Game Forms for Coalition Effectivity Functions

Colm Baston and Venanzio Capretta

Functional Programming Lab
School of Computer Science
University of Nottingham

TYPES, Oslo, 14/06/2019

Coalition Logic

A modal logic for reasoning about what groups of agents can achieve through collective action[†]:

- Assumes a finite set of agents N.
- Defines a coalition to be any $C \in \mathcal{P}(N)$.
- Introduces a modal operator $[C]\psi$ for each coalition with the interpretation "C can bring about ψ".
- Considers how agents' actions can interact rather than considering them in isolation.

[†]Marc Pauly, *"A modal logic for coalitional power in games"*, Journal of Logic and Computation, Vol. 12, 2002

Game Forms

Coalition logic has a semantics based on perfect-information strategic games where the players act simultaneously. A *game form* G is a tuple $\langle N, S, \{\mathcal{A}_i\}_{i \in N}, o \rangle$ where:

- N is a finite set of agents.
- S is a non-empty set of outcome states.
- $\{\mathcal{A}_i\}_{i \in N}$ is a family of sets of possible actions, one for each agent.
- o is a function $(\Pi_{i \in N}\ \mathcal{A}_i) \rightarrow S$ which associates each selection of actions with an outcome.

Effectivity Functions

Each game form admits an *effectivity function* $\mathcal{P}(N) \to \mathcal{P}(\mathcal{P}(S))$ that classifies which sets of outcomes a coalition can bring about:

- Let σ_C be a strategy profile for a coalition C; that is, a selection of actions $\Pi_{i \in C}\ \mathcal{A}_i$ for only those agents in C.
- Let $\sigma_C \oplus \sigma_{\overline{C}}$ denote the joining of two complementary strategy profiles into a global strategy profile for all agents.
- Define the effectivity function for a game form G:

$$E_G(C) = \{X \in \mathcal{P}(S) \mid \exists \sigma_C, \forall \sigma_{\overline{C}}, o(\sigma_C \oplus \sigma_{\overline{C}}) \in X\}$$

- If $X \in E_G(C)$, then the coalition C can force the outcome to be in X. We say "C is effective for X".

Playability

A derived effectivity function E_G is *playable*. An effectivity function E is called playable iff:

- $\varnothing \notin E(C)$
- $S \in E(C)$
- $\overline{X} \notin E(\varnothing) \Rightarrow X \in E(N)$
- $X \in E(C) \Rightarrow \overline{X} \notin E(\overline{C})$
- $X_1 \subseteq X_2 \Rightarrow X_1 \in E(C) \Rightarrow X_2 \in E(C)$
- $C_1 \subseteq C_2 \Rightarrow X \in E(C_1) \Rightarrow X \in E(C_2)$
- $C_1 \cap C_2 = \varnothing \Rightarrow X_1 \in E(C_1) \Rightarrow X_2 \in E(C_2) \Rightarrow X_1 \cap X_2 \in E(C_1 \cup C_2)$

Equivalence

Furthermore, Pauly proved that every playable effectivity function is the derived effectivity function of some game form:

- From some playable effectivity function E, construct a game form G, then show that $E = E_G$.
- We demonstrate both directions of this result type-theoretically, with the aim of formalisation in the proof assistants Coq and Agda.

Formalisation

In practice:

- $N = \{0,\ 1,\ \cdots,\ n-1\}$
- $S : Type$
- Game forms:

$$\sum_{\mathcal{A}:N \to Type} \left(\prod_{i:N} \mathcal{A}_i \right) \to S$$

- We require our subset type to have have decidable membership:

$$\mathcal{P}_{\mathsf{dec}}(A) = A \to \mathbb{2}$$

- We develop a library for working with them:

$$\varnothing = const\ \ false \qquad full = const\ true$$

$$X \cup Y = \cdots \qquad X \cap Y = \cdots \qquad \overline{X} = \cdots$$

Formalisation

- Coalition strategy profiles are represented by partial functions:

$$\sigma_C : \prod_{i:N} Maybe\ \mathcal{A}_i$$

$$\sigma_{C_1} \oplus \sigma_{C_2} = \cdots$$

- We can form the global strategy profile if we first show that the represented coalition $C = N$.
- From here, we can define $E_G : \mathcal{P}_{\mathsf{dec}}(N) \to \mathcal{P}_{\mathsf{dec}}(\mathcal{P}_{\mathsf{dec}}(S))$ and prove that it is playable:

$$playable\ E = \cdots \times \cdots \times \cdots \times \cdots \times \cdots$$

Game Construction

For the other direction, take a playable effectivity function E and construct a game form G:

- Define the family of action sets:

$$\mathcal{A}_i = \{\langle C, X, x, t\rangle \mid C \in \mathcal{P}_{\mathsf{dec}}(N),\ i \in C,\ X \in E(C),\ x \in X,\ t \in \mathbb{N}\}$$

Each agent chooses:

- C: a coalition they would like to be part of.
- X: a set of outcomes they would like that coalition to work towards.
- x: a specific outcome state in X that would be their preference.
- t: a natural number which will be used to decide which agent gets to choose their preference as the final outcome.

Game Construction

Given a fixed strategy profile σ we have $\langle C_i, X_i, x_i, t_i \rangle$ for each agent i.

- Call a coalition C σ-cooperative iff all $i \in C$ chose $C_i = C$ and also chose the same $X_i = X$.
- Let $\langle C_1, \ldots, C_m \rangle$ be all of the non-empty cooperative coalitions and let C_0 be all agents not in a cooperative coalition. $\langle C_0, \ldots, C_m \rangle$ is a partition of N.
- Let X_C denote the states selected by cooperative coalition C and define:

$$O(\sigma) = \bigcap_{k=1}^{m} X_{C_k}$$

Game Construction

- We will select an outcome state from $O(\sigma)$ by selecting an agent using the t_i values.
- Let d be the the agent $\sum_{i \in N} \ t_i \pmod{|N|}$.
- In general, d's choice of state x_i is not guaranteed to be in $O(\sigma)$. If it isn't, we revert to an arbitrary choice function $H : \Pi_{X \in E(N)} \ X$:

$$o(\sigma) = \begin{cases} x_d & \text{if } x_d \in O(\sigma) \\ H(O(\sigma)) & \text{otherwise} \end{cases}$$

- H exists constructively due to the playability property that ensures $\varnothing \notin E(C)$.

Equivalence Proof

From an effectivity function E, we have constructed a game form G. We must now show that $E = E_G$. From left-to-right:

- We assume $X \in E(C)$ and must show $X \in E_G(C)$.
- Expanding the definition: $\exists \sigma_C,\ \forall \sigma_{\overline{C}},\ o(\sigma_C \oplus \sigma_{\overline{C}}) \in X$.
- Choose σ_C such that $C_i = C$ and $X_i = X$ for all $i \in C$. The choices of x_i and t_i may be arbitrary.
- C is therefore a σ-cooperative coalition for any $\sigma_{\overline{C}}$, so it will be one of the classes in the partition used to define $O(\sigma)$.
- As $X_C = X$, and $O(\sigma)$ is defined by intersection, $O(\sigma) \subseteq X$, and therefore $o(\sigma) \in X$ as desired.

The right-to-left proof is omitted for this talk.

Summary

- Type-theoretic formalisation of the semantics of Pauly's coalition logic.
- Our game form construction uses a similar idea to Pauly's, but is simpler as well as being adapted for use in a constructive setting.
- Coq and Agda formalisations are work-in-progress.

Thank you!