Backtracking I

manuel.freire@uam.es

Contenido

- Estimando tiempos; necesidad de poda
- Idea general: búsqueda y poda
- Algunos ejemplos
  - Todos los subconjuntos
  - Todas las permutaciones
  - Eligiendo una buena representación: N-reinas

Estimando tiempos

- PC típico: ~1 Ghz
  - ~5 M estados/s a 200 ops/estado: pocas
  - 5 M < 2^{22} subconjuntos de 22 elementos
  - 5 M < 10! permutaciones con 10 elementos
- Cuando falla la fuerza bruta...
  - Evita revisitar simetrías, caminos alternativos
  - Evita visitar estados “inútiles” ¡poda!

Idea general

```
int backtrack(estado E, datos_problema D) {
   if (es_solucion(E, D)) {
      return procesa_solucion(E, D);
   }
   else {
      vector<estado> C;                    // facil y muy util
      construye_candidatos(E,D,C);
      for (int i=0; i<C.size(); i++) {
         int finalizado = backtrack(C[i],D);
         if (finalizado) return finalizado;
      }
   }
}
```

- Construir el 'estado' poco a poco
- Recursividad ~= búsqueda en profundidad en grafo - eficiente!
- Próximo día: revisitando A* - IA no queda *tan* lejos
- Usad “vector”, que manejar memoria no es divertido.
  g++ #include<vector> using namespace std;

Ejemplo: subconjuntos (1/3)

- estado

```
typedef struct estado {
   char e[N];   // e[i] == 1 => elemento i-esimo en uso
   int n;       // numero de procesados
};
```

- es_solucion

```
return E.n == N;
```

- procesa_solucion

```
for (int i=0; i<N; i++) if (E.e[i]) printf("%d ", i);
printf("\n");
```

- construye_candidatos

```
estado tmp = E;                    // copia de estructuras
tmp.n ++;
tmp[E.n] = 0; C.push_back(tmp);   // copia al introducir
tmp[E.n] = 1; C.push_back(tmp);
```

Ejemplo: permutaciones (2/3)

- estado

```
typedef struct estado {
    char e[N];        // la permutacion (llena hasta 'n')
    int n;
    char usado[N];    // elementos ya usados
    estado() {        // inicializa la estructura, 'constructor'
        for(int i=0; i<N; i++) usado[i] = 0;
    }
};
```

- es_solucion, procesa_solucion: siguen igual
- construye_candidatos

```
estado tmp = E;
tmp.n ++;
for (int i=0; i<N; i++) {
    if ( ! tmp.usado[N]) {
        tmp.e[E.n] = i; tmp.usado[i] = 1; C.push_back(tmp);
    }
}
```

Ejemplo: n-reinas (3/3)

- ¿Colocaciones de N reinas de ajedrez en un tablero de NxN? (por ejemplo, N=8)
  - Array de 8x8 elementos: 2^{64} posiblidades
  - Array de 8 posiciones de reina: $64^8 = 2^{48}$
  - Array de 8 posiciones, por filas: $8^8 = 2^{24}$
  - Igual, evitando repetidos: 8! = 40320

- Cuando falla la fuerza bruta...
  - Evita revisitar simetrías, caminos alternativos
  - Evita visitar estados "inútiles" ¡poda!