Metatheorem 2 and Pac-Man

Connor Lemp and Richard Carini

March 10, 2015

Metatheorem 2[1]

This week we will prove another meta-theorem relating to NP-Hardness. Then we can give an example of an application of the theorem. But first, some definitions:

- **Tokens** are items the avatar can carry and transport. A token is called **cumulative** if the avatar can carry more than one at a time, and **collectible** if they can be placed by the game's designer and can be picked up.
- A **Toll Road** is a path that can only be traveled across by paying a token the avatar is holding. These paths can be traversed as often as the player desires, but costs a token each time.

Metatheorem 2 has three parts:

1. a game is NP Hard if it has collectible tokens, location traversal and toll roads.

   Proof. Take any 3-regular planar graph G on n vertices. Now we construct a game level L that is beatable iff G has a Hamiltonian cycle. Create a location corresponding to each vertex in G that must be traversed (location traversal). Then, add a location f connected only to the starting vertex, s by a toll road. Place a collectible token in every location except f, and 2 tokens in s. Now connect locations corresponding to the edges in G with toll roads.

   Using this construction, if G has a Hamiltonian cycle C, then L is beatable. To do so, the player simply needs to follow the path through L corresponding to C. This allows the avatar to visit each location, as it means there is a path through locations of L that starts at s, visits every location corresponding to a vertex in G, returns to s and proceeds to f.

   Conversely, if L is beatable, then G has a Hamiltonian cycle. Let p be the player's path through L, and c be the path on G's vertices corresponding to p (meaning that an edge between u and v in $V(G)$ is in c iff the path connecting the locations in L corresponding to u and v is traversed in p). The location traversal property means all locations must be visited at least once by p. Therefore, c visits every vertex at least once. For a level with $n+1$ locations that must be visited, n tokens are required at minimum to move between the locations.

[1] Written by Connor Lemp

Because $n+1$ tokens are on the map, the player can visit one location twice, and the rest exactly once. Because f's only neighbor is s, any path p must visit s twice. Thus, c starts and ends at the same vertex, and visits no vertex more than once. Thus, c is a Hamiltonian cycle. □

2. a game is NP Hard if it has cumulative tokens, location traversal and toll roads.

   Proof. This proof is the same to the proof of part 1, with the exception that the player starts with $n+1$ tokens. All locations must be visited at least once, and the avatar has enough tokens to visit one location twice. This location must be s, otherwise the player will not have enough tokens to travel from s to f at the end. Thus the path through G, c, corresponding to the path through L, p, is a Hamiltonian cycle. □

3. a game is NP Hard if it has collectible cumulative tokens, toll roads and an exit location that must be visited last.

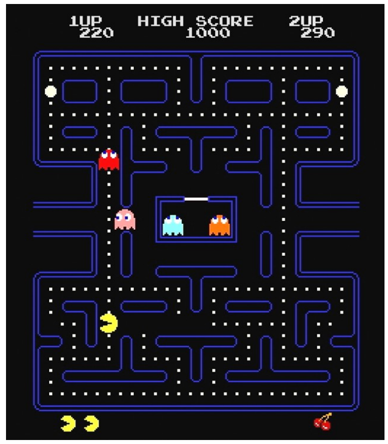
   Proof. This proof is similar to the previous 2 parts with a few modifications. 2 tokens are placed at every location in L except f, and the path from s to f consists of n toll roads in a row. The player needs to end up at s with atleast n tokens if they want to make the trek to the final chamber. The player gains a token by traveling to a location they haven't yet visited (costs 1 to travel and they gain 2 from the room itself) and loses a token by traveling to a room they have already visited (costs 1 to travel and doesn't gain any new tokens from the room). The only type of path through the n locations in L excluding f that will allow the player to win the game is a Hamiltonian cycle. This is because all n locations must be visited exactly once to get n coins, and the player must start and end at s so they can travel to f once they have all the tokens they need. If p satisfies these properties, then c is a Hamiltonian cycle. □

Pac-Man[2]

In the 80's, America became obsessed with a video arcade game by the name of *Pac-Man*. "Pac-Man Fever" took the form of quarters being poured into arcades, merchandise and tchotchkes, and even a hit single. The game features a yellow circle named Pac-Man opening and closing its mouth, which the player uses to navigate a maze and eat "Pac dots." The player, in addition to traveling through the maze and eating the dots, must avoid ghosts which spawn from a ghost house and chase Pac-Man.[3]

[2] Written by Richard Carini

[3] In the game, the ghosts are actually programmed to chase Pac-Man in differing ways. For instance, the red ghost (Blinky) is programmed to find the shortest route between its location and Pac-Man's, but the pink ghost (Pinky) is programmed to target the location four tiles in front of



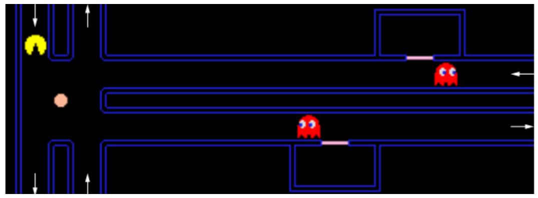
In order to assist Pac-Man in his quest to eat all of the Pac dots, some of the dots are larger (we call them Megadots) and, upon consumption, put the ghosts into "Scatter mode."

At this point, all of the ghosts turn a blue color, and turn around in the opposite direction that they were initially going, attempting to avoid Pac-Man. In this mode, Pac-Man is able to eat the blue ghost for bonus points, forcing the ghost to return to their ghost house. This is only temporary, however, since after the ghosts emerge from the ghost house, they return to "Chase mode" (even if they do not get eaten, they return to their Chase mode state after a while simultaneously). We will now prove that Pac-Man is NP-Complete by demonstrating that it has the characteristics of Metatheorem 2.

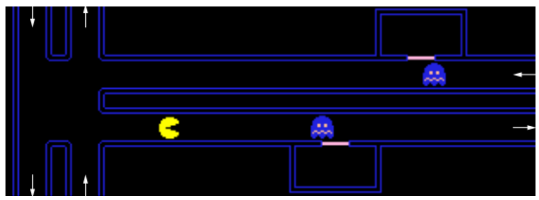
Pac-Man is NP Hard

Recall that by the first part of Metatheorem 2, we know that if construct a level in Pac-Man that features collectible tokens, location traversal, and toll roads, we can prove that the game is NP-Hard. Consider the following construction. Let every vertex in our 3-regular, planar graph G be of the following form, with a single Megadot in each vertex.

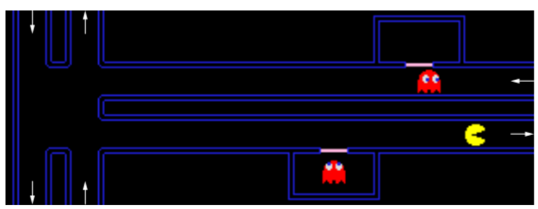
Pac-Man, resulting in an "ambush" technique (Pittman). For our purposes, we will assume all ghosts are red ghosts in order to make the game manipulable.



The Megadot will be our collectible tokens that must be eaten in order to finish the level. Additionally, we construct each edge as two parallel unit-wide pathways that each have a red ghost in the way, spawned by a ghost house halfway along the pathway on each side. This serves as an ideal toll road because Pac-Man only has one "token" (the Megadot) to use in order to traverse the edge.



Pac-Man chooses one pathway in the edge to travel down, sending one of the red ghosts back to its respective ghost house. Now Pac-Man can safely traverse the extent of the edge with no risk of losing a life to the red ghosts.



In order to make this proof more rigorous, we need to establish bounds on the size of the pathways that correspond to the edges of G. This is because there is currently no limit to how long or short the edges are, so ghosts may potentially travel outside of their respective pathways and ruin our construction by leaving a toll road open. Let the number of units that each ghost covers while in Scatter mode be s and the number of units that can be covered while in Chase mode be c (since the ghosts travel at different rates depending on what mode they are in). If n is the number of vertices in G, then the distance d of the distance between the vertex and the ghost house is given by $d = c + s + f(n+1)$. Each edge is no more than a distance of $2d$ in length. When the game starts, the ghosts are in Chase mode and begin traveling in one direction (the one in the direction of Pac-Man). When Pac-Man eats a Megadot, each of the ghosts reverse direction, and must travel in the same direction for the duration of the Scatter mode. Then the ghosts will end up traveling back and forth a distance of $c + s$ within the pathways. Since we know that Scatter mode can only

be entered in $n + 1$ times by our construction of $n + 1$ Megadots, we know that each ghost can never leave their pathway. Thus we have collectible tokens (our Megadots), toll roads (ghost-blocked pathways), and location traversal (since the collection of all Megadots is necessary to win the level; an additional regular Pac Dot can be added to the location f of our construction to enforce this). From this, and by Metatheorem 2, we know that Pac-Man is NP-Hard.
Since, given an arbitrary Pac-Man level and potential solution, the path can be checked in polynomial time, we also have that Pac-Man is in NP. Therefore, we conclude that Pac-Man is NP-Complete.

Sources

Play a (advertisements-laden) version of Pac Man!:
Pac Man
http://www.freepacman.org/

Details on Pac Man and nuances in the game:
Pittman, Jamey. *The Pac Man Dossier*
http://home.comcast.net/~jpittman2/pacman/pacmandossier.html

The proof that Pac Man is NP Complete, with the Metatheorem that was mentioned (same paper as last week's presentation):
Viglietta, Giovanni. *Gaming is a hard job, but someone has to do it!*
http://arxiv.org/pdf/1201.4995v5.pdf