Paradigmas y Lenguajes de Programación - UNPSJB Trelew

Laboratorio 01

Blackjack

Cátedra: + Prof: Lic. Romina Stickar + JTP: Lic. Lautaro Pecile

Alumnos: + Alzugaray Luciano + Krmpotic Lucas

- ## Estructura del proyecto
  - black.pl: Contiene la lógica para mandar comandos y ejecutar acciones.
  - cartas.pl: definición de las cartas y sus valores
  - utils.pl: reglas enunciadas en los objetivos preliminares
  - dealer.pl: reglas enunciadas en los objetivos intermedios
  - mejorJugada.pl: lógica para decidir la mejor jugada según la mano propia, la del crupier y las cartas jugadas.
  - cuentaUstonSS.pl: lógica de conteo de cartas según el sistema Uston SS
  - play.pl: archivo principal de la base de conocimientos
  - tests.pl: conjunto de casos de prueba para cada regla.

Lógica de la regla *play/3*

Argumentos:

- **ManoManoJugador:** lista con las cartas del jugador en una mano.
- **ManoManoCrupier:** lista con las cartas del crupier en una mano.
- **CartasJugadas:** lista de las cartas jugadas.

La regla **play/3** hace uso del functor **mejor_jugada/2** que, dada una **Mano** unifica con el **ValorMano** mas cercano a 21 pero no mayor a 21.

Nota: La lógica de **mejor_jugada/2** se encuentra declarada en el archivo mejorJugada.pl

Además se declaran 3 reglas que unifican con valores significativos para la decisión de seguir pidiendo cartas o no. Estas son:

1) **cota_inferior/1:** unifica con **ValorMano < 11** (11 es un valor que garantiza que pedir una carta no generará un **over** independientemente de como se componga ese 11)

2) **valor_umbral/1:** unifica con **ValorMano >= 17** (17 es un valor en el cual la decisión de pedir o no

carta dependerá de las probabilidades de carta baja o alta que arroje el estado de **Cuenta** según el sistema **Uston SS**)

3) **cota_superior/1:** unifica con **ValorMano >= 19** (a partir del 19 inclusive, convenimos en que la mejor jugada es plantarse independientemente de si la cuenta **Uston SS** arroja probabilidad de carta baja).

Con estos elementos, las posibilidades son de **play/3** son:

1) Pide cartas siempre que la mano del crupier sea superior.

```
play(ManoJugador, ManoCrupier, _):-
    mejor_jugada(ManoJugador, ValorManoJugador),
    mejor_jugada(ManoCrupier, ValorManoManoCrupier),
    not(ValorManoJugador > ValorManoManoCrupier).
```

2) Habiendo superado al crupier, pide carta si el valor de la mano es menor o igual a 11, salvo que sea con un **as** (también es 18 !!) y el conteo de cartas arroje que es probable que venga una carta baja.

```
play(ManoJugador, _, _):-
    hand(ManoJugador, ValorManoJugador),
    mejor_jugada(ManoJugador, ValorManoAux),
    cota_inferior(ValorManoJugador),
    not(valor_umbral(ValorManoAux)).
```

3)

```
play(ManoJugador, _, CartasJugadas):-
    mejor_jugada(ManoJugador, ValorMano),
    valor_umbral(ValorMano),
    not(cota_superior(ValorMano)),
    es_as_once(ManoJugador, ValorMano),
    contar_uston_ss(CartasJugadas, 1, Conteo),
    posibilidadDeCartaBaja(Conteo).
```

```
% Aca vemos la posibilidad de seguir sacando cartas para > 11 pero < 18.
play(ManoJugador, _, CartasJugadas):-
    mejor_jugada(ManoJugador, ValorMano),   % Veo la mano mas cercana a 21. Veo si
ese valor es
    ValorMano < 16,
    contar_uston_ss(CartasJugadas, 1, Conteo),  % es menor a 16 pido una carta
siempre y cuando el valor de
    not(posibilidadDeCartaAlta(Conteo)).               % el conteo me indique que
no hay posibilidades de sacar una carta alta.
```

Lógica de la regla *mejor_jugada/2*

Argumentos:

- **Mano:** lista de cartas de una mano. -**MejorValor**: el valor más cercano a 21 pero no mayor a 21 dentro de los valores posibles de Mano.

```
mejor_jugada(Mano, MejorValor):-
    findall(X, hand(Mano, X), ListaValoresDeLaMano),
    mejor_valor_mano(ListaValoresDeLaMano, 0, MejorValor).
```

Los posibles valores del **as** generan diferentes valores para una mano. Para decidir si se pide una nueva carta es necesario analizar el contexto, es decir, valores posibles de mano del Crupier y probabilidades de que la próxima carta sea baja (=<6) o alta (>6) según el sistema de conteo de cartas Uston SS.

La regla *mejor_jugada/2* define una lista con los valores posibles de la mano haciendo uso de *findall/3*, y llama a la funcion *mejor_valor_mano/3* que unifica con el valor de mano mas cercano a 21 pero no mayor a 21.

Sistema de conteo de cartas Uston SS

El sistema de conteo de cartas **Uston SS** define que una **carta baja es aquella cuyo valor es menor o igual a 6**, mientras que una **carta alta es aquella cuyo valor es estríctamente mayor a 6**.

Además, cada carta está asociada a un valor, que indica en cuanto hay que modificar la **Cuenta** cuando la carta es descubierta. Los valores asignados a cada carta son los siguientes:

2	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**J**	**Q**	**K**	**A**
+2	+2	+2	+3	+2	+1	0	-1	-2	-2	-2	-2	-2

El **valor inicial de Cuenta** se calcula multiplicando por **-2** la cantidad de barajas con las que se juega.

De este modo, el valor inicial de **Cuenta** se va alterando según el peso de cada carta que va saliendo. Así, **Cuenta** < -8 indica probabilidad de carta baja, mientras que **Cuenta** > 4 indica probabilidad de carta alta.

Comandos

Los comandos admitidos son:

- swipl -s black.pl play

Ejecuta el comando play con las cartas en juego. Un ejemplo de ejecución seria swipl -s black.pl play

hand jd ap dealer kt 2d cards jd ap kt 2d

- swipl -s black.pl dealer soft

Ejecuta el comando dealer_soft con las cartas del dealer. Un ejemplo de ejecución seria swipl -s black.pl dealer soft ad 6t

- swipl -s black.pl dealer hard

Ejecuta el comando dealer_hard con las cartas del dealer. Un ejemplo de ejecución seria swipl -s black.pl dealer hard ad 6t

Las reglas de parseado estan en el archivo black.pl. Cada comando tiene sigue un arbol de derivación, hasta comprobar que el comando sea correcto. En caso contrario, un mensaje de "comando incorrecto" aparece en pantalla.

En el caso de estar bien, ejecuta la regla hit o stand de acuerdo a si sigue jugando o se queda.

Para la traducción de las cartas está la regla string_to_card/2, que te devuelve una carta dado una token que signifique una carta.

La regla parse_args_play/2 es la que te parsea los argumentos de play y ejecuta el play.

Correr tests

El código contiene tests por cada función. Ante cualquier cambio o para prueba del código, se puede correr los tests con run_tests.