# Representation of Coalitional Games with Algebraic Decision Diagrams (Extended Abstract)

Karthik .V. Aadithya
Department of Electrical Engineering and Computer Sciences
The University of California, Berkeley, CA, USA
kv.aadithya@gmail.com

Tomasz P. Michalak
School of Electronics and Computer Science
University of Southampton, UK
tpm@ecs.soton.ac.uk

Nicholas R. Jennings
School of Electronics and Computer Science
University of Southampton, UK
nrj@ecs.soton.ac.uk

## ABSTRACT

With the advent of algorithmic coalitional game theory, it is important to design coalitional game representation schemes that are both compact and efficient with respect to solution concept computation. To this end, we propose a new representation for coalitional games, which is based on Algebraic Decision Diagrams (ADDs). Our representation is fully expressive, compact for many games of practical interest, and enables polynomial time Banzhaf Index, Shapley Value and core computation.

## Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multi-Agent Systems; I.2.4 [**Knowledge representation formalisms and methods**]; F.2 [**Theory of Computation**]: Analysis of Algorithms and Problem Complexity

## General Terms

Algorithms, Theory, Economics

## Keywords

Coalitional game theory, Algebraic Decision Diagrams

## 1. ALGEBRAIC DECISION DIAGRAMS

ADDs are highly optimized representations for ordered decision trees on boolean decision variables. In general, a decision tree is of size exponential in the number of decision variables. However, the observation is that *most practically encountered decision trees contain a significant amount of duplication*, i.e., there exist many subtrees within the decision tree that are isomorphic to one another.

For example, consider the ordered decision tree shown in Fig. 1 (a). In the figure, each terminal node (leaf node) is labelled with a real number, while each non-terminal node (decision node) is labelled with a boolean decision variable. Therefore, each decision node has exactly two edges leading away from itself: a dashed edge (leading to the decision node's *left child*) corresponding to the decision variable being set to `FALSE`, and a solid edge (leading to the decision node's *right child*) corresponding to the decision variable being set to `TRUE`. It is readily seen that this decision tree contains significant duplication (e.g., consider the identical sub-trees rooted at the nodes labelled $x_3$, as pointed out in Fig. 1 (a)).

**Cite as:** Representation of Coalitional Games with Algebraic Decision Diagrams (Extended Abstract), Karthik .V. Aadithya, Tomasz P. Michalak and Nicholas R. Jennings, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp.1121-1122.


The fundamental idea behind the ADD is that: *it is wasteful to maintain multiple identical copies of duplicated subtrees; instead, such isomorphic subtrees should be merged together*, thereby resulting in a much smaller (but equivalent) directed acyclic graph (DAG) [1, 2]. To this end, three *reduction rules* have been formulated for compressing a decision tree into a DAG [2]:

**Rule 1:** *Merge isomorphic terminal nodes.* That is, if two terminal nodes $u$ and $v$ carry the same value, delete $u$ and redirect all its incoming edges to $v$.

**Rule 2:** *Delete dummy nodes.* That is, if the left child of a decision node $u$ is the same as its right child, then delete $u$ and redirect all its incoming edges to this (only) child.

**Rule 3:** *Merge isomorphic decision nodes.* That is, if two decision nodes $u$ and $v$ have (a) identical labels, (b) identical left children and (c) identical right children, delete $u$ and redirect all its incoming edges to $v$.

For example, the decision tree of Fig. 1 (a) contains four isomorphic terminal nodes with value 1, six isomorphic terminal nodes with value 4 and four isomorphic terminal nodes with value 9. To get rid of all this duplication, Rule 1 (above) is applied 3+5+3=11 times in succession, resulting in the DAG of Fig. 1 (b). This DAG is not free from isomorphic nodes either. In fact, as shown in Fig. 1 (b), it has two sets of three isomorphic nodes each, which can be merged by applying Rule 3 four times in succession, thereby resulting in the DAG of Fig. 1 (c). This DAG again contains two isomorphic nodes (as shown in Fig. 1 (c)), which are merged by a single application of Rule 3. This results in the DAG of Fig. 1 (d), which is *maximally compressed* in the sense that it cannot be made smaller by any further application of Rules 1-3. Such a *maximally compressed* DAG (which can be shown to be a unique and canonical representation for the original decision tree) is called an Algebraic Decision Diagram.

## 2. REPRESENTING COALITIONAL GAMES

This section describes how ADDs can be used to represent coalitional games.

A coalitional game $g$ is defined as a tuple $g = \langle N, \nu \rangle$, where $N = \{x_1, x_2, \ldots, x_n\}$ is a set of agents and $\nu : 2^N \to \mathbb{R}$ is a characteristic function that maps every subset (or *coalition*) of $N$ to a real number, with $\nu(\emptyset) = 0$ [3].

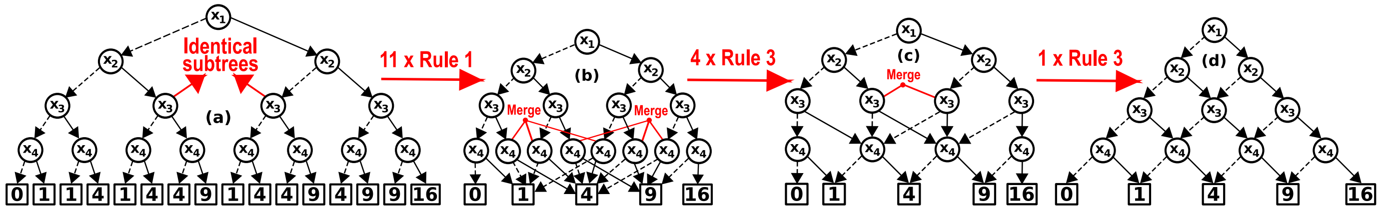


**Figure 1: Constructing an ADD from a decision tree.**

Note that the set of all coalitions of $N$ is in one to one correspondence with the set of truth assignments of the $n$ boolean variables $\{x_1, x_2, \ldots, x_n\}$, with the boolean variable $x_i$ being set to `TRUE` (`FALSE`) accordingly as the agent $x_i$ is present (absent) in the coalition. Thus, in effect, the characteristic function $\nu$ is a real-valued function of the boolean variables $\{x_1, x_2, \ldots, x_n\}$. So $\nu$ can be represented by an ordered decision tree over the same boolean variables, and this decision tree can be further compacted into an ADD (using the 3 rules of the previous section).

Therefore, every coalitional game $g$ can be represented by an ADD. For example, the coalitional game played by the set of 4 agents $N = \{x_1, x_2, x_3, x_4\}$, where $\nu(C) = (\text{size of } C)^2$ for every $C \subseteq N$, is represented by the ADD of Fig. 1 (d).

# 3. FORMAL DEFINITION

We now formally define our ADD-based representation for coalitional games.

In the ADD representation, a coalitional game $g = \langle N, \nu \rangle$ is specified by a tuple $\langle N, <, G(V, E, L_V, L_E) \rangle$, where

- $N$ is a finite set (the set of agents)
- $<$ is a strict total order defined on $N$
- $G(V, E, L_V, L_E)$ is a vertex-labelled, edge-labelled, directed acyclic graph (the ADD) that satisfies the following:
  - $V$ is a finite set (the set of ADD vertices)
  - $E \subset V \times V$ is a finite set (the set of ADD edges)
  - $L_V : V \to N \cup \mathbb{R}$ is a function that labels each ADD vertex with either an agent (for non-terminal vertices) or a real number (for terminal vertices)
  - $L_E : E \to \{$`SOLID, DASHED`$\}$ is a function that labels each ADD edge as either `SOLID` or `DASHED`
  - $G$ contains exactly one root/source vertex, i.e., exactly one vertex of in-degree zero
  - For all non-terminal vertices $u$ and $v$, if $(u, v) \in E$, then $L_V(u) < L_V(v)$
  - For each non-terminal vertex $u$, there exists exactly one vertex $v$, called the left child of $u$, such that $(u, v) \in E$ and $L_E((u, v)) =$ `DASHED`
  - For each non-terminal vertex $u$, there exists exactly one vertex $v$, called the right child of $u$, such that $(u, v) \in E$ and $L_E((u, v)) =$ `SOLID`
  - The reduction rules 1-3 of Section 1 cannot be used to simplify $G$ any further.

# 4. $\nu$(C) EVALUATION

We now formally outline an algorithm for evaluating the characteristic function in the ADD-based coalitional game representation.

Given an ADD representation $\langle N, <, G(V, E, L_V, L_E) \rangle$ for a coalitional game $g$, and a coalition $C \subseteq N$. Algorithm 1 formally specifies how to evaluate the characteristic function value $\nu(C)$.

**Algorithm 1:** Characteristic function evaluation with ADDs

**Inputs**: (a) Coalitional game $\Gamma = \langle N, <, G(V, E, L_V, L_E) \rangle$
(b) Coalition $C \subseteq N$.
**Output**: The characteristic function value $\nu(C)$.

```
ADDNode u = the root (source node) of G;
while u is not a terminal node of G do
    if agent L_V(u) ∉ C then
        u = left child of u;
    else
        u = right child of u;
    end
end
return L_V(u);
```

# 5. NOTEWORTHY PROPERTIES OF ADDS

Our ADD representation for coalitional games possesses the following properties:

**1.** ADDs are fully expressive (i.e., can be used to represent any coalitional game)

**2.** There are many games of practical interest whose ADD representations are exponentially more compact than their MC-Net representations (MC-Nets are described in [4]).

**3.** Banzhaf Indices and Shapley Values of all agents can be computed in time polynomial in the size of the ADD representation.

**4.** ADDs enable polynomial time algorithms for several core-related questions, such as testing if a given vector is in the core, checking if the core is empty and computing the smallest $\epsilon$ such that the strong-$\epsilon$ core is non-empty.

**5.** ADDs enable polynomial time Cost of Stability [5] computation.

Due to space constraints, we are unable to prove the above properties in this paper. Instead, we refer the reader to [6].

# 6. REFERENCES


[1] R.I. Bahar, E.A. Frohm, C.M. Gaona, G.D. Hachtel, E. Macii, A. Pardo, and F. Somenzi. Algebraic Decision Diagrams and their applications. *Formal Methods in System Design*, 10(2-3):171–206, 1997.

[2] R.E. Bryant. Symbolic boolean manipulation with ordered Binary Decision Diagrams. *ACM Computing Surveys*, 24(3):293–318, 1992.

[3] A. Rapoport. *N-person game theory: Concepts and applications*. Dover Pubs., 2001.

[4] S. Ieong and Y. Shoham. Marginal contribution nets: A compact representation scheme for coalitional games. In *ACM EC '05*, pages 193–202, 2005.

[5] Y. Bachrach, E. Elkind, R. Meir, D. Pasechnik, M. Zuckerman, J. Rothe, and J. Rosenschein. The cost of stability in coalitional games. In *Algorithmic Game Theory*, volume 5814 of *Lecture Notes in Computer Science*, pages 122–134. Springer, Berlin, 2009.

[6] K. V. Aadithya, T. P. Michalak, and N. R. Jennings. Representation of coalitional games with Algebraic Decision Diagrams. Technical report, Department of Electrical Engineering and Computer Sciences, The University of California, Berkeley, CA, USA, 2011. `http://www.eecs.berkeley.edu/Pubs/TechRpts/2011/EECS-2011-8.html`.