Chapter 13
Towards a Logical Analysis of *Adjusted Winner*

Eric Pacuit

13.1 Introduction

It is often convenient to view a computational procedure, or a program, as a relation on a set of states, where a state can be thought of as a function that assigns a value to every possible variable and a truth value to all propositions. This idea was proposed by Pratt [19] and extends the work of of Floyd [4] and Hoare [6]. Harel, Kozen and Tiuryn [5] provide a very thorough discussion of computational procedures from this point of view. In [12], Rohit Parikh suggests that a similar framework can be developed for studying *social* procedures, such as fair division algorithms or voting protocols.

In fact, the first example of such an analysis can be found in an earlier paper of Parikh [11]. In [11], a **PDL**-style logic, called **game logic**, is developed for reasoning about multi-agent strategic situations. It is then used to show that the Banach-Knaster last diminisher procedure to fairly divide a cake is "correct". Here "correct" means that each agent has a *strategy* to ensure that it receives a piece of the cake that is fair *from its point of view*. See [11, 16] for a discussion of this result and the current state of affairs of game logic.

Recently, there have been a number of attempts to answer Parikh's challenge to develop logical methods for studying social procedures. The different approaches can roughly be divided into two categories. The first category makes use of model checking techniques to verify that a given game-theoretic mechanism satisfies a particular *specification* (written in some logical language). The idea is to draw an analogy with the formal verification of computer systems via model checking techniques where the desired specifications are expressed in some temporal logic. In the game-theoretic setting, modal strategy logics, such as alternating temporal logic (see [1]) or coalitional logic (see [14]), are used to express the desired specification.

Eric Pacuit
Tilburg University, Tilburg Institute for Logic and Philosophy of Science, Warandelaan 2, 5037 AB Tilburg, The Netherlands, e-mail: e.j.pacuit@uvt.nl

See (Pauly M., and Wooldridge M. *Logic for Mechanisam Design – a Main festo.* Unpublished Manuscript.) [19], the recent dissertation [18] and references therein for details of this approach.

The second category of papers follow in the footsteps of Floyd [4], Hoare [6] and Pratt [19] and develops a formal calculus to reason about social procedures. This is the direction that Parikh took in [11] and more recently Pauly in [15]. A discussion of this approach can be found in Section 13.3.

In this paper, we apply the methods of [15] to fair division algorithms. In particular, we look at *Adjusted Winner* (*AW*) – an algorithm for "fairly " dividing n goods among two people invented by Steven Brams and Alan Taylor. See [2] for a discussion of *AW* and an excellent overview of other fair division algorithms. We will see that the techniques from [15] cannot be directly applied to the analysis of Adjusted Winner. The paper is organized as follows. The next section describes the Adjusted Winner procedure to "fairly" divide n divisible goods between two people. Section 13.3 outlines how to extend Pauly's framework to reason about Adjusted Winner and raises some difficulties.

13.2 The *Adjusted Winner* Procedure

We begin with an example that illustrates how *AW* works. Suppose there are two players, called Ann (A) and Bob (B), and n (divisible[1]) goods ($G_1, \dots, G_n$) which must be distributed to Ann and Bob. The goal of the Adjusted Winner algorithm is to *fairly* distribute the n goods between Ann and Bob. We begin by discussing an example which illustrates the Adjusted Winner algorithm.

Suppose Ann and Bob are dividing three goods: G_1, G_2, and G_3. *Adjusted Winner* begins by giving both Ann and Bob 100 points to divide among the three goods. Suppose that Ann and Bob assign these points according to the following table.

Item	Ann	Bob
G_1	$\underline{10}$	7
G_2	$\underline{65}$	43
G_3	25	$\underline{50}$
Total	100	100

The first step of the procedure is to give G_1 and G_2 to Ann since she assigned more points to those items, and item G_3 to Bob. However this is not an equitable outcome since Ann has received 75 points while Bob only received 50 points (each according to their personal valuation). We must now transfer some of Ann's goods to Bob. In order to determine which goods should be transfered from Ann to Bob, we look at the ratios of Ann's valuations to Bob's valuations. For G_1 the ratio is $10/7$ 1.43

[1] Actually all we need to assume is that *one* good is divisible. However, since we do not know before the algorithm begins *which* good will be divided, we assume all goods are divisible. See [2, 3, 9] for a discussion of this fact.

and for G_2 the ratio is $65/43 \quad 1.51$. Since 1.43 is less than 1.51, we transfer as much of G_1 as needed from Ann to Bob[2] to achieve equitability.

However, even giving all of item G_1 to Bob will not create an equitable division since Ann still has 65 points, while Bob has only 57 points. In order to create equitability, we must transfer part of item G_2 from Ann to Bob. Let p be the proportion of item G_2 that Ann will keep. p should then satisfy

$$65p = 100 - 43p$$

yielding $p = 100/108 = 0.9259$, so Ann will keep 92.59% of item G_2 and Bob will get 7.41% of item G_2. Thus both Ann and Bob receive 60.185 points. It turns out that this allocation (Ann receives 92.59% of item G_2 and Bob receives all of item G_1 and item G_3 plus 7.41% of item G_2) is *envy-free*, *equitable* and *efficient*, or *Pareto optimal*. In fact, Brams and Taylor show that Adjusted Winner *always* produces such an allocation [2]. We will discuss these properties in more detail below.

Suppose that $G_1, \ldots, G_n$ is a fixed set of goods, or items. A **valuation** of these goods is a vector of natural numbers $a_1, \ldots, a_n$ whose sum is 100. Let , , ,... denote possible valuations for Ann and , , ,... denote possible valuations for Bob. An **allocation** is a vector of n real numbers where each component is between 0 and 1 (inclusive). An allocation $= s_1, \ldots, s_n$ is interpreted as follows. For each $i = 1, \ldots, n$, s_i is the proportion of G_i given to Ann. Thus if there are three goods, then $1, 0.5, 0$ means, "Give all of item 1 and half of item 2 to Ann and all of item 3 and half of item 2 to Bob." Thus *AW* can be viewed as a function that accepts Ann's valuation and Bob's valuation and returns an allocation . It is not hard to see that every allocation produced by *AW* will have a special form: all components except one will be either 1 or 0.

We now give the details of the procedure. Suppose that Ann and Bob are each given 100 points to distribute among n goods as he/she sees fit. In other words, Ann and Bob each select a valuation, $= a_1, \ldots, a_n$ and $= b_1, \ldots, b_n$ respectively. For convenience rename the goods so that

$$a_1/b_1 \quad a_2/b_2 \quad \cdots a_r/b_r \quad 1 > a_{r+1}/b_{r+1} \quad \cdots a_n/b_n$$

Let $/$ be the above vector of real numbers (after renaming of the goods). Notice that this renaming of the goods ensures that Ann, based on her valuation , values the goods $G_1, \ldots, G_r$ at least as much as Bob; and Bob, based on his valuation , values the goods $G_{r+1}, \ldots, G_n$ more than Ann does. Then the *AW* algorithm proceeds as follows:

1. Give all the goods $G_1, \ldots, G_r$ to Ann and $G_{r+1}, \ldots, G_n$ to Bob. Let X, Y be the number of points received by Ann and Bob respectively. Assume for simplicity that $X \quad Y$.
2. If $X = Y$, then stop. Otherwise, transfer a portion of G_r from Ann to Bob which makes $X = Y$. If equitability is not achieved even with all of G_r going to Bob, transfer $G_{r-1}, G_{r-2}, \ldots, G_1$ in that order to Bob until equitability is achieved.

2 When the ratio is closer to 1, a unit gain for Bob costs a smaller loss for Ann.

Thus the *AW* procedure is a function from pairs of valuations to allocations. Let AW(,) = mean that is the allocation given by the procedure *AW* when Ann announces valuation and Bob announces valuation . In [2, 3], it is argued that *AW* is a "fair" procedure, where fairness is judged according to the following properties.

Let $= a_1,\dots,a_n$ and $= b_1,\dots b_n$ be valuations for Ann and Bob respectively. An allocation $= s_1,\dots,s_n$ is

- **Proportional** if both Ann and Bob receive at least 50% of their valuation. That is, $^n_{i=1} s_i a_i$ 50 and $^n_{i=1}(1-s_i)b_i$ 50.
- **Envy-Free** if no party is willing to give up its allocation in exchange for the other player's allocation. That is, $^n_{i=1} s_1 a_i$ $^n_{i=1}(1-s_i)a_i$ and $^n_{i=1}(1-s_i)b_i$ $^n_{i=1} s_i b_i$.
- **Equitable** if both players receive the same total number of points. That is $^n_{i=1} s_i a_i =$ $^n_{i=1}(1-s_i)b_i$.
- **Efficient** if there is no other allocation that is strictly better for one party without being worse for another party. That is for each allocation $= s_1,\dots,s_n$ if $^n_{i=1} a_i s_i >$ $^n_{i=1} a_i s_i$, then $^n_{i=1}(1-s_i)b_i <$ $^n_{i=1}(1-s_i)b_i$. (Similarly for Bob).

In order to simplify notation, let $V_A($, $)$ be the total number of points Ann receives according to valuation and allocation and $V_B($, $)$ the total number of points Bob receives according to valuation and allocation .

It is not hard to see that for two-party disputes, proportionality and envy-freeness are equivalent. For a proof, notice that

$$^n_{i=1} a_i s_i + {}^n_{i=1} a_i(1-s_i) = {}^n_{i=1} a_i s_i + {}^n_{i=1} a_i - {}^n_{i=1} a_i s_i = 100$$

Then if is envy free for Ann, then $^n_{i=1} a_i s_i$ $^n_{i=1} a_i(1-s_i)$. Hence, 2 $^n_{i=1} a_i s_i$ $^n_{i=1} a_i = 100$. And so, $^n_{i=1} a_i s_i$ 50. The argument is similar for Bob.

Conversely, suppose that is proportional. Then since $^n_{i=1} a_i s_i$ 50, $^n_{i=1} a_i s_i +$ $^n_{i=1} a_i s_i$ $100 =$ $^n_{i=1} a_i$. Then $^n_{i=1} a_i s_i +$ $^n_{i=1} a_i s_i -$ $^n_{i=1} a_i$ 0. Hence, $^n_{i=1} a_i s_i -$ $^n_{i=1} a_i(1-s_i)$ 0. And so, $^n_{i=1} a_i s_i$ $^n_{i=1} a_i(1-s_i)$. The proof is similar for Bob.

Returning to *AW*, it is easy to see the *AW* only produces equitable allocations (equitability is essentially built in to the procedure). Brams and Taylor go on to show that *AW*, in fact, satisfies all of the above properties.

Theorem 13.1 (Brams and Taylor [2]) *AW produces an allocation of the goods based on the announced valuations that is efficient, equitable and envy-free.*

A formal proof of this Theorem is provided in [2]. See also [9] for a number of new results about the Adjusted Winner procedure.

13.3 Towards a Logical Analysis

Hoare logic contains expression of the form $\{P\}$ $\{Q\}$ where is intended to be a program and P and Q are intended to be pre- and post-conditions respectively. The intended interpretation is that if starts in some state satisfying P, then halts (*if* it does halt) in a state satisfying Q. Here the program is a formal expression in some *programming language* (for example, the WHILE-language) and P and Q are formulas in some logical language (say first-order logic). Details are given below. Pauly's key idea in [16] is to extend the formal programming language with expressions $\mathsf{ch}_{\mathscr{A}}(\{x_i \mid i \quad \mathscr{A}\})$ intended to mean

"*each agent* $i \quad \mathscr{A}$ *independently chooses a value for the variable* x_i."

It is assumed that the agents make their choice *simultaneously*. Thus, $\mathsf{ch}_{\mathscr{A}}(\{x_i \mid i \quad \mathscr{A}\})$ is best thought of as a *strategic game form* (cf. [8]) where the choices for each agent $i \quad \mathscr{A}$ is the set of possible values for the variable x_i.

13.3.1 Relevant Details

Pauly extends the work of Hoare to develop a formal calculus for reasoning about game theoretic mechanisms. In the interest of space, we will not go into full details of Pauly's framework, but rather sketch the main ideas (the interested reader can consult [15]).

The Mechanism Programming Language: The mechanism programming language (*MPL*) is a simple extension of the well-known WHILE-language (cf. [7]). Assume $\mathscr{A}$ is a non-empty set of agents, $\mathscr{V}$ a set of variables, $\mathbb{F}$ a set of function symbols and $\mathscr{R}$ a set of relation symbols. **Terms** t and **boolean expressions** are defined in the usual way:

$$\begin{aligned} t &=_{\text{def}} x \mid f^k(t_1,\ldots,t_k) \\ &=_{\text{def}} \quad \mid R^k(t_1,\ldots,t_k) \mid \neg \quad \mid \quad _1 \quad _2 \end{aligned}$$

where $f \quad \mathbb{F}$ and $R \quad \mathscr{R}$ have arity $k \quad \mathbb{N}$. These boolean expressions and terms are used to create *game expressions*.

Definition 13.1 A **game expression** of *MPL* is generated according to the following grammar:

$$\begin{aligned} &=_{\text{def}} x := t \mid \quad _1; \quad _2 \mid \mathsf{if} \quad \mathsf{then} \quad _1 \; \mathsf{else} \quad _2 \mid \mathsf{while} \quad \mathsf{do} \quad \mid \\ &\quad \mathsf{ch}_{\mathscr{A}}(\{x_i \mid i \quad \mathscr{A}\}) \end{aligned}$$

where t is a term and a boolean expression.

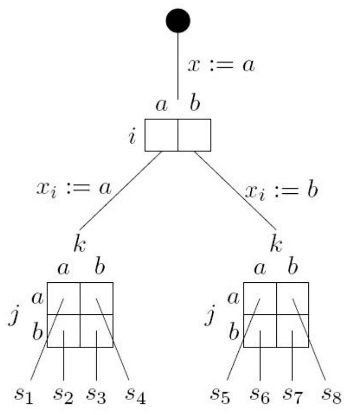
Pauly shows that this language is quite general in the sense that it can be used to describe a large number of game-theoretic mechanisms. For example, a solution to Solomon's Dilemma and various auctions can be described in this language (see [15] for details).

Semantics: As in Hoare logic, a **partial correctness assertion** has the form $\{P\}\ \ \{Q\}$, where P and Q are *predicates*. The intended interpretation is: *given an initial state* s_0*, there is a* subgame perfect equilibrium[3] *of the* game *described by consistent with the predicate* Q. We now make this precise.

Start with a standard first-order model containing a domain D and an interpretation $\mathscr{I}$ (which interprets elements of $\mathbb{F}$ and $\mathscr{R}$ as functions and relations over D respectively). A **state** is a function $s : \mathscr{V} \quad D$. Given an interpretation $\mathscr{I}$ and an initial state s_0, game expressions are interpreted as *extensive game forms*,[4] denoted $G(s_0, \ , \mathscr{I})$. Instead of giving a formal definition, we work through a simple example. Suppose that the domains consists of two elements $D = \{a, b\}$, there are four variables $\mathscr{V} = \{x, x_i, x_j, x_k\}$ and three agents $\mathscr{A} = \{i, j, k\}$. Consider the game expression

$$(x := a); \mathsf{ch}_{\{i\}}(\{x_i\}); \mathsf{ch}_{\{j,k\}}(\{x_j, x_k\}).$$

Suppose the initial state is $s_0 : \mathscr{V} \quad D$ is given by $s_0(v) = b$ for all $v \quad \mathscr{V}$. The game form corresponding to this game expression is pictured below:



Note that the matrices in the above tree represent the fact that agent j and agent k must *simultaneously* choose values for the variables x_j and x_k respectively. In the above picture, each s_i is a state where, for example, s_2 is the following function: $x \quad a$; $x_i \quad a$; $x_j \quad b$; and $x_k \quad a$. The if and while expressions have the usual interpretation (branching and iteration). Formally, game expressions are interpreted as extensive game forms with perfect information and simultaneous moves (cf. [8], Section 6.3.2).

As we just noted, game expressions do not provide any information about the agents' preferences over the terminal histories. This information is provided by the

[3] A strategy profile in an extensive game is a **subgame perfect equilibrium** if no agent has a reason to deviate from the profile *and this is true for all subgames*. Consult [8] for details.

[4] That is, an extensive game (i.e., a game tree) without the preferences over the outcomes.

pre- and post-conditions (P and Q above). To that end, elements of the domain are interpreted as possible outcomes and it is assumed that each agent has a (reflexive, transitive and complete) preference over the domain. A **predicate** is any[5] collection of states. Let o be an element of the domain. An ***e*-state** is a pair (s,o) and a ***e*-predicate** is any set of e-states. An e-predicate P can be used to turn a semi-game into a game as follows. For each (s,o) P, let f_o be the function that assigns o to each terminal history whose last state is s. Given an e-predicate P, let $\hat{P}$ denote the set of such functions. In general, for a given semi-game and e-predicate P, $\hat{P}$ may be empty or contain more than one function. What is important is that each f_o $\hat{P}$ turns the semi-game under consideration into a game. Let P and Q be e-predicates. Given this machinery we can be more precise about what it means to say that $\{P\}$ $\{Q\}$ is **valid in some first-order model**:

For each (s,o) P there is a outcome function f $\hat{Q}$ and a strategy profile such that in the game generated from and f, is a subgame perfect equilibrium and the last state on is mapped by f to o.

13.3.2 Formalizing *AW*

Describing *AW* in the above mechanism programming language is straightforward:

```
ch$_{\mathscr{A}}(\{x_a,x_b\})$;
$s :=$ wta$(x_a,x_b)$;
while $\neg Eq(s,x_a,x_b)$ do
    $s :=$ t$(s,x_a,x_b)$;
```

In the above program, $\mathscr{A} = \{a,b\}$ is the set of agents (Ann and Bob), the variables x_a and x_b are intended to represent Ann and Bob's valuations respectively and s represents the current allocation. To make this formal, it will be convenient to work in a two-sorted first-order structure with an allocation sort and a valuation sort. In what follows, we use $s,t,s_1,s_2,\ldots$ for allocation variables and $x,y,x_1,x_2,\ldots$ for valuation variables. The intended interpretation of the functions wta and t and the relation Eq is as expected:

- wta(x_a,x_b) is intended to represent the *winner take all* initial allocation given Ann's valuation x_a and Bob's valuation x_b. That is, wta will be interpreted as a function from pairs of valuations to allocations where the agent who assigns the most points to a particular item receives that item.
- t(s,x_1,x_2) is intended to represent the one-step transfer of goods as described in Section 13.2. The exact details of which good is transfered from which agent is described Section 13.2, so we will not repeat it here.

[5] Note that Pauly does *not* restrict to definable sets here. Thus he is provided an *extensional* semantics rather than an intensional semantics. Issues related to this, such as whether or not certain properties are expressible in a logical language (say first-order logic), is left for future work. We agree with Pauly that, although this raises some interesting questions, it will only complicate the current discussion.

- The relation Eq is intended to represent *equality* of each agents' valuation with respect to the current allocation. That is, $Eq(s, x_a, x_b)$ will hold whenever the allocation s is *equitable* with respect to the valuations x_a and x_b. Again, see Section 13.2 for details.

We also include in our language relation symbols Ef and *Pe* intended to mean envy-free and Pareto-efficient respectively. Formally, Ef will hold between an allocation and a pair of valuations and provided is envy-free with respect to and (see the definition in Section 13.2). Similarly for Pareto efficiency: *Pe* will hold[6] for , and provided the allocation is *Pareto efficient* with respect to and .

13.3.3 Discussion

After Ann an Bob make their choice, it is clear what we have to prove in order to show *AW* is "correct": we must show that after the `while`-loop the allocation contained in s is envy-free, equitable and Pareto efficient (with respect to the allocations chosen by Ann and Bob). This is exactly what Theorem 13.1 says.

The proof from [2] proceeds by first showing that *AW* produces a Pareto efficient allocation then noting that all Pareto efficient and equitable outcomes must be envy-free. Thus the real work is in showing that *AW* produces and Pareto efficient allocation. In this setting, this amounts to finding an appropriate *loop-invariant*. Here we can use the predicate $Pe(s, x_1, x_2)$ defined above. Thus the crucial part of the proof of Theorem 13.1 is showing that the *winner takes all* procedure produces a Pareto efficient allocation and that Pareto efficiency is preserved under repeated applications of the transfer function. Although not phrased this way, the relevant details can be found in [2] (Theorem 4.1, pp. 85–94).

Note that the discussion above is relevant *after* Ann and Bob have made their choices. However, the correctness assertions in both Parikh's framework and Pauly's framework are about outcomes that the agents *can achieve*. In [11], the Banach-Knaster last diminisher procedure[7] is proven correct by showing that a certain formula (in the language of game logic) is *derivable*. This formula essentially states that if certain preconditions are satisfied (i.e., the cake is big enough for the group) then each agent has a *strategy* to ensure it receives a piece of the cake that is fair from its point of view. Pauly's notion of correctness is similar except that there is

[6] Note that *Pe* is an atomic relation symbol. Alternatively, we may first define a relation (, ,) (, ,) iff either $_i\ _i\ _i >\ _i\ _i\ _i$ and $_i(1-\ _i)\ _i\ \ _i(1-\ _i)\ _i$ or $_i\ _i\ _i\ \ _i\ _i\ _i$ and $_i(1-\ _i)\ _i >\ _i(1-\ _i)\ _i$. Then the we can define the predicate *Pe* as follows: *Pe*(, ,) := s((, ,) (s, ,)). However, this formula is not in the language we described above as a quantifier is involved.

[7] In this cake-cutting algorithm, the first agent cuts a slice of the cake he considers fair. This piece is then inspected by each of the remaining agents in turn. Each agent can decide either leave the piece as is or diminish the piece and return the extra portion to the main part of the cake. The last person to diminish the piece receives that piece of the cake (or if no-one diminishes the piece, the cutter receives the piece). See [2] for details.

an additional requirement that the strategy profile must satisfy a certain equilibrium concept.

Formally verifying the correctness of Adjusted Winner in Pauly's framework provides us with an interesting challenge. The main issue is finding the correct notion of equilibrium for Ann and Bob. Verifying *AW* is the above framework can be broken down into two different tasks:

1. Given two valuations from Ann and Bob, prove that *AW* produces a equitable, envy-free and efficient allocation.
2. Argue that there is a joint strategy which is a subgame perfect equilibrium in the game generated by the *AW* program as described above. Of course, stating precisely what this means requires specifying a pre- and post-condition.

As argued above, the first step is relatively straightforward. However, solving the second problem raises some interesting issues. The central problem is making precise what it means to say that Ann and Bob have a *strategy* to ensure a fair outcome. A strategy for an agent amounts to choosing a particular valuation. Now if we assume that for each agent there is one valuation that is that agent's "true" valuation,[8] then Pauly's notion of correctness may not be applicable. The reason is because truthful announcement of valuations is not a Nash equilibrium.[9] This is illustrated from the following example from [2]. Suppose that Ann and Bob are dividing two paintings: one by Matisse and one by Picasso. Suppose that Ann and Bob's actual valuations are given by the following table.

Item	Ann	Bob
Matisse	75	25
Picasso	25	75
Total	100	100

Ann will get the Matisse and Bob will get the Picasso and each gets 75 of his or her points. However, this is *not* a Nash equilibrium. Suppose that Ann announces here valuation according to the following table.

Item	Ann	Bob
Matisse	26	25
Picasso	74	75
Total	100	100

So Ann will get the Matisse, receiving 26 of her announced (and insincere) points and Bob gets 75 of his announced points. Let p be the fraction of the Picasso that Ann will get, then we want

[8] This assumes that there are valuations which are, as a matter of *fact*, the agents' actual valuations. However, it may very well be that the players themselves cannot point to a valuation which they consider their "true" valuation. After all, valuations are simply a way to *represent* the players' preferences or utilities over the set of goods. See (Parikh R., and Pacit E. *Safe Votes, Sincere Votes, and Strategizing*. Unpublished Manuscript, 2005.) for a relevant discussion in the context of voting. Nonetheless, for this paper we assume that each player is associated with a unique "true valuation".

[9] Here we need only consider Nash equilibrium and not subgame perfect equilibrium as there is only one point in *AW* when the agents make a choice.

$$26 + 74p = 75 - 75p$$

Solving for p gives us $p = 0.33$ and each gets 50 of his or her announced preference. In terms of Ann's *true* preference, however, the situation is very different. She is getting from her true preference $75 + 0.33 \quad 25 = 83.33$.

There are two directions one can follow to extend Pauly's analysis to take into account the above issue. We do not go into details in this essay, but only sketch the main ideas.

Imperfect Information In order for Ann (or Bob) to take advantage of the fact that she is not playing her best response in the above example, she must *know* Bob's actual valuation *and that he will in fact play that valuation*. Note that this second point is important. For suppose that in the above example, that Bob has knowledge of Ann's valuation and decides to deceive Ann in the same way as Ann is deceiving him: Bob announces 74 for the Matisse and 26 for the Picasso. In this case, Ann will get the Picasso and Bob will get the Matisse each receiving 74 of their announced points. However, according to their actual valuations both Ann and Bob receive only 25 points! Thus, the information that each agent has about the other agent's valuation is relevant. This suggests two extensions to Pauly's framework. The first is to extend the basic model to include the information each agent has about the other agents' preferences.[10] The second is to allow agents to misrepresent their preferences and include "truthfulness" as a post-condition.

Safe Strategies As argued above, if Ann attempts to deceive Bob and is *wrong* about Bob's choice of valuation, then this can lead to devastating results for Ann (i.e., receiving less than 50 points). Thus while the honest strategy profile (each agent reports his/her "true" preference) is not necessarily a Nash equilibrium, it is a *safe strategy*, i.e., a strategy that guarantees at least 50 points. This is discussed in [2] and (Parikh R., and Pacuit E. *Safe Votes Sincere Votes, and Strategizing*. Unpublished Manuscript, 2005.). Using this notion of a safe strategy we can be more precise about what it means to say that a fair division algorithm is "correct". A fair division algorithm is correct if there is a *safe strategy* such that the outcome is envy-free, Pareto efficient and equitable.

13.4 Conclusion

Evidence of the usefulness of a rigorous analysis of social procedures can already be seen in many areas of economics and social choice theory. Classic results such as Arrow's Theorem have had profound effects on social choice theory and voting theory. A more concrete example is the analysis of the procedure used by King Solomon in the well-known biblical story. King Solomon is faced with two women each claiming that a baby is her own child. Solomon threatens to cut the baby in half causing one of the mothers to rescind her claim of motherhood; thus revealing

[10] Note that this does not mean moving to *imperfect information games*. Although this may be an interesting direction for future research, all that is needed here is a formal model of the agents information about the other agents' preferences.

herself as the true mother. However a formal analysis of this procedure reveals a mistake: it is possible for the false mother to misrepresent her actual preference and claim, as the true mother would, that the baby should be given to the other woman. What is often missing from the game-theoretic analyzes of social procedures is a clear and rigorous analysis of what it means for the social procedure to be *correct*.

The logics discussed in [15, 18, 19] (Pauly M., and Wooldridge M. *Logic for Mechanism Design – a Manifesto.* Unpublished Manuscript.) are all intended to be tools for providing such a rigorous analysis of social procedures. Focusing on the framework from [15], this essay highlights an important aspect of the analysis of many social procedures. This key component is the information, or knowledge, the agents have during the execution of the procedure (cf. [13] and [10] for an extended discussion of this point).

References

1. Alur R., Henzinger T., and Kupferman O. Alternating-time temporal logic, *JACM*, 49(5): 672–713, 2002.
2. Brams S. J., and Taylor A. D. *Fair Division: From Cake-Cutting to Dispute Resolution.* Cambridge University Press, Cambridge, 1996.
3. Brams S. J., and Taylor A. D. *The Win–Win Solution*. W. W. Norton and Company, New York, NY, 1999.
4. Floyd R. W. Assigning meanings to programs, *Proc. Symp. Appl. Math*., 19: 19–31, 1967.
5. Harel D., Kozen D., and Tiuryn J. *Dynamic Logic*. MIT Press, Cambridge, MA, 2000.
6. Hoare C. A. R. An axiomatic basis for computer programming, *Commun. ACM,* 12: 576–583, 1969.
7. Nielson H. R., and Nielson F. *Semantics with Applications.* Wiley, Chichester, 1992.
8. Osborne M., and Rubinstein A. *A Course in Game Theory*. The MIT Press, Cambridge, 1994.
9. Pacuit E., Parikh R., and Salame S. Some recent results on adjusted winner. In U. Endriss and J. Lange, editors, *Proceedings of Computational Social Choice*. ILLC Technical, Amsterdam, The Netherlands, 2006.
10. Pacuit E., and Parikh R. *Introduction to Formal Epistemology*. Coursenotes for ESSLLI 2007. Available at staff.science.uva.nl/_epacuit/formep esslli.html
11. Parikh R. The logic of games and its applications. In M. Karpinski and J. van Leeuwen, editors, *Topics in the Theory of Computation*, (vol. 24 of *Annals of Discrete Mathematics*), Elsevier, Amsterdam, 1985.
12. Parikh R. Language as social software (abstract). In *International Congress on Logic, Methodology and Philosophy of Science,* page 415. 1995.
13. Parikh R. Knowledge and structure in social algorithms (extended abstract). Presented at Stony Brook Conference on Game Theory, 2007.
14. Pauly M. A modal logic for coalitional power in games, *J. Log. Comput*., 12(1): 149–166, 2002.
15. Pauly M. Programming and verifying subgame perfect mechanisms, *J. Log. Comput.,* 15(3): 295–316, 2005.
16. Pauly M., and Parikh R. Game logic – An overview, *Stud. Log.,* 75(2): 165–182, 2003.
17. Pratt V. R. Semantical considerations on Floyd-Hoare logic. In *Proceedings of. 17th Symposium on Foundation of Computer Science*. IEEE, pages 109–121, 1976
18. van Otterloo S. *A Strategic Analysis of Multi-Agent Protocols*. PhD thesis, University of Liverpool, 2005.
19. Wooldridge M., Agotnes T., Dunne P., and van der Hoek, W. Logic for automated mechanism design — A progress report. In *Twenty-Second Conference on Artificial Intelligence* (AAAI-07), 2007.