More NP Complete Games

Richard Carini and Connor Lemp

March 3, 2015

Why Metatheorems?[1]

This week, we will be introducing the concept of **metatheorems** about NP-completeness. These theorems are intentionally general in order to encompass a wide variety of games. Some properties that are common to games will be described, and the games with these properties can be then fairly easily be proven to be NP-complete. In our writeup and our talk, we will only be giving the claim and the proof of the first metatheorem, however it can easily be seen that the metatheorems provide a very valuable tool for classifying the complexities of a group of games easily and quickly. First, we will need some definitions that will be used in proving our meta-theorems.

Definitions

- an **avatar** is an in-game entity that can be controlled by the player.
- A game has **location traversal** if some avatar in the game is able to move through the level, and is forced by the game designers to travel to certain locations (in any order and as many times as they wish) to beat the level. One location can be seen as the starting location, and another as the ending location.
- A **single use path** is exactly what it sounds like. This is some clever combination of game mechanics that form a path that can only be crossed once.[2]

Metatheorem 1

As we have shown previously, determining whether or not a Hamiltonian cycle exists in an arbitrary graph is NP-complete. This proof relies on the fact that finding a Hamiltonian Cycle within a 3-regular planar graph is also NP-complete. A planar graph is one where it is possible to arrange the vertices such that no edge across, and a k-regular graph is a graph where the degree of all vertices is k. Thus, we can show any game H is NP-complete by relating it to the 3-regular Hamiltonian Cycle problem.

(Metatheorem 1) **Claim:** Any game exhibiting both location traversal (with or without a starting location or an exit location) and single-use paths is NP-hard.

[1] Written by Connor Lemp

[2] Note that this is different from the one-way paths that we have introduced before. Whereas one-way paths can be crossed in one direction only once, single use paths are stronger: they can only be crossed once and then they are eliminated from the game. These usually can take the form of crumbling tiles in popular games

Proof. First let H represent a game with location traversal and single-use paths. For each vertex in an arbitrary 3-regular graph G, create a location in the game. Now create single use paths corresponding to the edges in G. Finally, create a finishing location f. f connects only to s, where s is the starting location. Because the graph is 3-regular, all locations corresponding to vertices in G have only three single-use paths that can be used. If the vertex is visited, it takes one path to get there and one to leave. Arriving at the vertex a second time is possible with the last single use path, but then the player cannot leave that location. Therefore, each location can only be entered and exited once. However, the location s has four single use paths: three that connect to the internal location of our level, and one that connects to the ending location f. Thus two of the single use paths are used in moving first off of s and then eventually back onto s, then the player can move directly onto the platform f through the final single-use path and finish the game. Thus, all rooms can only be visited if there exists a Hamiltonian Cycle in G beginning and ending with s. Additionally, G has a Hamiltonian Cycle if the level is completable since completing the level implies that the player has reached all locations and made it back to f. $\square$

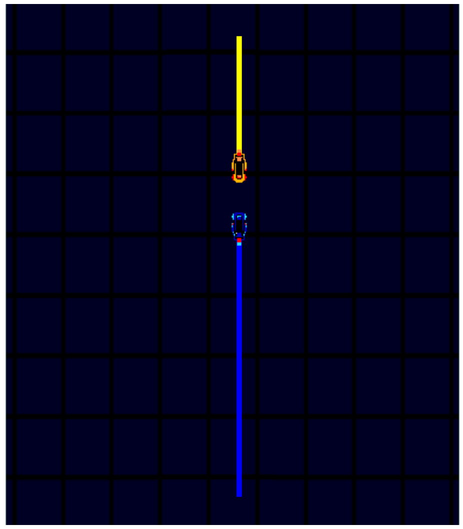
Applying Metatheorem 1[3]

To show the effectiveness and power of Metatheorem 1, we will be demonstrating that another game is NP-complete by simply showing that the game allows for the construction of location-traversal and single-use paths. By Metatheorem 1, we know that the fact that the game is NP-hard will follow naturally from this construction.

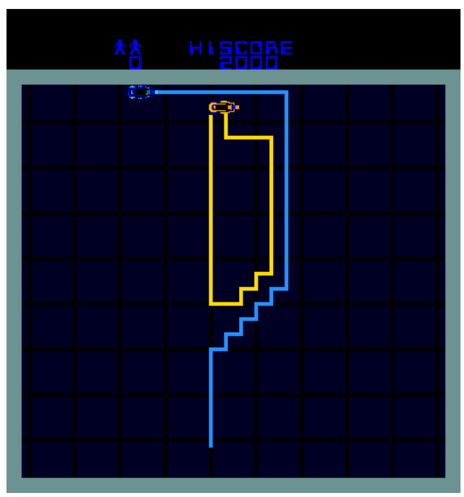
Tron

In 1982, Disney released both a movie and video game entitled Tron. Each was a promotional tool for the other, and had moderate success. about the video arcade game Tron. The game features four mini games that the player must complete in order to advance levels. One of the mini games of Tron is a light cycle duel. In this game, the player is a blue light cycle and the opponent(s) are orange light cycles.

[3]Written by Richard Carini



The player and the opponents maneuver their light cycles, leaving a trail of light behind themselves, and attempt to avoid crashing into the walls or the paths of light.[4] As the game progresses, the grid becomes filled with dangerous areas, and the area that can be traversed is lessened.



The game ends when either the player crashes or all of the opponents crash. Essentially, this creates a survival game where the winner is the last man standing.

Tron is NP Complete

Proof. The fact that Tron is in NP is evident: given an arrangement to follow, we can easily check to see if that results in winning the game or crashing the player's

[4]Additional note: once a light cycle crashes, their trail is removed from the board

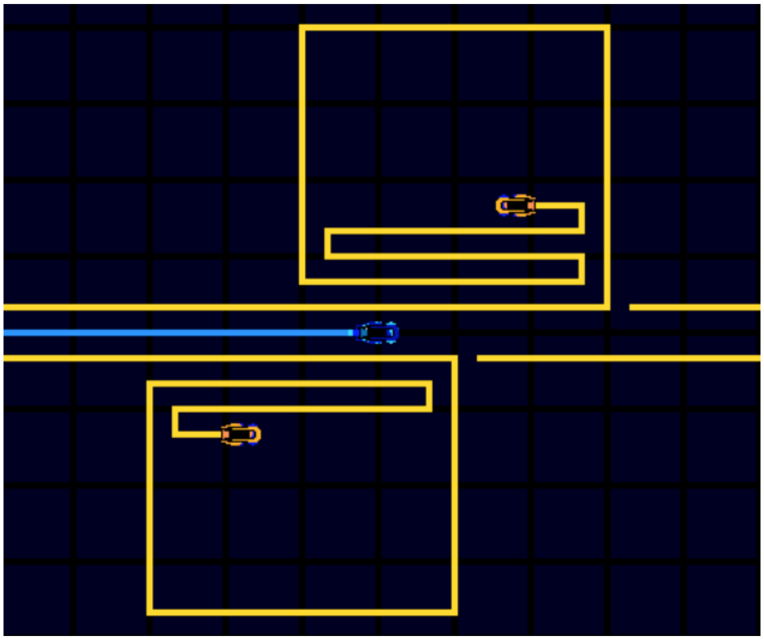
light cycle. To show that Tron is NP-Hard, we will use Metatheorem 1, proving the existence of location-traversal and single-use paths in the light cycle portion of the video game Tron. From this, the fact that Tron is NP-Hard will follow. From our construction, we will develop a way to make the opponent's light cycles both create a game grid and then destroy themselves after a predetermined threshold of time. If the player can outlast the opponent's light cycles, the player will win.
First we need a way to construct our graph with the opponent's light cycles. Rather than tracing the outline of a graph with a light cycle, we will be letting the opponents construct the area of the faces of our graph (including the outer face). This will, in a sense, trace the outline of a graph where the spaces between faces represents our initial graph's edges and vertices. Therefore, we have a game grid where the vertices are "empty spaces" that can be traversed by the player, and edges are bounded by two faces each.

1. Location-Traversal:

   Using the game board we described from above, we will show that a construction exists such that all vertices must be traversed by the player. Every vertex is an empty space being the intersection of three edges with a predefined area. Our paths must then be scaled up by a factor (while leaving the vertices' size the same) in order to make the total length of the paths negligible compared to the area of any one of the vertices.
   Our formal construction is as follows: Let a be the side length of each of the vertices (thus the area of a single vertex, A_v, is a^2) and our scaling factor be denoted by k. This factor must then be carefully chosen such that our desired conditions hold: firstly, we need to make the total length of the paths negligible compared to the area of any one vertex; secondly, we need to make the area of a vertex negligible compared to the area of a face (traced out by an opponent's light cycle). We do this by letting the area of each face (A_f) be significantly larger than the area of all of the vertices ($(n+1)A_v$), but the perimeter of each face (P_f) being significantly smaller than the area of any vertex (A_v):

   $$P_f < A_v < (n+1)A_v < A_f$$

   The game play begins then as follows: Each of the light cycles traces out the faces of our construction, taking a length of P_f. While this happens, the player "waits" by covering the starting vertex. Since P_f is less than A_v, we know that this is feasible. As soon as the opponent light cycles are finished creating their faces, they construct another area inside A_f which is both larger than $(n-1)A_v$ and smaller than nA_v. The opponents then begin spiraling towards the center of the rectangle on a suicide mission. The following image depicts this construction with the opponent's light cycles already spiraling inwards to their inevitable demise, and the player traveling on the edge between two faces.



The race for survival has thus begun with the player traversing the graph and traveling through the paths to visit each vertex. Since the opponents will self-destruct in a time that is greater than $(n-1)A_v$, we have forced the player to necessarily travel through each of the n vertices on our game grid.

2. Single-Use Paths

   The ability to have single-use paths has been hinted at in the above construction of location-traversal. Each edge of our initial graph will be sandwiched between two faces constructed by the opponent's light cycles. This path will have a width of one unit length. Since the player's own light path acts as a barrier to the player, we know that the path of unit length has effectively been "used" when the player traverses along the path exactly once.

Since we have shown location-traversal and single-use paths in Tron, we have demonstrated that Tron is NP-Hard. Since Tron is in NP and NP-Hard, we can conclude that Tron is NP-Complete.

$\square$

Sources:

The proof that Tron (and a lot of other games) is NP Complete, with the Metatheorem that was mentioned:
Viglietta, Giovanni. *Gaming is a hard job, but someone has to do it!*
http://arxiv.org/pdf/1201.4995v5.pdf

Another paper on the computational complexity of Tron, with some variations to determine PSPACE-completeness (not discussed in class or in this writeup, but interesting to look at):
Miltzow, Tillmann. *Tron, a combinatorial Game on abstract Graphs*
http://arxiv.org/pdf/1110.3211.pdf

Play Tron for yourself:
Steven Lisberger. Disney, 1982. *Classic Tron*.
http://games.disney.com/disneyxd-classic-tron