CPSC 506: Complexity of Computation

On the foundations of our field, connections between Science and Computing, where our field might be headed in the future

Outline for today

- Course topics
- Course work

- Introductions

- Website, logistics

- Course prerequisites
- Review of prerequisites: examples, exercises

CPSC 506: Course topics

- *Gauging the hardness of a problem*: time and space complexity, provably hard problems. How nondeterminism can help our thinking.
- *The power of randomness*: How does randomess help us compute more efficiently?
- *The power of interaction*: Proofs that use interaction and randomness.
- *Approximability and APX-hardness*: How to determine whether or not a problem has a good approximation algorithm.
- *Quantum complexity theory*: Novel ways of communicating and computing.
- *Chemical reaction systems*: Emerging theories of computing with molecules; energy as a computational resource.

CPSC 506: Course work

- Five homework assignments: 50%
- Student reading project and presentations: 20%
  (more on this soon…)
- Final exam: 30%

- Lots of problem-solving during class
- Opportunities for individualized learning

Introductions

- Your name, where you are in the program, what research areas you’re interested in, why you want to take the course, what you hope to get out of it

Prerequisites

I'll assume familiarity with (or willingness to quickly gain familiarity with)

- Turing machine (1-way or 2-way infinite tape(s), states, accepting state(s), configuration). A TM can be represented by a binary string.

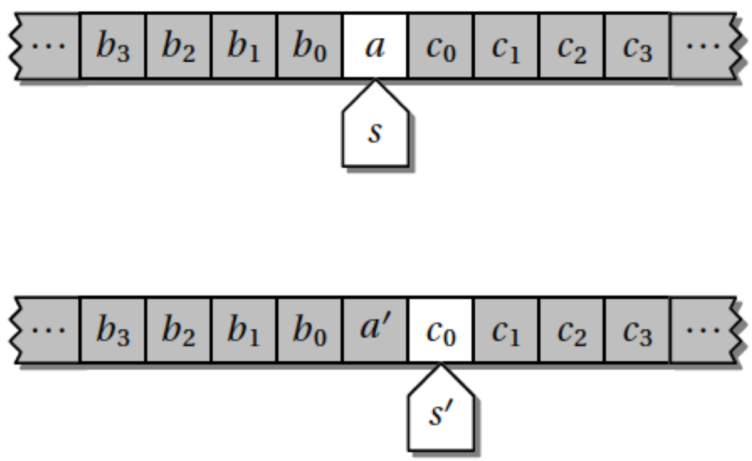


FIGURE 7.9: One step of a Turing machine. It changes the tape symbol from a to a', changes its internal state from s to s', and moves one step to the right.

Illustration from: The Nature of Computation, Chris Moore

Prerequisites

I'll assume familiarity with (or willingness to quickly gain familiarity with)

- Turing machine, time bounded Turing machine, universal Turing machine, decidable languages, undecidable languages

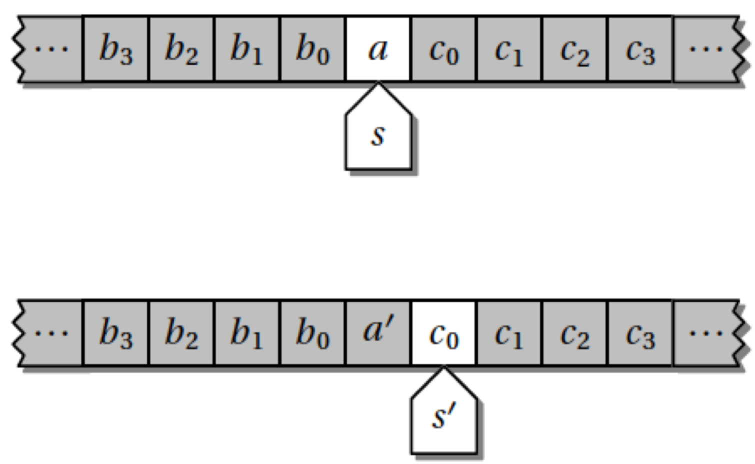


FIGURE 7.9: One step of a Turing machine. It changes the tape symbol from a to a', changes its internal state from s to s', and moves one step to the right.

Illustration from: The Nature of Computation, Chris Moore

Prerequisites

I'll assume familiarity with (or willingness to quickly gain familiarity with)

- Turing machine, time bounded Turing machine, universal Turing machine, decidable languages, undecidable languages
- The classes P and NP, what it means for a problem to be NP-complete, NP-hard. How polynomial-time reductions ($L \leq_p L'$) are used to establish NP-completeness results. How to derive simple reductions.
- What is a nondeterministic Turing machine (NTM), what it means for a NTM to accept a language, and the definition of NP in terms of NTMs.

Review: exercises, examples

- Can you give examples of problems that are in P, in NP, NP-complete, decidable, undecidable?

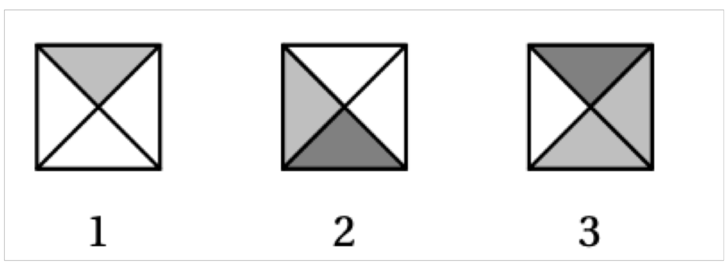
Review: exercises, examples

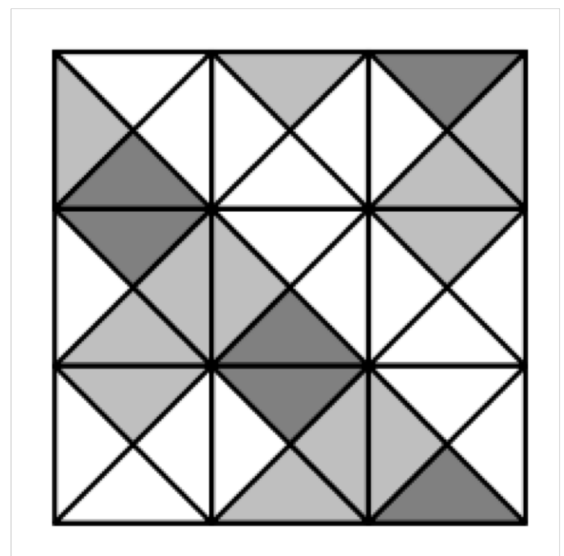
State whether must be true, there is a counter-example, or open question:

- If A is in P than A is NP-complete
- If language A is in P and language B is in P then A $\leq_p$ B
- If A is in P and B is in NP then A $\leq_p$ B
- SAT is in P

Wang tiling

- Let's look at an old problem that you might not have seen before, and determine its complexity. We'll also see how this problem has led to interesting new research problems.



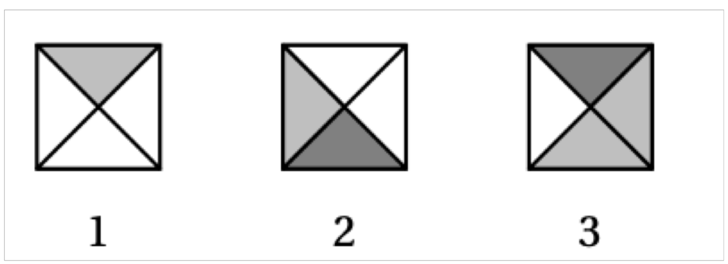


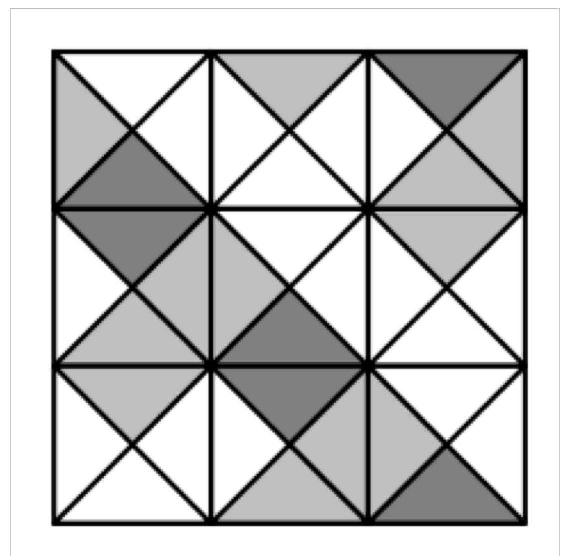
Illustrations from: The Nature of Computation, Chris Moore

Wang tiling

- **Instance**: A finite set T of Wang tiles: square tiles, each with a fixed orientation, edges labeled with colours.

- **Problem**: Can the infinite plane $\mathbb{Z}^2$ be tiled with tiles from T? Colours on adjacent tile edges must match, and tiles must be in their fixed orientation (no rotations or reflections). A tile can be used repeatedly.

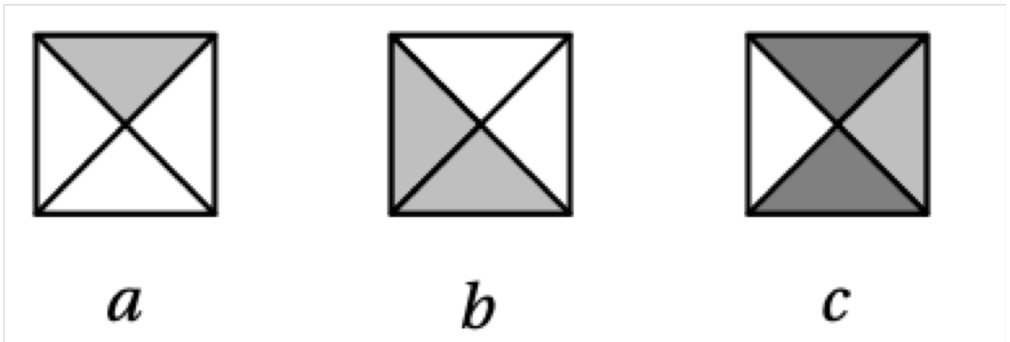




Illustrations from: The Nature of Computation, Chris Moore

Wang tiling

Another example:



One attempt at tiling doesn't work… can we conclude that this is a no instance of Wang Tiling?

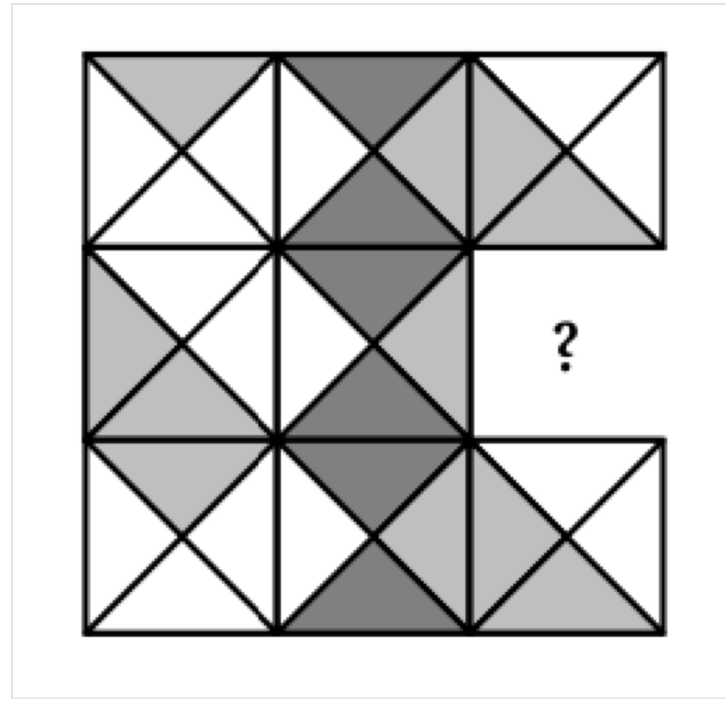


Illustration from: The Nature of Computation, Chris Moore

Periodic tilings

- Wang conjectured (1961) that if a set T can tile $\mathbb{Z}^2$, then T allows a periodic tiling of the plane, i.e., a tiling with infinitely repeatable square of tiles, or "period".

Explain why if Wang's conjecture is true, then the Wang Tiling problem is decidable.

Wang tiling: simulating a TM

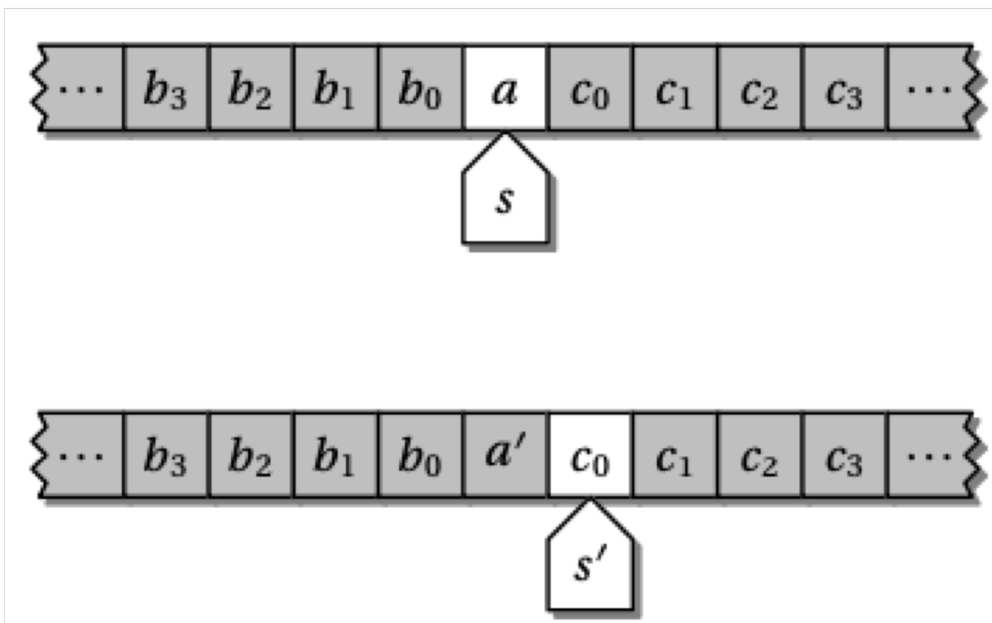
Robert Berger (1965) showed that the Wang Tiling problem is undecidable, thereby disproving Wang's conjecture. Let's construct ingredients of a reduction from the (undecidable) Blank Tape Halting problem to Wang Tiling.

The Blank Tape Halting problem is as follows:

- **Instance**: A description of a Turing machine (TM), with a two-way infinite tape.
- **Problem**: does the TM halt when the tape is initially blank?

Wang tiling: simulating a TM

- What tiles would be useful for TM simulation?
- First focus on tape positions *not* near the tape head. Design tiles to ensure that tiles in consecutive rows have the same tape contents. Your tile edges can be labeled with symbols in place of colours.



Wang tiling: simulating a TM

Now, consider the tape head position, plus the state, when the head moves to the *right*. Design tiles such that the next row of the Wang tiling is guaranteed to properly represent the configuration resulting from the TM transition. (A similar construction can handle transitions when the head moves to the left.)