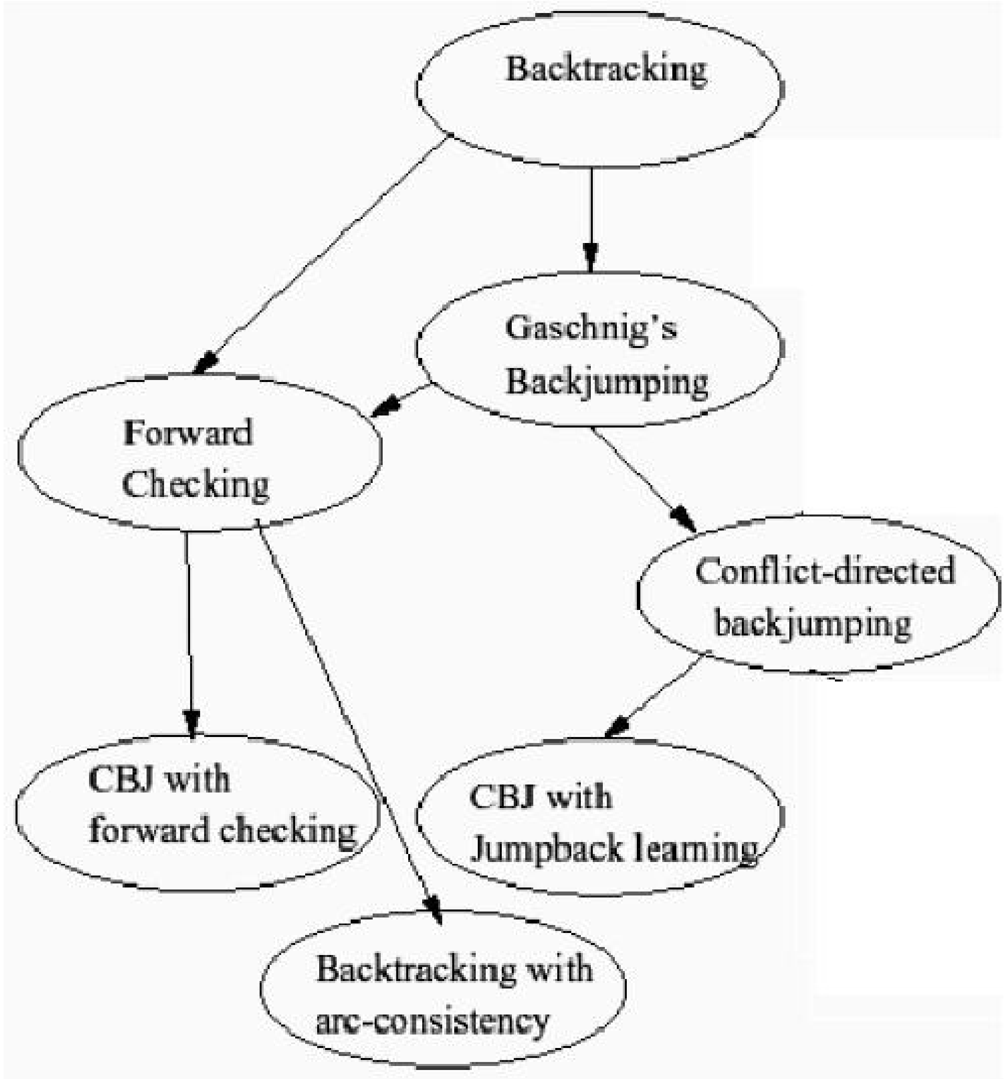
Srovnání prohledavacích algoritmů

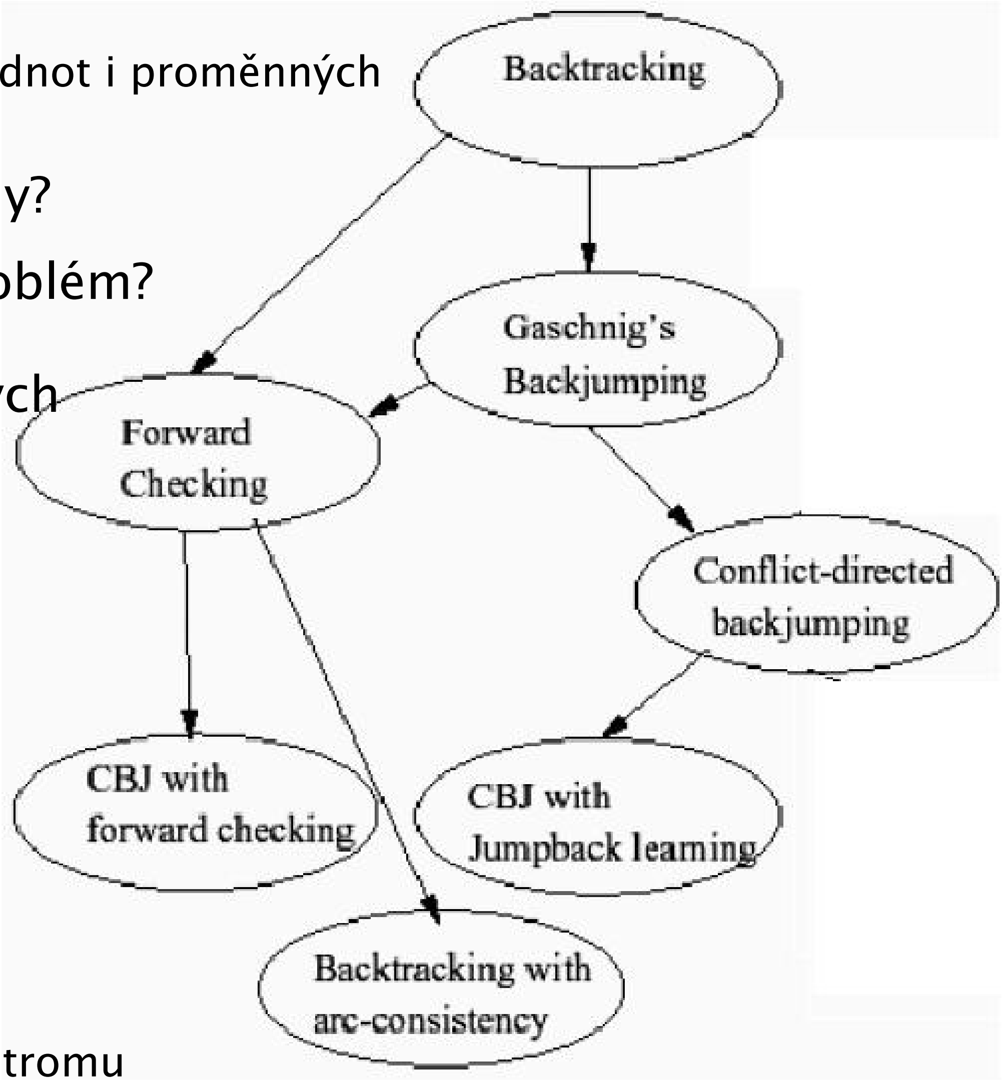
Srovnání prohledávacích algoritmů

- $A \rightarrow B$ znamená, že uzly prohledávacího stromu B jsou i v stromě A



Srovnání prohledávacích algoritmů

- $A \rightarrow B$ znamená, že uzly prohledávacího stromu B jsou i v stromě A
  - za předpokladu stejného uspořádání hodnot i proměnných
- Existuje srovnání i pro další algoritmy?
  Jaké algoritmy používat pro daný problém?
- **Experimentální porovnání** na různých sadách problémů ***(benchmarks)***
  - reálné problémy
  - nahodně vygenerované problémy
  - aplikačně založené náhodné problémy
- Kriteria
  - CPU čas
  - velikost generovaného prohledávacího stromu
  - počet volání procedury (např. `Consistent`)



Experimenty na reálných problémech

- Sady reálných problémů ***(benchmarks)***, na kterých lze algoritmy porovnávat
- CSPLib `http://www.csplib.org`
  - knihovna problémů pro omezující podmínky (otevřená pro nové problémy)
  - kolem 130 problémů z oblasti jako je rozvrhování, návrh a konfigurace, kombinatorika, bioinformatika, hry
  - popis problému, reference na jeho řešení, data, výsledky
    někdy i řešení nebo podrobné studie různých možností řešení
  - příklady
    - dopravní signalizace v čase na zadaných křižovatkách, výrobní linka, problém batohu, sledování cíle v distribuovaných senzorických sítích, ...
- Problém: výsledky lze stále velice obtížně zobecnit na další problémy
  - pro jeden problém je lepší jeden algoritmus, pro další problém jiný algoritmus

Náhodné problémy

- Algoritmy porovnávány na umělých, náhodně vygenerovaných problémech
  - lze generovat problémy různé obtížnosti (**fáze přechodu**)
  - libovolný počet datových instancí
  - lze testovat, co se stane (např. s parametry algoritmu) při změnách problému
- **Náhodné binární CSP (*random binary CSP*)**
  - parametry (n, m, p_1, p_2)
  - n počet proměnných
  - m počet hodnot v doméně proměnných
  - p_1 pravděpodobnost, že existuje omezení na páru proměnných
  - p_2 pravděpodobnost, že omezení povoluje daný pár hodnot

Aplikačně založené náhodné problémy

- Identifikace problémové domény
  - lze definovat parametrizovatelné problémy
  - problémy mají přitom specifickou (z aplikace vycházející) strukturu
  - problémy lze náhodně generovat s různým nastavením parametrů
- Výhody
  - zaměřené na reálné problémy
  - generování řady problémů umožňuje statistické porovnání
- Příklad: **shop scheduling** problémy
  - m strojů
  - n úloh, každá úloha se skládá z m operací prováděných na odlišných strojích
  - operace jedné úlohy nesmí být prováděny zároveň
  - podmínky na sekvencování operací úlohy (žádné, dáno pořadí, stejné pořadí pro všechny)
  - minimalizace dokončení poslední úlohy, minimalizace největšího zpoždění úlohy, . . .

Fáze přechodu

- Náhodný k-SAT problém
  - formule pevné délky jsou generovány výběrem m klauzulí
  - každá klauzule délky k je uniformně náhodně generována z množiny všech klauzulí
- **Obtížnost nalezení řešení**
  - při malém počtu klauzulí

Fáze přechodu

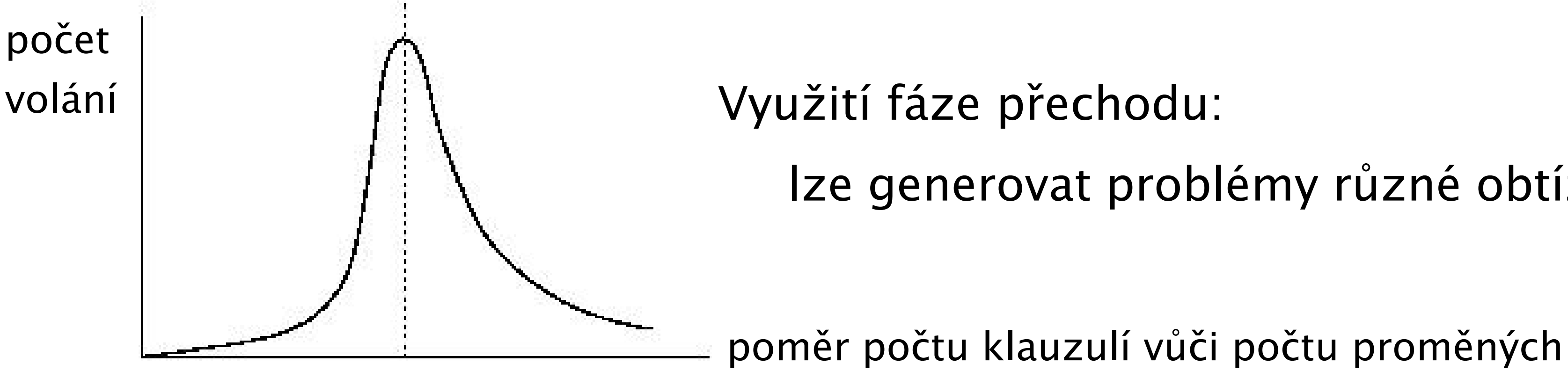
- Náhodný k-SAT problém
  - formule pevné délky jsou generovány výběrem m klauzulí
  - každá klauzule délky k je uniformně náhodně generována z množiny všech klauzulí
- **Obtížnost nalezení řešení**
  - při malém počtu klauzulí je většina problémů splnitelná a snadno řešitelná
  - při velkém počtu klauzulí

Fáze přechodu

- Náhodný k-SAT problém
  - formule pevné délky jsou generovány výběrem m klauzulí
  - každá klauzule délky k je uniformně náhodně generována z množiny všech klauzulí
- **Obtížnost nalezení řešení**
  - při malém počtu klauzulí je většina problémů splnitelná a snadno řešitelná
  - při velkém počtu klauzulí je detekována snadno nesplnitelnost většiny problémů
  - nalezení řešení je nejobtížnější za předpokladu, že cca 50 % problémů je splnitelných

Fáze přechodu

- Náhodný k-SAT problém
  - formule pevné délky jsou generovány výběrem m klauzulí
  - každá klauzule délky k je uniformně náhodně generována z množiny všech klauzulí
- **Obtížnost nalezení řešení**
  - při malém počtu klauzulí je většina problémů splnitelná a snadno řešitelná
  - při velkém počtu klauzulí je detekována snadno nesplnitelnost většiny problémů
  - nalezení řešení je nejobtížnější za předpokladu, že cca 50 % problémů je splnitelných
- Fenomén **fáze předchodu (*phase transition*)**
  - fáze přechodu z obtížně řešitelných problémů na snadno řešitelné problémů



Využití fáze přechodu:

lze generovat problémy různé obtížnosti

Optimalizace & soft omezení: modely

Optimalizační problém s podmínkami (COP)

- Problém splňování podmínek (X, D, C)
  - proměnné $X = \{x_1, \ldots, x_n\}$
  - domény $D = \{D_1, \ldots, D_n\}$
  - omezení $C = \{C_1, \ldots, C_n\}$
    - C_i je definováno na $Y_i \subseteq X$, $Y_i = \{x_{i_1}, \ldots, x_{i_k}\}$
    - C_i je podmnožina $D_{i_1} \times \ldots \times D_{i_k}$

Optimalizační problém s podmínkami (COP)

- Problém splňování podmínek (X, D, C)
  - proměnné $X = \{x_1, \ldots, x_n\}$
  - domény $D = \{D_1, \ldots, D_n\}$
  - omezení $C = \{C_1, \ldots, C_n\}$
    - C_i je definováno na $Y_i \subseteq X$, $Y_i = \{x_{i_1}, \ldots, x_{i_k}\}$
    - C_i je podmnožina $D_{i_1} \times \ldots \times D_{i_k}$
- **Objektivní funkce** $obj : Sol \rightarrow W$
- Základní definice:

  Optimalizační problém s podmínkami (*constraint optimization problem*)

  - nalezení řešení $\vec{d}$ pro (X, D, C) takové, že $obj(\vec{d})$ je optimální
    - optimální $\equiv$ maximální nebo minimální

COP: operační výzkum

- **Pevné (hard, required) omezení** $C_h = \{C_1, \ldots, C_n\}$ relace
  - C_i je definováno na $Y_i \subseteq X$, $Y_i = \{x_{i_1}, \ldots, x_{i_k}\}$
  - C_i je podmnožina $D_{i_1} \times \ldots \times D_{i_k}$
- **Měkké (soft) omezení** $C_s = \{F_1, \ldots, F_l\}$ funkce
  - F_j je definované nad $Q_j \subseteq X$, $Q_j = \{x_{j_1}, \ldots, x_{j_l}\}$
  - F_j je funkce do reálných čísel $D_{j_1} \times \ldots \times D_{j_l} \to \mathbb{R}^+$
- **Optimalizační problém s podmínkami (COP):** (X, D, C_h, C_s)

COP: operační výzkum

- **Pevné (hard, required) omezení** $C_h = \{C_1, \dots, C_n\}$ relace
  - C_i je definováno na $Y_i \subseteq X$, $Y_i = \{x_{i_1}, \dots, x_{i_k}\}$
  - C_i je podmnožina $D_{i_1} \times \ldots \times D_{i_k}$
- **Měkké (soft) omezení** $C_s = \{F_1, \dots, F_l\}$ funkce
  - F_j je definované nad $Q_j \subseteq X$, $Q_j = \{x_{j_1}, \dots, x_{j_l}\}$
  - F_j je funkce do reálných čísel $D_{j_1} \times \ldots \times D_{j_l} \to \mathbb{R}^+$
- **Optimalizační problém s podmínkami (COP):** (X, D, C_h, C_s)
- **Objektivní funkce** zjednodušení na $\sum$

$$F(\vec{d}) = \sum_{j=1}^{l} F_j(\vec{d}[Q_j])$$

$\vec{d}[Q_j] \ldots$ projekce $\vec{d}$ na Q_j

COP: operační výzkum

- **Pevné (hard, required) omezení** $C_h = \{C_1, \ldots, C_n\}$ relace
  - C_i je definováno na $Y_i \subseteq X$, $Y_i = \{x_{i_1}, \ldots, x_{i_k}\}$
  - C_i je podmnožina $D_{i_1} \times \ldots \times D_{i_k}$
- **Měkké (soft) omezení** $C_s = \{F_1, \ldots, F_l\}$ funkce
  - F_j je definované nad $Q_j \subseteq X$, $Q_j = \{x_{j_1}, \ldots, x_{j_l}\}$
  - F_j je funkce do reálných čísel $D_{j_1} \times \ldots \times D_{j_l} \rightarrow \mathbb{R}^+$
- **Optimalizační problém s podmínkami (COP):** (X, D, C_h, C_s)
- **Objektivní funkce** zjednodušení na $\sum$

$$F(\vec{d}) = \sum_{j=1}^{l} F_j(\vec{d}[Q_j])$$

$\vec{d}[Q_j] \ldots$ projekce $\vec{d}$ na Q_j

- **Řešení COP:** $\vec{d^o}$ splňující všechna omezení z C_h tak, že

  $F(\vec{d^o}) = \max_{\vec{d}} F(\vec{d})$ nebo $F(\vec{d^o}) = \min_{\vec{d}} F(\vec{d})$

Použití soft omezení

- Problémy optimalizační, příliš podmíněné, špatně definované problémy, ...
- Fuzzy preference, pravděpodobnosti, ceny, váhy, úrovně požadavků,...
- **Příliš podmíněné problémy:** řešení CSP neexistuje

F_1 `Prednaska < Cviceni` **@ 10**	tj. `pokud Prednaska<Cviceni pak` F_1=0
	`pokud Prednaska`$\geq$`Cviceni pak` F_1=10
F_2 `Prednaska in 4..5` **@ 6**	tj. `pokud Prednaska`$\in$`{4,5} pak` F_2=0
	`pokud Prednaska`$\notin$`{4,5} pak` F_2=6
F_3 `Cviceni in 1..4` **@ 4**	

Použití soft omezení

- Problémy optimalizační, příliš podmíněné, špatně definované problémy, ...
- Fuzzy preference, pravděpodobnosti, ceny, váhy, úrovně požadavků,...
- **Příliš podmíněné problémy:** řešení CSP neexistuje

  F_1 `Prednaska < Cviceni` @ 10   `tj. pokud Prednaska<Cviceni pak` F_1=0
  `pokud Prednaska`$\geq$`Cviceni pak` F_1=10

  F_2 `Prednaska in 4..5` @ 6   `tj. pokud Prednaska`$\in$`{4,5} pak` F_2=0
  `pokud Prednaska`$\notin$`{4,5} pak` F_2=6

  F_3 `Cviceni in 1..4` @ 4

- **Problémy s nejistotou**
  - Je 0.7 nezbytné, abych přišla do středy. po..st 0.7, ct..ne 0
  - Je nezbytné, abych nepřišla příliš později než ve středu. po..st 0.7, ct 0.5, pa 0.3, so..ne 0

Použití soft omezení

- Problémy optimalizační, příliš podmíněné, špatně definované problémy, . . .
- Fuzzy preference, pravděpodobnosti, ceny, váhy, úrovně požadavků,. . .
- **Příliš podmíněné problémy:** řešení CSP neexistuje

  F_1 `Prednaska < Cviceni` @ 10 — tj. `pokud Prednaska<Cviceni pak` F_1=0
  `pokud Prednaska≥Cviceni pak` F_1=10

  F_2 `Prednaska in 4..5` @ 6 — tj. `pokud Prednaska∈{4,5} pak` F_2=0
  `pokud Prednaska∉{4,5} pak` F_2=6

  F_3 `Cviceni in 1..4` @ 4

- **Problémy s nejistotou**
  - Je 0.7 nezbytné, abych přišla do středy. po..st 0.7, ct..ne 0
  - Je nezbytné, abych nepřišla příliš později než ve středu. po..st 0.7, ct 0.5, pa 0.3, so..ne 0
- **Špatně definované problémy:** není zřejmé, která omezení definují CSP

  `Zitra = pekne` @ 80%

  `Zitra = zamraceno` @ 30%

Přístupy pro soft omezení

- Vybrané přístupy
  - základní: MAX-CSP, omezení s váhami, fuzzy omezení
  - zobecňující: omezení nad polookruhy (*semiring-based*)

Přístupy pro soft omezení

- Vybrané přístupy
  - základní: MAX-CSP, omezení s váhami, fuzzy omezení
  - zobecňující: omezení nad polookruhy (*semiring-based*)
- Rozlišení systémů na základě: (v závorkách popis pro CSP)
  - **omezení** – rozšíření klasického omezení (c = relace)
  - **problém** – rozšíření CSP (trojice (V, D, C))
  - **úroveň splnění** – jak přiřazení hodnot splňuje problém ($\bigwedge c\theta$)
  - **řešení** – které přiřazení je (optimálním) řešením (splňují všechna omezení)
  - **úroveň konzistence problému** – jak je možné nejlépe splnit problém
    tj. jak (optimální) řešení splňuje problém (true)

Omezení s váhami, MAX-CSP

- **Omezení s váhami**:
  - Váha/cena spojená s každým omezením
  - Omezení – dvojice $(c, w(c))$
  - Problém – trojice (V, D, C_w)
  - Úroveň splnění – funkce na množině přiřazení $\omega(\theta) = \sum_{\theta \models \neg c} w(c)$

    $\Rightarrow$ **součet vah nesplněných omezení**
  - Řešení – přiřazení θ s **minimální** $\omega(\theta)$
  - Úroveň konzistence – úroveň splnění řešení

Omezení s váhami, MAX-CSP

- **Omezení s váhami**:
  - Váha/cena spojená s každým omezením
  - Omezení – dvojice $(c, w(c))$
  - Problém – trojice (V, D, C_w)
  - Úroveň splnění – funkce na množině přiřazení $\omega(\theta) = \sum_{\theta \models \neg c} w(c)$

    $\Rightarrow$ **součet vah nesplněných omezení**
  - Řešení – přiřazení θ s **minimální** $\omega(\theta)$
  - Úroveň konzistence – úroveň splnění řešení

- **MAX-CSP (maximální CSP)**
  - Váha je rovna jedné $\Rightarrow$ **maximalizace počtu splněných omezení**

Omezení s váhami: příklad

```
Prednaska < Cviceni @ 10
Prednaska in 4..5 @ 6
Cviceni in 1..4 @ 4
```

- Přiřazení: σ = {Prednaska/3, Cviceni/4}
- Úroveň splnění σ odpovídá součtu vah nesplněných omezení při σ, tj. 6

Omezení s váhami: příklad

```
Prednaska < Cviceni @ 10
Prednaska in 4..5 @ 6
Cviceni in 1..4 @ 4
```

- Přiřazení: σ = {Prednaska/3, Cviceni/4}
- Úroveň splnění σ odpovídá součtu vah nesplněných omezení při σ, tj. 6
- Řešení: θ = {Prednaska/4, Cviceni/5}
- Úroveň splnění θ: 4
- Úroveň konzistence odpovídá úrovni splnění θ: 4

Fuzzy CSP

- **Fuzzy množiny**: příslušnost prvku k množině zadána číslem z intervalu $[0, 1]$
- **Fuzzy omezení:** fuzzy relace $\mu_c(d_1, \ldots, d_k) \in \langle 0, 1 \rangle$ udává **úroveň preference**

$D_{\mathrm{A}} = D_{\mathrm{B}} = \{1, 2, 3\}$

c1: A = 1 @(1,0.7) tj. pokud A=1, pak μ_{c1}=1
pokud A≠1, pak μ_{c1}=0.7

c2: min(abs(A - B)), abs(A - B) = 0 => @1   tj. μ_{c2}=1
= 1 => @0.5 tj. μ_{c2}=0.5
= 2 => @0.1 tj. μ_{c2}=0.1

c3: max(A + B) @(A + B)/10   tj. μ_{c3}=(A + B)/10

Fuzzy CSP

- **Fuzzy množiny**: příslušnost prvku k množině zadána číslem z intervalu $[0, 1]$
- **Fuzzy omezení:** fuzzy relace $\mu_c(d_1, \ldots, d_k) \in \langle 0, 1 \rangle$ udává **úroveň preference**

$D_{\mathrm{A}} = D_{\mathrm{B}} = \{1, 2, 3\}$

```
c1: A = 1 @(1,0.7)  tj. pokud A=1, pak μc1=1
                        pokud A≠1, pak μc1=0.7
c2: min( abs(A - B) ), abs(A - B) = 0 => @1   tj. μc2=1
                                  = 1 => @0.5 tj. μc2=0.5
                                  = 2 => @0.1 tj. μc2=0.1
c3: max(A + B) @(A + B)/10   tj. μc3=(A + B)/10
```

- **Fuzzy CSP** (X, D, C_f)
  - C_f je množina fuzzy omezení
  - X uspořádaná množina proměnných

Kombinace a projekce omezení

- **Projekce n-tic** $(d_1, \ldots, d_l) \downarrow^{Y}_{X}$ příklad: $(1, 2, 3, 4, 5) \downarrow^{(A,B,C,D,E)}_{(D,A,E)} = (4, 1, 5)$

Kombinace a projekce omezení

- **Projekce n-tic** $(d_1, \ldots, d_l) \downarrow^Y_X$ příklad: $(1,2,3,4,5) \downarrow^{(A,B,C,D,E)}_{(D,A,E)} = (4,1,5)$
- **Kombinace** $c = c_X \oplus c_Y$, $dom(c) = Z = X \cup Y$ c, c_X, c_Y omezení nad Z, X, Y

$$\mu_c(d_1, \ldots, d_k) = \min(\mu_{c_X}((d_1, \ldots, d_k) \downarrow^Z_X), \mu_{c_Y}((d_1, \ldots, d_k) \downarrow^Z_Y))$$

  - udává, jaká je úroveň splnění všech přiřazení Z vzhledem k c_X a c_Y
  - příklad (pokračování): kombinace c1 ⊕ c2 ⊕ c3 pro $(1,3)$

$$\mu_{c1 \oplus c2 \oplus c3}(1,3) = \min(\mu_{c1}((1)), \mu_{c2}((1,3)), \mu_{c3}((1,3))) = \min(1, 0.1, 0.4) = 0.1$$

Kombinace a projekce omezení

- **Projekce n-tic** $(d_1, \dots, d_l) \downarrow^{Y}_{X}$ příklad: $(1,2,3,4,5) \downarrow^{(A,B,C,D,E)}_{(D,A,E)} = (4,1,5)$
- **Kombinace** $c = c_X \oplus c_Y$, $dom(c) = Z = X \cup Y$ c, c_X, c_Y omezení nad Z, X, Y

$$\mu_c(d_1, \dots, d_k) = \min(\mu_{c_X}((d_1, \dots, d_k) \downarrow^{Z}_{X}), \mu_{c_Y}((d_1, \dots, d_k) \downarrow^{Z}_{Y}))$$

  - udává, jaká je úroveň splnění všech přiřazení Z vzhledem k c_X a c_Y
  - příklad (pokračování): kombinace c1 $\oplus$ c2 $\oplus$ c3 pro $(1,3)$

$$\mu_{c1\oplus c2\oplus c3}(1,3) = \min(\mu_{c1}((1)), \mu_{c2}((1,3)), \mu_{c3}((1,3))) = \min(1, 0.1, 0.4) = 0.1$$

- **Projekce** $c = c_Y \Downarrow_X$, $dom(c) = X, X \subseteq Y$ c, c_X, c_Y omezení nad X, X, Y

$$\mu_c(d_{x1}, \dots, d_{xk}) = \max_{((d_{y1},\dots,d_{yl})\in D_{y1}\times\cdots\times D_{yl})\wedge((d_{y1},\dots,d_{yl})\downarrow^{Y}_{X}=(d_{x1},\dots,d_{xk}))} \mu_{c_Y}(d_{y1}, \dots, d_{yl})$$

  - udává, jaká je úroveň splnění všech přiřazení X vzhledem k c_Y
  - příklad (pokračování): projekce c3 $\Downarrow_{(B)}$ na (1)

$$\mu_{c3\Downarrow_{(B)}}(1) = \max(\mu_{c3}(1,1), \mu_{c3}(2,1), \mu_{c3}(3,1)) = \max(0.2, 0.3, 0.4) = 0.4$$

Řešení fuzzy CSP

- **Úroveň splnění** přiřazení $(d_1, \ldots, d_n)$ dána jako $\mu_{\oplus C}(d_1, \ldots, d_n)$

Řešení fuzzy CSP

- **Úroveň splnění** přiřazení $(d_1, \ldots, d_n)$ dána jako $\mu_{\oplus C}(d_1, \ldots, d_n)$
- **Řešení** – přiřazení $(d_1, \ldots, d_n)$ takové, že

$$\max_{(d_1,\ldots,d_n)\in D_1\times\cdots\times D_n} \mu_{\oplus C}(d_1, \ldots, d_n) = \mathbb{C}(P_\mu)$$

Řešení fuzzy CSP

- **Úroveň splnění** přiřazení $(d_1, \ldots, d_n)$ dána jako $\mu_{\bigoplus C}(d_1, \ldots, d_n)$
- **Řešení** – přiřazení $(d_1, \ldots, d_n)$ takové, že

$$\max_{(d_1,\ldots,d_n) \in D_1 \times \cdots \times D_n} \mu_{\bigoplus C}(d_1, \ldots, d_n) = \mathbb{C}(P_\mu)$$

  - příklad: $\bigoplus C$ = c1 $\oplus$ c2 $\oplus$ c3 pro všechna (A, B)

    $(3,3) \ldots 0.6$, $(2,2) \ldots 0.4$, $(1,1) \ldots 0.2$

    $(2,3)$ a $(3,2) \ldots 0.5$, $(2,1)$ a $(1,2) \ldots 0.3$

    $(1,3)$ a $(3,1) \ldots 0.1$

Řešení fuzzy CSP

- **Úroveň splnění** přiřazení $(d_1, \ldots, d_n)$ dána jako $\mu_{\bigoplus C}(d_1, \ldots, d_n)$
- **Řešení** – přiřazení $(d_1, \ldots, d_n)$ takové, že

$$\max_{(d_1,\ldots,d_n) \in D_1 \times \cdots \times D_n} \mu_{\bigoplus C}(d_1, \ldots, d_n) = \mathbb{C}(P_\mu)$$

  - příklad: $\bigoplus C$ = c1 $\oplus$ c2 $\oplus$ c3 pro všechna (A, B)

    $(3,3)\ldots 0.6$, $(2,2)\ldots 0.4$, $(1,1)\ldots 0.2$

    $(2,3)$ a $(3,2)\ldots 0.5$, $(2,1)$ a $(1,2)\ldots 0.3$

    $(1,3)$ a $(3,1)\ldots 0.1$ $\Rightarrow$ $(3,3)$ je řešení, $\mathbb{C}(P_\mu) = 0.6$

Řešení fuzzy CSP

- **Úroveň splnění** přiřazení $(d_1, \ldots, d_n)$ dána jako $\mu_{\oplus C}(d_1, \ldots, d_n)$
- **Řešení** – přiřazení $(d_1, \ldots, d_n)$ takové, že

$$\max_{(d_1,\ldots,d_n)\in D_1\times\cdots\times D_n} \mu_{\oplus C}(d_1, \ldots, d_n) = \mathbb{C}(P_\mu)$$

  - příklad: $\bigoplus C$ = c1 $\oplus$ c2 $\oplus$ c3 pro všechna (A, B)

    $(3,3)\ldots 0.6$, $(2,2)\ldots 0.4$, $(1,1)\ldots 0.2$

    $(2,3)$ a $(3,2)\ldots 0.5$, $(2,1)$ a $(1,2)\ldots 0.3$

    $(1,3)$ a $(3,1)\ldots 0.1$ $\Rightarrow$ $(3,3)$ je řešení, $\mathbb{C}(P_\mu) = 0.6$

- **Úroveň nekonzistence** $1 - \mathbb{C}(P_\mu)$
- $\mathbb{C}(P_\mu)$ je **úroveň konzistence**

  také jako projekce na prázdnou množinu proměnných $\bigoplus C \Downarrow_\emptyset$

Příklad: řešení fuzzy CSP

- Viz dříve: $\bigoplus C = \mathrm{c1} \oplus \mathrm{c2} \oplus \mathrm{c3}$ pro všechna (A, B)

  $(3,3)\ \ldots 0.6,\ (2,2) \ldots 0.4,\ (1,1) \ldots 0.2,$

  $(2,3)$ a $(3,2)\ \ldots 0.5,\ (2,1)$ a $(1,2) \ldots 0.3,\ (1,3)$ a $(3,1) \ldots 0.1$

- $\bigoplus C = \bigoplus C \Downarrow_{\{\mathsf{A},\mathsf{B}\}}$

  $\bigoplus C \Downarrow_{\{\mathsf{A},\mathsf{B}\}} (3,3) = \max(\mu_{\bigoplus C}(3,3)) = 0.6,$

  $\bigoplus C \Downarrow_{\{\mathsf{A},\mathsf{B}\}} (2,2) = 0.4, \ldots$

Příklad: řešení fuzzy CSP

- Viz dříve: $\bigoplus C$ = c1 $\oplus$ c2 $\oplus$ c3 pro všechna (A, B)

  $(3,3)\ \ldots 0.6,\ (2,2) \ldots 0.4,\ (1,1) \ldots 0.2,$

  $(2,3)$ a $(3,2)\ \ldots 0.5,\ (2,1)$ a $(1,2) \ldots 0.3,\ (1,3)$ a $(3,1) \ldots 0.1$

- $\bigoplus C = \bigoplus C \Downarrow_{\{A,B\}}$

  $\bigoplus C \Downarrow_{\{A,B\}} (3,3) = \max(\mu_{\bigoplus C}(3,3)) = 0.6,$

  $\bigoplus C \Downarrow_{\{A,B\}} (2,2) = 0.4, \ldots$

- $\bigoplus C \Downarrow_{\{A\}}$

  $\bigoplus C \Downarrow_{\{A\}} (1) = \max(\mu_{\bigoplus C}(1,1), \mu_{\bigoplus C}(1,2), \mu_{\bigoplus C}(1,3)) = \max(0.2, 0.3, 0.1) = 0.3$

  $\bigoplus C \Downarrow_{\{A\}} (2) = \max(\mu_{\bigoplus C}(2,1), \mu_{\bigoplus C}(2,2), \mu_{\bigoplus C}(2,3)) = \max(0.3, 0.4, 0.5) = 0.5$

  $\bigoplus C \Downarrow_{\{A\}} (3) = \max(\mu_{\bigoplus C}(3,1), \mu_{\bigoplus C}(3,2), \mu_{\bigoplus C}(3,3)) = \max(0.1, 0.5, 0.6) = 0.6$

Příklad: řešení fuzzy CSP

- Viz dříve: $\bigoplus C$ = c1 $\oplus$ c2 $\oplus$ c3 pro všechna (A, B)

  $(3,3)\ldots 0.6$, $(2,2)\ldots 0.4$, $(1,1)\ldots 0.2$,

  $(2,3)$ a $(3,2)\ldots 0.5$, $(2,1)$ a $(1,2)\ldots 0.3$, $(1,3)$ a $(3,1)\ldots 0.1$

- $\bigoplus C = \bigoplus C \Downarrow_{\{A,B\}}$

  $\bigoplus C \Downarrow_{\{A,B\}} (3,3) = \max(\mu_{\bigoplus C}(3,3)) = 0.6,$

  $\bigoplus C \Downarrow_{\{A,B\}} (2,2) = 0.4, \ldots$

- $\bigoplus C \Downarrow_{\{A\}}$

  $\bigoplus C \Downarrow_{\{A\}} (1) = \max(\mu_{\bigoplus C}(1,1), \mu_{\bigoplus C}(1,2), \mu_{\bigoplus C}(1,3)) = \max(0.2, 0.3, 0.1) = 0.3$

  $\bigoplus C \Downarrow_{\{A\}} (2) = \max(\mu_{\bigoplus C}(2,1), \mu_{\bigoplus C}(2,2), \mu_{\bigoplus C}(2,3)) = \max(0.3, 0.4, 0.5) = 0.5$

  $\bigoplus C \Downarrow_{\{A\}} (3) = \max(\mu_{\bigoplus C}(3,1), \mu_{\bigoplus C}(3,2), \mu_{\bigoplus C}(3,3)) = \max(0.1, 0.5, 0.6) = 0.6$

- $\bigoplus C \Downarrow_{\emptyset}$

  $$\bigoplus C \Downarrow_{\emptyset} () = \max(\mu_{\bigoplus C}(1,1), \mu_{\bigoplus C}(1,2), \mu_{\bigoplus C}(1,3), \mu_{\bigoplus C}(2,1), \mu_{\bigoplus C}(2,2),$$
  $$\mu_{\bigoplus C}(2,3), \mu_{\bigoplus C}(3,1), \mu_{\bigoplus C}(3,2), \mu_{\bigoplus C}(3,3)) = 0.6$$