma also applies to the types of compressions introduced by Fitzsimons-Ji-Vidick-Yuen in quantum information, including those in $\mathrm{MIP}^* = \mathrm{RE}$. So one just needs to prove the compression exists and then quote the general lemma. The compression also only needs to operate on succinct representations of single games instead of computable infinite sequences of games.

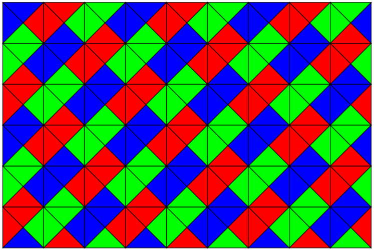
- Nothing here is due to me, except any errors are mine.

Wang tiles

A **Wang tile** is a square tile with a color on each side.



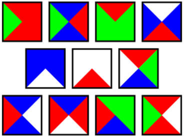
We arrange copies of the tiles side by side with matching colors on the sides of adjacent tiles.



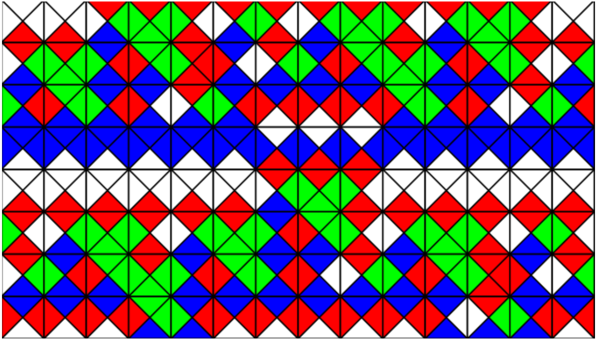
Q: Given a finite set of Wang tiles, if we are allowed infinitely many copies of them, can they tile the infinite plane?

Another example

Tileset T:



A tiling of the plane:



Wang tiling is undecidable

Theorem (Berger, 1966)

There is a computable function mapping each Turing machine M to a finite set of tiles T_M so that M halts $\iff$ there is no tiling of the plane using T_M. So the Wang tiling problem is undecidable.

Proof sketch (Durand, Romashchenko, Shen, 2008):

Exercise: A tileset T tiles the plane iff T tiles every $n \times n$ square.

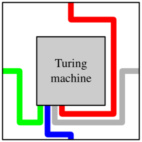
Given a tileset T, let $f(T)$ be the smallest n so that T cannot tile an $n \times n$ square. So $f(T) = \infty$ iff T tiles the plane.

Show there is a polynomial time program "Compress" so that given a Turing machine M that computes a tileset T of size $\leq 2^n$ in time $\mathrm{poly}(n)$, we can compute a tileset $T' = \text{Compress}(M, n)$ of $\mathrm{poly}(n)$ size so that

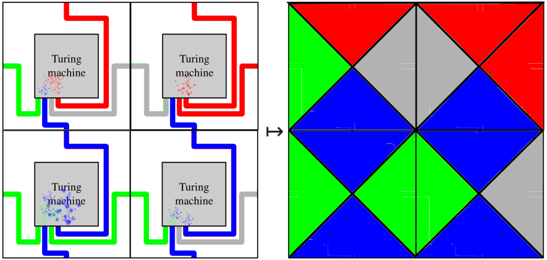
- T tiles the plane $\iff$ T' tiles the plane. .
- $f(T') \geq \max(f(T), n)$.

Finish using the recursive compression lemma. $\square$

The idea of the compression procedure



The tiles T' force a "grid" pattern of size $\mathrm{poly}(n)$. Inside each grid square is a "cpu" that runs the Turing machine M for $\mathrm{poly}(n)$ steps to check its inputs are valid colors from a tile in T. If not, then the tiling can't be continued. Otherwise, the tiles propagate the input colors to the edge of the grid square. In this way, each grid square is a "macrotile", and there is a bijection between these macrotiles and tiles in T.



Succinct representations of strings

Suppose M is a Turing machine and $n, t \in \mathbb{N}$. (M, n, t) is a **succinct representation** of a string x of length $\leq 2^n$ if for all $m < 2^n$, $M(m)$ halts in time $\leq t$ and

$$M(m) = \begin{cases} 0 & \text{if } x(m) = 0 \\ 1 & \text{if } x(m) = 1 \\ 2 & \text{if } m \geq \text{length}(x) \end{cases}$$

This is the usual type of succinct representation in complexity theory. It is (polytime) equivalent to representing a string of length $\leq 2^n$ using a circuit with n input bits. (We regard natural numbers in our succinct representations as being coded in unary)

By coding other objects (e.g. games, tilesets, etc.) by strings we can also reasonably talk about succinct representations of them.

E.g. SUCCINCT-3COLOR is NEXP-complete (Papadimitriou-Yannakakis 1986), as used by Ito-Vidick (2012).

Lemma (Compression lemma)

Suppose $f\colon \{0,1\}^* \to [0,\infty]$, $A \subseteq \{x \in \{0,1\}^* \colon f(x) < \infty\}$ *is nonempty, and there is a polynomial time computable function* Compress *so that if* $\hat{x}$ *is a succinct description of* x *and* $\operatorname{Compress}(\hat{x}, n) = y$, *then:*

1. $f(y) \geq \max(f(x), n)$
2. *If* $x \in A$, *then* $y \in A$.

Then there is a polynomial time reduction g *from the halting problem to* A *that maps each Turing machine* M, *to* $g(M) \in \{0,1\}^*$ *so if* M *halts, then* $g(M) \in A$, *and if* M *does not halt, then* $g(M) \in \{x \colon f(x) = \infty\}$.

E.g. $A = \{G \colon \mathrm{val}^*(G) = 1\}$, $f(G) = \log \mathscr{E}(G, 1/2)$.

$A = \{T \colon T$ can tile the plane$\}$, $f(T)$ is the least n so T cannot tile an $n \times n$ square.

- There is no assumption f is computable.
- If the input to Compress is not a succinct description, there are no assumptions on how it behaves.
- Generalizes beyond polytime compressions provided the amount of succinctness is large compared to the complexity of compression.

Proof

Fix $a_0 \in A$. Consider the following polynomial-time computable function $h(M, n)$. $h(M, n)$ first runs M for n steps. If M halts in $\leq n$ steps, then $h(M, n)$ outputs a_0. If M does not halt in n steps, then $h(M, n)$ outputs $\text{Compress}(\widehat{h(M, n+1)}, n)$, where $\widehat{h(M, n+1)}$ is a succinct description of $h(M, n+1)$.

Note: justifying the self-referential call of h uses the polynomial-time version of Kleene's recursion theorem (see e.g. Westrick, 2017), and a careful calculation of the runtime of $h(M, n)$. (There is a recursive relationship between the runtime of $h(M, n)$ and the runtime of $h(M, n+1)$).

If M does not halt in n steps, then $f(h(M, n)) \geq \max\{f(h(M, n+1), n\}$ by assumption (1). So inductively, $f(h(M, 0)) = \infty$.

If M halts in n steps, then $h(M, n) = a_0$, so by assumption (2) and induction $h(M, n) \in A$, so $h(M, n-1) \in A$, ..., so $h(M, 0) \in A$.

Define the polynomial time reduction g by $g(M) = h(M, 0)$. So if M does not halt, $f(g(M)) = \infty$. If M does halt, then $g(M) \in A$. $\square$

Open problems/requests

- Tell me any instances of compression in the literature that I might not know, and check this abstract compression lemma applies to all of them. (Find the right general compression lemma if it doesn't).
- Find compression proofs of some classical undecidability proofs in mathematics e.g. the undecidability of the word problem for groups.
- Find compression proofs of some new undecidability results e.g. computing the operator norm (see Fritz-Netzer-Thom).

Thanks!

- B. Durand, A. Romashchenko, A. Shen, *Fixed Point and Aperiodic Tilings* 12th International Conference on Developments in Language Theory, Kyoto : Japan (2008).
  `https://doi.org/10.1007/978-3-540-85780-8_22`
  `arXiv:0802.2432`
- L. Westrick, *Seas of squares with sizes from a* Π^0_1 *set*. Israel Journal of Mathematics, 222(1):431–462, 2017.
  `https://doi.org/10.1007/s11856-017-1596-6`
  `arXiv:1602.04481`.

Being careful – the complexity really is important

If we do not ensure the amount of succinctness is more than the complexity of the compression procedure, the compression lemma is false.

Here is a good example to keep in mind. Let $f(x) = \mathrm{length}(x)$. If $\hat{x}$ is a succinct description of x, let $\mathrm{Compress}(\hat{x}, n) = x0^n$, i.e. x with n zeroes appended. Note Compress takes exponential time to compute. Let $A = \{0,1\}^*$. The hypotheses of the compression lemma are satisfied (except for Compress being polynomial time computable), but clearly there cannot be a polynomial time reduction of the halting problem to A.

With a little more work it is possible to construct counterexamples where Compress outputs strings of polynomial length (but runs in exponential time).