Capítulo 1

Introducción

1.1. Historia del ajedrez computacional

El periodo de tiempo comprendido entre 1949 y 1959 es considerado como el nacimiento del ajedrez computacional. En el año de 1949 el matemático Claude Shannon, escribió el artículo *Programming a Computer for Playing Chess* [37]. Este artículo fue el primero en contener los principios básicos para hacer un programa que jugara ajedrez, los cuales aún son usados en los programas de ajedrez de la actualidad. Igualmente, Shannon describió las posibles estrategias de búsqueda necesarias debido a la enorme complejidad del ajedrez (estas estrategias serán descritas es la sección 2). Un año después, en 1950, el matemático ingles Alan Turin[44] inventó un algoritmo enfocado a enseñar a una máquina a jugar ajedrez, pero en ese tiempo la computadora que pudiera ser programada con dicho algoritmo aún no existía. Turing llevó a cabo su algoritmo usando una pluma y un pedazo de papel para jugar en contra de uno de sus colegas. Si bien el programa perdió, este partido dio comienzo al ajedrez computacional.

En el mismo año, John Von Neumann creó una enorme máquina llamada MINIAC, la cual podía calcular alrededor de 100,000 instrucciones por segundo, y también era programable. Su tamaño y poder de cálculo eran inmensos para ese tiempo, ya que fue construida con el objetivo de realizar los cálculos necesarios para poder producir bombas atómicas. En lugar de inmediatamente trabajar en los cálculos para las bombas, los científicos empezaron a experimentar con la máquina. Una de las primeras cosas que hicieron fue escribir un programa de ajedrez para la computadora: fue para un tablero reducido de 6x6 y no contaba con alfiles. El programa necesitaba alrededor de 12 minutos a una profundiad de 4 para realizar un movimientos, y de haber contado con los alfiles hubiera requerido alrededor de 3 horas. El programa jugó

solamente 3 partidas: una contra él mismo (ganaron blancas); el segundo contra un jugador fuerte (el juego duró 10 horas y terminó con la victoria del maestro); finalmente el programa jugó contra una joven mujer que aprendió el juego una semana antes, el programa ganó en solamente 23 movimientos. Fue la primera vez que un humano perdía ante una computadora en un juego de habilidad intelectual[32].

Durante 1958 un grupo de científicos americanos de la Universidad Carnegie-Mellon, desarrollaron mejoras al algoritmo de búsqueda básico al cual llamaron algoritmo α–β (es descrito en el capítulo 3). Este algoritmo permitió búsquedas a mayores profundidades, lo que facultaba a las computadoras jugar ajedrez más eficientemente. Durante la década siguiente, Ken Thompson contruyó una máquina cuyo único objetivo era jugar ajedrez llamada Belle [10]. Ésta, era mucho más fuerte que cualquier programa de ajedrez existente y conservó este título entre todos los progamas que jugaban ajedrez por un gran periodo.

El progreso en el ajedrez computacional, fue enfocado en esa época principalmente en incrementar el poder de cómputo de los programas, a través de hardware dedicado. El poder de juego de los programas de ajedrez ya se podría equiparar al de varios maestros, sin embargo, el juego era aún dominado por los humanos. Al final de 1980, un grupo independiente de estudiantes crearon su propio programa de ajedrez llamado *Deep Thought*, el cual fue uno de los más fuertes durante esos años. *Deep Thought* fue el prototipo usado para crear *Deep Blue*, el cual consistía en hardware especializado para jugar ajedrez, y podía calcular 200 millones de posiciones por segundo, en comparación a las 5 jugadas por segundo en promedio que puede analizar una gran maestro, esto da una idea de lo eficientes que son los humanos para jugar ajedrez, sin contar con una capacidad de cálculo tan grande como las computadoras. *Deep Blue* fue construida por un equipo de IBM con el solo propósito de derrotar al campeón reinante en esa época del ajedrez: Garry Kasparov.

En 1996 se llevó a cabo un primer encuentro contra Garry Kasparov el campeón del mundo. El encuentro terminó con una aplastante victoria para Garry Kasparov. Un año más tarde y con una nueva versión mejorada de *Deep Blue*, la computadora derrotó a Garry Kasparov. Fue la primera vez que un programa de ajedrez derrotaba al campeón mundial, lo que generó gran expectativa y sorpresa en su época. Años después numerosos encuentros entre hombre y máquina fueron llevados a cabo, principalmente contra los campeones del mundo, en los que destaca el encuentro entreVladimir Kramnik vs *Deep Fritz* en el 2002, el cual terminó con una victoria para la máquina. Es importante resaltar que estos programas corrían en computadoras personales y por lo tanto no contaban con el enorme poder de cálculo de sus predecesoras como *Deep Blue*, lo que significa que la ventaja debida a un hardware especializado puede ser

remplazada por el diseño y aplicación eficiente de algoritmos de búsqueda sofisticados.

En la actualidad el programa más fuerte es el programa *Rybka*, el cual compite al mismo nivel de los grandes maestros del ajedrez y ha ganado cada uno de los torneos donde ha participado, incluso ha derrotado a grandes maestros, aún con *handicap* en su contra. Los días en que los humanos superaban a las máquinas en un juego de habilidad mental como el ajedrez han quedado atrás, y ahora éstas compiten a un nivel igual o superior a los grandes maestros en este juego. Esto es posible gracias al esfuerzo invertido en la investigación y desarrollo de algoritmos y heurísticas de búsqueda sofisticados para este tipo de problemas, durante los pasados 60 años, sobre todo por parte de la inteligencia artificial.

1.2. Conceptos básicos

El ajedrez es un juego de suma–cero, determinista, finito y de información completa. Suma-cero significa las metas de los competidores son contrarias, una victoria para uno es una derrota para el otro. Supongamos que una posición puede ser evaluada con un valor de 20 para un jugador, entonces el valor para su oponente debe ser -20. Determinista significa que para cada estado y cada acción que tomemos sabemos siempre cual será el estado siguiente. Finito, ya que los juegos no pueden extenderse por siempre, debido a las reglas del ajedrez como: las 3 repeticiones de una posición resulta en un empate y también a la regla de los 50 movimientos, que dice que si en 50 movimientos no hay una captura o movimiento de peón entonces es un empate. Información completa significa que no hay información oculta o incertidumbre como en un juego de cartas. Adicionalmente el juego es secuencial y se juega por turnos.

Un juego de dos jugadores como el ajedrez, puede ser representado por un árbol, donde cada nodo es una posición en el tablero, comenzando desde la raíz con la posición inicial del juego. Los vértices entonces serían los posibles movimientos, que llevan a los siguientes nodos (posiciones). De esta manera, cada posible estado puede ser generado y agregado al árbol. Un juego, después que ha sido jugado, puede ser visto como un camino en el árbol de juego.

Hay que destacar que el árbol del juego del ajedrez es en realidad un grafo acíclico dirigido. No es cíclico, porque las reglas del ajedrez prohíben bucles eternos y como resultado, cada secuencia de posiciones es única. Debido a que diferentes caminos pueden conducir a la misma posición, en realidad es un grafo y no un árbol. Sin embargo, dado que un grafo puede ser representado como un árbol, se le refiere a menudo como un árbol y los métodos de búsqueda se usan como tal. La complejidad en el contexto del ajedrez describe el tiempo computacional o el espacio de búsqueda que es necesitado para resolver este juego. Resolver un juego requiere

una secuencia de movimientos que lleven a una posición final (ganadora o perdedora). Para darse una idea numérica del espacio de búsqueda del juego del ajedrez, algunas cifras son presentadas a continuación.

En cada posición en un juego de ajedrez existen alrededor de 30 a 40 movimientos legales [37]. Los 20 vértices del nodo raíz son los primeros movimientos posibles para el jugador de piezas blancas. Todos los posibles estados generados por estos movimientos en el primer nivel del árbol de juego, pueden ser denominados como el espacio de búsqueda cuando se ve una jugada adelante. Para preveer una jugada en el futuro requiere que nos fijemos en las 30 configuraciones resultantes (caso promedio), 2 jugadas requiere otros 30 movimientos por cada configuración anterior, por lo que para este nivel serían $30^2 = 900$ posiciones posibles para el espacio de búsqueda, para 6 jugadas en el futuro, el espacio contendrá $30^6 = 729,000,000$ posiciones posibles.

Tomando una media de 30 posibles movimientos por posición, y una duración de 80 jugadas (40 por cada jugador) [23], hay $30^{80} = 1.47808829 \times 10^{118}$ posibles juegos de ajedrez, esto indica que hay más posiciones en el ajedrez que átomos en el universo. Este número también es conocido como el número de Shannon, en honor al padre del ajedrez computacional. Debido a este espacio de búsqueda es tan enorme, es casi imposible generar todos los posibles estados, aunque la velocidad de cómputo sea mejorada de manera inimaginable. Esto es debido a la teoría de la medida más pequeñas de tiempo y espacio observable, el tiempo Planck y el espacio Plack que son respectivamente 10^{-43} segundos y 1.5×10^{-35}. Suponiendo que una computadora pudiera ser construida con esas dimensiones, y usando esa cantidad de tiempo para generar una posición, se podría llegar a generar todos los estados posibles de esta forma en $3.16887646 \times 10^{69}$ años. Si bien el árbol de juego existe desde el punto de vista teórico, en la práctica éste no puede ser generado completamente. Inclusive un árbol con profundidad limitada a 8 jugadas es difícil de generar desde el punto de vista práctico.

1.3. Módulos del ajedrez computacional

La publicación de 1950 de Shannon *Programming a Computer for Playing Chess* aún describe las bases del ajedrez computacional actual, ya que desde entonces no mucho ha cambiado respecto a la forma en que las computadoras son programadas para jugar ajedrez. Un programa de ajedrez normalmente contiene 3 módulos que hacen posible jugar ajedrez para una computadora: generador de movimientos, función de evaluación y algoritmos de búsqueda.

El módulo de generación de movimientos es usado para generar recursivamente los movi-

mientos del árbol de juego (aristas). La función de evaluación asigna un valor a las hojas del árbol para que, mediante los métodos de búsqueda, se encuentre el camino a la mejor posición en el árbol. Lógicamente, cuando una posición ganadora es encontrada (estado terminal), la función de evaluación tiene que retornar el valor más alto posible. Los valores para el oponente deberían ser exactamente estos valores pero negados, ya que es un juego de suma-cero. Todos los demás estados no ganadores o no finales obtendrán un valor numérico entre estos valores máximos, debido a que los algoritmos de búsqueda usan esta información para encontrar el mejor movimiento.

En su publicación, Shannon distinguió entre 2 estrategias diferentes que debían seguirse en las búsquedas para el juego de ajedrez, la de tipo A, donde todas las combinaciones de movimientos son examinadas hasta una profundidad dada, y la de tipo B, en las cual sólo los movimientos prometedores son explorados con la esperanza de reducir las ramificaciones del árbol de búsqueda. Estas dos estrategias son estudiadas en los siguientes capítulos.

1.4. Objetivo y metas del trabajo de tesis

El objetivo de este trabajo fue desarrollar un motor de ajedrez, denominado *Buhochess*, el cual sirviera como una base para el estudio e implementación de algunos de los métodos de búsqueda más importantes en juegos deterministas de suma cero. Si bien, para la implementación del motor de ajedrez *Buhochess* fue necesario implementar los tres módulos que contiene todo motor de ajedrez, el trabajo se centra principalmente en el desarrollo y análisis de los métodos de búsquedas.

Entre las metas definidas para alcanzar el objetivo general se cuentan el desarrollo de un motor de ajedrez, desarrollado desde cero, en el cual se implementaran todos los métodos y algoritmos que utiliza un motor de ajedrez moderno. Otra de las metas fue el estudio, análisis e implementación de los principales algoritmos de búsqueda para juegos deterministas de suma cero. Otra meta del trabajo fue el estudio e implementación de algoritmos de búsqueda avanzados y de heurísticas que permitieran realizar una búsqueda con una profundidad de, al menos, 10, dentro de las restricciones de tiempo reglamentarias.

El desarrollo no se consideró completo hasta que el motor fue capaz de jugar una partida completa de ajedrez a un nivel «decente» contra un jugador humano con todas las reglas del mismo, asi como cualquier restricción de tiempo. El término «decente» es subjetivo, por lo que se consideró que el motor *Buhochess* no se consideraría terminado hasta que nunca perdiera al jugar contra un jugador humano de nivel intermedio. Esto se verificó al oponer a *Buhochess*

con los jugadores pertenecientes al club de ajedrez de la Unison. Igualmente, se consideró su nivel de juego como «decente», al ser capaz de vencer sistemáticamente a otros motores de ajedrez con un poder de juego medio, como el *Big Bang Chess* de *Apple Inc.*

1.5. Organización del trabajo

El capítulo 1 de este trabajo de introducen los conceptos básicos del ajedrez computacional y se definen los objetivos y metas del trabajo realizado. Debido a que la tesis se concentra principalmente en el desarrollo y prueba de los algoritmos de búsqueda, la mayor parte del trabajo se concentra en el desarrollo e implementación de éstos. Sin embargo el desarrollo de los módulos de generación de movimientos y función de evaluación son esenciales para el buen desempeño del motor de ajedrez. En el capítulo 2 se presentan estos componentes básicos y se muestra cómo fueron implementadas en *Buhochess*.

En el capítulo 3, se muestran los algoritmos de búsqueda básicos, así como extensiones del algoritmo para mejorar la toma de decisión. Estos algoritmos son muy sensibles a la ordenación de movimientos en la búsqueda. En el capítulo 4, se presentan las heurísticas utilizadas en la ordenación de movimientos, así como mejoras de los algoritmos básicos que permiten desarrollar búsquedas de tipo A con el menor número de nodos posible. Sin embargo, para poder realizar una búsqueda más eficiente (sobre todo en el medio juego) en el capítulo 5 se presentan técnicas implementadas en *Buhochess* conocidas como *poda hacia adelante*, las cuales son estrategias de búsqueda tipo B. Con el fin de analizar en forma experimental el beneficio de cada algoritmo y heurística implementada, en el capítulo 6 se presentan algunos ejemplos ilustrativos. Por último, se presentan las conclusiones y posibles trabajos futuros.