Evaluation Protocol for Assessing Reasoning Correctness in Game-Theoretic Dilemmas

1 Preliminaries

This section presents payoff matrices used in the experiment, the log file format, and the type of errors that we are looking for. Please familiarise yourself with the content of this section before conducting the evaluation.

1.1 Payoff matrices for considered games

P1/P2	**R (Betray)**	**B (Confess)**
R (Betray)	$(1, 1)$	$(5, 0)$
B (Confess)	$(0, 5)$	$(3, 3)$

Table 1: The payoff matrix for the Prisoner's Dilemma (PD).

P1/P2	**R (Hare)**	**B (Stag)**
R (Hare)	$(1, 1)$	$(3, 0)$
B (Stag)	$(0, 3)$	$(5, 5)$

Table 2: The payoff matrix for the Stag Hunt (SH).

P1/P2	**R (Hawk)**	**B (Dove)**
R (Hawk)	$(0, 0)$	$(5, 1)$
B (Dove)	$(1, 5)$	$(3, 3)$

Table 3: The payoff matrix for the Hawk-Dove (HD).

Tables 1-3 present the payoffs for Prisoner's Dilemma, Stag Hunt, and Hawk-Dove, respectively. Please note that the matrices are inverse compared to the typical presentation of payoff matrices for these games.

1.2 Log file format

The log file (a reasoning sample) structure is as follows:

1. Attempt: each file contains between 1 and 5 attempts. The beginning of each attempt is marked by "###ATTEMPT##[attempt number]~". Attempts are numbered from 0 to 4. Each attempt contains the following sections:

(a) **RESPONSE##**: This is the LLMs' output that you need to evaluate.

(b) **PREDICATES##**: This is the list of predicates corresponding to the response, not relevant for this evaluation.

(c) **FAILED QUERIES##**: List of failed predicates, not relevant for this evaluation.

(d) **CORRECTING PROMPT##**: Correcting prompt based on failed predicates, not relevant for this evaluation.

Your goal is to evaluate the **RESPONSE##** section of the first and last attempt.

1.3 Error types

The reasoning sample should be marked as "incorrect" if it contains any error related to reasoning about payoffs. Such errors can be classified as follows:

- **Payoff assignment error (pa)**. Wrong assignment of payoffs for actions, e.g. "Me choosing 'R' and them choosing 'B' will result in payoff $5 for me."
- **Action risk/gain assessment error (rga)**. Incorrectly identifying an action as guaranteeing better worse-case/best-case scenario outcome, e.g. "Action 'B' is safer, because it maximizes my minimal gain."
- **Payoff comparison error (pc)**. Incorrect comparison of numerical values, e.g. "Choosing 'B' when they choose 'R' gives me $0 which is better than $1."

If you encounter an error related to reasoning about payoffs that do not belong to any of the above categories, mark it as **Unclassified (u)** type of error. Only the errors related to reasoning about payoffs should lead to classifying a sample as "incorrect". Ignore other types of errors, including:

- The incorrect use of game theory concepts, such as dominant strategy or Nash equilibrium,
- Other statements incorrect from the game theory point of view, e.g. "The rational choice is to select an action that maximizes mutual payoff."
- Wrong identification of a game, e.g. recognizing Stag Hunt as Prisoner's Dilemma.

2 CSV Evaluation Form

The CSV file you will fill has the following columns:

- **file:** The name of the log file (reasoning sample, filled in automatically).
- **game:** The type of game played ('sh', 'pd', 'hd', filled in automatically).
- **attempts:** The number of reasoning attempts (filled in automatically).
- **orig_choice:** The original choice made by the reasoner ({R}, {B}, filled in automatically).

- **final_choice:** The final choice made by the reasoner ({R}, {B}, filled in automatically).
- **orig_cor:** Indicates whether the original reasoning was correct (binary: 0 or 1, filled in by you).
- **fin_cor:** Indicates whether the final reasoning was correct (binary: 0 or 1, filled in by you).
- **err_type:** The type of errors encountered in reasoning (‘pa’: payoff assignment error, ‘pc’: payoff comparison error, ‘rga’: action risk/gain assessment error, ‘u’: unclassified), separated by “#” (filled in by you).
- **sentence:** Sentence(s) containing faults in reasoning, separated by “#” (filled in by you).
- **con_mat:** The confusion matrix field (’TP’, ’FN’, ’FP’, ’TN’, filled in automatically after evaluation).
- **remarks:** Additional remarks or comments.
- **failed_queries:** The failed Prolog queries (filled in automatically).

3 Steps of Evaluation

1. Open the log file (reasoning sample) provided in the **file** column.
2. Read the **RESPONSE##** content in “###ATTEMPT##0∼” section.
   (a) If the reasoning is correct fill in **orig_cor** column with ‘1’.
   (b) If the reasoning is incorrect fill in **orig_cor** column with ‘0’, paste the sentences containing faulty reasoning separated by ‘#’ into the column **sentence** and fill in the **err_type** column (‘pa’, ‘pc’, ‘rga’, or ‘u’, see Section 1.3). If there is more then one type of error, use ‘#’ as a separator.
3. If there is more than one reasoning attempt, assess the correctness of the final attempt and fill in **final_cor** with ‘1’ for correct reasoning and ‘0’ for incorrect reasoning. If there is only one reasoning attempt, fill in **final_cor** with the same value as in **orig_cor**.
4. If you have any remarks, fill in **remarks** column (optional).

Below, see an example of the filled row of the CSV evaluation form.

file	...	orig_cor	fin_cor	err_type	sentence	...
log_1.txt	...	0	1	pa#rga	"First."#"Second."	...

Table 4: Sample row of the evaluation form.