**Capítulo 5 – problemas de satisfacción de restricciones**

**5.1 – Problemas de satisfacción de restricciones**

* Tratar estados como más que sólo pequeñas cajas negras conduce a la invención de una nueva variedad de métodos de búsqueda
* Los capítulos anteriores exploraron la idea de que los problemas puedes resolverse buscando en un espacio de estados.
  + Los estados pueden evaluarse con heurísticas específicas del dominio y probados para ver si son estados objetivo
  + Desde el punto de vista del algoritmo de búsqueda, cada estado es una **caja negra**
    - Solo se le puede acceder con las rutinas especificas del problema
* En este capítulo se examinan **problemas de satisfacción de restricciones**
  + Los estados y test objetivo forman una representación simple, estándar y estructurada
  + Se utilizan heurísticas de propósito general mas que heurísticas especificas al problema

**Problemas de satisfacción de restricciones**

* Un PSR está definido por
  + un conjunto de variables X1, X2…Xn
  + Un conjunto de restricciones C1, C2..Cm
* Cada variable X tiene un **dominio** de **valores** posibles no vacío
* Cada restricción implica algún subconjunto de variables y especifica las combinaciones aceptables de valores para ese subconjunto
* Un estado del problema está definido por una **asignación** de valores a una o todas las variables
  + Una asignación que no viola ninguna restricción es asignación **consistente**
  + Una asignación completa es una asignación en la que se menciona cada variable
  + Una **solución de un PSR** es una asignación completa que satisface todas las restricciones
* Es bueno visualizar un PSR como un **grafo de restricciones**
  + Los nodos del grafo corresponden a variables del problema y los arcos corresponden a restricciones
* Al PSR se le puede dar una **formulación incremental** como en un problema de búsqueda estándar:
  + Estado inicial: asignación vacía { }, todas las variables no están asignadas
  + Función de sucesor: un valor se puede asignar a cualquier variable no asignada
  + Test objetivo: asignación actual es completa
  + Costo del camino: un costo constante para cada paso
* Cada solución debe ser una asignación completo, aparecen a profundidad n si hay n variables
  + Los algoritmos de búsqueda primero en profundidad son populares para PSRs
* El camino que alcanza una solución es irrelevante, podemos usar una **formulación completa de estados**
  + Cada estado es un asignación completa que podría o no satisfacer las restricciones
* La clase más simple de PSR implica variables discretas dominios finitos
* El número de posibles asignaciones completas es exponencial en el número de variables
* Los PSR con dominio finito incluyen **PSRs booleanos**
  + Las variables pueden ser verdaderas o falsas
* Las variables discretas también pueden tener **dominios infinitos**
  + No se puede describir restricciones enumerando todas las combinaciones permitidas de valores
  + En vez de eso, se usa un **lenguaje de restricción**
* Los problemas de satisfacción de restricciones con dominios continuos son muy comunes en el mundo real y son ampliamente estudiados en IO
  + Los **problemas de programación lineal** son la categoría más conocida de PSRs en dominios continuos
    - Las restricciones deben ser desigualdades lineales
* Además de tipos de variables, hay tipos de restricciones
  + **Restricción unaria:** restringe los valores de una sola variable
  + **Restricción binaria:** relaciona dos variables

**Palabras clave**

|  |  |
| --- | --- |
| **Caja negra** | Estado sin estructura perceptible interna. Estructura de datos arbitraria a la que se puede acceder sólo con las rutinas especificas del problema |
| **Problemas de satisfacción de restricciones** | Definido por un conjunto de variables y restricciones |
| **Consistente** | Asignación que no viola ninguna restricción |
| **Grafo de restricciones** | Los nodos del grafo corresponden a variables del problema y los arcos corresponden a restricciones |
| **PSRs booleanos** | PSR donde las variables pueden ser verdaderas o falsas |
| **Restricción unaria** | restringe los valores de una sola variable |
| **Restricción binaria** | Relaciona dos variables |