```ts
// This file is used to define the Type Effectiveness of a move against a Pokémon.
// It calculates the effectiveness of a move based on the attacking type and the
defending Pokémon's types.

// Pokemon types as union type and value object
type PokemonType =
  | "normal"
  | "fire"
  | "water"
  | "electric"
  | "grass"
  | "ice"
  | "fighting"
  | "poison"
  | "ground"
  | "flying"
  | "psychic"
  | "bug"
  | "rock"
  | "ghost"
  | "dragon"
  | "dark"
  | "steel"
  | "fairy";

const PokemonType = {
  NORMAL: "normal",
  FIRE: "fire",
  WATER: "water",
  ELECTRIC: "electric",
  GRASS: "grass",
  ICE: "ice",
  FIGHTING: "fighting",
  POISON: "poison",
  GROUND: "ground",
  FLYING: "flying",
  PSYCHIC: "psychic",
  BUG: "bug",
  ROCK: "rock",
  GHOST: "ghost",
  DRAGON: "dragon",
  DARK: "dark",
  STEEL: "steel",
  FAIRY: "fairy",
} as const;

// Type effectiveness chart - only storing non-neutral matchups for efficiency
const TYPE_CHART: Record<PokemonType, Partial<Record<PokemonType, number>>> = {
  [PokemonType.NORMAL]: {
    [PokemonType.ROCK]: 0.5,
    [PokemonType.GHOST]: 0,
    [PokemonType.STEEL]: 0.5,
```

```
  },
  [PokemonType.FIRE]: {
    [PokemonType.FIRE]: 0.5,
    [PokemonType.WATER]: 0.5,
    [PokemonType.GRASS]: 2,
    [PokemonType.ICE]: 2,
    [PokemonType.BUG]: 2,
    [PokemonType.ROCK]: 0.5,
    [PokemonType.DRAGON]: 0.5,
    [PokemonType.STEEL]: 2,
  },
  [PokemonType.WATER]: {
    [PokemonType.FIRE]: 2,
    [PokemonType.WATER]: 0.5,
    [PokemonType.GRASS]: 0.5,
    [PokemonType.GROUND]: 2,
    [PokemonType.ROCK]: 2,
    [PokemonType.DRAGON]: 0.5,
  },
  [PokemonType.ELECTRIC]: {
    [PokemonType.WATER]: 2,
    [PokemonType.ELECTRIC]: 0.5,
    [PokemonType.GRASS]: 0.5,
    [PokemonType.GROUND]: 0,
    [PokemonType.FLYING]: 2,
    [PokemonType.DRAGON]: 0.5,
  },
  [PokemonType.GRASS]: {
    [PokemonType.FIRE]: 0.5,
    [PokemonType.WATER]: 2,
    [PokemonType.GRASS]: 0.5,
    [PokemonType.POISON]: 0.5,
    [PokemonType.GROUND]: 2,
    [PokemonType.FLYING]: 0.5,
    [PokemonType.BUG]: 0.5,
    [PokemonType.ROCK]: 2,
    [PokemonType.DRAGON]: 0.5,
    [PokemonType.STEEL]: 0.5,
  },
  [PokemonType.ICE]: {
    [PokemonType.FIRE]: 0.5,
    [PokemonType.WATER]: 0.5,
    [PokemonType.GRASS]: 2,
    [PokemonType.ICE]: 0.5,
    [PokemonType.GROUND]: 2,
    [PokemonType.FLYING]: 2,
    [PokemonType.DRAGON]: 2,
    [PokemonType.STEEL]: 0.5,
  },
  [PokemonType.FIGHTING]: {
    [PokemonType.NORMAL]: 2,
    [PokemonType.ICE]: 2,
    [PokemonType.POISON]: 0.5,
    [PokemonType.FLYING]: 0.5,
```

```
    [PokemonType.PSYCHIC]: 0.5,
    [PokemonType.BUG]: 0.5,
    [PokemonType.ROCK]: 2,
    [PokemonType.GHOST]: 0,
    [PokemonType.DARK]: 2,
    [PokemonType.STEEL]: 2,
    [PokemonType.FAIRY]: 0.5,
  },
  [PokemonType.POISON]: {
    [PokemonType.GRASS]: 2,
    [PokemonType.POISON]: 0.5,
    [PokemonType.GROUND]: 0.5,
    [PokemonType.ROCK]: 0.5,
    [PokemonType.GHOST]: 0.5,
    [PokemonType.STEEL]: 0,
    [PokemonType.FAIRY]: 2,
  },
  [PokemonType.GROUND]: {
    [PokemonType.FIRE]: 2,
    [PokemonType.ELECTRIC]: 2,
    [PokemonType.GRASS]: 0.5,
    [PokemonType.POISON]: 2,
    [PokemonType.FLYING]: 0,
    [PokemonType.BUG]: 0.5,
    [PokemonType.ROCK]: 2,
    [PokemonType.STEEL]: 2,
  },
  [PokemonType.FLYING]: {
    [PokemonType.ELECTRIC]: 0.5,
    [PokemonType.GRASS]: 2,
    [PokemonType.ICE]: 0.5,
    [PokemonType.FIGHTING]: 2,
    [PokemonType.BUG]: 2,
    [PokemonType.ROCK]: 0.5,
    [PokemonType.STEEL]: 0.5,
  },
  [PokemonType.PSYCHIC]: {
    [PokemonType.FIGHTING]: 2,
    [PokemonType.POISON]: 2,
    [PokemonType.PSYCHIC]: 0.5,
    [PokemonType.DARK]: 0,
    [PokemonType.STEEL]: 0.5,
  },
  [PokemonType.BUG]: {
    [PokemonType.FIRE]: 0.5,
    [PokemonType.GRASS]: 2,
    [PokemonType.FIGHTING]: 0.5,
    [PokemonType.POISON]: 0.5,
    [PokemonType.FLYING]: 0.5,
    [PokemonType.PSYCHIC]: 2,
    [PokemonType.GHOST]: 0.5,
    [PokemonType.DARK]: 2,
    [PokemonType.STEEL]: 0.5,
    [PokemonType.FAIRY]: 0.5,
```

```
  },
  [PokemonType.ROCK]: {
    [PokemonType.FIRE]: 2,
    [PokemonType.ICE]: 2,
    [PokemonType.FIGHTING]: 0.5,
    [PokemonType.GROUND]: 0.5,
    [PokemonType.FLYING]: 2,
    [PokemonType.BUG]: 2,
    [PokemonType.STEEL]: 0.5,
  },
  [PokemonType.GHOST]: {
    [PokemonType.NORMAL]: 0,
    [PokemonType.PSYCHIC]: 2,
    [PokemonType.GHOST]: 2,
    [PokemonType.DARK]: 0.5,
  },
  [PokemonType.DRAGON]: {
    [PokemonType.DRAGON]: 2,
    [PokemonType.STEEL]: 0.5,
    [PokemonType.FAIRY]: 0,
  },
  [PokemonType.DARK]: {
    [PokemonType.FIGHTING]: 0.5,
    [PokemonType.PSYCHIC]: 2,
    [PokemonType.GHOST]: 2,
    [PokemonType.DARK]: 0.5,
    [PokemonType.FAIRY]: 0.5,
  },
  [PokemonType.STEEL]: {
    [PokemonType.FIRE]: 0.5,
    [PokemonType.WATER]: 0.5,
    [PokemonType.ELECTRIC]: 0.5,
    [PokemonType.ICE]: 2,
    [PokemonType.ROCK]: 2,
    [PokemonType.STEEL]: 0.5,
    [PokemonType.FAIRY]: 2,
  },
  [PokemonType.FAIRY]: {
    [PokemonType.FIRE]: 0.5,
    [PokemonType.FIGHTING]: 2,
    [PokemonType.POISON]: 0.5,
    [PokemonType.DRAGON]: 2,
    [PokemonType.DARK]: 2,
    [PokemonType.STEEL]: 0.5,
  },
};

/**
 * Calculates the type effectiveness multiplier for an attack
 * @param attackingType - The type of the attacking move
 * @param defendingTypes - The type(s) of the defending Pokemon (1 or 2 types)
 * @returns The damage multiplier (0, 0.25, 0.5, 1, 2, or 4)
 */
function calculateTypeEffectiveness(
```

```
  attackingType: PokemonType,
  defendingTypes: PokemonType | PokemonType[]
): number {
  const defenderTypes = Array.isArray(defendingTypes)
    ? defendingTypes
    : [defendingTypes];

  let multiplier = 1;

  for (const defenderType of defenderTypes) {
    const effectiveness = TYPE_CHART[attackingType]?.[defenderType] ?? 1;
    multiplier *= effectiveness;
  }

  return multiplier;
}

// Usage examples:
// Single type defender
const effectiveness1 = calculateTypeEffectiveness(
  PokemonType.FIRE,
  PokemonType.GRASS
); // Returns 2 (super effective)

// Dual type defender
const effectiveness2 = calculateTypeEffectiveness(PokemonType.ELECTRIC, [
  PokemonType.WATER,
  PokemonType.FLYING,
]); // Returns 4 (2 × 2, doubly super effective)

export { PokemonType, calculateTypeEffectiveness };
```