COMPSYS 726 Expert Systems – Pokémon Showdown

Woojin Jeon (wjeo779)

**Introduction to Pokemon Showdown and Expert System**

In general an expert system has 5 following components

In general view, an expert system includes the following components: a [knowledge base](https://en.wikipedia.org/wiki/Knowledge_base), an [inference engine](https://en.wikipedia.org/wiki/Inference_engine), an explanation facility, a knowledge acquisition facility, and a user interface.[[48]](https://en.wikipedia.org/wiki/Expert_system" \l "cite_note-:0-48)[[49]](https://en.wikipedia.org/wiki/Expert_system" \l "cite_note-49)

Pokemon -> Shortened to PK from this point

**State Representation**

The provided code presents us with various methods to collect data allowing us to track the full state of all different areas of the current ongoing battle such as Health Points, Weather and Pokémon types. However, our inference engine will not require all the data available, hence, we will only extract data that our engine can utilize henceforth known as the “battlestate”.

Reduces Truth Maintenance overhead (as we now have a single source of truth and some others) and keeps rule evaluation deterministic.

To track “battlestate” a new BattleState dataclass that will extract the relevant details. The table beneath outlines the states that will be stored in the BattleState and a short reasoning to its inclusion.

|  |  |  |  |
| --- | --- | --- | --- |
| **State** | **Reasoning** | **State** | **Reasoning** |
| turn : int | Total turns taken | Speed advantage: bool | Who will move first |
| My PK Species : str | My pokemon that is active | Opponent (Opp) PK Species :str | Opponent pokemon that is active |
| My PK Health : int | My PK remaining health (fraction) | Opp PK Health : int | Opponents PK remaining health (frac) |
| My PK Type : List[str] | My PK type(s) | Opp PK Type :List[str] | Opponents PK type(s) |
| My PK status : str | My PK status (if any) | Opp PK status : str | Opp PK status (if any) |
| My PK boosts: Dict[str, int] | Accuracy, atk, def, evasion, spa, spd, spe | Opp PK boosts: Dict[str,int] | Accuracy, atk, def, evasion, spa, spd, spe |
| My PK moves : List[str] | List of all the moves my pokemon can do | Opp PK moves: List[str] | List of revealed moves the opp has used |
| My PK item : str | My PK item (if any) | Opp PK item : str | Opp PK item (if any) |
| My PK tera type : str | My PK tera (if any) | Opp PK tera type: str | My PK tera (if any) |
| My remaining PK : int | How many pokemon in my team are alive | Opp remaining PK : int | How many pokemon in opp team are alive |
| Weather : str | Current Weather | Terrain: str | Current Terrain |
| Hazards on my side : Dict[str, int] | Hazards on my side | Hazards on opp side: Dict[str,int] | Hazards on opps side |
| Screens on my side: Dict[str, int] | Screens on my side | Screens on opp side: Dict[str, int] | Screens on opps side |

The created dataclass is frozen to ensure that the dataclass will be immutable meaning that once the object has been created it is read only. This means that every turn we will create a new BattleState instance, allowing easier truth maintenance and allows us to easily track previous BattleStates.

**Derived Helper Functions**

By utilizing the BattleState data we can derive multiple tactical facts. These facts compute and capture known obvious domain knowledge allowing our engine to react to the changing battlefield. The table below defines the functions and their usage.

|  |  |  |  |
| --- | --- | --- | --- |
| **Function name** | **Input** | **Output** | **Description** |
| Estimate\_damage() | Move, My Pokemon, Opponent Pokemon | Float | Calculate how much damage this move would inflict on the opponent’s Pokemon |
| Best\_move() | My Pokemon, Opponent Pokemon, legal\_moves | Move | Returns the best move (highest expected damage) to use against the opponent’s Pokemon. |
| Can\_KO\_Opponent() | Move, |  |  |
|  |  |  |  |
|  |  |  |  |

**Forward Chaining Design**

**Forward Chaining Results**

**Backward Chaining Design**

**Backward Chaining Results**

**Results and Comparison**

There exists a log that prints the turn and other information

We can use it to see winner, how many checks are done, who is quicker? (actual game has a time limit)

**Potential Hybrid Prototype (Depending on findings)**

Forward was good at x, Backwards bad at x. Then we can use forward and backward???

**Reflection**

Why one mode is better for our specific pokemon showdown experiement?