A dynamic metalogic argumentation framework implementation

J. Eriksson Lundström[1] G. Aceto[2] A. Hamfelt[1]

[1]Dept. of Informatics and Media
Uppsala University, Sweden

[2]Faculty of Engineering
University of Bologna, Italy

RuleML2011, 5th International Web Rule Symposium, July 19-21, Barcelona, Spain, 2011

Outline

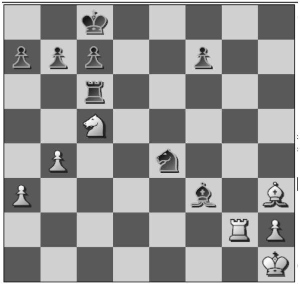
1. Introduction
   - Point of departure
   - Argumentation
   - Methodology
   - Formal setup
2. Implementation
3. Considerations on Game Tree Analysis
   - Complexity
4. Results/Contribution
   - Definitions and properties
   - A Possible Solution - OR/OR-trees
5. Conclusion and Future Work

Approach

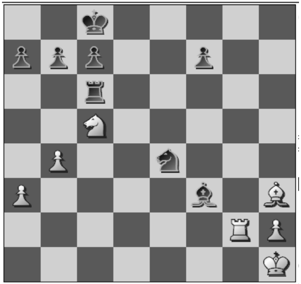
- An approach to logical analysis and formalization of argumentation and dispute as game trees, using a metalogic defeasible framework.
- Formalization of the interplay between the logical layer of defeasible argumentation and the dynamic progression in the argumentation.
- Draws on analogies between tactical chess game notions and notions in adversarial argumentation games.
- Implemented as a non-ground metainterpreter for defeasible logic.

Outline

Chess - A two-party zero-sum board game

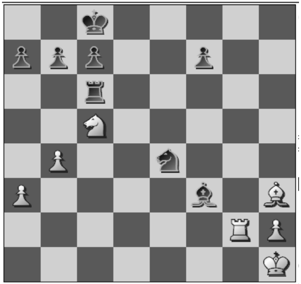


Game Playing in Board Games Like Chess



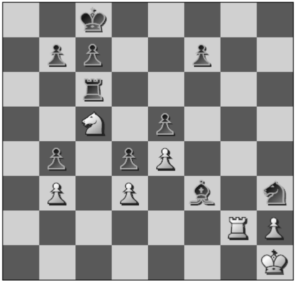
- Involves movement of physical objects on a board
- Two parties
  - white
  - black

Game Playing in Board Games Like Chess

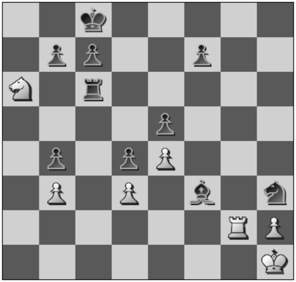


- Involves movement of physical objects on a board
- Two parties
  - white
  - black

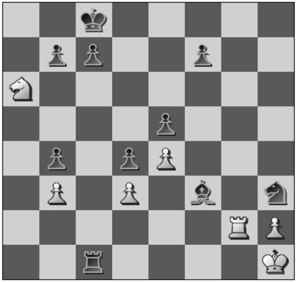
Chess - A two-party zero-sum board game revisited



Chess - A two-party zero-sum board game revisited



Chess - A two-party zero-sum board game revisited



Requires completion criterion

- won
- lost
- draw - infinite path or no legal moves

Algorithmical game-analysis: State space search and game trees

- state space
- a game tree is a directed (acyclic) graph
- game playing can be modelled as a search problem

Relevance

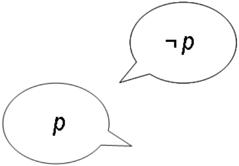
What does chess have to do with defeasible computation of legal rules and norms?

Outline

Argumentation games

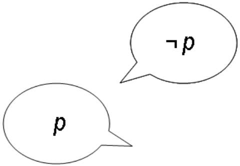
- The proponent and the opponent of a disputed key-claim put forward utterances (speech acts) according to protocol schemes.
- Each agent is equipped with a set of private arguments/knowledge.
- The purpose is to conduct a winning strategy adhering to:
  - admissible arguments for each given state of the argumentation game.
  - appropriate winning/losing conditions in analogy to checkmate conditions in chess.

Adversarial Argumentation like Legal Dispute



- Involves putting forward speech acts/utterances according to protocols for the litigation
- Two parties
  - proponent
  - opponent

Adversarial Argumentation like Legal Dispute



- Involves putting forward speech acts/utterances according to protocols for the litigation
- Two parties
  - proponent
  - opponent

Argumentation Characteristics

Argumentation is:

- *dynamical*
- *defeasible*
- *creative*

Thus:

- Argumentation - an adversarial form of human problem solving.

Argumentation Characteristics

Argumentation is:

- *dynamical*
- *defeasible*
- *creative*

Thus:

- Argumentation - an adversarial form of human problem solving.

Argumentation Characteristics

Argumentation is:

- *dynamical*
- *defeasible*
- *creative*

Thus:

- Argumentation - an adversarial form of human problem solving.

Argumentation Characteristics

Argumentation is:

- *dynamical*
- *defeasible*
- *creative*

Thus:

- Argumentation - an adversarial form of human problem solving.

State Space Search and Game Trees for Argumentation Analysis

- state space
- a game tree is a directed (acyclic) graph
- *argumentation* can be modeled as a search problem with speech acts replacing moves

Outline

Methodology

- Computational Logic
- Metalogic
- Metalogic for formalization of argumentation games

Metalogic for formalization of argumentation games

- Allows for separation of domain knowledge from control knowledge :
  - background knowledge and speech acts forming object language clauses encoded as terms
- Explicit representation of the control knowledge as a metalogic program for managing dynamical aspects eg.
  - dialectic progression
  - nondeterministic nature of the argumentation process

An Example

Definition (gambit)

A *gambit* is a sacrifice by corroborating some sub-argument of the adversary in order to support the own claim.

Gambit in Argumentation - A sample legal case on prescriptive rights

- Content of common knowledge base:
  $possession \Rightarrow ownership$
  $possession \wedge permission \rightarrow \neg ownership$
  $permission \rightarrow possession$

  A sample argumentation game:
- Repository of party 1 (proponent):
  $\Rightarrow possession$
- Repository of party 2 (opponent):
  $\Rightarrow \neg possession$
  $\rightarrow permission$

Gambit in Argumentation - A sample legal case on prescriptive rights

- Content of common knowledge base:
  $possession \Rightarrow ownership$
  $possession \wedge permission \rightarrow \neg ownership$
  $permission \rightarrow possession$

  A sample argumentation game:
- Repository of party 1 (proponent):
  $\Rightarrow possession$
- Repository of party 2 (opponent):
  $\Rightarrow \neg possession$
  $\rightarrow permission$

Gambit in Argumentation - A sample legal case on prescriptive rights

- Content of common knowledge base:
  $possession \Rightarrow ownership$
  $possession \wedge permission \rightarrow \neg ownership$
  $permission \rightarrow possession$

  A sample argumentation game:
- Repository of party 1 (proponent):
  $\Rightarrow possession$
- Repository of party 2 (opponent):
  $\Rightarrow \neg possession$
  $\rightarrow permission$

Argumentation game analysis

- The proponent initiates the argumentation by uttering her sole clause. $\Rightarrow$ *possession*
- The opponent can choose between the two clauses available as speech acts in her repository.

Speech act opt. A	Speech act opt.B
$\Rightarrow \neg$*possession*	$\rightarrow$ *permission*
$\therefore$ The dispute becomes *a draw*	$\therefore$ The game is *won* by the opponent.

Conclusion:
The upshot of this argumentation game analysis is that the opponent should play her second choice, cf. Gambit, to win the argumentation.

Argumentation game analysis

- The proponent initiates the argumentation by uttering her sole clause. $\Rightarrow$ *possession*
- The opponent can choose between the two clauses available as speech acts in her repository.

Speech act opt. A	Speech act opt.B
$\Rightarrow \neg$*possession*	$\rightarrow$ *permission*
$\therefore$ The dispute becomes *a draw*	$\therefore$ The game is *won* by the opponent.

Conclusion:
The upshot of this argumentation game analysis is that the opponent should play her second choice, cf. Gambit, to win the argumentation.

Argumentation game analysis

- The proponent initiates the argumentation by uttering her sole clause. $\Rightarrow$ *possession*
- The opponent can choose between the two clauses available as speech acts in her repository.

Speech act opt. A	Speech act opt.B
$\Rightarrow \neg$*possession*	$\rightarrow$ *permission*
$\therefore$ The dispute becomes *a draw*	$\therefore$ The game is *won* by the opponent.

Conclusion:
The upshot of this argumentation game analysis is that the opponent should play her second choice, cf. Gambit, to win the argumentation.

Argumentation game analysis

- The proponent initiates the argumentation by uttering her sole clause. $\Rightarrow$ *possession*
- The opponent can choose between the two clauses available as speech acts in her repository.

Speech act opt. A	Speech act opt.B
$\Rightarrow \neg$*possession*	$\rightarrow$ *permission*
$\therefore$ The dispute becomes *a draw*	$\therefore$ The game is *won* by the opponent.

Conclusion:
The upshot of this argumentation game analysis is that the opponent should play her second choice, cf. Gambit, to win the argumentation.

Argumentation game analysis

- The proponent initiates the argumentation by uttering her sole clause. $\Rightarrow$ *possession*
- The opponent can choose between the two clauses available as speech acts in her repository.

Speech act opt. A	Speech act opt.B
$\Rightarrow \neg$*possession*	$\rightarrow$ *permission*
$\therefore$ The dispute becomes *a draw*	$\therefore$ The game is *won* by the opponent.

Conclusion:
The upshot of this argumentation game analysis is that the opponent should play her second choice, cf. Gambit, to win the argumentation.

Outline

Metalogical Setup

In predicate logic we specify won and lost with the following two mutually recursive clauses exploring the game tree using existential and universal quantification alternately.

$$won(S) \leftarrow \exists S'(move(S,S') \wedge lost(S'))$$
$$lost(S) \leftarrow \forall S'(move(S,S') \rightarrow won(S'))$$

In Prolog this may become:

$$won(S) \leftarrow move(S,S') \wedge lost(S')$$
$$lost(S) \leftarrow \; not\ defensible(S)$$
$$defensible(S) \leftarrow move(S,S') \wedge \; not\ won(S')$$

move implements the argumentation state transition with a selected speech act according to appropriate contextual and termination criteria for the argumentation.

Metalogical Setup

In predicate logic we specify won and lost with the following two mutually recursive clauses exploring the game tree using existential and universal quantification alternately.

$$won(S) \leftarrow \exists S'(move(S,S') \wedge lost(S'))$$
$$lost(S) \leftarrow \forall S'(move(S,S') \rightarrow won(S'))$$

In Prolog this may become:

$$won(S) \leftarrow move(S,S') \wedge lost(S')$$
$$lost(S) \leftarrow \ not\ defensible(S)$$
$$defensible(S) \leftarrow move(S,S') \wedge \ not\ won(S')$$

move implements the argumentation state transition with a selected speech act according to appropriate contextual and termination criteria for the argumentation.

Interacting with the defeasible logic prover

The game tree explorer is to appeal to the defeasible logic prover with a termination criterion (cf. checkmate) checking whether the key claim or its denial has become irrefutably defeasible in the current argumentation state.

We are inspired by the defeasible logic prover from Governatori *et al.* 2004, is adapted to our dynamic framework by introducing the argumentation state S as parameter.

$$
\begin{aligned}
&\mathit{definitely}(X, S) \leftarrow \mathit{member}(\mathit{strict}(R, X, [Y_i, \ldots, Y_n], S) \\
&\qquad\qquad \wedge \textstyle\bigwedge_i^n \mathit{definitely}(Y_i, S)) \\
&\mathit{defeasible}(X, S) \leftarrow \mathit{definitely}(X, S) \\
&\mathit{defeasible}(X, S) \leftarrow \mathit{negation}(X, X') \wedge \\
&\qquad\qquad \mathit{not}\ \mathit{definitely}(X', S) \wedge \\
&\qquad\qquad \mathit{member}(\mathit{supporter}(R, X, [Y_1, \ldots, Y_n]), S) \wedge \\
&\qquad\qquad \textstyle\bigwedge_i^n \mathit{defeasible}(Y_i, S)
\end{aligned}
$$

Termination criteria

	won	**lost**
pro	$S_i \vdash \kappa$	$S_i \vdash \neg\kappa$
	$S_i \mid\!\sim \kappa \wedge$ opp-rep exhausted in S_{i+1}	$S_i \not\mid\!\sim \kappa \wedge$ prop-rep exhausted in S_{i+1}
	some act leads to lost for opp (at turntaking)	all acts lead to won for opp (at turntaking)
opp	$S_i \vdash \neg\kappa$	$S_i \vdash \kappa$
	$S_i \mid\!\sim \neg\kappa \wedge$ prop-rep exhausted in S_{i+1}	$S_i \mid\!\sim \kappa \wedge$ opp-rep exhausted in S_{i+1}
	some act leads to lost for prop (at turntaking)	all acts lead to won for prop (at turntaking)

Table: Won and lost dialogues

The Defeasible Logic Prover

Point of departure: the Prolog-defeasible logic implementation of Nute 1998

- `flies(X):- bird(X).`
- `flies(X) := bird(X).`
- negation as failure `neg flies(X) :- house(X)` and classical negation `neg/1`
- Strict and defeasible derivation
- rebutting and undercutting attacks
- ambiguity propagation or ambiguity blocking

The Defeasible Logic Prover ctd

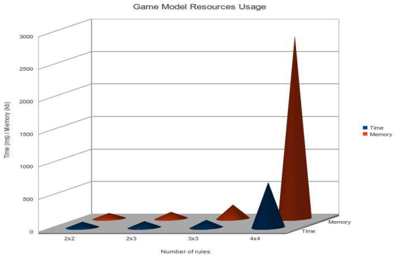
- defeasible incompatibility rules
- defeasible superiority relations
- multiple superiority relations evaluation
- loop detector equipped to handle inconsistent superiority relations.

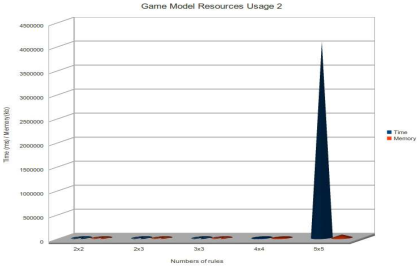
The Argumentation Game Model

- a non-ground representation, a Vanilla-like meta-interpreter
- the control of the game reasoning mechanism captures the nondeterministic nature of an adversarial argumentation dispute
- we implement the argumentation game as an AND/OR-tree.
- we distinguish among each player's private repository of information and a public knowledge called CKB.

Outline

Complexity





(on a Centrino 2 T2300@1.66Ghz, 1Gb Ram, kernel Linux 2.6.32.22-generic, for which the stack limitation was set to unlimited performance.)

Outline

Definitions 1/1

Definition (balanced theory)

We say that a theory T (consisting of the two private knowledge bases) is *balanced* iff the arguments of the parties are equally many for each party or include at most one more rule for the proponent, there is no possibility for a party to put forward several arguments in one turntake, and there is no possibility for a party to refrain from putting forward an argument besides from exhaustion of the private knowledge base.

Definitions 2/2

Definition (persistent theory)

We say that a theory T is *persistent* iff all arguments put forward are unchanged as to their rule strength assignment throughout the dispute, and, all arguments put forward from of the private repositories are a) valid given the protocol during the entire dispute, and b) T is balanced

Properties of AND/OR-trees

- an AND/OR-tree is a proper representation for adversarial argumentation games in the cases when the theory T is persistent.
- when AND/OR tree can be used a theory T can be explored used a depth-first strategy that reduce complexity from O(k!q!) to O(k+q).

Non-Persistent Cases

- *Case of changed rule strength* A non-persistent theory T could contain arguments R of which the rule strength could be changed in a given context and prune the tree accordingly. Depending on the contexts we can have the following states: $R \in T$ and it remains defeasibly derivable; $R \in T$ and it becomes strict;
- *Case of rejected arguments* If an argument is rejected in a given context, in another context it may still be applicable and as a result, the system reaches a different conclusion.
- *Case of an unbalanced theory* If we have an unbalanced theory it should be noted that branches could reach different final steps caused by the fact that they employ different sets of arguments.

Outline

OR/OR-trees

- deal with sub-chains of arguments and analyzes its conclusions rather than evaluating the relations (AND-level) among branches

Issues of OR/OR-trees

- no means of heuristic search/heuristic function to use for pruning of the OR/OR-tree is available
- well established pruning algorithms such as the alpha beta pruning applied to branch and bound algorithms may be still be applied for OR/ORtrees as these algorithms operate on the meta-semantic layer of the heuristic function over the tree representation.

Implementation

Example	Elapsed time
Tweety AND/OR 4x2 rules	0.020s
Tweety OR/OR 4x2 rules	0.070s
Tweety OR/OR 4x3 rules	0.310s

Conclusion

- We motivated our approach to the logical analysis and formalization of legal argumentation
- We discussed limitations of AND/OR-tree representations
- We have introduced OR/OR-trees and elaborated on these as a viable solution for implementations of adversarial argumentation games by allowing general cases to be executed without building the complete argumentation derivation tree.
- We (briefly) introduced a full implementation of defeasible adversarial argumentation that incorporates the OR/OR-trees as a complement to AND/OR-trees.

Future Work

- Comprehensively evaluate OR/OR-trees, especially for larger examples.
- Optimization of the implementation.
- Analysis of strategic aspects of argumentation games.
- Preprocessing mechanism by means of reflection as formalized in provability relation to be able to filter acceptable rules.
- Evaluation of extended real-world examples.

Thank You for Your Attention!

A dynamic metalogic argumentation framework implementation

J. Eriksson Lundström[1] G. Aceto[2] A. Hamfelt[1]

[1]Dept. of Informatics and Media
Uppsala University, Sweden

[2]Faculty of Engineering
University of Bologna, Italy

RuleML2011, 5th International Web Rule Symposium, July 19-21, Barcelona, Spain, 2011