Department of Information Studies

International Order Reshaping Based on Argumentation Mechanism Design (*ArgMD*)

This dissertation is submitted in partial fulfilment of the requirements for the Master's degree in MSc Knowledge Information and Data Science

INST0062 MSc Dissertation

SRN: 21105759

Supervised by Dr Antonis Bikakis

Chicago Referencing

Word count: 13,149

September 2022

Declaration

I have read and understood the college and departmental statements and guidelines concerning plagiarism.

I declare that:

- This submission is entirely my own original work.
- Wherever published, unpublished, printed, electronic or other information sources have been used as a contribution or component of this work, these are explicitly, clearly and individually acknowledged by appropriate marks, citations, references and statements in the text. It is 13,149 words in length.

Contents

Acknowledgement

I would like to express my most sincere gratitude to Dr Antonis Bikakis, who offers remarkable support, professional guidance and warm encouragement throughout this study. Our journey together is invaluable treasure to me.

This dissertation is dedicated to the people of Ukraine, who are suffering from dehumanising war crimes yet showing the most admirable courage of mankind; and the people of Taiwan, who are exposed to frequent military threats and political harassments yet never willing to bow down to the dictators on the other side of the strait. May democracy prevail, may there be peace.

“We shall never surrender!”

–Sir Winston Leonard Spencer Churchill

Abstract

Recent geopolitical conflicts in Ukraine and Taiwan urge the democratic world to take prompt actions to defend democracy, to promote humanity and to prevent hot wars. Taking the geopolitical crisis into the perspective of game theory, International Order is a game played by the democratic world and the authoritarian regimes. Argumentation Mechanism Design in Answer Set Programming offers an automatic, flexible, programmable tool to lead the game towards desirable outcomes by modifying the game rules.

This study abstracts International Order Reshaping as a Mechanism Design problem, tries and compares two approaches of mapping games into Answer Set Programming, then conducts an Argumentation Mechanism Design case study. Focused on International Order Reshaping, the case study re-designs the games with "Battle of Sexes" model, maps the games into Dung's Argumentation Frameworks, then encodes the Game-Based Argumentation Frameworks into Answer Set Programming, where Argumentation Mechanism Design solutions can be automatically enumerated. The case study also promotes two methods to increase the interpretability of Argumentation Mechanism Design solutions—assigning compulsory attack relations and limiting the complexity of attack relation set.

Results demonstrate support for Argumentation Mechanism Design in Answer Set Programming as an effective tool of automatically solving complex real-world issues. Code-generated Argumentation Mechanism Design solutions are interpretable as practical suggestions for International Order Reshaping.

Keywords: Game Theory, Mechanism Design, Argumentation Framework, Answer Set Programming

List of Abbreviations

AF	Argumentation Framework
ArgMD	Argumentation Mechanism Design
ASP	Answer Set Programming
GBAF	Game-Based Argumentation Framework
MD	Mechanism Design
SSA	Strategy-Set Argument

1 Introduction

Research Problem. Aggravated geopolitical conflicts in east Europe and east Asia since January 2022 not only expose the world to growing threat of new round of massive wars, but also kick off a game where both the democratic world and the autocratic dictators wish to prevail. International Order Reshaping is inevitable in the foreseeable future and prompt actions must be taken to lead it towards a desirable outcome where democracy and humanity remain mainstream while hot wars can be prevented. Taking geopolitics into the perspective of game theory, International Order Reshaping is in fact a typical Mechanism Design (*MD*) problem concerning the game and desirable outcome mentioned above. Therefore, to take prompt actions in International Oder Reshaping, a flexible, automatic and programmable solution for *MD* problems is of great demand. With such a solution, International Order Reshaping can be led into a certain desirable direction with instructions automatically generated by programmes. Finding a solution which meets this demand is the main research problem of this study.

Theoretical Framework. A *MD* problem is concerned with the following question: *What game rules guarantee a desirable social outcome when each self-interested agent selects the best strategy for itself?* Specifically, International Order Reshaping is a *MD* problem of the game of global politics. Therefore, to offer a programmable *MD* solution for International Order Reshaping, games should be encoded as models in suitable programming languages. Answer Set Programming (*ASP*), as a form of declarative programming based on fast satisfiability solvers for propositional logic and oriented towards difficult search problems, is particularly useful in knowledge-intensive applications such as game theory. Mapping games into *ASP* stable model semantics will build the foundation of the solution to the main research problem. In this research, two pathways are built up to facilitate games → *ASP* mapping process.

Pathway 1 (extensive-form game → *ASP*) is to analyse a game G in its extensive form Γ (extensive-form game Γ may be derived from equivalent normal-form game), then map Γ into a node-edge style roadmap and express the roadmap with stable model semantics. In this way, it is made possible to perform *MD* simply by modifying edges between nodes (blocking, reconnecting, obstructing etc.). Previously conducted studies have offered support to the construction of this pathway: It is stated in [VanDamme1984] that proper equilibria in a normal-form game each induces a quasi-perfect equilibrium in every extensive-form game having this normal form; while application of *ASP* in roadmap expressions and designs can be found both in navigation system developing [Hu2013] and multi-agent pathfinding [Gómez2021].

Pathway 2 (normal-form game → *AF* → *ASP*) is to formalise a normal-form game G, along with its game rules, as abstract Argumentation Framework (*AF*), then directly express *AF* as stable model semantics in *ASP*. Previous studies have shed light on this approach by summarising the mapping of multi-agent abstract argumentation as an instance of a *MD* problem, matching *MD* concepts and corresponding Argumentation Mechanism Design (*ArgMD*) instantiation [Rahwan2009, Fan2016, Young2019]. Furthermore, the implementation of *AF* using *ASP* has also been realised with *ASPARTIX* system [Egly2008, Dvořák2020].

Brief Review of Methodology. Process of this research consists of the following five major parts.

1. Analyse international order preference and available strategies of two groups of countries (seven in total)—Group 1: the United Kingdom, United States, France, Germany, Japan; Group 2: Russia, P. R. China.
2. Use hypothetical utility functions, strategy profiles and build up an extensive-form game represented by a "decision tree" based on the scenario. Transfer the extensive-form game into normal-form.
3. Map the game into *ASP* semantics, via two pathways respectively. Evaluate and compare the two pathways. Then, select the more effective, interpretable pathway.
4. Conduct a *MD* case study on International Order Reshaping in the selected mapping pathway. Offer solutions to International Order Reshaping.
5. Increase the interpretability of International Order Reshaping solutions. Reveal the game rules for desirable social outcomes and explain them with respect to the scenario of International Order.

Game theory mainly studies the strategic interactions between two or more self-interested agents. In the scenario in Part 1, agents are the two groups of countries. Preference of each group is expressed as a utility function $u_i(o_1, \theta_i)$, the independent variables of which are the outcome o and the country's agent type θ_i. $u_i(o_1, \theta_i) > u_i(o_2, \theta_i)$ when group of countries i prefers outcome o_1 to outcome o_2. The game outcomes that will arise are determined by solution concepts, assuming all groups of countries are strategic and rational. The most widely-used solution concept is the Nash Equilibrium, the centrepiece of game theory. A Nash Equilibrium is a strategy profile where each agent of the game follows a strategy which maximises the value of its own utility function, given its agent type and the strategies of the other agents [Nash1950].

MD studies how to ensure that desirable game outcomes or decisions are made, given a group of self-interested agents who have preferences over game outcomes. That is, the game outcome depends on the preferences of the countries. When a scenario is mapped into a game, a social choice function can be used to capture *MD* dependency.

Research Purposes and Key Questions. After mapping games into *ASP* stable model semantics via the two pathways listed above, this research will attempt to compare the two pathways by evaluating their efficiency, interpretability and simplicity respectively. The best among the two pathways will be selected and presented as the solution to the research problem. *MD* case study which focuses on International Order Reshaping and involves seven countries will be then conducted in the form of *ASP* queries, whose outputs will be explained and testified in reference to human understanding of global politics and diplomacy.

In brief, this research has two major purposes: 1. To find the better way of mapping games into *ASP*; 2. To lead International Order Reshaping towards desirable direction with a programmable Mechanism Design method.

To fulfil the research purposes and to create lasting value from the implications expected from the study, there are several questions worth paying attention to throughout this research. During the process of mapping games into *ASP*, this research shall attempt to answer how to select the dominant criterion among comprehensiveness, interpretability and simplicity etc. in terms of making a good mapping method with respect to the purposes of *MD*. In the *MD* case study focused on International Order Reshaping, this research should answer how *MD* solutions can be interpretated as practical geopolitical instructions, and how interpretability of *MD* solutions can be increased.

Significance of the Study. From the perspective of politics and humanity: Considering the danger of collapse and fragmentation that the democratic world is exposed to, and the global political unrest the world is experiencing in the year of 2022, it is urgent and meaningful to find a way, or at least point the direction, to reshape the international order into a status which promotes democracy, defends human rights and poses minimised threat to peace. Such status is the desirable social outcome when the global politics is considered the game, while the corresponding game rules are left for this research to solve. From the perspective of programming technologies: Despite the successful application of *ASP* in the field of argumentation, *MD* has not yet been reported mapped into *ASP* for automatic and flexible game rule solutions. This research is dedicated to filling this blank.

2 Literature Review

The first three subsections respectively review several classic solution concepts in the field of game theory, some core concepts in the domain of argumentation framework and the interplay between games and argumentation. Sub-section 2.4 summarises fundamental concepts and implementations of *MD*, while Sub-section 2.5 reviews Answer Set Programming encodings for Dung style abstract argumentation. The first five subsections together lay the theoretical foundation of Pathway 2 of mapping games to *ASP* (normal-form game → *AF* → *ASP*). At last, subsection 2.6 demonstrates the feasibility of Pathway 1 (extensive-form game → *ASP*), a pathway to a large extent remains unexplored and brings compelling research opportunities.

2.1 Game Theory: Normal-Form Games

Game theory provides mathematical models for the analysis of strategic interactions between agents [Gibbons1992, Shoham2008, Myerson2013]. The normal form of games is the most widely used representation of strategic interactions in game theory. In a normal-form game, each player can select a single action and execute it—where a strategy is called a pure strategy. In pure strategy games, each strategy represents taking an action with 100% probability. A pure strategy game can be defined as follows [Shoham2008]:

Definition GT-1. Normal-form game. *A finite, n-person normal-form game is a tuple* (N, S, u)*, where:*

- N *is a finite set of* n *players, indexed by* i*;*
- $S = S_1 \times \cdots \times S_n$*, where* S_i *is a finite set of strategies available to player* i*, each vector* $s = (s_1, \ldots, s_n)$ *is called a strategy profile;*
- $u = (u_1, \ldots, u_n)$*, where* $u_i: S \to \mathbb{R}$ *is a real-valued utility (or payoff) function for player* i*.*

The available strategies for player i are denoted as s_i, while the set of pure strategies for player i is denoted as S_i. A pure-strategy profile, denoted as s, record choices of strategy from all players. Formally, $s_{-i} = (s_1, s_2, s_3, \ldots s_{i-1}, s_{i+1}, \ldots s_n)$ defines a strategy profile s without the strategy of player i. Hence, $s = (s_i, s_{-i})$, while S_{-i} is the set of all strategy profiles of all players except for player i.

Game theory contains many solution concepts [Shoham2008], among which the most important ones are defined as follows.

Definition GT-2. Strictly dominant strategy. *Let* s_i *and* s_i' *denote two strategies of player* i*, then* s_i *strictly dominates* s_i' *if* $\forall s_{-i} \in S_{-i},\ u_i(s_i, s_{-i}) > u_i(s_i', s_{-i})$*. A strategy is strictly dominant for an agent if it strictly dominates every other strategy for that agent.*

Definition GT-3. Equilibrium in strictly dominant strategies. *A strategy profile* $s = (s_1, \dots, s_n)$ *in which every* s_i *is strictly dominant for player* i *is an equilibrium in strictly dominant strategies.*

Definition GT-4. Best response. $s_i \in S_i$ *is player* i*'s best response to the strategy profile* s_{-i} *if* $u_i(s_i^*, s_{-i}) \geq u_i(s_i, s_{-i})$ *for all strategies* $s_i \in S_i$.

Definition GT-5. Nash Equilibrium. *Given a strategy profile* $s = (s_1, \dots, s_n)$*, if* s_i *is a best response to* s_{-i} *for all agents* i*, the strategy profile* s *is a Nash Equilibrium of the game.*

An equilibrium in strictly dominant strategies is by definition the unique Nash Equilibrium. Nash Equilibria can be divided into strict Nash Equilibria and weak Nash Equilibria. A Nash Equilibrium is a strict equilibrium only when every agent's strategy constitutes a unique best response to other agents' strategies. Otherwise, it is a weak equilibrium.

Definition GT-6. Strict Nash Equilibrium. *A strategy profile* $s = (s_1, \dots, s_n)$ *is a strict Nash Equilibrium if, for all agents* i *and for all strategies* $s_i' \neq s_i,\ u_i(s_i, s_{-i}) > u_i(s_i', s_{-i})$.

Definition GT-7. Weak Nash Equilibrium. *A strategy profile* $s = (s_1, \dots, s_n)$ *is a strict Nash Equilibrium if, for all agents* i *and for all strategies* $s_i' \neq s_i,\ u_i(s_i, s_{-i}) \geq u_i(s_i', s_{-i})$.

Strict Nash Equilibria are intuitively more stable than weak Nash Equilibria, for that in the latter case one or more players can swerve away from Nash Equilibria. In a game, there can be at most one strict Nash Equilibrium. That is, for each game, strict Nash Equilibrium is either non-existing or unique.

2.2 Abstract Argumentation Framework

The conceptions of argumentation framework (*AF*) are inventively introduced in [Dung1995]. With nodes representing arguments and edges representing the attack relations, an *AF* can be visualised as a directed map-like graph.

Definition AF-1. **An argumentation framework** *is a pair* $AF = \langle \mathcal{A}, \mathcal{R} \rangle$*, where* $\mathcal{A}$ *is a set of arguments, and* $\mathcal{R}$ *is a binary relation over* $\mathcal{A}$*, i.e.,* $\mathcal{R} \subseteq \mathcal{A} \times \mathcal{A}$*.*

$(\alpha, \beta) \in \mathcal{R}$ denotes that argument α attacks argument β. Given an *AF*, the statuses of the arguments the *AF* contains are evaluated contingent on the following three notions [Liao2021], producing different type of *AF* extensions—the sets of arguments acceptable together.

Definition AF-2. *Given* $AF = \langle \mathcal{A}, \mathcal{R} \rangle$*, and* $\mathcal{B} \subseteq \mathcal{A}$

- *A set of arguments* $\mathcal{B}$ *is* **conflict-free** *if* $\nexists \alpha, \beta \in \mathcal{B}$ *such that* $(\alpha, \beta) \in \mathcal{R}$*.*
- *An argument* $\alpha \in \mathcal{A}$ *is* **acceptable** *w.r.t. a set* $\mathcal{B}$ *(*α *is defended by* $\mathcal{B}$*), if* $\forall (\beta, \alpha) \in \mathcal{R}$ $(\beta \notin B, \beta \neq \alpha), \exists \gamma \in \mathcal{B}$ *such that* $(\gamma, \beta) \in \mathcal{R}$*.*
- *A conflict-free set of arguments* $\mathcal{B}$ *is* **admissible** *if each argument in* $\mathcal{B}$ *is an acceptable argument w.r.t.* $\mathcal{B}$*.*

Extension-based argumentation semantics can be interpretated as pre-defined criteria. The acceptability of arguments in an *AF* can be determined according to these criteria.

Definition AF-3. *Given an argument framework* $AF = \langle \mathcal{A}, \mathcal{R} \rangle$*, and an admissible set of arguments* $\mathcal{E} \subseteq \mathcal{A}$*.*

- $\mathcal{E}$ *is a* **complete** *extension of* AF*, denoted as* $\mathcal{E}_{\mathcal{CO}}(AF)$*, if* $\mathcal{E}$ *contains all arguments in* $\mathcal{A}$ *that is acceptable w.r.t.* $\mathcal{E}$*.*
- $\mathcal{E}$ *is a* **grounded** *extension of* AF*, denoted as* $\mathcal{E}_{\mathcal{GR}}(AF)$*, if* $\mathcal{E}$ *is a minimal complete extension w.r.t. set inclusion.*
- $\mathcal{E}$ *is a* **stable** *extension of* AF*, denoted as* $\mathcal{E}_{\mathcal{ST}}(AF)$*, if* $\mathcal{E}$ *is conflict-free and* $\forall \beta \in \mathcal{A} \backslash \mathcal{E}, \exists \alpha \in \mathcal{E}$ *such that* $(\alpha, \beta) \in R$*.*

2.3 Interplay between Games and Argumentation

The interplay between games and argumentation goes in two directions, one is analysing agents' behaviour in argumentation according to game theory, the other is applying argumentation framework to games. Many previous studies have focused on the first direction [Prakken2005, Rahwan2008, Rahwan2009, Riveret2008].

As for the second direction, Dung introduces in his seminal paper [Dung1995] the basic procedures of applying argumentation frameworks to n-person cooperative games, with the stable marriage problem (SMP) as example. Dung maps $\langle IMP, \rightarrow \rangle$ as an abstract *AF* while dealing with n-person cooperative games. In such an abstract *AF*, each argument represents an imputation (given the payoff distribution among agents) while each attack represents the domination between imputations. In his later lectures recorded in [Narahari2014], Dung encourages applying different definitions of arguments and attack relations to different type of games, pointing out that cooperative games and normal-form games are naturally different: in cooperative games, joint actions of groups of players are the basic elements (also known as primitives); in non-cooperative games, basic elements are the actions of individual players.

Dung's endeavour to apply argumentation to games is advanced by [Bistarelli2020] which focuses on n-person cooperative games, along with [Young2019] which focuses on SMP and further reveals the correspondence between game-theoretical solution concepts and Dung's argumentation semantics in cooperative games. In the SMP, an argument is denoted as (m, w), which represents a man m marries woman w, while (m, w) is attacked by (m', w') if the following conditions are established: (1) $m' = m$ while m prefers w' over w; (2) $w' = w$ while w prefers m' over m.

Fan and Toni extend the second direction by taking normal-form games into the perspective of argumentation [Fan2014] and revealing the potential of using assumption-based argumentation to solve normal-form games through dialogue. They map normal-form games into assumption-based *AF* and have each strategy profile translated into an assumption $d(\sigma_\alpha, \sigma_\beta)$ which is interpretated as "this strategy profile is a Nash Equilibrium", and a conclusion $nD(\sigma_\alpha, \sigma_\beta)$ which is interpretated as "this strategy profile is not a Nash Equilibrium". An argument with a conclusion $nD(\sigma_\alpha, \sigma_\beta)$ attacks an argument $d(\sigma_\alpha, \sigma_\beta) \vdash d(\sigma_\alpha, \sigma_\beta)$. Unlike in Dung's work, each argument represents a strategy profile instead of an imputation in Fan and Toni's work. However, an imputation and a strategy profile both involve all players in a game. From this angle, Fan and Toni's approach is similar to Dung's. Such approaches prove to be rather successful in cooperative games but not so in normal-form games. The explanation to this difference is that in cooperative games, strict correspondences can be established between solutions of games and semantics of abstract AF while such correspondences become weak in normal-form games.

To apply *AF* to normal-form games, Game-Based Argumentation Framework (*GBAF*) is promoted and constructed with the five definitions below in [Cheng2021]. In *GBAF*, an available strategy for a player is directly transformed into an argument in the corresponding *AF*, while the best response relations are transformed and embedded as attack relations in the corresponding *AF*.

Definition IP-1. Game-based argument. *Given a normal-form game $G = (N, S, u)$, a_i is a game-based argument which denotes "player i should choose strategy s_i". $\mathcal{A}_i$ is a set of available game-based arguments for player i. $\mathcal{A}_G$ denotes The set of available game-based arguments for all players is denoted by $\mathcal{A}_G$, that is, $\mathcal{A}_G = \cup_{i=1}^{n} \mathcal{A}_i$.*

Definition IP-2. Game-based attack relation. *Given a normal-form game $G = (N, S, u)$ and the corresponding set of available game-based arguments $\mathcal{A}_G$, denote the set of game-based attack relations as $\mathcal{R}_G \subseteq \mathcal{A}_i \times \mathcal{A}_j$, $(i \neq j)$. $(a_j, a_i) \in \mathcal{R}_G$ if s_i is not the best response to all s_{-i} that contain s_j.*

$\mathcal{R}_G$ contains attack relations each of which is between two game-based arguments from different players. Given a_i, a_i' and a_j (corresponding to strategies s_i, s_i' and s_j), a_j attacks a_i if s_i' results in higher payoff than s_i w.r.t. s_j.

Definition IP-3. Strategy profile argument set. *Given a normal-form game $G = (N, S, u)$ and the corresponding set of available game-based arguments $\mathcal{A}_G$, $\mathcal{A}_{sp}$ denotes a strategy profile arguments set which represents a strategy profile such that $|\mathcal{A}_{sp}| = n$. For any pair of arguments $a, b \in \mathcal{A}_{sp}$ $(a \neq b)$, there does not exist a player i such that $a \in \mathcal{A}_i$ and $b \in \mathcal{A}_i$. That is, every argument in $\mathcal{A}_{sp}$ belongs to a strategy of a distinct player.*

$\mathcal{A}_{SP}$ denotes the set containing all the possible $\mathcal{A}_{sp}$. Game-Based Argumentation Framework (*GBAF*) is defined as follows:

Definition IP-4. Game-Based Argumentation Framework. *Given a normal-form game $G = (N, S, u)$, the corresponding GBAF is defined as $AF_G = (\mathcal{A}_G, \mathcal{A}_{SP}, \mathcal{R}_G)$, where $\mathcal{A}_G$ is the set of game-based arguments, $\mathcal{A}_{sp}$ is the set of strategy profile arguments sets, and $\mathcal{R}_G$ is the set of game-based attacks.*

Definition IP-5. GBAF extension semantics. *Given a normal-form game $G = (N, S, u)$, the corresponding game-based framework $AF_G = (\mathcal{A}_G, \mathcal{A}_{SP}, \mathcal{R}_G)$ and a semantic σ, $\mathcal{E}_\sigma(AF_G) = \mathcal{E}_\sigma((\mathcal{A}_G, \mathcal{R}_G)) \cap \{\mathcal{A} | \mathcal{A} \in 2^{\mathcal{A}_G} \text{and } \forall i \in N, \exists a_i \in \mathcal{A}\}$.*

According to the Definition IP-4 and Definition IP-5, argument sets corresponding to game-theoretical solution concepts in normal-form games should be included in both $\mathcal{A}_{SP}$ and a particular extension of AF_G according to a selected semantics.

Based on the structure of *GBAF*, correspondences between solution concepts in the field of game theory and semantics in the domain of argumentation are given as follow.

Correspondences in *GBAF*

1. *Given a normal-form game $G = (N, S, u)$ and the corresponding AF_G, if there does not exist a set $\mathcal{A}_{sp}$ in the AF_G which is conflict-free, then there does not exist $s = (s_1, \dots, s_n)$ which is an equilibrium.*
2. *Given a normal-form game $G = (N, S, u)$ where $N = \{1,2\}$ and the corresponding AF_G, a_1 is attacked by a_2 iff s_1 is not the best response to s_2.*
3. *Given a normal-form game $G = (N, S, u)$ where $N = \{1,2\}$ and the corresponding AF_G, a strategy s_1 is dominant iff a_1 is unattacked, i.e., $\nexists a_2 \in \mathcal{A}_2$, such that $(a_2, a_1) \in \mathcal{R}_G$.*

4. *Given a normal-form game* $G = (N, S, u)$ *where* $N = \{1,2\}$ *and the corresponding* AF_G*, a strategy profile* $s = (s_1, s_2)$ *is an equilibrium in strictly dominant strategies, iff* $\{a_1, a_2\} = \mathcal{A}_{SP} \cap \mathcal{E}_{\mathcal{GR}}(AF_G)$.

5. *Given a normal-form game* $G = (N, S, u)$ *where* $N = \{1,2\}$ *and the corresponding* AF_G*, a strategy profile* $s = (s_1, s_2)$ *is a Nash Equilibrium iff in the* AF_G *transferred from* G*, the corresponding* $\mathcal{A}_{sp} = \{a_1, a_2\}$ *is conflict-free.*

6. *Given a normal-form game* $G = (N, S, u)$ *where* $N = \{1,2\}$ *and the corresponding* AF_G*, a strategy profile* $s = (s_1, s_2)$ *is a strict Nash Equilibrium iff* $\{a_1, a_2\} \in \mathcal{A}_{SP} \cap \mathcal{E}_{ST}(AF)$.

2.4 Mechanism Design

Mechanism Design (*MD*) is concerned with how to guarantee that desirable game outcomes or decisions are made, given a group of self-interested agents who have preferences over the outcomes. To be specific, one often wishes that the game outcome depends on the preferences of the agents, a type of game-theoretical dependency that can be captured by a social choice function.

Definition MD-1. Social choice function. *A social choice function is a rule* $f : \Theta_1 \times \ldots \times \Theta_n \to \mathcal{O}$ *, that selects some outcome* $f(\theta) \in \mathcal{O}$ *, given agent types* $\theta = (\theta_1, \ldots, \theta_n)$.

The challenge is that the agent types (θ_i) are usually only known to the agents themselves. Therefore, it relies on the agents revealing their true types to select game outcomes with the social choice function. However, given a social choice function, some agents may find that it is better off if they lie about their agent types, since by doing so they may mislead the social choice function to choosing game outcomes they prefer. Instead of assuming the agents to truthfully reveal their agent types, mechanism can be employed to force the game towards the desired outcome.

A mechanism $\mathcal{M} = \langle \Sigma, g(\cdot) \rangle$ is the set of strategies that agents are allowed to choose. $\Sigma = \Sigma_1 \times \cdots \times \Sigma_n$, where Σ_i is the strategy set for agent i, covers all possible strategy profiles; while $g(s)$, known as outcome function, specifies the game outcome o for each strategy profile $s = (s_1, \ldots, s_n) \in \Sigma$. This defines a game where each agent i is free to choose any strategy in Σ_i and to select a strategy leading to game outcomes which maximise its utility. Social choice function f is implemented by mechanism $\mathcal{M}$ if the game outcomes produced by $\mathcal{M}$ is exactly the game outcomes would have been returned by f in the case that all agent types are truthfully revealed.

Definition MD-2. Implementation. *A mechanism $\mathcal{M} = (\Sigma, g(\cdot))$ implements social choice function f if there exists an equilibrium s^* such that $\forall \theta \in \Theta, g\big(s^*(\theta)\big) = f(\theta)$.*

While agents' strategy spaces are not restricted by the definition of mechanism, the strategies of the agents are to claim a type, $\theta_i^{'}$ to the mechanism in an important category of mechanisms, the direct-revelation mechanisms.

Definition MD-3. Direct-revelation mechanism. *A direct-revelation mechanism is a mechanism in which $\Sigma_i = \Theta_i$ for all i, and $g(\theta) = f(\theta)$ for all $\theta \in \Theta$.*

While $\theta_i^{'} = \theta_i$ is not necessarily true, the Revelation Principle states that if a social choice function f can be implemented, then it is necessarily implementable by a direct mechanism in which all agents reveal their true agent type. The social choice function f is called incentive compatible in such situation.

Definition MD-4. Incentive compatible. *The social choice function $f\,(\cdot)$ is incentive compatible (or truthfully implementable) if the direct mechanism $\mathcal{M} = (\Sigma, g(\cdot))$ has an equilibrium s^* such that $s_i{}^*(\theta_i) = \theta_i$.*

If the equilibrium concept is the dominant-strategy equilibrium, then the social choice function is strategy-proof. A mechanism is called incentive-compatible or strategy-proof when the social choice function that the mechanism implements is incentive-compatible or strategy-proof.

A new approach "Argumentation Mechanism Design" (*ArgMD*) is reported in [Rahwan2009]. Given an argumentation framework $AF = \langle \mathcal{A}, \mathcal{R} \rangle$ with a set of arguments $\mathcal{A}$ and a set of binary attack relations $\mathcal{R}$, a mechanism is defined with respect to AF and semantic $\mathcal{S}$. Mapping of multi-agent (n self-interested agents) abstract *AF* as an instance of a *MD* problem is summarised as shown in Table 1.

Table 1. Abstract Argumentation Framework as a mechanism.

MD **Concept**	***ArgMD*** **Instantiation**
Agent type $\theta_i \in \Theta_i$	Agent's arguments $\theta_i = \mathcal{A}_i \subseteq \mathcal{A}$
Outcome $o \in \mathcal{O}$	Accepted arguments $Acc(\cdot) \subseteq \mathcal{A}$
Utility $u_i(o,\ \theta_i)$	Preference over $2^{\mathcal{A}}$ (what arguments end up being accepted)
Social choice function $f\colon \Theta_1 \times \ldots \times \Theta_n \to \mathcal{O}$	$f(\mathcal{A}_1, \ldots \mathcal{A}_n) = Acc(\langle \mathcal{A}_1 \cup \ldots \cup \mathcal{A}_n, \mathcal{R} \rangle, \mathcal{S})$ by some argument acceptability criterion
Mechanism $\mathcal{M} = (\Sigma, g(.))$ where $\Sigma = \Sigma_1 \times \ldots \times \Sigma_n$ and $g\colon \Sigma \to \mathcal{O}$	Σ_i is an argumentation strategy, $g\colon \Sigma \to 2^{\mathcal{A}}$
Direct mechanism: $\Sigma_i = \Theta_i$	$\Sigma_i = 2^{\mathcal{A}}$ (every agent reveals a set of arguments)
Truth revelation	Revealing $\mathcal{A}_i$

2.5 Answer Set Programming Encodings for Dung's *AF*

Answer Set Programming Argumentation Reasoning Tool (*ASPARTIX*) is a system which provides *ASP* encodings usable in a standard *ASP*-solver for reasoning tasks and semantics in Dung's *AF*. Basic workflow of *ASPARTIX*, as summarised in [Dvořák2020], is shown in Figure 1.

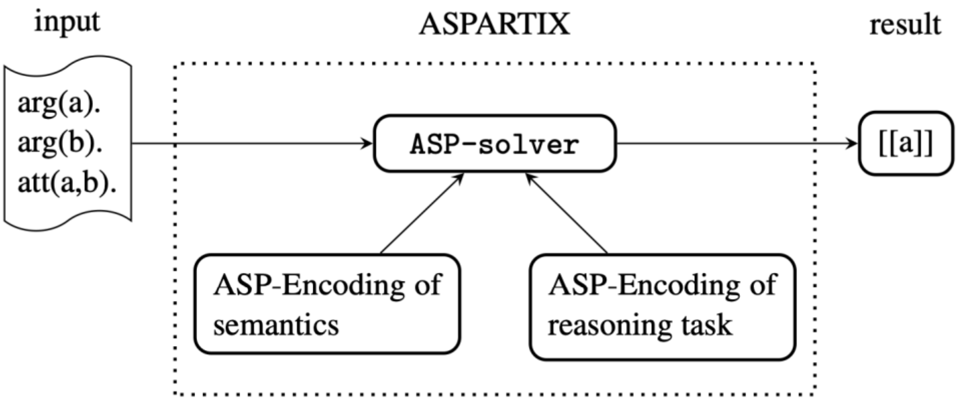


Figure 1. Basic workflow of ASPARTIX.

ASPARTIX is inventively introduced in Sarah Gaggl's Master Thesis [Gaggl2009]. With a DLV program implemented, *ASPARTIX* is capable of computing the standard extensions not only for classic Dung's *AF*, but also for Preference-Based *AF*, Value-Based *AF* and Bipolar *AF*. In the latter cases, *ASPARTIX* is able to enumerate complete and save extensions, as well as to distinguish s-admissible (for stable), c-admissible (for closed) and d-admissible (classical, following Dung) extensions. Also, the preferred extensions are available respectively.

Given an *AF* in the apx format (facts in *ASP* language, example of which shown in Figure 2) as input, *ASPARTIX* delegates the main reasoning to an *ASP* solver (DLV, Clingo, e.g.), with answer set programs encoding the argumentation semantics and reasoning tasks.

```
arg(a).      % a is an argument

att(a, b).   % a attacks b
```

The input file should record the arguments and attack relations in the *AF* to evaluate. To be specific, the *AF* is encoded by a series of statements, such as that in the code trunk above, where the first sequence of statements (`arg/1`)encodes the arguments while the second sequence (`attack/2`) encodes the attack relations in the *AF*.

***AF* argument graph**	***ASPARTIX* input in apx format**

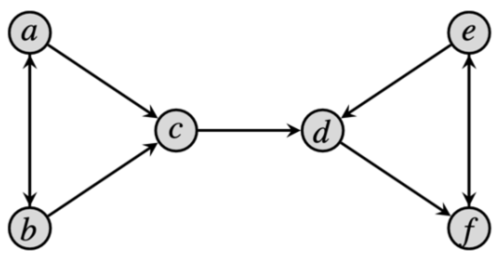
	`arg(a).`
	`arg(b).`
	`arg(c).`
	`arg(d).`
	`arg(e).`
	`arg(f).`
	
	`att(a, b).`
	`att(b, a).`
	`att(a, c).`
	`att(b, c).`
	`att(c, d).`
	`att(d, f).`
	`att(e, d).`
	`att(e, f).`
	`att(f, e).`

Figure 2. Example ASPARTIX input in apx format.

In this research, Clingo serves as the *ASP* solver in *ASPARTIX*. Usage of Clingo is instructed in [TuWien2021] as follows. The `semantics.dl` here is a dummy file which can be replaced by *ASP* fact files given in Appendix 1, corresponding to the concepts of complete, grounded and stable extensions as summarised in Definition AF-3.

Enumerate Extensions. With the following command run in Terminal, *ASPARTIX* will enumerate all extensions of the *AF* to evaluate (encoded as `input.af`) w.r.t. a particular semantics (encoded as `semantics.dl`).

```
$ clingo input.af semantics.dl filter.lp 0
```

`filter.lp` is used to pick out and show the predicates denoting the arguments which the extension contains. To enumerate not all but only a number of extensions, replace 0 with N (the number of extensions).

Credulous Reasoning. With the following command *ASPARTIX* enumerates all arguments which are credulously accepted.

```
$ clingo input.af semantics.dl filter.lp -e brave
```

Sceptical Reasoning. With the following command *ASPARTIX* enumerates all arguments which are sceptically accepted.

```
$ clingo input.af semantics.dl filter.lp -e cautious
```

2.6 Research Gap: Directly Mapping Games to *ASP*

Subsections 2.3 and 2.4 summarise the methods of mapping normal-form games into *AF* and the encodings of *AF* in *ASP* respectively. Combining these techniques, a pathway of connecting games to *ASP* (Pathway 2) can be constructed for automatic and flexible solving of *MD*. However, such pathway must “call at” *AF*, is there a “direct train” between games and *ASP*?

Looking back at the *ASP* encoding process in Figure 2, what is essentially crucial is to construct in advance a node-edge style graph which can be easily expressed in *ASP* language in a straightforward manner. *AF* argument graph is one but not the only node-edge style graph that can be used for *ASP* encodings. In fact, a game can be represented in the extensive form, a node-edge style decision tree graph such as Figure 3, where each non-terminal node represents a player with its game status, each edge represents a strategical action that the player at its starting point can execute, and each terminal node represents a game payoffs to all players. Therefore, by replacing *AF* argument graph with decision tree graph, *ASP* can directly encode games.

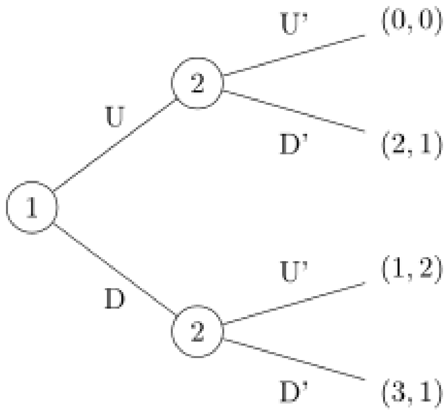
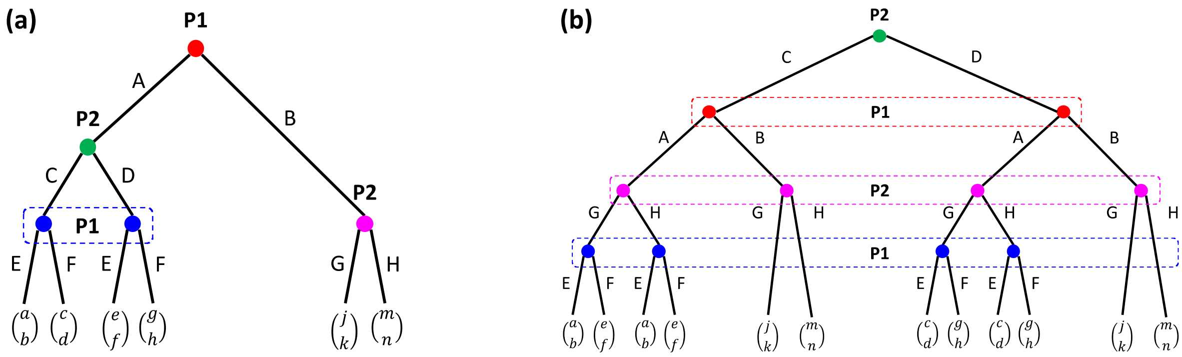


Figure 3. A game represented in extensive form (decision tree).

Extensive form plays an important role in the early history of game theory, while normal form is introduced as a less complex, amenable transformation of extensive form for the convenience of analysis. Every finite extensive-form game has a unique normal-form representation [Cressman2003]. However, the converse of the statement is not true: A normal-form game can have more than one extensive-form representations, which means the transformation from extensive form to normal form causes loss of information. A case in point is that normal-form game in Figure 4(c) can be derived from both extensive-form games in Figure 4(a) and 4(b). Therefore, when employing Pathway 2 to map a figurative game (such as International Order Reshaping in this research) into *ASP*, the game should be presented in extensive form directly for minimum loss of information. Involving the normal form would be unnecessary.



(C)

P2 / P1	CG	CH	DG	DH
AE	*a,b*	*a,b*	*e,f*	*e,f*
AF	*c,d*	*c,d*	*g,h*	*g,h*
BE	*j,k*	*m,n*	*j,k*	*m,n*
BF	*j,k*	*m,n*	*j,k*	*m,n*

Figure 4. Deriving a normal-form game from extensive-form games.

However, in some cases, starting point of the analysis of a game is abstract, usually in normal form. Given a game in the normal form, one could reverse-engineer it back to extensive form and map the game into *ASP*. The correspondences between solution concepts in normal form and in extensive form are clarified during the formation of normal form by many previous studies [VanDamme1984, Mailath1991, Seidenfeld1995, Cressman2003]. The most essential and the most widely used correspondence is that a proper equilibrium of a normal form game induces a quasi-perfect equilibrium in every extensive form game having this normal form [VanDamme1984].

3 Methodology

This section describes the novel methodology used in approaching solutions to research problem. After systematic literature review, *ASP* proves to be a feasible, flexible and straightforward way of automatically solving *MD* problems. Therefore, employing *ASP*, the research problem can be transformed as finding and evaluating methods of mapping game and *MD* problems to *ASP* semantics, where game and *MD* problems are drawn from International Order Reshaping scenario.

The scenario of international order is summarised according to recent political and social science research. Countries are selected to be involved in the scenario according to their international political and economic influence. Seven countries are selected and divided into two groups according to level of democracy, similarities in ideology and multilateral co-operations. The two groups can be seen as two self-interested *players*. Each player takes part in five independent chicken-game-modified “Prisoner’s Dilemma” games representing five major fields of national strengths—economy, culture, information society, political systems and military forces respectively. Utility functions vary in different games according to the significance in national strengths of the field the game represents. For each game, game outcome changes according to players’ actions, which are also known as the players’ strategy profiles. A player’s combine utility function is calculated from five independent utility functions from five games. Scenario of International Order and corresponding game will be further discussed in detail in Sub-section 4.1.

The goals of International Order Reshaping are to defend democratic values and to prevent massive hot wars. Correspondingly, *MD* should guarantee that value of combined utility function is larger for the group of countries of higher level of democracy and that value of combined utility functions should not reach the bottom of range for either group. Goals of International Order Reshaping and corresponding *MD* problems will be further discussed in detail in Sub-section 4.2.

With scenario and goals of International Order Reshaping clarified, mapping game to *ASP* is the next move towards realising automatic, flexible *MD* problem solving. Mapping game to *ASP* is conducted in two pathways, which will be introduced in detail in Section 5.

In Pathway 1 (Extensive-form Game → *ASP*), mapping extensive-form game to *ASP* is straightforward when seeing decision-tree as a node-edge style roadmap. The game can be encoded in *ASP* by simply sorting out the relations and connections between all nodes.

In Pathway 2 (Normal-form Game → *AF* → *ASP*), mapping normal-form game to *AF* requires finding corresponding arguments for all available strategies and sorting out binary relations (attack) between arguments. While the former is rather straightforward, the latter needs careful examination of the game outcomes and utility functions. *AF* acquired from normal-form game is presented as an argument graph and will be further encoded in *ASP*. *ASPRATIX* techniques introduced in [TuWien2021] are employed, with arguments as `arg(X)`, binary relations as `attack(X,Y)` (X attacks Y) and Nash Equilibria as stable extensions of the *AF* [Dung1995]. A stable extension is a conflict-free argumentation set $\mathrm{E} \subseteq \mathrm{A}$ which attacks all arguments in A/E, whose semantics is encoded in *ASP* as ‘stable.dl’. Encoding of *AF* in *ASP* adopts Clingo *ASP* syntax, and queries are made in Clingo environment (version 5.5.1). Pathway 2 will be introduced in detail in Sub-section 5.2.

The two pathways are juxtaposed to be evaluated and compared in Sub-section 5.3, with respect to feasibility of *MD* and comprehensiveness of game information. Pathway 2 is selected for *MD* in *ASP*.

ArgMD in *ASP* is illustrated in Section 6. First off, game scenario used in Section 4 is examined based on its returned Nash Equilibria, where “Prisoner’s Dilemma” model proves to be incapable of fulfilling the goals of *MD* or properly reflecting the reality of International Order Reshaping. “Battle of Sexes” model is used instead in the redesigned games. *ArgMD* is then conducted in *ASP*, narrowing Nash Equilibria to only desirable outcomes. *ASP* encoding is modified to push forward analysis towards interpretable *ArgMD* solutions by assigning compulsory attack relations and limiting complexity of solutions. An interpretation example of *ArgMD* solutions is presented, offering practical instructions to International Order Reshaping.

4 Scenario and Goals of International Order Reshaping

This section introduces the practical scenario of current International Order and the preliminary “Prisoner’s Dilemma” games used to model the scenario. The goals of International Order Reshaping are claimed regarding the realistic political scenario and translated into mathematical expressions in Mechanism Design of games.

4.1 Scenario of International Order and Corresponding Game

MODEL

To analyse the international political and economic influence of countries, consider the following two sets.

Set 1. Permanent Members of the United Nations Security Council [UnitedNations2021]

$$U_1 = \{UK, US, France, Russia, China\}$$

Set 2. Five largest economies in the world in year 2021 [WorldBank2022]

$$U_2 = \{US, China, Japan, Germany, UK\}$$

The union of Set 1 and Set 2 consists of seven countries.

$$U_1 \cup U_2 = \{UK, US, France, Germany, Japan, Russia, China\}$$

These seven countries are involved in the summarized scenario and divided into two groups according to level of democracy [WorldPopulationReview2022], similarities in ideology and multilateral co-operations: Group 1 consists of five countries which are graded either "Full Democracy" or "Flawed Democracy"; Group 2 consists of the rest two countries which are both graded "Authoritarian Regime". Group 1 and Group 2 can be seen as two self-interested *players*, P_1 and P_2.

$$P_1 = \{UK, US, France, Germany, Japan\}$$

$$P_2 = \{Russia, China\}$$

"Prisoner's Dilemma" model is a widely used game model in analysing international conflicts—a case in point is recent research on US-China trade war [Yin2018]. In this section, the scenario of current International Order is built up as a five-dimensional "Prisoner's Dilemma" game.

Each player can take part in five independent "Prisoner's Dilemma" games which represent the competitions in the field of economy (G_1), culture (G_2), information society (G_3), political systems (G_4) and military forces (G_5) respectively. As shown in

Figure 5, the five games all have similar decision-tree structures but different utility functions for P_1 and P_2. For each player in each game there are two available decisions—*compromise* and *intransigence*, where

1. *compromise* means that the player is willing to modify its current status, make compromises and seek peace with the other player;

2. *intransigence* means that the player is unwilling to change its current status and is ready to start or upgrade conflicts with the other player.

Considering that China has already stirred up chaos in Hong Kong in 2019, and that Russia has started invading Ukraine and put the entire Europe in threat in early 2022, P_2 will always take the first move in all five games.

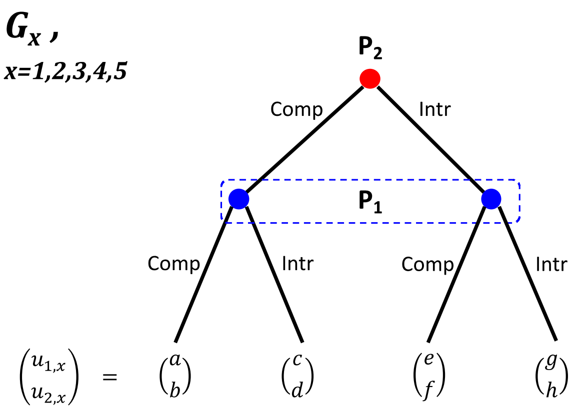


Figure 5. Extensive form of an independent main game between the two players.

SEQUENTIAL CHICKEN-GAME

To enforce peace in relationship, a player may sometimes shift to intransigence strategies to punish the other player for intransigence. Such punishment causes damage of interest on both sides but is usually temporary due to the greater comparative advantage of peace. However, such strategic shifts in the repeated

“Prisoner's Dilemma” model can be problematic due to reciprocal trigger strategies which prevent co-operation from resuming due to signalling problems and flaws in repeated play structure [Axelrod2000]. To avoid such problems, when mutual intransigence occurs, the game can be shifted into a chicken-game (as shown in Figure 6) instead, representing the dilemma of conflicts between two players. Players can choose either to

accept status quo: remain calm, prevent escalation and encourage reconciliation

or to

upgrade conflicts: change the disagreement into actual harm.

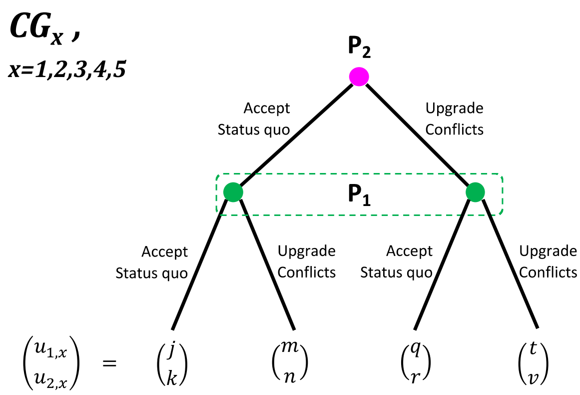


Figure 6. Extensive form of a chicken-game between the two players.

Figure 7 below shows the extensive form of the repeated “Prisoner's Dilemma” game with a modification to the chicken-game in the mutual intransigence outcome.

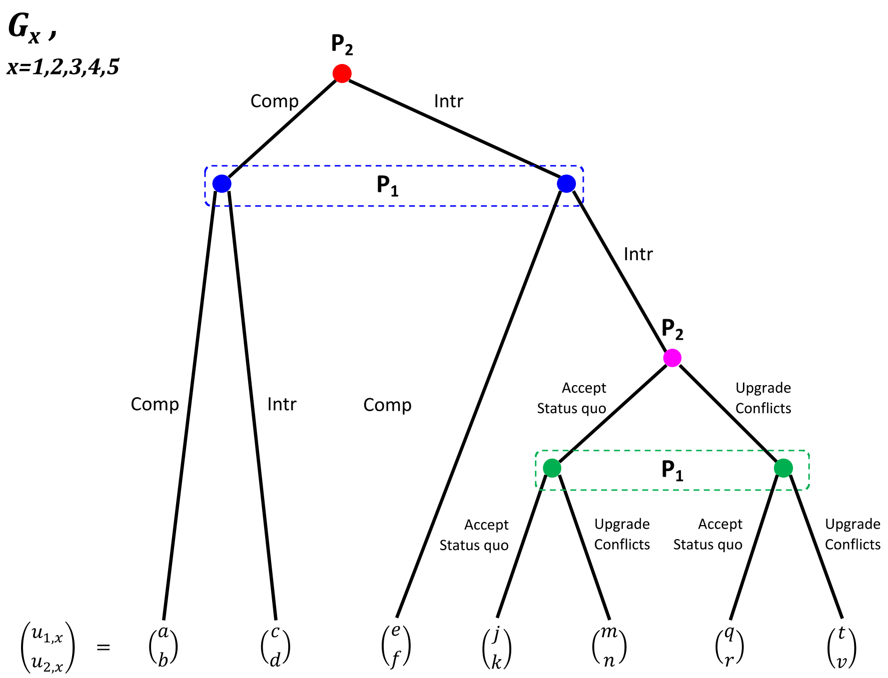


Figure 7. Extensive form of a chicken-game-modified "Prisoner's Dilemma" main game between the two players.

Normal form of chicken-game-modified "Prisoner's Dilemma" main game is given in Table 2.

Table 2. Normal form of a chicken-game-modified "Prisoner's Dilemma" main game between the two players.

<table>
<tr><td>P_2 / P_1</td><td>compromise</td><td colspan="3">intransigence</td></tr>
<tr><td>compromise</td><td>(a, b)</td><td colspan="3">(e, f)</td></tr>
<tr><td rowspan="3">intransigence</td><td rowspan="3">(c, d)</td><td>P_2 / P_1</td><td>accept status quo</td><td>upgrade conflicts</td></tr>
<tr><td>accept status quo</td><td>(j, k)</td><td>(q, r)</td></tr>
<tr><td>upgrade conflicts</td><td>(m, n)</td><td>(t, v)</td></tr>
</table>

PAYOFF ALTERNATIONS

Pairs of values of utility functions in the normal-form game shown in Figure 7 correspond to the following scenarios in the field represented by the game:

(a, b): Peace, with equality of positions.

(c, d): Unilateral declaration of conflicts, with P_1 in advantageous position.

(e, f): Unilateral declaration of conflicts, with P_2 in advantageous position.

(j, k): Temporary peace agreement to restrain conflicts, with equality of positions.

(m, n): Unilateral promotion of conflicts, with P_1 in advantageous position.

(q, r): Unilateral promotion of conflicts, with P_2 in advantageous position.

(t, v): Upgraded and lasting bilateral conflicts.

The five pairs of independent utility functions, as shown in Table 3, are given with respect to the significance in national strength of field each main game represents. Utilities in Table 3 are arbitrarily chosen to simulate the realistic influences of changes in multilateral international order on the two groups of countries.

Table 3. Utility functions in five chicken-game-modified main games.

	(a, b)	**(c, d)**	**(e, f)**	**(j, k)**	**(m, n)**	**(q, r)**	**(t, v)**
G_1	(8, 8)	(4, 0)	(0, 4)	(0, 0)	(5, -5)	(-5, 5)	(-30, -30)
G_2	(4, 4)	(2, 0)	(0, 2)	(0, 0)	(1, -1)	(-1, 1)	(-10, -10)
G_3	(5, 5)	(4, 0)	(0, 4)	(0, 0)	(5, -5)	(-5, 5)	(-15, -15)
G_4	(4, 4)	(3, 0)	(0, 3)	(0, 0)	(2, -2)	(-2, 2)	(-15, -15)
G_5	(30, 30)	(20, 0)	(0, 20)	(0, 0)	(10, -10)	(-10, 10)	(-100, -100)

The combined utility functions are calculated regarding the impact on civilization when conflicts in each field occur. Economy (G_1), culture (G_2), information societies (G_3) and political systems (G_4) together construct various shapes and levels of civilizations regardless of peace or conflicts in these four games. However, conflicts in G_5, which is also known as Hot War, sabotage human civilization on both sides. Therefore, the combined utility function can be designed as Equation 1.

$$u_i = n(U_i) \cdot u_{i,5} \cdot \exp\left(\sum_{k=1}^{4} u_{i,k}\right) \qquad (Equation\ 1)$$

In Equation 1, $n(U_i)$ denotes the number of unique elements in the set of countries.

4.2 Goals of Reshaping and Corresponding Mechanism Design

The goals of International Order Reshaping are to

1. Defend democratic values.

2. Prevent massive hot wars or ruin of civilization.

Correspondingly, *MD* problems of International Order Reshaping should guarantee that

1. The group of countries of higher level of democracy (P_1) prevails, having larger value of combined utility function than the other group (P_2):

$$u_1 > u_2 \qquad (Equation\ 2)$$

2. Value of combined utility functions should be greater than 0 for both groups:

$$\begin{cases} u_1 > 0 \\ u_2 > 0 \end{cases} \qquad (Equation\ 3)$$

Hence, we have $u_1 > u_2 > 0$, that is

$$n(U_1) \cdot u_{1,5} \cdot \exp\left(\sum_{k=1}^{4} u_{1,k}\right) > n(U_2) \cdot u_{2,5} \cdot \exp\left(\sum_{k=1}^{4} u_{2,k}\right) > 0 \qquad (Equation\ 4)$$

Noticing two properties of function $\exp(x)$: $\forall x, exp(x) > 0$ and $\forall x, \frac{\mathrm{d}}{\mathrm{d}x} exp(x) > 0$, A necessary condition for Equation 4 to hold is

$$\begin{cases} u_{1,5} > 0 \\ u_{2,5} > 0 \end{cases} \qquad (Equation\ 5)$$

Specially, when $n(U_1) \cdot u_{1,5} \leq n(U_2) \cdot u_{2,5}$, to guarantee that Equation 4 holds, there must be

$$\sum_{k=1}^{4} u_{1,k} > \sum_{k=1}^{4} u_{2,k}. \qquad (Equation\ 6)$$

Equation 5 and 6 form the mathematical basis of desirable game outcome selection for *MD* in answer set programming, which will be discussed in detail in Section 6.

5 Mapping Games to Answer Set Programming

Taking the preliminary "Prisoner's Dilemma" games as the test subject, this section tries out two different pathways of mapping games to *ASP*—"Extensive-form Game → *ASP*" and "Normal-form Game → *AF* → *ASP*". With the features of the two pathways juxtaposed, "Normal-form Game → *AF* → *ASP*" proves to be more suitable for the purpose of Mechanism Design.

5.1 Pathway 1: Extensive-form Game → *ASP*

Mapping extensive-form game to *ASP* is straightforward when seeing decision-tree as a node-edge style roadmap. Each edge is a strategy; each internal node of the decision-tree is a player's corresponding game status; and each terminal node (leaf) is a game outcome.

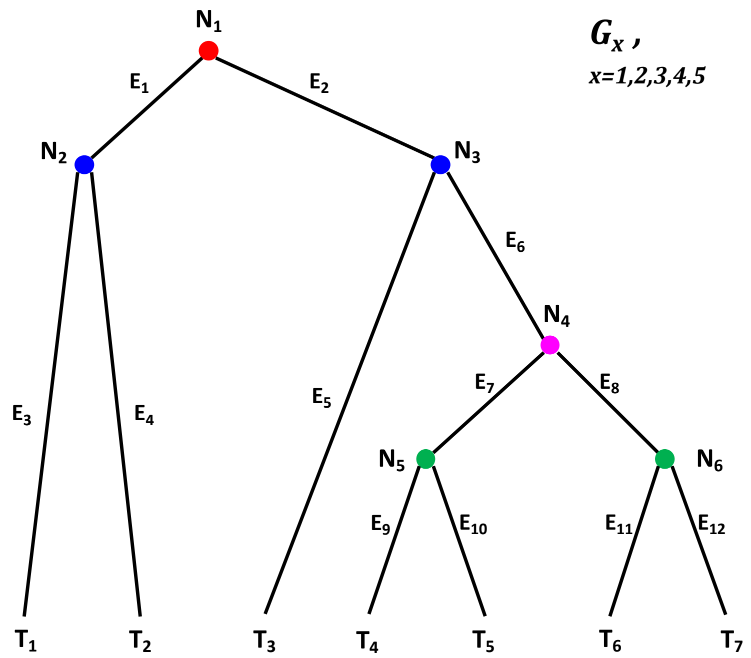


Figure 8. Marked edge-node style roadmap of International Order extensive-form game.

Decision-tree in Figure 7, with internal and terminal nodes marked as in Figure 8, is encoded in *ASP* as the following code trunks.

First off, all edges and nodes, as well as how nodes are connected by edges are recorded as follow.

```
strategy(e1).
...
strategy(e12).

node(n1).
```

```
  ...
  node(n6).

  node(t1).
  ...
  node(t7).

  proceed(n1, n2, e1).
  proceed(n1, n3, e2).
  proceed(n2, t1, e3).
  proceed(n2, t2, e4).
  proceed(n3, t3, e5).
  proceed(n3, n4, e6).
  proceed(n4, n5, e7).
  proceed(n4, n6, e8).
  proceed(n5, t4, e9).
  proceed(n5, t5, e10).
  proceed(n6, t6, e11).
  proceed(n6, t7, e12).
  -proceed(X,Y,E) :- node(X), node(Y), strategy(E), not
proceed(X,Y,E).
```

Then, class structure and categorisation are recorded as follow. Both game status and game outcomes are denoted as nodes in the decision tree and must be distinguished. Among all game outcomes, Nash Equilibria and Quasi Nash Equilibria should also be especially marked out.

```
  class(node).
  class(gamestatus).
  class(outcome).
  class(nash_equilibrium).
  class(quasi_nash_equilibrium).

  is_subclass(gamestatus, node).
  is_subclass(outcome, node).
  is_subclass(nash_equilibrium, outcome).
  is_subclass(quasi_nash_equilibrium, outcome).

  subclass(C1,C2) :- is_subclass(C1,C2).
  subclass(C1,C2) :- is_subclass(C1,C3), subclass(C3,C2).
  -subclass(C1,C2) :- class(C1), class(C2), not
subclass(C1,C2).

  member(X,C) :- in(X,C).
```

```
  member(X,C) :- in(X,C0), subclass(C0,C).
  siblings(C1,C2) :- is_subclass(C1,C), is_subclass(C2,C),
C1 != C2.
  -member(X,C2) :- member(X,C1), siblings(C1,C2).

  in(n1, gamestatus).
  ...
  in(n6, gamestatus).

  in(t1, nash_equilibrium).
  in(t2, outcome).
  in(t3, outcome).
  in(t4, outcome).
  in(t5, quasi_nash_equilibrium).
  in(t6, quasi_nash_equilibrium).
  in(t7, outcome).
```

Then, predicate for retrieving reachable outcomes is given based on simple logic that a game outcome (t_i) is reachable only if at least one route starting from game-start (n_1) while ending at t_i exist.

```
  route(X,Y) :- proceed(X,Y,M), not restricted(M).
  route(X,Y) :- proceed(X,Z,M), not restricted(M), route(Z,Y).
  -route(X,Y) :- node(X), node(Y), not route(X,Y).

  reachable(T) :- member(T,outcome), route(n1,T).
  -reachable(T) :- member(T,outcome), not route(n1,T).

  reachable_equilibrium(Q) :- in(Q, nash_equilibrium),
reachable(Q).
  -reachable_equilibrium(Q) :- in(Q, nash_equilibrium), not
reachable(Q).

  reachable_quasi_equilibrium(Q) :- in(Q,
quasi_nash_equilibrium), reachable(Q).
  -reachable_quasi_equilibrium(Q) :- in(Q,
quasi_nash_equilibrium), not reachable(Q).
```

Next step is where *MD* can be conducted in Pathway 1. Certain edges (strategies) can be restricted here by mechanism designer, so that part of game outcomes becomes unreachable and only desirable game outcomes are kept.

```
%e.g.% restricted(e7).
```

The last part of code allows reachable game outcomes, as well as reachable Nash Equilibria and reachable Quasi Nash Equilibria to be shown in solved answer sets.

```
#show reachable/1.
#show -reachable/1.

#show reachable_equilibrium/1.
#show -reachable_equilibrium/1.

#show reachable_quasi_equilibrium/1.
#show -reachable_quasi_equilibrium/1.
```

The full code for Pathway 1 of mapping games to *ASP* is given in Appendix 2.

5.2 Pathway 2: Normal-form Game → *AF* → *ASP*

NORMAL-FORM GAME → *AF*

As the first step of portraying the argumentative scenario, each available independent strategy (a single move in the game) can be denoted as a corresponding argument. That is to say, an “argument” defined in Definition IP-1 is understood as an independent strategy.

a_1: P_2 chooses compromise.

a_2: P_2 chooses intransigence.

a_3: P_1 chooses compromise.

a_4: P_1 chooses intransigence.

a_5: P_2 ready to accept status quo (limited potential conflicts).

a_6: P_2 ready to upgrade conflicts.

a_7: P_1 ready to accept status quo (limited potential conflicts).

a_8: P_1ready to upgrade conflicts.

For agent P_2, its argument set $\mathcal{A}_2$ could be prepared as one of the following: $\{a_1\}$, $\{a_2, a_5\}$, $\{a_2, a_6\}$. For agent P_1, its argument set $\mathcal{A}_1$ could be prepared as one of the following: $\{a_3\}$, $\{a_4, a_7\}$, $\{a_4, a_8\}$. In this case, the normal form of game in Section 4 could be written as the following format.

Table 4. Argumentative payoff alterations.

$\mathcal{A}_1$ \ $\mathcal{A}_2$	$\{a_1\}$	$\{a_2, a_5\}$	$\{a_2, a_6\}$
$\{a_3\}$	(a, b)	(e, f)	(e, f)
$\{a_4, a_7\}$	(c, d)	(j, k)	(q, r)
$\{a_4, a_8\}$	(c, d)	(m, n)	(t, v)

Considering each independent strategy as an argument, the binary attack relation set $\mathcal{R}$ in this *AF* can be clarified as shown in Figure 9.

It is worth mentioning that unlike the case of "Battle of Sexes" models (e.g. model in [Rahwan2009]), additional predicates need to be added to accurately map the "Prisoner's Dilemma" model into *ASPARTIX* input. In a "Prisoner's Dilemma" model such as the one used in Section 4, a player's available strategies at each decision node (game step) are mutually exclusive. The corresponding arguments do not form attack relations with each other due to non-self-attack principles. However, these arguments are also mutually exclusive and should be prevented from co-existing in the extensions of *AF*.

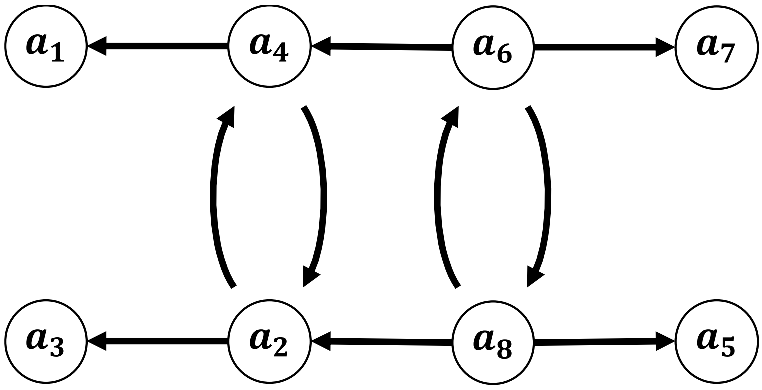


Figure 9. AF argument graph of a chicken-game-modified "Prisoner's Dilemma" main game between the two players., transformed from normal-form game with each argument representing an independent strategy.

In Figure 9, as it is in the previous research, each game-based argument in $\mathcal{A}_G$ is interpreted as "choosing a certain independent strategy" (such as a_2) instead of "choosing a strategy set consisting of multiple moves in the game" (such as $\{a_2, a_5\}$). Such interpretation is problematic in terms of solving game theory or mechanism design problems in *ASP*—On one hand, in an *AF* based on such interpretation, Nash Equilibria of game do not have uniform corresponding extension semantics, meaning that Nash Equilibria cannot be enumerated in *ASPARTIX*. On the other hand, such interpretation does not comply with Definition IP-2, *"Given a normal-form game* $G = (N, S, u)$ *and its set of available game-based arguments* $\mathcal{A}_G$*, denote the set of game-based attack relations as* $\mathcal{R}_G \subseteq \mathcal{A}_i \times \mathcal{A}_j$*,* $(i \neq j)$*.* $(a_j, a_i) \in \mathcal{R}_G$ *if* s_i *is not the best response to all* s_{-i} *that contain* s_j*"*, since the concept of "best response" is not based on single moves in game but the dominant relations between strategy sets each consisting of a series of moves.

Hence, to make *GBAF* solvable in *ASPARTIX* and to comply with Definition IP-2, the concept of "argument" in Definition IP-1 should be interpretated as "choosing strategy set" instead of "choosing independent strategy". To clarify and to distinguish, an argument representing "strategy set" is denoted as ***strategy set argument*** **(SSA)** in this paper. Each SSA represents an available strategy set for a player. For example, in the normal-form game in Figure 9, denote

$SSA_1 = \{a_1\}$, $SSA_3 = \{a_2, a_5\}$, $SSA_5 = \{a_2, a_6\}$,

$SSA_2 = \{a_3\}$, $SSA_4 = \{a_4, a_7\}$, $SSA_6 = \{a_4, a_8\}$.

Scanning through all cells of the payoff alteration table, the attack relations between SSAs are determined according to best response. Take the shaded cell (SSA_2, SSA_3) as an example. Scan the row where the cell is located, when and only when SPA_3 is NOT the best response to SSA_2 ($f \neq \max(b, f, f)$), SSA_3 is attacked by SSA_2. Scan the column where the cell is located, only when SSA_2 is NOT the best response to SSA_3 ($e \neq \max(e, j, m)$), then SSA_2 is attacked by SSA_3.

Table 5. Attack relations between SSAs.

$\mathcal{A}_2$ / $\mathcal{A}_1$	SSA_1	SSA_3	SSA_5
SSA_2	(a, b)	(e, f)	(e, f)
SSA_4	(c, d)	(j, k)	(q, r)
SSA_6	(c, d)	(m, n)	(t, v)

Correspondingly, the *ASPARTIX* input in apx format should be as shown in the code trunk below according to *ASPARTIX* encodings of Dung's *AF*.

Denote the *AF* as $GBAF = \langle \mathcal{A}_G, \mathcal{R}_G \rangle$, SSA set $\mathcal{A}_G$ is recorded as follows. For the convenience of analysis on attack relations, Nash Equilibria and *MD*, each SSA is marked which player it belongs to.

```
arg(ssa1).
...
arg(ssa6).

strategy_set_of(ssa1,p2).
...
strategy_set_of(ssa6,p1).
```

Attack relation set $\mathcal{R}_G$ is recorded as follows.

```
att(ssa1,ssa4).
att(ssa1,ssa6).
att(ssa3,ssa4).
att(ssa3,ssa6).
att(ssa5,ssa4).
att(ssa5,ssa6).

att(ssa2,ssa3).
att(ssa2,ssa5).
att(ssa4,ssa3).
att(ssa4,ssa5).
att(ssa6,ssa3).
att(ssa6,ssa5).
```

Correspondence 5 in *GBAF* states that "*Given a normal-form game $G = (N, S, u)$ where $N = \{1,2\}$ and the corresponding AF_G, a strategy profile $s = (s_1, s_2)$ is a Nash Equilibrium if in the AF_G transferred from G, the corresponding $\mathcal{A}_{sp} = \{a_1, a_2\}$ is conflict-free".* Also, Dung points out Nash Equilibria should correspond to stable extensions of game-based *AF* [Dung1995]. Hence, to query for Nash Equilibrium in the game of International Order Reshaping, enumerate stable extension(s) of corresponding *AF* above w.r.t. semantics given in 'stable.dl'. Full code for mapping chicken-game-modified “Prisoner’s Dilemma” main games via Pathway 2 is given in Appendix 3.

5.3 Evaluation and Comparison of Pathways

The two pathways have the following distinctive features.

1. Pathway 1 is straightforward in logic while Pathway 2 involves solution concept transformations.
2. Pathway 1 retains all information of game’s process while Pathway 2 only preserves the corelations between strategy profiles and game outcomes. Hence, modification of the game process, without change of game model, can be easily captured by Pathway 1.
3. Code volume is significantly larger in Pathway 1 than in Pathway 2.
4. Nash Equilibria are rather difficult to locate in Pathway 1 but very easy to detect in Pathway 2. In a normal-form game, whether a game outcome is a Nash Equilibrium can be judged by simply comparing values of utility functions within the row and the column where the game outcome is located.

Considering that *MD* wishes the game outcome depends on the preferences of the players, while Nash Equilibria are the reconciliation of different players’ preferences, *MD* is impossible to be performed without accurate locations of Nash Equilibria. Also, *MD* may involve change of game model when the original model does not generate any desirable outcomes. Hence, for the purpose of *MD*, Pathway 2 (Normal-form Game → *AF* → *ASP*) proves to be a better pathway of mapping game to *ASP*.

6 Mechanism Design with Answer Set Programming

This section conducts *MD* with the application of *ASP*. First off, the game model is re-designed in order to fulfil the goals of *MD*. The preliminary “Prisoner’s Dilemma” model, which fails to comprehensively reflect the International Order scenario and allows no room for *MD*, is replaced by the “Battle of Sexes” model. Then, *ArgMD* is realised in *ASP* based on the *GBAF* derived from the re-designed game. This section also promotes two methods to increase the interpretability of *ArgMD* solutions—assigning compulsory attack relations and limiting complexity of solutions. An example of interpretation of *ArgMD* solutions is presented to demonstrate how *ArgMD* in *ASP* is capable of providing practical, pragmatic instructions to International Order Reshaping.

6.1 Failure of “Prisoner’s Dilemma” Model

With *ASP* semantics set as ‘stable.dl’, Nash Equilibria of a game are revealed as stable sets in the corresponding *GBAF*. As for the *GBAF* of the game in Section 4, *ASP* gives its stable extensions as follows.

```
Solving...
Answer: 1
nash(spa2,spa1)
Answer: 2

Answer: 3

SATISFIABLE

Models       : 3
```

That is, for the game in Section 4 with “Prisoner’s Dilemma” model, there is one and only one Nash Equilibrium, meaning that there is no room for *MD* without change of model—*MD* is voided when there is no alternative Nash Equilibrium. Moreover, it is necessary to examine the meaning of this Nash Equilibrium in order to answer

1. whether the game outcome reaches the goals of *MD*.
2. whether the selection of strategies profiles is in line with reality.

As for the first question above, recall that the goals of *MD* are to defend democratic values and to prevent massive hot wars or ruin of civilization, mathematically $u_1 > u_2 > 0$. Game outcome at the returned Nash Equilibrium is $(u_{1,x}, u_{2,x}) = (a, b)$ where $a \equiv b > 0$, meaning that $u_1 = u_2 > 0$. Therefore, the returned Nash Equilibrium fails to reach the goals of *MD*—wars are prevented yet democracy does not prevail.

As for the second question above, recall the meanings of a_1~a_8. The returned Nash Equilibrium only occurs when the two players conduct the following strategy sets: ***P_1* *and P_2 both make compromise, willing to modify its current status and seek peace with the other player; even in potential cases where conflicts are triggered, both P_1 and P_2 remain calm, prevent escalation and encourage reconciliation instead of upgrading conflicts.*** In other words, to maximise their benefits, the two self-interested groups of countries will find "the ideal peace" with each other without any modifications to current International Order. Beautiful as it may sound, "the ideal peace" is more of wishful thinking than probable reality, let alone that peace has already been broken at the moment.

To summarise the above, "Prisoner's Dilemma" games fail to solve International Order Reshaping problems. The reason of this failure lies in either the utility functions or the model structure. As for utility functions, one notable characteristic is that $c = f < a = b$ (see Table 3), meaning that both groups of countries prefer realising "the ideal peace" over taking "the advantageous position", which is against the essence of competitive relationship between the two rival camps. However, if the mathematical relations are changed into $c = f > a = b$ to keep in line with the reality, a more vital problem occurs—there will be no Nash Equilibrium in the game (see Table 4). Hence, the failure of “Prisoner’s Dilemma” games is caused by the model itself—despite that “Prisoner’s Dilemma” model is useful in analysing repetitive competitions, its structure is not capable of providing the foundations of *MD*—more than one Nash Equilibria.

Moreover, the “Prisoner’s Dilemma” model deprives players of the rights to knowing the decisions made by each other, making their strategies blind and simultaneous. This is again against the reality since that countries take actions in turn and usually take actions based on the intelligence on the actions already made by the other.

6.2 Redesigned Games with “Battle of Sexes” Model

To relocate the *MD* problem, intentions of the two players should be more clearly revealed in the games. In the initial failed model, each player has intention for either “war” or “peace”. However, intention for “peace” has proved to be problematic in *MD* and in revealing the player’s type. Practically speaking, when groups of countries go in competition, instead of taking a draw, they would naturally wish to win and seize the advantageous position. If a group of countries fails to achieve such an ambition, it will have to surrender the advantageous position to the opposite group. Both groups of countries are allowed to make forceful counterattack to regain the advantageous position. Worst case scenario, both groups are unwilling to accept an inferior position, conflicts occur. That is to say, there should be three unique game outcomes in total.

Also, both groups should be informed and aware of the moves already made by the opposite group. This should be guaranteed by multilateral consultations and negotiations.

"Battle of Sexes" model, one of the classic models in Game Theory, can meet all the requirements mentioned above. Definition of the two players and the calculation of combined utility functions remain the same as those in the initial model used in Section 4. With game strategies denoted as *AF* arguments as follow, the redesigned games are given in extensive form (Figure 10), normal form (Table 6) and Dung's *AF* (Figure 11).

{ }: P_1 or P_2 start conflicts with the opposite player without negotiation.

α_1: The world should accept in peace that P_2 is in advantageous position.

α_2: The world should accept in peace that P_1 is in advantageous position.

α_3: P_2 is unsatisfied with inferiority to P_1. P_2 will make counterattack to force P_1 to hand over the advantageous position.

α_4: P_1 is unsatisfied with inferiority to P_2. P_1 will make counterattack to force P_2 to hand over the advantageous position.

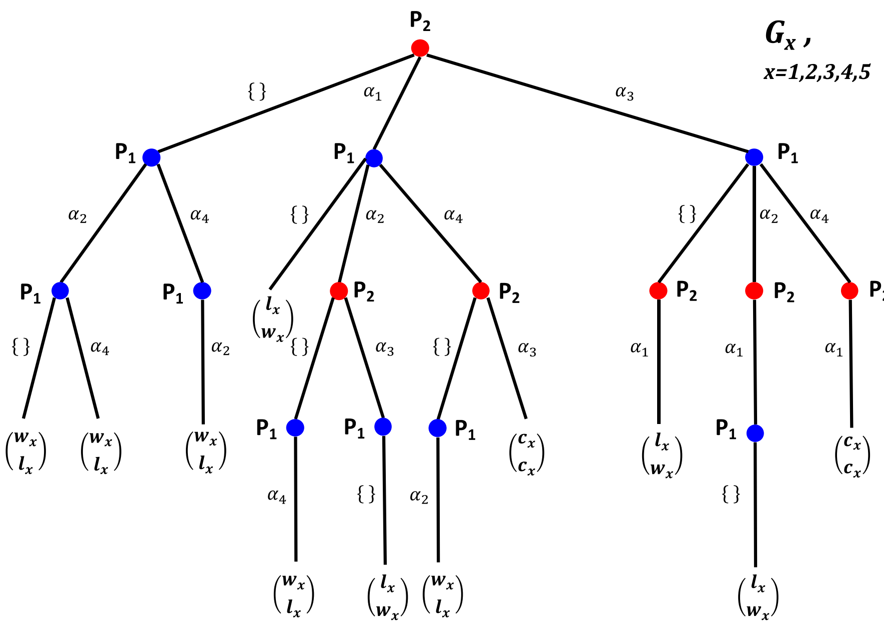


Figure 10. Extensive form of a redesigned main game (with Battle-of-Sexes model) between the two players.

Table 6. Normal form of a redesigned main game (with Battle-of-Sexes model) between the two players.

$\mathcal{A}_1$ \ $\mathcal{A}_2$	$SSA_1 = \{\alpha_1\}$	$SSA_3 = \{\alpha_3\}$	$SSA_5 = \{\alpha_1, \alpha_3\}$	$SSA_7 = \{\ \}$
$SSA_2 = \{\alpha_2\}$	(c_x, c_x)	(c_x, c_x)	(l_x, w_x)	(w_x, l_x)
$SSA_4 = \{\alpha_4\}$	(c_x, c_x)	(c_x, c_x)	(c_x, c_x)	(c_x, c_x)
$SSA_6 = \{\alpha_2, \alpha_4\}$	(w_x, l_x)	(c_x, c_x)	(c_x, c_x)	(w_x, l_x)
$SSA_8 = \{\ \}$	(w_x, l_x)	(c_x, c_x)	(w_x, l_x)	(c_x, c_x)

In Table 6, c_x denotes player's utility when conflicts are started in field x, l_x denotes player's utility when it is in peace but with a disadvantageous position in field x, while w_x denotes player's utility when it is in peace with an advantageous position in field x. Values of c_x, l_x and w_x in five "Battle of Sexes" main games are given in Table 7 below.

Table 7. Utility functions in five "Battle of Sexes" main games.

	c_x	l_x	w_x
G_1	-10		4
G_2	-5		2
G_3	-10	0	3
G_4	-3		2
G_5	-100		40

$$\begin{pmatrix} u_{1,x} \\ u_{2,x} \end{pmatrix} = \begin{cases} (c_x, c_x), & P_1\ P_2\ in\ conflicts \\ (w_x, l_x), & P_1\ advantageous \\ (l_x, w_x), & P_2\ advantageous \end{cases}, \quad x = 1, 2, 3, 4, 5.$$

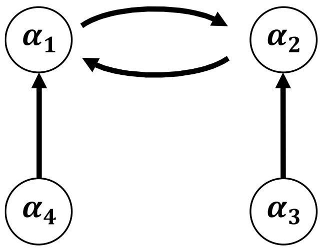


Figure 11. AF argument graph of a redesigned main game (with Battle-of-Sexes model) between the two players., transformed from normal-form game with each argument representing an independent strategy.

Based on the normal form in Table 5 and Definition IP-2, the correspondingly *ASPARTIX* input in apx format should be as shown in the code trunk below according to *ASPARTIX* encodings of Dung's *AF*.

SSA set $\mathcal{A}_G$ is recorded as follows. For the convenience of analysis on attack relations, Nash Equilibria and *MD*, each SSA is marked which player it belongs to—SSA_1, SSA_3, SSA_5 and SSA_7 belong to P_2, while the rest belong to P_1.

```
arg(ssa1).
...
arg(ssa8).

strategy_set_of(ssa1,p2).
...
strategy_set_of(ssa8,p1).
```

Attack relation set $\mathcal{R}_G$ is recorded as follows. Each attack relation `att(X,Y)` is derived from the fact that Y is not the best response to X.

```
att(ssa1,ssa2).
att(ssa1,ssa4).
att(ssa1,ssa8).
att(ssa3,ssa2).
att(ssa3,ssa4).
att(ssa3,ssa6).
att(ssa3,ssa8).
att(ssa5,ssa4).
```

```
att(ssa5,ssa6).
att(ssa7,ssa2).
att(ssa7,ssa4).
att(ssa7,ssa8).

att(ssa2,ssa1).
att(ssa2,ssa3).
att(ssa2,ssa7).
att(ssa4,ssa1).
att(ssa4,ssa3).
att(ssa4,ssa5).
att(ssa4,ssa7).
att(ssa6,ssa3).
att(ssa6,ssa5).
att(ssa8,ssa1).
att(ssa8,ssa3).
att(ssa8,ssa7).
```

Implementing stable extension semantics in *ASP* given by 'stable.dl', four Nash Equilibria are returned: $(\{\alpha_2,\alpha_4\},\{\alpha_1\})$, $(\{\alpha_2,\alpha_4\},\{\ \})$, $(\{\alpha_2\},\{\alpha_1,\alpha_3\})$, $(\{\ \},\{\alpha_1,\alpha_3\})$. Full code for mapping "Battle of Sexes" main games via Pathway 2 is given in Appendix 4.

```
Solving...
Answer: 1

Answer: 2
nash(ssa6,ssa1) nash(ssa6,ssa7)
Answer: 3
nash(ssa2,ssa5) nash(ssa8,ssa5)
Answer: 4

SATISFIABLE

Models       : 4
```

6.3 Argumentation Mechanism Design in Answer Set Programming

MD wishes to guarantee desirable system-wide game outcomes when there is a group of self-interested agents who have preferences over the outcomes. In other words, *MD* wishes to narrow Nash Equilibria down to only desirable game outcomes.

The goals of International Order Reshaping are to prevent defend democratic values and to prevent massive hot wars. Examining separate utility functions given in Table 7 and the combine utility functions $u_i = u_{i,5} \cdot \exp\left(\sum_{k=1}^{4} u_{i,k}\right)$, only two among the returned four Nash Equilibria in G_5, $(\{\alpha_2, \alpha_4\}, \{\alpha_1\})$ and $(\{\alpha_2, \alpha_4\}, \{\ \})$ achieve the goals of International Order Reshaping. The practical meaning of $(\{\alpha_2, \alpha_4\}, \{\alpha_1\})$ and $(\{\alpha_2, \alpha_4\}, \{\ \})$ can be interpretated as "democratic countries (Player 1) have fully executed all available strategies to secure their advantageous position, while Russia and China (Player 2) have taken no or only shallow actions to fight back", which is a highly desirable status of global politics and perfectly matches the values that the two International Order Reshaping goals represent. Therefore, $(\{\alpha_2, \alpha_4\}, \{\alpha_1\})$ and $(\{\alpha_2, \alpha_4\}, \{\ \})$ should be considered as desirable game outcomes. In other words, they should be exactly the Nash Equilibria in *MD* solutions.

One major way of conducting Mechanism Design in Argumentation Frameworks, or realising *ArgMD*, is to modify attack relations $\mathcal{R}_G$ in $GBAF = \langle \mathcal{A}_G, \mathcal{R}_G \rangle$ so that each Nash Equilibrium enumerated based on stable extension semantics is a desirable game outcome selected by the game designer, vice versa. In *ASP*, such modification of attack relations can be implemented based on the following logic.

Step 1. Retrieve the maximal attack relation set $\mathcal{R}_{G,max}$ which consists of all possible attack relations. That is, any pair of arguments from two different players may attack each other. Step 2. Guess a set of attack relations $\mathcal{R}'_G \subseteq \mathcal{R}_{G,max}$. According to mechanism designer's requests, some particular attack relations in $\mathcal{R}_{G,max}$ can be compulsory and must be included in $\mathcal{R}'_G$. Step 3. Enumerate the stable extensions of corresponding $GBAF' = \langle \mathcal{A}_G, \mathcal{R}'_G \rangle$, detect Nash Equilibria. Step 4. Compare the set of desirable game outcomes $\mathcal{O}$ and the set of Nash Equilibria $\mathcal{N}'$, only when $\mathcal{O} = \mathcal{N}'$, $\mathcal{R}'_G$ is accepted as a solution of *ArgMD*.

Modification of attack relations in *ArgMD* is encodable in *ASP*. In the case of "Battle of Sexes" main games, Step 1 and Step 2 above can be encoded as the following code trunk. All returned answers of `may_att/2` form $\mathcal{R}_{G,max}$, all returned answers of `att_in/2` form $\mathcal{R}'_G$ while all returned answers of `att_out/2` make up $\mathcal{R}_{G,max} \backslash \mathcal{R}'_G$.

```
  may_att(X,Y) :- strategy_set_of(X,P1),
strategy_set_of(Y,P2), P1!=P2.
  att_in(X,Y) :- not att_out(X,Y), may_att(X,Y).
  att_out(X,Y) :- not att_in(X,Y), may_att(X,Y).
```

Compulsory attack relations assigned by mechanism designer should be directly added here, for example, `att_in(ssa5,ssa6)`.

```
  att_in(ssa5,ssa6).
```

ASP coding for Step 3 is similar to that for retrieving Nash Equilibria of the initial games in Sub-section 7.2. Basically, this step is to enumerate stable extensions of the new $GBAF' = \langle \mathcal{A}_G, \mathcal{R}'_G \rangle$ by applying the semantics given by 'stable.dl', , the only difference is that `att/2` is replaced by `att_in/2` to encode $\mathcal{R}'_G$ instead of the initial $\mathcal{R}_G$. Then, among the returned answers of `in/1`, select all pairs of SPAs where the first SSA belongs to Player 1 while the second SSA belongs to Player 2. The selected pairs of SPAs are the Nash Equilibria for the modified new game corresponding to $GBAF' = \langle \mathcal{A}_G, \mathcal{R}'_G \rangle$. Selection of Nash Equilibria is encoded by the following part of code.

```
  nash(X,Y) :- strategy_set_of(X,p1), strategy_set_of(Y,p2),
in(X), in(Y).
  -nash(X,Y) :- strategy_set_of(X,p1), strategy_set_of(Y,p2),
not nash(X,Y).
```

To encode Step 4 in *ASP*, the set of desirable outcomes $\mathcal{O}$ should be given by mechanism designer as a beginning. As for the case of "Battle of Sexes" model International Order Reshaping, the desirable game outcomes are $(\{\alpha_2,\alpha_4\},\{\alpha_1\})$ and $(\{\alpha_2,\alpha_4\},\{\ \})$, which correspond to $(SSA6, SSA1)$ and $(SSA6, SSA7)$ according to Table 6. To accurately encode the set of desirable game outcomes $\mathcal{O}$, it must be pointed out in *ASP* that any other cell in Table 6 is not a desirable game outcome. *ASP* encoding for $\mathcal{O}$ is as follows.

```
  desirable_outcome(ssa6,ssa1).
  desirable_outcome(ssa6,ssa7).
  -desirable_outcome(SS1,SS2) :- strategy_set_of(SS1,p1),
strategy_set_of(SS2,p2), not desirable_outcome(SS1,SS2).
```

The equality between the set of desirable outcomes and the set of Nash Equilibria is equivalent to "No game outcome is a Nash Equilibrium but not a desirable outcome, vice versa". Mathematically, $\mathcal{O} = \mathcal{N}' \Leftrightarrow \{\{\nexists o | o \notin \mathcal{O}, o \in \mathcal{N}'\}$ and $\{\nexists o | o \notin \mathcal{N}', o \in \mathcal{O}\}\}$. Therefore, the equality between $\mathcal{O}$ and $\mathcal{N}'$ can be encoded as follows.

```
  :- nash(SS1,SS2), not desirable_outcome(SS1,SS2).
  :- desirable_outcome(SS1,SS2), not nash(SS1,SS2).
```

In the case of "Battle of Sexes" model International Order Reshaping where $(\{\alpha_2,\alpha_4\},\{\alpha_1\})$ and $(\{\alpha_2,\alpha_4\},\{\ \})$ are desirable game outcomes, *ASP* code of *ArgMD* attack relation modification is given in Appendix 5.

6.4 Towards Interpretable *ArgMD* Solutions

INCREASE INTERPRETABILITY OF *ArgMD SOLUTIONS*

When no compulsory attack relations are assigned by mechanism designer, *ASP* code in Appendix 5 returns as many as 18,984,375 solutions, which all guarantee desirable game outcomes to be exactly the Nash Equilibria.

```
Solving...
Answer: 1

...
Answer: 18984375
...
SATISFIABLE

Models       : 18984375
```

In a *GBAF*, the more available strategy sets for each player, the more elements the maximal attack relation set $\mathcal{R}_{G,max}$ contains. For an $n \times n$ normal-form game, $\mathcal{R}_{G,max}$ consists of $2n^2$ attack relations. Therefore, the number of all possible $\mathcal{R}'_G$ is $2^{2n^2} = 4^{n^2}$, which double-exponentially blows up as n increases. For example, there are 256 possible $\mathcal{R}'_G$ when $n = 2$, while there are 4,294,967,296 possible $\mathcal{R}'_G$ when $n = 4$. Consequently, given the same number of desirable game outcomes, number of *ArgMD* solutions also rockets as the game complicates. This is very problematic when it comes to explaining *ArgMD* solutions or choosing *MD* strategies.

Assigning compulsory attack relations adds extract constraints to the model and naturally cut down the number of *ArgMD* solutions. However, assigning compulsory attack relations does not necessarily reduce the complexity of each *ArgMD* solution—"the total number of answer sets” is a very different concept from “the number of att_in/1 in each answer set”.

To push forward analysis towards interpretable *ArgMD* solutions, mechanism designers should first pay attention to the conciseness of solutions. Hence, this research promotes another approach, where not only fewer *ArgMD* solutions are returned but finding the most concise solution(s) is feasible. For an *ArgMD* solution, the fewer attack relations $\mathcal{R}'_G$ contains, the more concise and interpretable the corresponding $GBAF'$ is. Mechanism designer can claim a maximum acceptable number of attack relations (e.g. `max_num_ar=10`), the following *ASP* code should allow users to rule out all answer sets that construct a more complicated $\mathcal{R}'_G$ with more attack relations.

```
#const max_num_ar=10.
:- #count{X,Y:att_in(X,Y)}>max_num_ar.
```

Mechanism designer can assign various value to `max_num_ar` to further analyse different clusters of *ArgMD* solutions. The minimum value of `max_num_ar` that allows the model to be SATISFIABLE is the size of $\mathcal{R}'_G$ in the most concise *ArgMD* solution(s). In the case studied in this research, the size of the most concise $\mathcal{R}'_G$ is 10. Model solved in *ASP* as follows.

```
Solving...
Answer: 1

...
Answer: 6144
...
SATISFIABLE

Models       : 6144
```

Despite that some attack relations can be difficult to explain in the first place, the necessity of other attack relations can be rather obvious if mechanism designers reflected on reality of the game background. For example, in the case of International Order Reshaping, it is apparently against both player's interest to have conflicts started without any negotiation at all. That is, $SSA_7 = \{\,\}$ and $SSA_8 = \{\,\}$ are supposed to attack each other. Also, it would be the worst case scenario for global politics if conflicts become inevitable—both players have chosen all available strategies. That is, $SSA_5 = \{\alpha_1, \alpha_3\}$ and $SSA_6 = \{\alpha_2, \alpha_4\}$ are supposed to attack each other. Therefore, as for the compulsory attack relations, the following predicates can be given in *ASP*.

```
att_in(ssa7,ssa8).
att_in(ssa8,ssa7).
att_in(ssa5,ssa6).
att_in(ssa6,ssa5).
```

With consideration of both conciseness of solutions and compulsory attack relations, only 192 stand out from the total 18,984,375 *ArgMD* solutions. Upgraded *ASP* code which generates the following solutions is recorded in Appendix 6.

```
Solving...
Answer: 1
att_in(ssa7,ssa8) att_in(ssa8,ssa7) att_in(ssa5,ssa6)
att_in(ssa6,ssa5) att_in(ssa6,ssa3) att_in(ssa2,ssa1)
att_in(ssa1,ssa2) att_in(ssa4,ssa3) att_in(ssa1,ssa4)
att_in(ssa3,ssa6)
...
Answer: 192
att_in(ssa7,ssa8) att_in(ssa8,ssa7) att_in(ssa5,ssa6)
att_in(ssa6,ssa5) att_in(ssa6,ssa3) att_in(ssa7,ssa2)
att_in(ssa1,ssa4) att_in(ssa3,ssa4) att_in(ssa4,ssa5)
att_in(ssa2,ssa7)
SATISFIABLE

Models        : 192
```

Each of the 192 solutions is not only highly efficient—reaching the goals of *MD* with a minimum number of attack relations, but also of the highest interpretability among all *ArgMD* solutions.

INTERPRETATION EXAMPLE

Taking Answer 1 above as an example, it can be shown how an *ArgMD* solution reflects back on the game reality and gives instructions to International Order Reshaping. Recalling the definition of attack relations between SSAs: SSA_i attacks SSA_j when and only when SSA_j is NOT the best response to SSA_i, Answer 1 can be interpretated as follows.

1. $SSA_1 = \{\alpha_1\}$ is not the best response to $SSA_2 = \{\alpha_2\}$, vice versa.
2. $SSA_5 = \{\alpha_1, \alpha_3\}$ is not the best response to $SSA_6 = \{\alpha_2, \alpha_4\}$, vice versa.
3. $SSA_7 = \{\ \}$ is not the best response to $SSA_8 = \{\ \}$, vice versa.
4. $SSA_3 = \{\alpha_3\}$ is not the best response to $SSA_6 = \{\alpha_2, \alpha_4\}$, vice versa.
5. $SSA_3 = \{\alpha_3\}$ is not the best response to $SSA_4 = \{\alpha_4\}$.
6. $SSA_4 = \{\alpha_4\}$ is not the best response to $SSA_1 = \{\alpha_1\}$.

Table 8. Normal form of G_5 for ArgMD attack relation modification.

$\mathcal{A}_1$ \ $\mathcal{A}_2$	$\boldsymbol{SSA_1} = \{\boldsymbol{\alpha_1}\}$	$\boldsymbol{SSA_3} = \{\boldsymbol{\alpha_3}\}$	$\boldsymbol{SSA_5} = \{\boldsymbol{\alpha_1}, \boldsymbol{\alpha_3}\}$	$\boldsymbol{SSA_7} = \{\ \}$
$\boldsymbol{SSA_2} = \{\boldsymbol{\alpha_2}\}$	(a_1, a_2)	(b_1, b_2)	(c_1, c_2)	(d_1, d_2)
$\boldsymbol{SSA_4} = \{\boldsymbol{\alpha_4}\}$	(e_1, e_2)	(f_1, f_2)	(g_1, g_2)	(h_1, h_2)
$\boldsymbol{SSA_6} = \{\boldsymbol{\alpha_2}, \boldsymbol{\alpha_4}\}$	(i_1, i_2)	(j_1, j_2)	(k_1, k_2)	(l_1, l_2)
$\boldsymbol{SSA_8} = \{\ \}$	(m_1, m_2)	(n_1, n_2)	(o_1, o_2)	(p_1, p_2)

With respect to the normal form of G_5 in Table 8, Answer 1 can be further translated into the follow set of inequalities.

1. $a_2 \neq \max(a_2, b_2, c_2, d_2)$, $a_1 \neq \max(a_1, e_1, i_1, m_1)$,

2. $k_2 \neq \max(i_2, j_2, k_2, l_2)$, $k_1 \neq \max(c_1, g_1, k_1, o_1)$,
3. $p_2 \neq \max(m_2, n_2, o_2, p_2)$, $p_1 \neq \max(d_1, h_1, l_1, p_1)$,
4. $j_2 \neq \max(i_2, j_2, k_2, l_2)$, $j_1 \neq \max(b_1, f_1, j_1, n_1)$,
5. $f_2 \neq \max(e_2, f_2, g_2, h_2)$,
6. $e_1 \neq \max(a_1, e_1, i_1, m_1)$.

New utility functions $u_{1,5}$ and $u_{2,5}$ can be designed according to the inequalities above. To speak in context of global politics, it should be made sure that both groups of countries are aware of the following information.

1. When one group of countries demands that "the world should accept in peace that it is in advantageous position" without further plan of counterattack if its demand is not met immediately, the other group of countries should not make the same bold yet shallow statement in order to protect its own interest.
2. When one group of countries chooses to execute all available strategies, the other group will harm its own interest if it also executes all available strategies in return. That is, when one group of countries already announces that "the world should accept in peace that it is in advantageous position" and that "it will be unsatisfied with inferiority and will make counterattack to force the other player to hand over the advantageous position", the other group should avoid doing the same in order to protect its own interest.
3. Having conflicts started without any negotiation at all will harm the interest of both groups of countries.
4. When the democratic countries (Player 1) choose to execute all available strategies, Russia and China (Player 2) will harm their own interest if they only state that "they will be unsatisfied with inferiority and will make counterattack to

force the other player to hand over the advantageous position" directly. Also, if Russia and China make the statement above, democratic countries will suffer loss if execute all available strategies.

5. When the democratic countries only state that "they will be unsatisfied with inferiority and will make counterattack to force the other player to hand over the advantageous position" directly, Russia and China will harm their interest if they take the same strategy.
6. When Russia and China make the bold yet shallow demand that "the world should accept in peace that they are in advantageous position", the democratic countries should not state that "they will be unsatisfied with inferiority and will make counterattack to force the other player to hand over the advantageous position" directly in order to protect their own interest.

7 Conclusions

The main research problem is to offer a flexible, automatic and programmable solution for *MD* problems so that International Order Reshaping can be led into a certain desirable direction with instructions automatically generated by programmes. Solving the main research problem can break down into two main research purposes: 1. To find the better way of mapping games into *ASP*; 2. To lead International Order Reshaping towards desirable direction with a programmable Mechanism Design method. Conclusions of this research respond to the main research problem and evaluate the process of addressing the problem.

7.1 Answer to the Main Research Problem

The solution to the main research problem given in this research can be briefly described as follows. First off, use the “Battle of Sexes” model to construct games which simulate the scenario of International Order. Then, map the normal form of the “Battle of Sexes” games into Dung’s Argumentation Frameworks, considering each available strategy set as an argument (SSA) and deriving attack relations from non-best-responses in the normal-form games. For each Game-Based Argumentation Framework, employ *ASPARTIX* system to enumerate all Nash Equilibria, which are exactly the stable extensions of the Game-Based Argumentation Framework. Argumentation Mechanism Design is then conducted by automatically (in Answer Set Programming) modifying the attack relation set of the Game-Based Argumentation Framework so that the set of Nash Equilibria equals to the set of desirable game outcomes chosen by mechanism designers, with Mechanism Design solutions given in the form of re-designed attack relation set. To increase the interpretability of Argumentation Mechanism Design solutions, mechanism designer can assign compulsory attack relations and limit the complexity of re-designed attack relation set. In this way, Argumentation Mechanism Design solutions, which are returned as answers of queries in *ASPARTIX* system, should be able to reflect on global politics reality and offer pragmatic instructions on reaching the goals of International Order Reshaping.

The two main research purposes are fulfilled in answering the main research problem. Compared to Pathway 1 (Extensive-form Game → *ASP*), Pathway 2 (Normal-form Game → *AF* → *ASP*) proves to be a better way of mapping games into *ASP*, for that it is more efficient and interpretable in terms of laying the foundation of Mechanism Design. Argumentation Mechanism Design in *ASPARTIX*, as a flexible and automatic programmable method, is capable of leading International Order Reshaping towards desirable direction.

7.2 Findings and Contributions

Findings. Beside the answer to the main research problem, this research has the following findings that may make a difference in the development of game theory, argumentations and political sciences.

(1) Although containing more comprehensive information of game process, extensive form is not suitable for mapping games into Dung's Argumentation Frameworks. Game-Based Argumentation Frameworks rely on the concept of "best response" to derive the attack relation set. Such concept is straightforward only in the normal form of games.

(2) To make *GBAF* solvable in *ASPARTIX* with uniform semantics as well as to reconcile Definition IP-1 and Definition IP-2, the concept of "argument" in Game-Based Argumentation Framework should be interpretated as "choosing strategy set (a series of moves in game)" instead of "choosing independent strategy (a single move in game)"

(3) The "Prisoner's Dilemma" game model is unrealistic and powerless when it comes to Mechanism Design for International Order Reshaping. In the fields of politics and economy, the "Prisoner's Dilemma" game model has been widely used when game theory is employed to analyse potential conflicts between countries. However, the "Prisoner's Dilemma" model is only effective when the actual conflicts are not started yet. Previous researches based on the "Prisoner's Dilemma" model, in which "ideal peace" is usually the only Nash Equilibrium, often come to a rather wishful

conclusion that “players will all stay calm for the sake of their own interest”. However, once actual conflicts have been triggered, the model is voided and no longer suitable for simulating the real scenario, where “ideal peace” has already been broken. Moreover, with the “Prisoner’s Dilemma” model, Mechanism Design is not capable of forcing any desirable game outcomes where a particular player prevails, since the model is naturally promoting ideal “win-win” situation. On the contrary, the “Battle of Sexes” game model better simulates the prior-inferior relations between players, waiving unrealistic “win-win” situation and leaving room for Mechanism Design. Therefore, the “Battle of Sexes” game model is more suitable for game-theoretical analysis during conflicts.

Contributions. The following contributions are what distinguish this study and the reasons why lasting values can be expected.

(1) Through the case study of International Order Reshaping, the application of Game-Based Argumentation Frameworks is remarkably pushed forward—mapping games into Argumentation Frameworks plays an important role in laying the foundations of programmable Mechanism Design.

(2) This research fills the blank of encoding Argumentation Mechanism Design in Answer Set Programming, allowing game rule solutions to be automatically and flexibly generated in *ASPARTIX* system.

(3) Also, this study is one of the pioneering works that take global politics into the perspective of argumentations and Answer Set Programming. Hopefully, the concepts and methods introduced in this research will open up a new interdisciplinary area between artificial intelligence and political sciences.

7.3 Limitations and Future Works

The pioneering new methods promoted in this research are not yet mature and need to be comprehensively testified in more practical applications. Limitations of the methods in this research include, but are not limited to, the following:

(1) When mapping games into Game-Based Argumentation Frameworks, the derivation of attack relation set essentially follows the same logic of the search for Nash Equilibria. Such underlying logic results in attack relations between arguments representing strategy sets instead of independent strategies. The binary relations between game-based arguments corresponding to independent strategies are only interpretable to human understanding but not involved in programmable solutions. Future works on Argumentation Mechanism Design are suggested to pay more attention to finding more straightforward mapping methods between games and Argumentation Frameworks—methods in which attack relations can be built between game-based arguments representing independent strategies.

(2) Which semantics in Game-Based Argumentation Frameworks most accurately maps the concept of Nash Equilibria in game theory remains in dispute. This research references Dung's approach, using the semantics of stable extensions to enumerate Nash Equilibria. However, the semantics of maximal extensions, which is used in several other studies, can lead to the same results in all Answer Set Programming projects introduced in this study. Future works should set the standards of mapping the concept of Nash Equilibria to Argumentation Framework extension semantics.

References

[Bistarelli2020] Bistarelli, Stefano, and Francesco Santini. "Abstract argumentation and (optimal) stable marriage problems." Argument & Computation 11, no. 1-2 (2020): 15-40.

[Cheng2021] Cheng, You, Beishui Liao, and Jieting Luo. "Applying Abstract Argumentation to Normal-Form Games." In International Workshop on Explainable, Transparent Autonomous Agents and Multi-Agent Systems, pp. 314-328. Springer, Cham, 2021.

[Cressman2003] Cressman, Ross, Christopher Ansell, and Ken Binmore. Evolutionary dynamics and extensive form games. Vol. 5. MIT Press, 2003.

[Dung1995] Dung, Phan Minh. "On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games." Artificial intelligence 77, no. 2 (1995): 321-357.

[Dvořák2020] Dvořák, Wolfgang, Anna Rapberger, Johannes P. Wallner, and Stefan Woltran. "ASPARTIX-V19-an answer-set programming based system for abstract argumentation." In International Symposium on Foundations of Information and Knowledge Systems, pp. 79-89. Springer, Cham, 2020.

[Egly2008] Egly, Uwe, Sarah Alice Gaggl, and Stefan Woltran. "Aspartix: Implementing argumentation frameworks using answer-set programming." In International Conference on Logic Programming, pp. 734-738. Springer, Berlin, Heidelberg, 2008.

[Fan2014] Fan, Xiuyi, and Francesca Toni. "On computing explanations in abstract argumentation." In ECAI 2014, pp. 1005-1006. IOS Press, 2014.

[Fan2016] Fan, Xiuyi, and Francesca Toni. "On the interplay between games, argumentation and dialogues." 2016 International Conference on Autonomous Agents & Multiagent Systems, 2016.

[Gaggl2009] Gaggl, Sarah Alice. Solving argumentation frameworks using answer set programming. na, 2009.

[Gibbons1992] Gibbons, Robert S. "Game theory for applied economists." In Game Theory for Applied Economists. Princeton University Press, 1992.

[Gómez2021] Gómez, Rodrigo N., Carlos Hernández, and Jorge A. Baier. "A Compact Answer Set Programming Encoding of Multi-Agent Pathfinding." IEEE Access 9 (2021): 26886-26901.

[Hu2013] Hu, Shunfu, and Ting Dai. "Online map application development using Google Maps API, SQL database, and ASP .NET." International Journal of Information and Communication Technology Research 3, no. 3 (2013).

[Liao2021] Liao, Beishui, Michael Anderson, and Susan Leigh Anderson. "Representation, justification, and explanation in a value-driven agent: an argumentation-based approach." AI and Ethics 1, no. 1 (2021): 5-19.

[Mailath1991] Mailath, George J., Larry Samuelson, and J. Swinkels. "Extensive form reasoning in normal form games." Foundations in Microeconomic Theory (1991): 451.

[Myerson2013] Myerson, R. B. "Game theory. USA: Harvard University Press." (2013).

[Narahari2014] Narahari, Yadati. Game theory and mechanism design. Vol. 4. World Scientific, 2014.

[Nash1950] Nash Jr, John F. "Equilibrium points in n-person games." Proceedings of the national academy of sciences 36, no. 1 (1950): 48-49.

[Prakken2005] Prakken, Henry. "Coherence and flexibility in dialogue games for argumentation." Journal of logic and computation 15, no. 6 (2005): 1009-1040.

[Rahwan2008] Rahwan, Iyad, and Kate Larson. "Pareto optimality in abstract argumentation." In AAAI, pp. 150-155. AAAI Press, 2008.

[Rahwan2009] Rahwan, Iyad, and Kate Larson. "Argumentation and game theory." In Argumentation in artificial intelligence, pp. 321-339. Springer, Boston, MA, 2009.

[Riveret2008] Riveret, Régis, Antonino Rotolo, and Giovanni Sartor. "Heuristics in Argumentation: A Game-Theorical Investigation." (2008).

[Seidenfeld1995] Seidenfeld, Teddy. "When normal and extensive form decisions differ." Studies in Logic and the Foundations of Mathematics 134 (1995): 451-463.

[Shoham2008] Shoham, Yoav, and Kevin Leyton-Brown. Multiagent systems: Algorithmic, game-theoretic, and logical foundations. Cambridge University Press, 2008.

[TuWien2021]"ASPARTIX for Dung AFs", dbai.tuwien.ac.at, 2022. [Online]. Available: https://www.dbai.tuwien.ac.at/research/argumentation/aspartix/dung.html.

[UnitedNations2021] United Nations, 2021. Current Members | United Nations Security Council. [online] Un.org. Available at: <https://www.un.org/securitycouncil/content/current-members>.

[VanDamme1984] Van Damme, Eric. "A relation between perfect equilibria in extensive form games and proper equilibria in normal form games." International Journal of Game Theory 13, no. 1 (1984): 1-13.

[WorldBank2022] World Bank, 2022. Gross domestic product 2021. [online] Worldbank.org. Available at: <https://databank.worldbank.org/data/download/GDP.pdf>.

[WorldPopulationReview2022] World Population Review, 2022. Democracy Countries 2022. [online] Worldpopulationreview.com. Available at: <https://worldpopulationreview.com/country-rankings/democracy-countries>.

[Yin2018] Yin, Jason Z., and Michael H. Hamilton. "The conundrum of US-China trade relations through game theory modelling." Journal of Applied Business and Economics 20, no. 8 (2018): 133-150.

[Young2019] Young, Anthony P., David Kohan Marzagao, and Josh Murphy. "Applying abstract argumentation theory to cooperative game theory." arXiv preprint arXiv:1905.10922 (2019).

Appendices

Appendix 1. Encoding for *AF* Semantics in Clingo-Based *ASPARTIX*

Naïve Extension (‘naive.dl’):

```
%% Guess a set S \subseteq A
in(X) :- not out(X), arg(X).
out(X) :- not in(X), arg(X).

%% S has to be conflict-free
:- in(X), in(Y), att(X,Y).

%% Check Maximality
okOut(X) :- in(Y), att(Y,X).
okOut(X) :- in(Y), att(X,Y).
okOut(X) :- att(X,X).
:-  out(X), not okOut(X).
```

Complete Extension (‘comp.dl’):

```
%% Guess a set S \subseteq A
in(X) :- not out(X), arg(X).
out(X) :- not in(X), arg(X).

%% S has to be conflict-free
:- in(X), in(Y), att(X,Y).

%% The argument x is defeated by the set S
defeated(X) :- in(Y), att(Y,X).

%% The argument x is not defended by S
not_defended(X) :- att(Y,X), not defeated(Y).

%% admissible
:- in(X), not_defended(X).

%% Every argument which is defended by S belongs to S
:- out(X), not not_defended(X).
```

Grounded Extension (‘ground.dl’):

```
% An argument is in the grounded extension if all attackers
are labelled out
in(X):-arg(X), defeated(Y):att(Y,X).

% An argument is labelled out if one of its attackers is
labelled in
defeated(X):-arg(X), in(Y), att(Y,X).
```

Stable Extension (stable.dl):

```
%% Guess a set S \subseteq A
in(X) :- not out(X), arg(X).
out(X) :- not in(X), arg(X).

%% S has to be conflict-free
:- in(X), in(Y), att(X,Y).

%% The argument x is defeated by the set S
defeated(X) :- in(Y), att(Y,X).

%% S defeats all arguments which do not belong to S
:- out(X), not defeated(X).
```

Nash Equilibria (filter.lp):

```
  nash(SS1,SS2) :- SS1>SS2, in(SS1), in(SS2),
strategy_set_of(SS1,P1), strategy_set_of(SS2,P2), P1!=P2.

  #show nash/2.
```

Appendix 2. *ASP* Code Mapping Chicken-Game-Modified "Prisoner's Dilemma" Main Games via Pathway 1

```
%-----------%
%- The Map -%
%-----------%
strategy(e1).
strategy(e2).
strategy(e3).
strategy(e4).
strategy(e5).
strategy(e6).
strategy(e7).
strategy(e8).
strategy(e9).
strategy(e10).
strategy(e11).
strategy(e12).

node(n1).
node(n2).
node(n3).
node(n4).
node(n5).
node(n6).

node(t1).
node(t2).
node(t3).
node(t4).
node(t5).
node(t6).
node(t7).

proceed(n1, n2, e1).
proceed(n1, n3, e2).
proceed(n2, t1, e3).
proceed(n2, t2, e4).
proceed(n3, t3, e5).
proceed(n3, n4, e6).
proceed(n4, n5, e7).
proceed(n4, n6, e8).
proceed(n5, t4, e9).
proceed(n5, t5, e10).
proceed(n6, t6, e11).
proceed(n6, t7, e12).
-proceed(X,Y,E) :- node(X), node(Y), strategy(E), not
proceed(X,Y,E).
```

```
%---------------------------------------%
%- Class Structure and Categorisation. -%
%---------------------------------------%
class(node).
class(gamestatus).
class(outcome).
class(nash_equilibrium).
class(quasi_nash_equilibrium).

subclass(C1,C2) :- is_subclass(C1,C2).
subclass(C1,C2) :- is_subclass(C1,C3), subclass(C3,C2).
-subclass(C1,C2) :- class(C1), class(C2), not subclass(C1,C2).

member(X,C) :- in(X,C).
member(X,C) :- in(X,C0), subclass(C0,C).
siblings(C1,C2) :- is_subclass(C1,C), is_subclass(C2,C), C1 !=
C2.
-member(X,C2) :- member(X,C1), siblings(C1,C2).

is_subclass(gamestatus, node).
is_subclass(outcome, node).
is_subclass(nash_equilibrium, outcome).
is_subclass(quasi_nash_equilibrium, outcome).

in(n1, gamestatus).
in(n2, gamestatus).
in(n3, gamestatus).
in(n4, gamestatus).
in(n5, gamestatus).
in(n6, gamestatus).

in(t1, nash_equilibrium).
in(t2, outcome).
in(t3, outcome).
in(t4, outcome).
in(t5, quasi_nash_equilibrium).
in(t6, quasi_nash_equilibrium).
in(t7, outcome).


%----------------------%
%- Reachable Outcomes -%
%----------------------%
route(X,Y) :- proceed(X,Y,M), not restricted(M).
route(X,Y) :- proceed(X,Z,M), not restricted(M), route(Z,Y).
-route(X,Y) :- node(X), node(Y), not route(X,Y).

reachable(T) :- member(T,outcome), route(n1,T).
```

```
-reachable(T) :- member(T,outcome), not route(n1,T).

reachable_Equilibrium(Q) :- in(Q, nash_equilibrium),
reachable(Q).
-reachable_Equilibrium(Q) :- in(Q, nash_equilibrium), not
reachable(Q).


reachable_quasi_Equilibrium(Q) :- in(Q,
quasi_nash_equilibrium), reachable(Q).
-reachable_quasi_Equilibrium(Q) :- in(Q,
quasi_nash_equilibrium), not reachable(Q).


%==========================%
%= Strategy Restrictions =%
%==========================%
restricted(e7).


%====================%
%= Show Reachables =%
%====================%
#show reachable/1.
#show -reachable/1.

#show reachable_Equilibrium/1.
#show -reachable_Equilibrium/1.

#show reachable_quasi_Equilibrium/1.
#show -reachable_quasi_Equilibrium/1.
```

Appendix 3. *ASP* Code Mapping Chicken-Game-Modified "Prisoner's Dilemma" Main Games via Pathway 2

```
arg(a1).
arg(a3).
arg(a2a5).
arg(a4a7).
arg(a2a6).
arg(a4a8).

att(a1,a4a7).
att(a1,a4a8).
att(a2a5,a4a7).
att(a2a5,a4a8).
att(a2a6,a4a7).
att(a2a6,a4a8).

att(a3,a2a5).
att(a3,a2a6).
att(a4a7,a2a5).
att(a4a7,a2a6).
att(a4a8,a2a5).
att(a4a8,a2a6).


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Encoding for stable extensions
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Guess a set S \subseteq A
in(X) :- not out(X), arg(X).
out(X) :- not in(X), arg(X).

%% S has to be conflict-free
:- in(X), in(Y), att(X,Y).

%% The argument x is defeated by the set S
defeated(X) :- in(Y), att(Y,X).

%% S defeats all arguments which do not belong to S
:- out(X), not defeated(X).


%% Rule out "one player, more than one strategy sets"
strategy_of(a1,p2).
strategy_of(a3,p1).
strategy_of(a2a5,p2).
strategy_of(a4a7,p1).
```

```
strategy_of(a2a6,p2).
strategy_of(a4a8,p1).
nash(SS1,SS2)     :-     SS1>SS2,     in(SS1),     in(SS2),
strategy_of(SS1,P1), strategy_of(SS2,P2), P1!=P2.

#show nash/2.
```

Appendix 4. *ASP* Code Mapping "Battle of Sexes" Main Games via Pathway 2

```
arg(ssa1).
arg(ssa2).
arg(ssa3).
arg(ssa4).
arg(ssa5).
arg(ssa6).
arg(ssa7).
arg(ssa8).

strategy_of(ssa1,p2).
strategy_of(ssa2,p1).
strategy_of(ssa3,p2).
strategy_of(ssa4,p1).
strategy_of(ssa5,p2).
strategy_of(ssa6,p1).
strategy_of(ssa7,p2).
strategy_of(ssa8,p1).

att(ssa1,ssa2).
att(ssa1,ssa4).
att(ssa1,ssa8).
att(ssa3,ssa2).
att(ssa3,ssa4).
att(ssa3,ssa6).
att(ssa3,ssa8).
att(ssa5,ssa4).
att(ssa5,ssa6).
att(ssa7,ssa2).
att(ssa7,ssa4).
att(ssa7,ssa8).

att(ssa2,ssa1).
att(ssa2,ssa3).
att(ssa2,ssa7).
att(ssa4,ssa1).
att(ssa4,ssa3).
att(ssa4,ssa5).
```

```
att(ssa4,ssa7).
att(ssa6,ssa3).
att(ssa6,ssa5).
att(ssa8,ssa1).
att(ssa8,ssa3).
att(ssa8,ssa7).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Encoding for stable extensions
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Guess a set S \subseteq A
in(X) :- not out(X), arg(X).
out(X) :- not in(X), arg(X).

%% S has to be conflict-free
:- in(X), in(Y), att(X,Y).

%% The argument x is defeated by the set S
defeated(X) :- in(Y), att(Y,X).

%% S defeats all arguments which do not belong to S
:- out(X), not defeated(X).


%% Rule out "one player, more than one strategy sets"
nash(SS1,SS2)   :-   strategy_of(SS1,p1),   strategy_of(SS2,p2),
in(SS1), in(SS2).

#show nash/2.
```

Appendix 5. *ASP* Code for *ArgMD* Attack Relation Modification

("Battle of Sexes" Main Games in International Order Reshaping)

```
arg(ssa1).
arg(ssa2).
arg(ssa3).
arg(ssa4).
arg(ssa5).
arg(ssa6).
arg(ssa7).
arg(ssa8).

strategy_set_of(ssa1,p2).
strategy_set_of(ssa2,p1).
strategy_set_of(ssa3,p2).
strategy_set_of(ssa4,p1).
strategy_set_of(ssa5,p2).
strategy_set_of(ssa6,p1).
strategy_set_of(ssa7,p2).
strategy_set_of(ssa8,p1).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Encoding for stable extensions
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Guess a set S \subseteq A
in(X) :- not out(X), arg(X).
out(X) :- not in(X), arg(X).

%% S has to be conflict-free
:- in(X), in(Y), att_in(X,Y).

%% The argument x is defeated by the set S
defeated(X) :- in(Y), att_in(Y,X).

%% S defeats all arguments which do not belong to S
:- out(X), not defeated(X).
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Nash Equilibria
nash(X,Y) :- strategy_set_of(X,p1), strategy_set_of(Y,p2),
in(X), in(Y).
-nash(X,Y) :- strategy_set_of(X,p1), strategy_set_of(Y,p2),
not nash(X,Y).


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% Mechanism Design
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Guess a set of attack relations
may_att(X,Y) :- strategy_set_of(X,P1), strategy_set_of(Y,P2),
P1!=P2.
att_in(X,Y) :- not att_out(X,Y), may_att(X,Y).
att_out(X,Y) :- not att_in(X,Y), may_att(X,Y).

%% Rule out "a SP of Player 1 attacks no SP of Player 2"
:- att_out(SS1,ssa1), att_out(SS1,ssa3), att_out(SS1,ssa5),
att_out(SS1,ssa7).
%% Rule out "a SP of Player 2 attacks no SP of Player 1"
:- att_out(SS2,ssa2), att_out(SS2,ssa4), att_out(SS2,ssa6),
att_out(SS2,ssa8).

%% Desirable game outcomes
desirable_outcome(ssa6,ssa1).
desirable_outcome(ssa6,ssa7).
-desirable_outcome(SS1,SS2) :- strategy_set_of(SS1,p1),
strategy_set_of(SS2,p2), not desirable_outcome(SS1,SS2).

%% Each Nash Equilibrium is a desirable game outcome, vice
versa
:- nash(SS1,SS2), not desirable_outcome(SS1,SS2).
:- desirable_outcome(SS1,SS2), not nash(SS1,SS2).
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

#show att_in/2.
```

Appendix 6. *ASP* Code for *ArgMD* Attack Relation Modification with Consideration of Conciseness of Solutions and Compulsory Attack Relations

```
arg(ssa1).
arg(ssa2).
arg(ssa3).
arg(ssa4).
arg(ssa5).
arg(ssa6).
arg(ssa7).
arg(ssa8).

strategy_set_of(ssa1,p2).
strategy_set_of(ssa2,p1).
strategy_set_of(ssa3,p2).
strategy_set_of(ssa4,p1).
strategy_set_of(ssa5,p2).
strategy_set_of(ssa6,p1).
strategy_set_of(ssa7,p2).
strategy_set_of(ssa8,p1).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Encoding for stable extensions
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Guess a set S \subseteq A
in(X) :- not out(X), arg(X).
out(X) :- not in(X), arg(X).

%% S has to be conflict-free
:- in(X), in(Y), att_in(X,Y).

%% The argument x is defeated by the set S
defeated(X) :- in(Y), att_in(Y,X).

%% S defeats all arguments which do not belong to S
:- out(X), not defeated(X).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Nash Equilibria
nash(X,Y) :- strategy_set_of(X,p1), strategy_set_of(Y,p2),
in(X), in(Y).
-nash(X,Y) :- strategy_set_of(X,p1), strategy_set_of(Y,p2),
not nash(X,Y).
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Mechanism Design
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%% Guess a set of attack relations
may_att(X,Y) :- strategy_set_of(X,P1), strategy_set_of(Y,P2),
P1!=P2.
att_in(X,Y) :- not att_out(X,Y), may_att(X,Y).
att_out(X,Y) :- not att_in(X,Y), may_att(X,Y).

%% Compulsory attack relations
att_in(ssa7,ssa8).
att_in(ssa8,ssa7).
att_in(ssa5,ssa6).
att_in(ssa6,ssa5).

%% Rule out "a SP of Player 1 attacks no SP of Player 2"
:- att_out(SS1,ssa1), att_out(SS1,ssa3), att_out(SS1,ssa5),
att_out(SS1,ssa7).
%% Rule out "a SP of Player 2 attacks no SP of Player 1"
:- att_out(SS2,ssa2), att_out(SS2,ssa4), att_out(SS2,ssa6),
att_out(SS2,ssa8).

%% Desirable game outcomes
desirable_outcome(ssa6,ssa1).
desirable_outcome(ssa6,ssa7).
-desirable_outcome(SS1,SS2) :- strategy_set_of(SS1,p1),
strategy_set_of(SS2,p2), not desirable_outcome(SS1,SS2).

%% Each Nash Equilibrium is a desirable game outcome, vice
versa
:- nash(SS1,SS2), not desirable_outcome(SS1,SS2).
:- desirable_outcome(SS1,SS2), not nash(SS1,SS2).

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Select concise ArgMD solutions
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

#const max_num_attack_relations=10.
:- #count{X,Y:att_in(X,Y)}>max_num_attack_relations.

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

#show att_in/2.
```