Solución de Problemas con CCP
Consistencia

slides basados en el curso “constraint Programming” de Christian Schulte [2]
Profesor: Camilo Rueda [1]

[1]Universidad Javeriana-Cali,
[2]KTH Royal Institute of Technology, Sweden

PUJ 2008

Fortaleza de la propagación

Propagación: mecanismo para podar valores que no pueden participar en una solución

- En el caso de “distinct” (restricción de “todos distintos”)
  - el mecanismo es valor-consistencia
  - cuando una variable queda con un solo valor, removerlo de los dominios de las otras
  - es la forma ingenua de propagador para “distinct”
  - complejidad: lineal en el número de variables
- Mejora?
  - compromiso fortaleza/complejidad

Problemas de propagación ingenua

- Ejemplo:

  $$distinct(x, y, z)$$
  $$x \in \{1, 2\} \quad y \in \{1, 2\} \quad z \in \{1, 2\}$$

  hay solución?

Problemas de propagación ingenua

- Ejemplo:

$$distinct(x, y, z)$$
$$x \in \{1, 2\} \quad y \in \{1, 2\} \quad z \in \{1, 2\}$$

- no hay solución y la propagación ingenua no lo nota
- se debe recurrir a la exploración

Problemas de propagación ingenua (2)

- Ejemplo:

  $$distinct(x, y, z)$$

  $$x \in \{1, 2\} \quad y \in \{1, 2\} \quad z \in \{1, 2, 3\}$$

  hay solución?

Problemas de propagación ingenua (2)

- Ejemplo:

$$distinct(x, y, z)$$
$$x \in \{1, 2\} \quad y \in \{1, 2\} \quad z \in \{1, 2, 3\}$$

- claramente si.
- Además, $z = 3$
- pero distinct ingenuo no lo nota
- se debe recurrir a exploración

Problemas de propagación ingenua (2)

- Ejemplo:

$$distinct(x, y, z)$$
$$x \in \{1, 2\} \quad y \in \{1, 2\} \quad z \in \{1, 2, 3\}$$

- claramente si.
- Además, $z = 3$
- pero distinct ingenuo no lo nota
- se debe recurrir a exploración

La solución es hacer más fuerte
la poda de dominios: dominio-consistencia

Dominio-consistencia

- Para cada restricción r, propagar de modo que al final de la propagación

> para cada variable x
> y para cada valor v en el dominio de x
> exista una solución con $x = v$

La fortaleza de un propagador es parámetro fundamental que debe ajustarse

Dominio-consistencia: ejemplo

$$x = 3y + 5z$$
$$x \in \{2, 3, ..., 7\} \quad y \in \{0, 1, 2\} \quad x \in \{-1, 0, 1, 2\}$$

- Propagación de dominio-consistencia:
  $x \in$
  $y \in$
  $z \in$
- soluciones:
- complejidad:

Dominio-consistencia: ejemplo

$$x = 3y + 5z$$
$$x \in \{2, 3, ..., 7\} \quad y \in \{0, 1, 2\} \quad x \in \{-1, 0, 1, 2\}$$

- Propagación de dominio-consistencia:
  $x \in \{3, 5, 6\}$
  $y \in$
  $z \in$
- soluciones:
- complejidad:

Dominio-consistencia: ejemplo

$$x = 3y + 5z$$
$$x \in \{2, 3, ..., 7\} \quad y \in \{0, 1, 2\} \quad x \in \{-1, 0, 1, 2\}$$

- Propagación de dominio-consistencia:
  $x \in \{3, 5, 6\}$
  $y \in \{0, 1, 2\}$
  $z \in$
- soluciones:
- complejidad:

Dominio-consistencia: ejemplo

$$x = 3y + 5z$$
$$x \in \{2, 3, ..., 7\} \quad y \in \{0, 1, 2\} \quad x \in \{-1, 0, 1, 2\}$$

- Propagación de dominio-consistencia:
  $x \in \{3, 5, 6\}$
  $y \in \{0, 1, 2\}$
  $z \in \{0, 1\}$
- soluciones:
  $(3, 1, 0), (5, 0, 1), (6, 2, 0)$
- complejidad:

Dominio-consistencia: ejemplo

$$x = 3y + 5z$$
$$x \in \{2, 3, ..., 7\} \quad y \in \{0, 1, 2\} \quad x \in \{-1, 0, 1, 2\}$$

- Propagación de dominio-consistencia:
  $x \in \{3, 5, 6\}$
  $y \in \{0, 1, 2\}$
  $z \in \{0, 1\}$
- soluciones:
  $(3, 1, 0), (5, 0, 1), (6, 2, 0)$
- complejidad:
  $o(n^2)$

Descomposición de restricciones

- En principio toda restricción puede decomponerse en colección de binarias
  - pero es mala idea!
- $distinct(x_1, ..., x_n)$ puede descomponerse en n^2 propagadores:
  $x_i \neq x_j \quad$ para $1 \leq i < j \leq n$
  - no hay mejora en propagación
  - requiere complejidad de $o(n^2)$ en espacio
  - se pierde la visión global de la restricción

Restricciones globales

- Son restricciones sobre varias variables
  - sus propagadores requieren menos memoria
  - podan más valores
  - usan algoritmos con conocimiento del área de aplicación
- rama de investigación muy activa

Propagación de restricciones aritméticas

- Sea un propagador para
  $2 \times x = y$
  $x \in \{1,2,3,4\}$ $y \in \{1,2,3,4,5,6\}$
- propagación de dominio obtendría
  $x \in \{1,2,3\}$ $y \in \{2,4,6\}$
- el usual de Gecode/J obtiene (ver código)
  $x \in \{1,2,3\}$ $y \in \{2,3,4,5,6\}$

Propagación de restricciones aritméticas de Gecode

- Usa límite-consistencia
  - solamente opera sobre los límites del intervalo
  - se conoce también como propagación intervalar
  - obtiene consistencia solamente para los límites de cada intervalo
- típicamente es suficiente en muchas aplicaciones

Costos de la propagación

- Restricción “distinct” con n variables
  - valor-consistencia: $o(n)$
  - límite-consistencia: $o(n \log n)$
    en muchos casos, $o(n)$
  - dominio-consistencia: $o(n^{2,5})$
- Restricción aritmética con n variables
  - valor-consistencia(inútil): $o(n)$
  - límite-consistencia: $o(n)$
  - dominio-consistencia: *exponencial*

Problemas de satisfacción de restricciones

$$\boxed{CSP = \langle V, U, R \rangle}$$

- Un conjunto V de variables
  $V = \{x_1, x_2, ..., x_n\}$
- Un universo U de valores posibles (tipo)
  todas las variables toman valores de U
- Un conjunto R de restricciones
  - cada restricción involucra algunas de las variables
  - establecen valores aceptables (sus soluciones)

Componentes de las restricciones

una restricción r define:

- sus variables
  $$var(r) = (x_1, ..., x_k) \in V^k$$
- sus soluciones
  $$sol(r) \subseteq U^k$$

Asignaciones

- una asignación a define qué valor toma cada variable:
  $a : V \rightarrow U$
- ua asignación a es solución de una restricción r (se escribe $a \in r$) ssi
  - $var(r) = (x_1, ..., x_k)$
  - $(a(x_1), ..., a(x_k)) \in sol(r)$

Ejemplo

$V = \{x, y, z\}$	variables
$U = \{1, 2, 3\}$	universo
$a(x) = 2,\ a(y) = 3,\ a(z) = 2$	asignación

La asignación también se escribe:

$a = \{x \mapsto 2, y \mapsto 3, z \mapsto 2\}$

Solución de un CSP

- una asignación $a \in V \rightarrow U$ es una solución de un CSP $P = \langle V, U, R \rangle$ si
  - $a \in r$ para toda $r \in R$
- las soluciones $sol(P)$ de P se definen como

$$sol(P) = \{s : V \rightarrow U \mid s \text{ es solución de } P\}$$

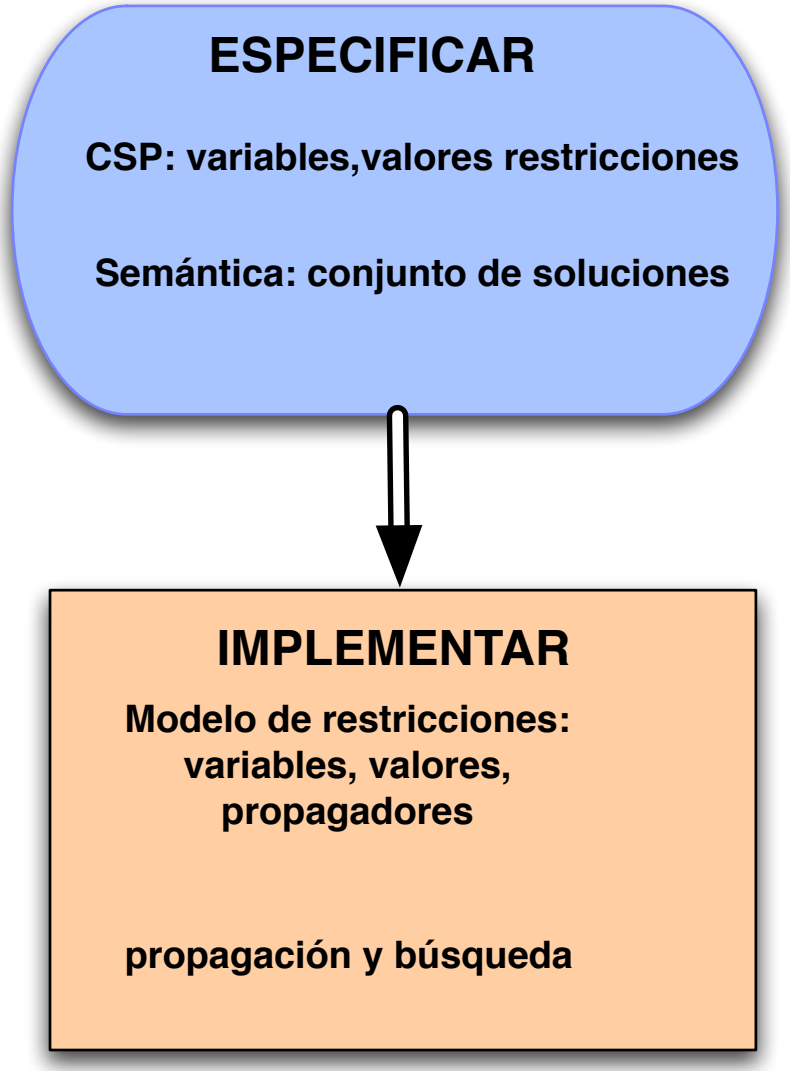
Ejemplo

- $P = \langle V, U, R \rangle$ con
  - $V = \{x, y, z\}$
  - $U = \{1, 2, 3\}$
  - $R = \{r_1, r_2, r_3\}$ donde

$var(c_1) = (x, y)$ $sol(c_1) = \{(1,2), (1,3), (2,1), (2,3), (3,1), (3,2)\}$
$var(c_2) = (x, z)$ $sol(c_2) = sol(c_1)$
$var(c_3) = (y, z)$ $sol(c_3) = sol(c_1)$

$$sol(P) = \left\{ \begin{array}{l} (1,2,3), (1,3,2), (2,1,3) \\ (2,3,1), (3,1,2), (3,2,1) \end{array} \right\}$$

Especificar vs implementar



Especificar vs implementar (2)

- Un modelo implementa un CSP
  - cuando tienen el mismo conjunto de soluciones
- propiedades de un propagador N_r de una restricción r : sean $var(r) = (x_1, ..., x_k)$, $D_1, D_2 \subseteq U^k$,

$N_r(D_1) \subseteq D_1$	contracción
$sol(r) \cap D_1 \subseteq N_r(D_1)$	validez
$D_1 \subseteq D_2 \Rightarrow N_r(D_1) \subseteq N_r(D_2)$	monotonicidad

Resolver un CSP

- Encontrar una implementación: el modelo de restricciones
- El modelo usa propagadores
- Los propagadores deben cumplir ciertas propiedades

Modelo de restricciones

- Ofrece una implementación de un CSP
  - asegurarse de que efectivamente implementa el CSP
- en lugar de restricciones se tienen propagadores
  - qué es un propagador?
  - los propagadores operan sobre un store de restricciones
  - qué es exactamente un store de restricciones?

Store de restricciones

- función que asocia variables con conjuntos de valores:
  $s \in V \to \mathbb{P}(U)$
  - un conjunto de stores se denota $S = V \to \mathbb{P}(U)$
- Los propagadores y ramificadores operan sobre stores

Fortaleza de un store

- un store s_1 es más fuerte que un store s_2 (escrito $s_1 \leq s_2$) si y solo si
  $$s_1(x) \subseteq s_2(x) \quad \text{para toda } x \in V$$
- s_1 es estrictamente más fuerte que s_2 (escrito $s_1 < s_2$) ssi
  $$s_1 \leq s_2 \wedge s_1 \neq s_2$$

Fortaleza de un store

- un store s_1 es más fuerte que un store s_2 (escrito $s_1 \leq s_2$) si y solo si

  $$s_1(x) \subseteq s_2(x) \quad \text{para toda } x \in V$$

- s_1 es estrictamente más fuerte que s_2 (escrito $s_1 < s_2$) ssi

  $$s_1 \leq s_2 \wedge s_1 \neq s_2$$

Ejemplo :

$V = \{x, y\}, \quad U = \{1, 2, 3\}$
$s_1 = \{x \mapsto \{1, 2\}, y \mapsto \{2, 3\}\}$
$s_2 = \{x \mapsto \{2\}, y \mapsto \{2, 3\}\}$
$s_3 = \{x \mapsto \{2, 3\}, y \mapsto \{1, 2, 3\}\}$

entonces, $s_2 < s_1$ y $s_2 < s_3$
pero no se cumple $s_3 \leq s_1$
ni tampoco $s_1 \leq s_3$

Propiedades de los propagadores

- contractor: debe calcular stores más fuertes
- propagador: función de stores a stores, $p \in S \rightarrow S$
  - $p(s) \leq s$ para todo store s
- si p implementa la restricción r,
  - p nunca elimina una solución de r
  - cómo definir formalmente lo anterior?
    problema: p relaciona stores, mientras que *sol* trata de asignaciones

Propiedades de los propagadores (2)

- Escribimos $a \in s$, para una asignación a y store s si
  - $a(x) \in s(x)$ para toda variable $x \in V$
- Definimos
  - $store_a(x) = \{a(x)\}$
    ($store_a$ es la asignación a vista como store)
  - propiedad: $a \in s \Leftrightarrow store_a \leq s$

Propiedades de los propagadores (2)

- Escribimos $a \in s$, para una asignación a y store s si
  - $a(x) \in s(x)$ para toda variable $x \in V$
- Definimos
  - $store_a(x) = \{a(x)\}$
    ($store_a$ es la asignación a vista como store)
  - propiedad: $a \in s \Leftrightarrow store_a \leq s$

Ejemplo :

$V = \{x, y\}, \quad U = \{1, 2, 3\}$
$a = \{x \mapsto 2, y \mapsto 3, z \mapsto 1\}$
$store_a = \{x \mapsto \{2\}, y \mapsto \{3\}, z \mapsto \{1\}\}$

Ejemplo de propagador

$V = \{x, y\}, \quad U = \{0, ..., 5\}$

propagador $p_{\leq}$ *para* $x \leq y$:

$$p_{\leq}(s) = \{x \mapsto \{d \in s(x) \mid d \leq max(s(y))\} \\ y \mapsto \{d \in s(y) \mid d \geq min(s(x))\}\}$$

Ejemplo de propagador (2)

Para el store

$$s = \{x \mapsto \{1, 3, 4, 5\}, y \mapsto \{0, 1, 2, 4\}$$

El propagador $P_{\leq}$ retorna

$$\begin{aligned} p_{\leq}(s) = \{x &\mapsto \{d \in s(x) \mid d \leq 4\} \\ y &\mapsto \{d \in s(y) \mid d \geq 1\}\} \\ &= \{x \mapsto \{1, 3, 4\}, y \mapsto \{1, 2, 4\}\} \end{aligned}$$

Implementar restricción

- el propagador p implementa la restricción c si

$$\boxed{a \in c \text{ si y solo si } p(store_a) = store_a}$$

  o sea,
  - las soluciones de c son puntos fijos de p
- Lo anterior no basta

mantener soluciones

- Supongamos que p implementa c, y $a \in c$
- Se requiere: si $a \in s$ entonces $a \in p(s)$

$$
\begin{aligned}
& a \in s \Leftrightarrow store_a \leq s \\
& \quad ???? \\
& \quad \Leftrightarrow store_a \leq p(s) \\
& \quad \Leftrightarrow a \in p(s)
\end{aligned}
$$

propagador inválido

- Supongamos el propagador
  $p(s) =$ **if** $s(x) = \{1,2,3$ **then** $\{x \mapsto \{1\}\}$
  else s
  y los stores

$$s_1 = \{x \mapsto \{1,2,3\}\}$$
$$s_2 = \{x \mapsto \{1,2\}\}$$

  Entonces,

  $s_1 > s_2$ pero $p(s_1) < p(s_2)$

  - haría la propagación dependiente del orden !!
  - tal propagador debe prohibirse

los propagadores deben ser monotónicos

$p \in S \to S$ es

- contractor $p(s) \leq s$
- monotónico $s_1 \leq s_2 \Rightarrow p(s_1) \leq p(s_2)$

$$a \in s \Leftrightarrow store_a \leq s$$
$$\Rightarrow p(store_a) \leq p(s)$$
monotonicidad
$$\Leftrightarrow store_a \leq p(s)$$
$$\Leftrightarrow a \in p(s)$$

asignaciones no válidas

- suponga que p implementa c, que $a \notin c$ y sea $s = store_a$. Entonces p falla en s

$$
\begin{aligned}
& a \notin s \Leftrightarrow p(s) \neq s \\
& \quad \Rightarrow p(s) < s \\
& \qquad \text{(contractor)} \\
& \quad \Rightarrow \exists x \in V \text{ tal que } p(s)(x) \subset s(x) \\
& \quad \Rightarrow \exists x \in V \text{ tal que } p(s)(x) = \emptyset \\
& \quad \Rightarrow p \text{ falla en } s
\end{aligned}
$$

Modelo de restricciones

- un modelo de restricciones $M = \langle V, U, P \rangle$ se define mediante
  - un conjunto de variables V
  - un conjubto de valores U
  - un conjunto de propagadores P

Soluciones de un propagador

- Las soluciones $sol(p)$ de un propagador p se definen como $\{a \in V \to U \mid store_a = p(store_a)\}$
- Las soluciones $sol(M)$ de un modelo de restricciones $M = \langle V, U, P \rangle$ se definen como $\{a \in V \to U \mid a \in sol(p) \text{ para todo } p \in P\}$

Un modelo de restricciones $M = \langle V, U, P \rangle$ implementa un CSP C si
$sol(M) = sol(C)$

Soluciones (2)

- Interesan las soluciones a un modelo M obtenidas a partir de un store s:

  $$sol(M, s)$$

  definidas como

$$\boxed{sol(M, s) = \{a \in sol(M) \mid a \in s\}}$$

Propiedades de un propagador
para todo modelo $M = \langle V, U, P \rangle$ y store s un propagador p debe cumplir $sol(M, s) = sol(M, p(s))$
los propagadores preservan las soluciones

Función de propagación (ingenua)

propagar((V,U,P),s)
  while $p \in P$ *and* $p(s) \neq s$ *do*
    $s := p(s)$;
  return s;

- qué devuelve como resultado?
- termina?

Función de propagación (ingenua)

propagar((V,U,P),s)
  while $p \in P$ *and* $p(s) \neq s$ *do*
    $s := p(s)$;
  return s;

- qué devuelve como resultado?
- termina?
  si, por contracción y propiedad de orden bien-fundado

El resultado calculado

- supongamos $propagar(\langle V, U, P\rangle, s) = s_r$

  $sol(\langle V, U, P\rangle, s) = sol(\langle V, U, P\rangle, s_r)$
  (no se eliminan soluciones)

  para todo $p \in P : p(s_r) = s$
  (no hay más propagación posible)
  (el punto fijo simultáneo más grande)

El resultado calculado (2)

- supongamos $propagar(\langle V, U, P\rangle, s) = s_r$, entonces

s_r es el máximo punto fijo simultáneo con $s_r \leq s$
es decir, para todo $p \in P$: $p(s_r) = s$

Por qué propagador ingenuo

- Busca siempre entre todo P un propagador que pueda contraer estrictamente
  - mantiene propagadores que se sabe han llegado a su punto fijo
  - puede invocar propagadores que no hacen contracción

Mejorar el propagador

- Idea: un propagador poda dominios de algunas pocas variables
  - repropagar solamente propagadores que comparten alguna de ellas
- mantener un conjunto de propagadores “activos”
  - no se sabe si ya llegaron a su punto fijo en el store actual
  - todos los demás ya lo hicieron
  - propagar solamente los “activos”

Variables de un propagador

- Variables $var(p)$ del propagador p
  - variables de interés para p
- No se consideran entradas de otras variables
  para todo $s_1, s_2 \in S$
  si ocurre que
  $\forall x \in var(p) : s_1(x) = s_2(x)$
  entonces, $\forall x \in var(p)$:
  $$p(s_1)(x) = p(s_2)(x) \ )$$
- No se afectan otras variables con los resultados:
  para todo $s \in S$ y para toda $x \in (V - var(p))$
  $$p(s)(x) = s(x)$$

Mejorar el propagador (2)

- Mantener un conjunto A de propagadores activos
- Escoger de A el propagador que debe ejecutarse y quitarlo de A
- calcular las variables modificadas
- agregar a A los propagadores que comparten variables modificadas
  - incluyendo el actual!

Propagador mejorado

$propague((V, U, P), s_0)$
$s := s_0;\ A := P;$
while $A \neq \emptyset$ *do*
escoja $p \in A$;
$s' := p(s);\ A := A - \{p\};$
$VM := \{x \in V \mid s(x) \neq s'(x)\};$
$PA := \{q \in P \mid \exists x \in var(q) : x \in VM\};$
$A := A \cup PA;$
$s := s';$
return s;

Propiedades

- Qué calcula
  - puntos fijos simultáneos?
  - el más grande?
  - definir invariante del ciclo
- Terminación?
  - los stores ya no son estrictamente más fuertes

Invariante

- El ciclo mantiene la propiedad
  - para todo $p \in P - A \Rightarrow p(s) = s$
  - después de terminación ($A = \emptyset$):
    para todo $p \in P \Rightarrow p(s) = s$