

REPRESENTACIÓN Y LÓGICA DEL MONOPOLIO EN PROLOG

Práctica 1: Conocimiento y Razonamiento Automatizado

(Adaptado para la asignatura de *Conocimiento y Razonamiento Automatizado*, Curso 2025-26)

Preámbulo

El Monopolio es un juego de tablero basado en la compraventa de propiedades, gestión de recursos económicos y toma de decisiones estratégicas bajo reglas formales.

El objetivo de esta práctica es que el estudiante desarrolle habilidades avanzadas en la manipulación de listas en Prolog, utilizando el juego como modelo para representar estados dinámicos y aplicar reglas de transformación sobre estructuras lineales.

No se pretende construir una versión comercial del juego, ni utilizar librerías externas ni predicados avanzados. La práctica debe desarrollarse exclusivamente con los mecanismos básicos del lenguaje Prolog, obligando al estudiante a implementar manualmente la gestión de estructuras y estados.

Esta es una práctica sobre representación simbólica y manipulación de listas, tomando como caso de estudio el juego del Monopoly. Vea el Ejemplo 1.



Ejemplo 1. Tablero del juego de monopolio típico

1. Objetivos

El objetivo de esta práctica es que el estudiante desarrolle habilidades en la manipulación de listas en Prolog, comprendiendo su importancia en la representación de problemas complejos mediante el desarrollo de un juego. Además, se busca que practique la aplicación iterativa de reglas para la simplificación progresiva de la solución, explorando técnicas de búsqueda, sustitución y filtrado dentro de estructuras lineales. En general se pretende gestionar y manipular listas en Prolog tomando como caso de uso la representación y simplificación (parcial o total) del juego de monopolio. Para lograrlo, se plantean las siguientes actividades:

1. Representación del Tablero: El tablero se representará como una lista lineal de 40 elementos, donde cada casilla corresponde a una propiedad, servicio, impuesto o casilla especial (Salida, Cárcel, Parking, Carta, etc). Vea el Ejemplo 2.

```
[ salida,  
  propiedad(marron1, 60, marron),  
  carta,  
  propiedad(marron2, 60, marron),  
  impuesto(200),  
  ...  
 ]
```

Ejemplo 2. Ejemplo conceptual de la representación del tablero como lista.

2. Estado de los Jugadores: Representación de los jugadores (mínimo 2) mediante una lista de estructuras que contenga su posición actual, dinero disponible y una sublista de propiedades adquiridas. Por ejemplo, el estado se pudiera representar como: “*estado(Jugadores, Tablero, Turno)*” y el jugador como: “*jugador(Nombre, Posicion, Dinero, Propiedades)*”.
3. Lógica de Movimiento: Implementación de predicados que simulen el lanzamiento de dados y actualicen la posición del jugador mediante la manipulación de los índices de la lista del tablero, gestionar paso por salida, controlar la posición circular (módulo 40).
4. Reglas de Interacción (Simplificación de Estados): Una vez que un jugador cae en una casilla, se aplican reglas de forma iterativa y coordinada:
 - Regla 0 (Compra): Si la casilla es una propiedad sin dueño y el jugador tiene fondos, se añade a su lista de propiedades.
 - Regla 1 (Alquiler): Si la propiedad tiene dueño, se resta el monto del dinero del jugador y se añade al dueño.
 - Regla 2 (Monopolio): Si un jugador posee todas las propiedades de un mismo color (sublista), se habilita la opción de construir casas.
 - Regla 3 (Bancarrota): Si el dinero es negativo, se ejecutan predicados de liquidación de activos o eliminación del jugador de la lista global.

Notas:

- No se utilizarán generadores aleatorios externos; el dado puede simularse mediante hechos o listas predefinidas.
- Las reglas no son independientes. La aplicación de una puede activar otras.
- Se debe implementar un mecanismo de iteración por turnos.

2. Práctica a Entregar

La práctica consistirá en **dos partes principales**:

1. Programa Principal (main.pl)

- Donde se incluyan todos los predicados necesarios para:
 - Representación del tablero.
 - Representación inicial de jugadores.
 - Predicados de movimiento.
 - Aplicación de reglas (Regla 0, 1, 2 y 3).
 - Control de turnos.
 - Métricas
 - Aplicación de reglas adicionales
 - Se recomienda sea modular por lo tanto el main.pl puede ser un script de entrada a la aplicación. Vea Formato del archivo en el Apartado 5.

2. Breve Informe (Informe.pdf)

- Explicando:
 - Objetivos alcanzados.
 - Representación elegida.
 - Descripción detallada de las reglas y cómo están implementadas.
 - Análisis de ejecución.
 - Mejoras implementadas o extensiones (ver Sección 4).
 - Fuentes consultadas.
 - Distribución de tareas en el equipo de trabajo.

3. Conjunto de ejemplos

Cada grupo debe proponer 5 escenarios iniciales de juego. Ejemplos sugeridos:

- Escenario 1:
Dos jugadores, primeras compras iniciales.
- Escenario 2:
Un jugador con monopolio formado.
- Escenario 3:
Jugador próximo a bancarrota.
- Escenario 4:
Estado con múltiples alquileres consecutivos.
- Escenario 5:
Simulación completa de 10 turnos.

4. Mejoras del Programa

A. Posibles Mejoras

1. Interfaz ASCII del tablero.
2. Reglas Avanzadas: Incluir técnicas adicionales para resolver más casos o implementar funcionalidades no exploradas en el juego.
3. Librería clpfd: explorar posibilidades con clpfd y comparar con análisis funcionalidad y eficiencia.
4. Iteración y Métricas: Automatizar la iteración de reglas y medir el número de iteraciones.
5. Detección de Conflictos: Implementar verificación automática para identificar conflictos.
6. Interfaz Interactiva: Permitir la entrada manual y mostrar el progreso paso a paso.
7. Construcción de casas.
8. Hipotecas.
9. Gestión de cartas.
10. Patrimonio total por jugador.
11. Ranking dinámico.
12. Detección de estancamiento del juego.
13. Simulación automática de partidas completas.
14. Interfaz avanzada en Python.

B. Metodología de Implementación de mejoras

1. Revisión teórica.
2. Propuesta de implementación.
3. Desarrollo del código en Prolog.
4. Pruebas y validación.
5. Análisis de resultados.
6. Conclusiones.
7. Documentación.

5. Detalles de la Entrega

- **Equipo:** De 3 a 5 personas, con al menos dos mejoras implementadas. Los equipos con más de cinco personas deben adicionar una mejora por cada integrante adicional.
- **Cada equipo debe presentar resultados parciales cada dos semanas (minutas) donde especificara tarea realizada, resultados obtenidos y próximo paso.**
- **Formato del archivo:**
El archivo .zip debe tener la siguiente estructura:
 - apellido1Apellido2Apellido3_PLX.zip
 - └── main.pl
 - └── reglal.pl
 - └── metricas.pl
 - └── Informe.pdf
 - Use camelCase para el nombre del archivo
- **Informe (15-20 páginas):**
 - Portada.
 - Resumen.
 - Índice.
 - Introducción.

- Objetivos.
 - Desarrollo:
 - Descripción de la representación.
 - Aplicación de las reglas de simplificación (Reglas 0, 1, 2 y 3).
 - Mejoras implementadas y propuestas adicionales.
 - Otra información técnica de interés.
 - Resultados y análisis:
 - Resultados de lo desarrollado.
 - Análisis de los resultados.
 - Análisis de los resultados en la evolución de ejemplos concretos.
 - Métricas.
 - Complejidad
 - Conclusiones.
 - Bibliografía (**OJO: es importante**)
 - Anexos. Todo el código desarrollado. Se valora positivamente si crean y usan convenientemente un repositorio de Github.
 - **Nota:** el uso de la IA no se prohíbe pero lo generado se debe comentar, desde el prompt utilizado y la respuesta dada.
-
- **Entrega:**
 - Plataforma: Campus Virtual de la UAH.
 - Semana del PEC1

6. Aspectos de Evaluación y Rúbrica para la Práctica

- Puntuación: Cada práctica vale 100 puntos y se promedia la nota final aplicando el porcentaje correspondiente para prácticas de laboratorio.
- Rúbrica: se usa para evaluar todo el proceso de la práctica.
- Proceso de Evaluación: El profesor puede consultar sobre la práctica en cualquier momento. Al final de la práctica se realiza un proceso de defensa de la misma. El estudiante tiene derecho de defender su práctica.
- Implementación de la práctica: Debe estar operativa para evaluar el informe.
- Minutas: Se deben entregar al menos 3 minutas de la práctica. Estas contienen información sobre las tareas realizadas por el estudiante para que avance la práctica. Es individual. En ella se indica la tarea realizada por el estudiante, resultado obtenido y la próxima tarea a realizar. La no entrega de minutas hace que el estudiante pierda la práctica. Si no recibes feedback de la minuta, se considera aprobada.
- Cronograma de Minutas: Primera minuta después de la explicación del profesor (siguiente semana obligatorio). Entrega cada dos semanas como máximo. La última minuta puede ser la entrega de la práctica final. En cada entrega se debe indicar la tarea a realizar.