

En flexibel schack AI baserad på case-baseD Reasoning

A Case-Based Reasoning approach to flexible chess AI

Examensarbete inom huvudområdet Datavetenskap

Grundnivå 30 högskolepoäng

Vårtermin 2015

Johannes Qvarford

Handledare: Peter Sjöberg

Examinator: Anders Dahlbom

Sammanfattning

[Direkt efter titelsidan ska det finnas en sammanfattning, som omfattar högst 150 ord. Denna sida ingår inte i ordinarie sidnumrering. Sammanfattningen ska i kort form sammanfatta hela arbetet inklusive bakgrund, problemformulering, genomförande/implementation, resultat och framtida arbete. Den ska därmed vara det sista som skrivs i rapporten. Den behöver därför endast vara med i den rapport som lämnas in för opponering (och den slutgiltiga rapporten). I tidigare inlämningar kan denna text lämnas oförändrad.]

**Nyckelord:** [minst tre nyckelord men inte allt för många, helst inte fler än sex stycken]

Innehållsförteckning

[1 Introduktion 1](#_Toc410984136)

[2 Bakgrund 2](#_Toc410984137)

[2.1 Schack 2](#_Toc410984138)

[2.1.1 Regler 2](#_Toc410984139)

[2.1.2 Portable Game Notation 3](#_Toc410984140)

[2.2 Case-based Reasoning 4](#_Toc410984141)

[2.3 Citat 4](#_Toc410984142)

[3 Problemformulering 6](#_Toc410984143)

[3.1 Problembeskrivning 6](#_Toc410984144)

[3.1.1 Generera falldatabaser 6](#_Toc410984145)

[3.1.2 CBR-tolken 6](#_Toc410984146)

[3.2 Metodbeskrivning 6](#_Toc410984147)

[4 Genomförande/Implementation/ Projektbeskrivning 7](#_Toc410984148)

[4.1 Research / Förstudie 7](#_Toc410984149)

[5 Utvärdering 8](#_Toc410984150)

[5.1 Presentation av undersökning 8](#_Toc410984151)

[5.2 Analys 8](#_Toc410984152)

[5.3 Slutsatser 8](#_Toc410984153)

[6 Avslutande diskussion 9](#_Toc410984154)

[6.1 Sammanfattning 9](#_Toc410984155)

[6.2 Diskussion 9](#_Toc410984156)

[6.3 Framtida arbete 9](#_Toc410984157)

[Referenser 10](#_Toc410984158)

# Introduktion

Schack har en lång historia som sträcker sig ända till början av 600-talet e.Kr.. Schack har traditionellt spelats mellan två människor, men under det senaste århundradet har även maskiner utvecklats för att spela spelet. Dessa maskiner har visat sig kunna mäta sig med mänskliga spelare, och redan under 70-talet utvecklades en maskin som kunde besegra en stormästare (Hapgood, 1982). Sedan dess har utvecklingen bara fortsatt, och 1997 besegrades den då regerande världsmästaren Garri Kasparov av en schackspelande maskin vid namn ”Deep Blue” (Cambell, Hoane & Hsu, 2001).

Schackmaskiner har förbättrats genom historien, men inte genom att efterlikna människor. Hapgood (1982) påstod att schackmaskinerna spelade fult, men vann genom att utnyttja små misstag som motståndaren gjorde. Inte mycket forskning har gjorts kring området att skapa realistisk artificiell intelligens (AI) som kan spela schack på olika skicklighetsnivåer. För att tackla detta område presenteras en AI-agent som använder *Case-based reasoning* (CBR) för att efterlikna mänskligt beteende. CBR är en teknik för att utveckla AI-agenter, som bygger på att lösa problem baserat på lösningar av tidigare, liknande problem. När en agent ska göra ett drag kan den härma vad en expert gjort i samma läge, genom att konsultera en falldatabas. Alla möjliga fall kan inte lagras eftersom det enligt Shannon (1950) lär finns uppemot 1054 fall i schack. AI-agenten måste därför ibland basera sitt val på det expertläge som är mest likt det nuvarande läget.

I det här arbetet ska en schack AI-agent skapas som använder CBR. Agenten ska kunna använda falldatabaser från spel spelade av spelare av olika rank för att justera dess svårighetsgrad. Svårighetsgraden ska överensstämma med spelarnas rank, så att agenten är svårare att besegra om dess falldatabas är baserad på en högt rankad spelare än om den är baserad på en lågt rankad spelare. Falldatabaser kommer kunna skapas baserade på tidigare spelade partier dokumenterade i *Portable Game Notation* (PGN); ett format som används av bland annat World Chess Federation (FIDE) och United States Chess Federation (USCF). Arbetet ska kunna användas som grund för att utveckla AI-agenter för spel med justerbar svårighetsgrad.

# Bakgrund

I denna sektion visas presenteras bakgrundsinformation till CBR och hur det tidigare har applicerats inom forskning. Här visas även spelet schack, dess regler och det vida använda formatet för att beskriva matcher: PGN.

## Case-based Reasoning

CBR är en teknik för problemlösning som går ut på att använda lösningar på tidigare, liknande problem (”case based reasoning”, n.d). Detta kan vara användbart när det inte finns någon heuristik för att utvärdera möjliga handlingar som en AI-agent kan utföra, eller att den är svår att programmera. Nära besläktat är tekniken *Case-Based Planning* (CBP) som går ut på att planera hur ett mål ska nås, baserat på tidigare handlingar som utfördes för att nå målet.

CBR och CBP har applicerats på datorspel i ett antal tidigare arbeten. XXX presenterar hur CBP kan användas för att lära agenter att spela ett fotbollsspel tillsammans (Learning collab) genom att låta dem observera hur ett motståndarlag spelar. I (”Learning to Win”) visas ett sätt att hantera den stora mängd fall som uppstår i realtidsstrategispelet (RTS) Wargus när CBP appliceras. I XXX (”Case-Based Planning and Execution”) studeras även hur CBR kan appliceras på Wargus, men i det arbetet ligger fokus på hur annoterade fall kan användas för att hitta beroenden mellan mål, och resonera fram planer för att nå dessa mål. Fallen annoterades i detta arbete med det mål som experten försökte uppnå i varje fall när den agerade som den gjorde. Det turbaserade strategispelet (TBS) Call to Power || (CTP2) användes som provunderlag i (“Game AI for a Turn-based Strategy Game with Plan Adaptation and Ontology-based retrieval”). I arbetet undersöks hur sökningen av relevanta fall kan förbättras genom att bara lagra den del av spelläget som är relevant för att utföra handlingen i ett visst fall.

## Schack

### Regler

Schack är ett turbaserat brädspel för två spelare där målet är att besegra sin motståndare. Spelet utspelar sig på en 8x8-rutors spelplan, där varje spelare kontrollerar varsin armé av spelpjäser. I Figur 1 visas en bild av spelplanen i början av spelet.



Bild av spelplanen i början av spelet.

Spelarna turas om att flytta spelpjäser i sina arméer. En spelare får bara flytta en spelpjäs per drag. Om en spelare flyttar en av sina spelpjäser på en ruta ockuperad av en motståndarpjäs, så fångas motståndarpjäsen och lämnar spelplanen för resten av matchen. Högst en pjäs i taget får ockupera en ruta, och en pjäs får generellt inte flytta till en ruta om andra pjäser står i vägen till rutan.

Bonden (♟) kan flytta sig ett steg rakt framåt (sett från den ägande spelarens håll), eller ett steg diagonalt framåt om draget är ett fångande drag. Springaren (♞) kan flytta sig två steg horisontellt eller vertikalt, och ett steg på den resterande axeln. Springaren kan flytta till en ruta även om det finns pjäser som blockerar vägen. Löparen (♝) kan röra sig diagonalt. Tornet (♜) kan röra sig horisontellt eller vertikalt. Drottningen (♛) kan antingen röra sig horisontellt, vertikalt eller diagonalt. Kungen (♚) kan röra sig ett steg horisontellt, vertikalt eller diagonalt.

Om en bonde når den sista raden sedd ur ägarens perspektiv, så kan den omvandlas till vilken annan pjäs som helst.

Bonden kan flytta två steg rakt framåt på sitt första drag. Om en spelare flyttar en bonde två steg, så kan bonden betraktas som om den bara tog ett steg, om den fångas av en motståndarbonde nästa drag. Detta kallas *en passant* och illustreras i Figur 2.



1. Bild av *an passant*. Om vit flyttar sin bonde två rutor framåt kan den svarta bonden fånga den genom att flytta till rutan som den röda pilen indikerar.

En spelare kan göra så kallad rockad med sin kung och ett torn, om det inte finns några pjäser mellan tornet och kungen, och varken tornet eller kungen har flyttats förut. Rockaden går till så att kungen flyttas två steg i tornets riktning, och tornet flyttas i kungens riktning så att den hamnar en ruta på andra sidan av kungens nya position. Figur 3 illustrerar hur detta kan se ut om den vita kungen gör rockad med det närmaste tornet, och Figur 4 illustrerar rockad med tornet längst bort.



Bild som visar hur pjäserna flyttas när vit gör kort rockad.



1. Bild som visar hur pjäserna flyttas när vit gör lång rockad.

En spelare får aldrig göra ett drag som leder till att motståndaren kan fånga spelarens kung nästa drag. Om en spelare gör ett drag så att motståndarens kung kan fångas på spelarens nästa drag så kallas det för schack, och om motståndaren inte kan förhindra detta vinner spelaren, vilket kallas för schack matt. Ett exempel av schack matt visas i Figur 5.



1. Bild som visar hur vit kan göra schack matt. Kungen hotas att fångas av tornet på andra raden, samtidigt som den inte kan flytta sig utan att bli tagen nästa drag.

Om motståndarens kung inte kan fångas av spelaren nästa drag i nuläget, men samtidigt inte kan göra något drag utan att motståndaren kan fånga kungen nästa drag så blir det lika, vilket även kallas för patt. Ett exempel av patt visas i Figur 6.



1. Bild som visas hur vit kan göra patt. Den svarta kungen hotas inte, men samtidigt kan den inte flytta sig någonstans utan att hotas av tornet eller den vita kungen.

### Portable Game Notation

PGN är ett format som utvecklades för att spara och beskriva schackmatcher, som skulle vara lätt för både datorer och människor att läsa (källa). Ett PGN-dokument kan innehålla ett antal matcher, och varje match innehåller metainformation om matchen, och dragen som utfördes i matchen. Informationen kan gälla när/var matchen spelades och av vilka. Dragen skrivs med algebraisk notation (AN). AN är en notation som beskriver drag så kortfattat till den grad att de inte är tvetydiga.

Raderna numreras från vits håll med bokstäver från a till h, och kolumnerna med siffrorna 1 till 8. En position på spelplanen kan då beskrivas med dess tillhörande rad och kolumn t.ex. e4 eller a2. Ett drag har ett prefix med stor bokstav som beskriver vilken sorts pjäs som flyttades. N för springare, R för torn, B för löpare, Q för drottning, K för kung, medan bonde saknar prefix. Detta följs av positionen som pjäsen flyttades till. Exempel: Ke1, Nf3, c4. Om draget är ett fångande drag så sätts ett x framför positionen som pjäsen flyttades till. De drag som leder till schack har ett plustecken som suffix.

I de fall då ett drag är tvetydigt, t.ex. om två springare på e4 respektive e6 kan flytta till c5, så följs pjäsbokstaven av radkoordinaten eller kolumnkoordinaten beroende på vilken som kan uttrycka draget unikt (Nee5 är inte unikt i detta fall, medan N4e5 är det). Rockad med närmaste torn representeras med ”O-O” och rockad med torn längst bort representeras med ”O-O-O”. Efter det sista draget i matchen visas resultatet 1-0, 0-1, eller ½-½ om vit vann, förlorade, respektive kom lika med svart. I Figur 7 visas ett exempel av en match beskriven i PGN.

[Event "F/S Return Match"]  
[Site "Belgrade, Serbia Yugoslavia|JUG"]  
[Date "1992.11.04"]  
[Round "29"]  
[White "Fischer, Robert J."]  
[Black "Spassky, Boris V."]  
[Result "1/2-1/2"]

1. e4 e5 2. Nf3 Nc6 3. Bb5 a6 {This opening is called the Ruy Lopez.}  
4. Ba4 Nf6 5. O-O Be7 6. Re1 b5 7. Bb3 d6 8. c3 O-O 9. h3 Nb8 10. d4 Nbd7  
11. c4 c6 12. cxb5 axb5 13. Nc3 Bb7 14. Bg5 b4 15. Nb1 h6 16. Bh4 c5 17. dxe5  
Nxe4 18. Bxe7 Qxe7 19. exd6 Qf6 20. Nbd2 Nxd6 21. Nc4 Nxc4 22. Bxc4 Nb6  
23. Ne5 Rae8 24. Bxf7+ Rxf7 25. Nxf7 Rxe1+ 26. Qxe1 Kxf7 27. Qe3 Qg5 28. Qxg5  
hxg5 29. b3 Ke6 30. a3 Kd6 31. axb4 cxb4 32. Ra5 Nd5 33. f3 Bc8 34. Kf2 Bf5  
35. Ra7 g6 36. Ra6+ Kc5 37. Ke1 Nf4 38. g3 Nxh3 39. Kd2 Kb5 40. Rd6 Kc5 41. Ra6  
Nf2 42. g4 Bd3 43. Re6 1/2-1/2

1. En schackmatch i PGN-formatet. Notera att numreringen inte ökar för varje drag, utan varje par av drag.

[Bakgrundskapitlet innehåller teoretisk bakgrund som är intressant för problemområdet. Det inleds ofta med mera generell bakgrund för att mot slutet av bakgrundskapitlet avsluta med underkapitel som är mera specifika för problemet. Bakgrunden ska innehålla referenser.

Bakgrunden ska inledningsvis hållas generell för att i senare avsnitt gå in på mer specifika frågor och närma sig problemformuleringen.

Referenser i texten ska anges enligt harvardsystemet (exempel på referenslista sist i mallen):

Artiklar med en författare kan refereras på följande sätt:

… Ernefeldt (2008) menar att...

… I en tidigare studie (Ernefeldt, 2008) argumenteras det för att...

Två författare:

… Nya synsätt (Salen & Zimmerman, 2005) påverkar hur …

... Salen och Zimmerman (2005) menar att …

Om flera författare: första gången man anger en referens med flera författare kan man ange samtliga för att senare använda m.fl. Om det är väldigt många författare är det tillåtet att endast använd m.fl. varianten.

… Lundell, Lings och Syberfeldt (2011) visade att…

… Lundell m.fl. (2011) visade att…]

## Citat

Citat används när den exakta ordalydelsen i refererade texter behöver lyftas fram. Det är väldigt viktigt att det tydligt framgår var citatet är hämtat (källhänvisning med sidnummer) och texten får inte ändras på något sätt. Kortare citat kan komma löpande i texten och skrivs då omgivet med citattecken.

Ett exempel på kortare citat: Roberts, Furst, Dorn och Isbell (2009, s. 25) beskriver sitt ramverk som ett "web-based mixed-media choose-your-own-adventure-style interactive storytelling system".

Längre citat ska anges som ett eget stycke som är indraget. I mallen kan stilen "Citat" användas. Det kan till exempel se ut så här:

Our framework for evaluating our approach is a web-based mixed-media choose-your-own-adventure-style interactive storytelling system. Our system displays a sequence of authored text and videos that comprise narrative units, or events, that are linked together by explicit decision points for the player.

Roberts, Furst, Dorn & Isbell, 2009, s. 25

Källhänvisningen placeras direkt under citatet och ska inte vara kursiv. I mallen kan stilen "CitatRef" användas.

# Problemformulering

## Problembeskrivning

Syftet med arbetet är att skapa ett system som gör det möjligt att skapa AI-agenter med hjälp av ett antal dokumenterade matcher. Det finns två steg i denna process, att generera falldatabaser, och att utveckla CBR-tolken av AI-agenten, som kan applicera CBR på falldatabaser.

### Generera falldatabaser

För att kunna skapa stora och varierande falldatabaser måste en stor mängd data samlas in från experter. Det finns flera schackmatchdatabaser som innehåller flera tusentals matcher sparade i PGN, men att tolka PGN under körtid kan ta en del processkraft under körtid. För att komma runt detta kan ett antal PGN-matcher konverteras till en falldatabasfil i förväg, som är lättare och snabbare att använda av CBR-tolken.

Positionen för en pjäs representeras med två heltal för positionens rad och kolumn. Ett drag som flyttar en pjäs från en position till en annan representeras av från- och tillpositionen.Ett läge representeras med en tvådimensionell matris, där ett element representerar vilken sorts pjäs som finns på positionen med koordinaterna eller att det inte finns någon. Ett fall representeras av ett läge och dess tillhörande drag.

### CBR-tolken

CBR-tolkens uppgift är att bestämma ett drag att utföra i ett visst läge, givet dess falldatabas. Den gör detta genom att undersöka alla fall i listan, och göra det drag i fallet vars läge är mest likt det givna läget, och vars drag går att utföra i det givna läget. En liknelsefunktion används för att gradera hur pass lika två lägen är. Den är utformad som följande:

Där a och b är två lägen och P är en funktion som ger 1 om båda pjästyperna är lika, och noll om inte (om båda saknar pjäs så är de lika). **(bra med ekvationer? Mer/mindre?)**

CBR-tolken ska integreras i schackmotorn Stockfish(källa)(bara en url eller mer invecklat?)(GPL licenserad, men programmet behöver aldrig distribueras utan källkoden så det är inga problem eller?) och det ska vara möjligt att spela matcher mot motståndare i motorn med olika utvecklade AI-agenter.

[Problemformuleringen ska detaljerat redogöra för det problem examensarbetet är baserat på. Problemet ska först beskrivas övergripande för att senare i problemformuleringen beskrivas mer detaljerat.]

## Metodbeskrivning

[Problemformulering ska även innefatta en metodbeskrivning som ger en tydlig bild av hur frågeställningen ska besvaras, undersökas och arbetet utvärderas.]

# Genomförande/Implementation/ Projektbeskrivning

# Utvärdering

# Avslutande diskussion

Referenser

Campbell, M., Hoane, A. J., & Hsu, F. H. (2002). Deep blue. *Artificial intelligence*, 134(1), s. 57-83.

case based reasoning. (n.d.). The Free On-line Dictionary of Computing. Tillgänglig på Internet: http://dictionary.reference.com/browse/case based reasoning [Hämtad Februari 2, 2015].

Hapgood, F. (December 1982) Computer chess bad-human chess worse. *New Scientist*, 96(1337), s. 827–830.

Richter, M. M., Weber, R. O. (2013) *Case-based reasoning A textbook*. Berlin: Springer-Verlag

Shannon, C. E. (1950) Programming a Computer for Playing Chess. Philosophical Magazine. 41(314)

Electronic Arts (2011) *Battlefield 3* (Version: 1.0) [Datorprogram]. Electronic Arts. Tillgänglig på Internet: http://www.battlefield.com/battlefield3.

Ernefeldt, E. (2008) *Phun - a 2D physics playground - GameDev.net*. 13 February 2008. Phun -a 2D physics playground. Tillgänglig på Internet: http://www.gamedev.net/topic/482775-phun---a-2d-physics-playground/ [Hämtad December 13, 2011].

Lundell, B., Lings, B. & Syberfeldt, A. (2011) Practitioner perceptions of Open Source software in the embedded systems area. *Journal of Systems and Software*. 84 (9), s. 1540–1549.

Roberts, D.L., Furst, M.L., Dorn, B. & Isbell, C.L. (2009) Using influence and persuasion to shape player experiences. *Proceedings of the 2009 ACM SIGGRAPH Symposium on Video Games. Sandbox ’09. New York*, NY, USA, ACM. s. 23–30.

Salen, K. & Zimmerman, E. (red.) (2005) *The game design reader: a Rules of play anthology*. Cambridge, MA: MIT Press.

[Kontrollera noga med sökfunktionen att alla referenser som anges i referenslistan faktiskt finns i brödtexten. Kontrollera på samma sätt med sökfunktionen att alla referenser i brödtexten finns i referenslistan. Var noga med att samtliga referenser formateras enligt Harvard. Ovan finns exempel på ett antal referenser som följer skolans format på referenslistor enligt Martin G Eriksson (se kursens filsamling).

Referenserna ovan är av typen: datorprogram, webbsida, journalartikel, konferensartikel samt bok.

Datorspel ska refereras till som datorprogram även om de spelas på spelkonsol. I automatiserade referenshanteringssystem så används oftast versionsnumret för att skilja på böcker och datorprogram. Spelen bör därför ha ett versionsnummer, speciellt om spelen finns i flera olika versioner.

Referenslistan ska vara uppställd i bokstavsordning, med utgångspunkt från författarens/redaktörens efternamn. Referenserna ska utgöras av en enhetlig uppställning och *inte* delas in i sektioner beroende på typ, webbsidor, böcker och artiklar av olika slag.

Notera att det bara är referenser som *endast* finns på internet som ska vara listade som Tillgänglig på internet. Alltså ska referenserna inte ha ”Tillgänglig på internet” om referensen finns i tryckt form. URL:er ska inte vara understrukna eller blå.]

1. Designdokument etc.

[Appendix ska fungera som referenslistan - dvs det ska finnas referenser till den från texten. Appendix ska inte vara numrerade utan ska namnges med: Appendix A, Appendix B osv. De ska vara sidnumrerade (I, II, III ...) men de ska inte finnas med i innehållsförteckningen. Varje nytt appendix ska börja på toppen av sidan.]