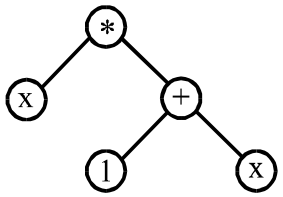
5. prednáška

Genetické programovanie (GP)

Na prelome 80-tých a 90-tých rokov americký informatik John Koza [1,2] (Stanford University) navrhol originálnu modifikáciu genetického algoritmu, ktorú nazval **genetické programovanie**. V tomto prístupe sú chromozómy - znakové reťazce nahradené zložitejšími štruktúrami - funkciami.



Nech $A=\{(x_i,y_i);\ i=1,2,...,p\}$ je **tréningová množina** obsahujúca p bodov (x_i,y_i). Naším cieľom je nájsť takú funkciu $t(x)$ (reprezentovanú syntaktickým stromom), ktorá minimalizuje rozdiel (jeho kvadrát alebo absolútnu hodnotu) vypočítaných a zadaných hodnôt y z tréningovej množiny.

Dva typy regresií

(1) **Parametrická regresia** (modelová funkcia je fixná, optimalizujú sa len jej parametre)

(2) **Symbolická regresia** (Koza, optimalizuje sa tvar modelovej funkcie)

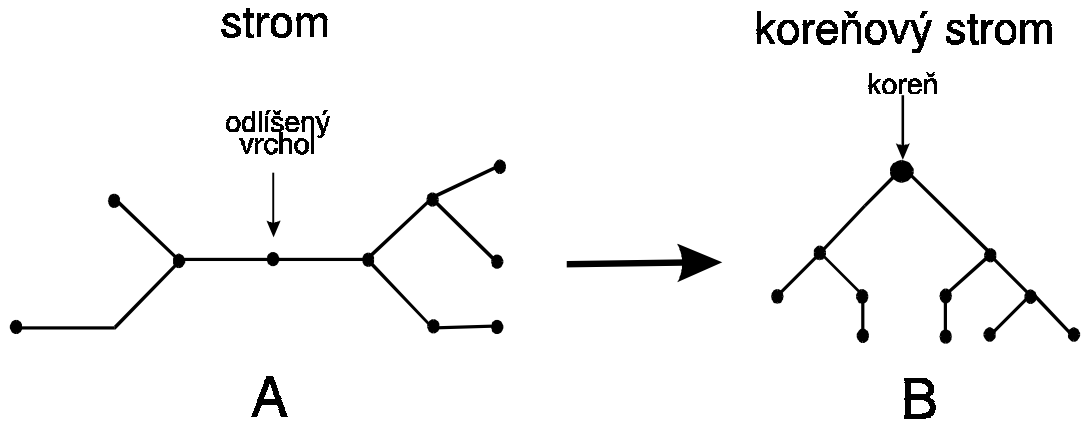
Symbolická regresia je len jedna možná interpretácia genetického programovania.

Iný pohľad na genetické programovanie je chápanie funkcie $t(x)$ ako "programu" pre správanie sa nejakého objektu (agenta - robota) v prostredí.

Koreňovéstromy

Nech $\boldsymbol{G}=(V,E)$ je *strom* (súvislý acyklický graf [3]), kde $V=\{v_1,v_2,...,v_p\}$ je neprázdna *vrcholová množina* a $E=\{e_1,e_2,...,e_q\}$ je *hranová množina*,

$$|V|=|E|+1$$



Ak jeden vrchol v strome je odlíšený od ostatných vrcholov, potom strom sa nazýva **koreňový strom** a odlíšený vrchol sa nazýva **koreň.**

Valentnosť $val(v)$ je nezáporné celé číslo, ktoré určuje počet hrán incidentných s vrcholom v.

$$\sum_{v \in V} val(v) = 2q = 2(p-1)$$

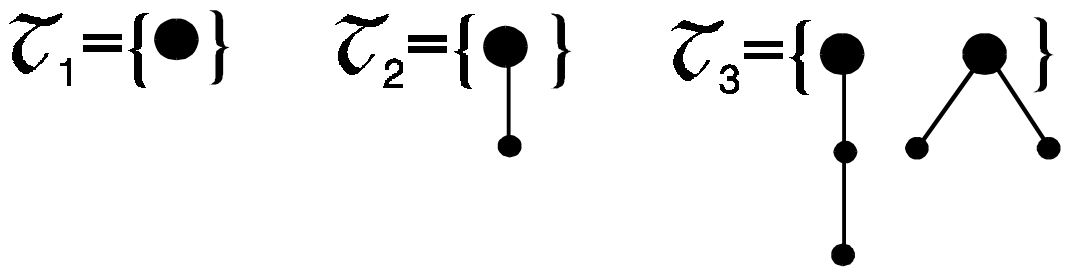
Koreňový strom je formálne určený ako usporiadaná trojica

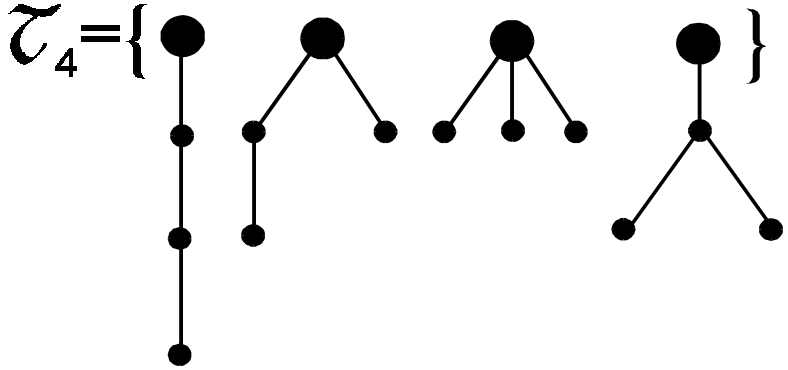
$$T = (V, E, v)$$

kde $v \in V$ je koreň.

Nech $\mathcal{T}_i$ je množina obsahujúca všetky možné koreňové stromy, ktoré sú zložené z i vrcholov

$$\mathcal{T} = \mathcal{T}_1 \cup \mathcal{T}_2 \cup \ldots$$





Nech koreňový strom z $\mathcal{T}$ je ohodnotený reálnym číslom

$$t : \mathcal{T} \to R$$

Funkcia t ohodnotí každý koreňový strom reálnym číslom, ktoré vyjadruje v určitom priblížení "topológiu" koreňového stromu.

Požadovaná hodnota indexu nech je označená t_{req} , potom môžeme definovať **účelovú funkciu**

$$f(T) = | t(T) - t_{req} |$$

Nulová hodnota účelovej funkcie odpovedá globálnemu minimu nad priestorom všetkých možných koreňových stromov,

$$T_{opt} = \arg \min_{T \in \mathcal{T}} f(T)$$

Koreňový strom $\boldsymbol{T}_{opt}$ odpovedá takému koreňovému stromu, ktorý minimalizuje účelovú funkciu nad celým priestorom koreňových stromov T.

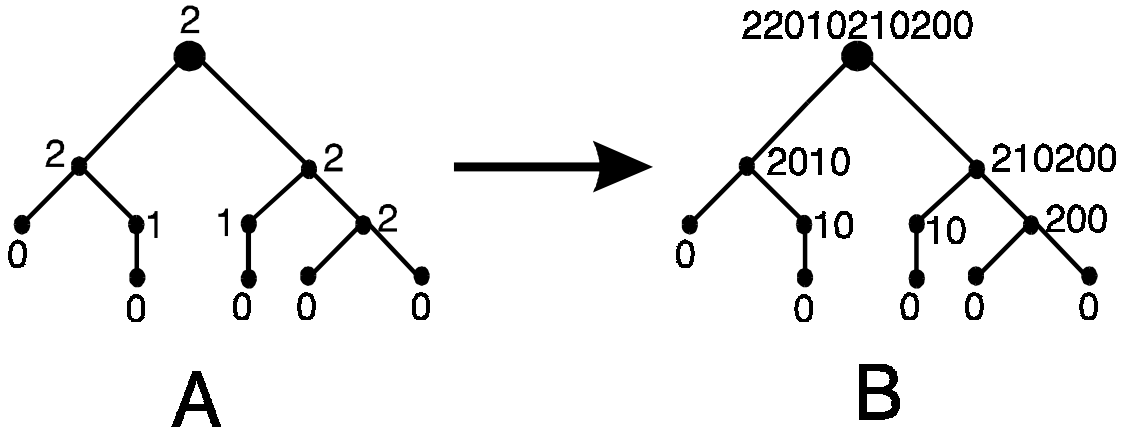
Readov lineárny kód

Readov kód je často používaná v teórii grafov pre konštruktívnu enumeráciu stromových štruktúr.

Readov kód code(***T***) je reťazec (sekvencia) celých čísel

$$\alpha = \left(\alpha_1, \alpha_2, ..., \alpha_p\right)$$

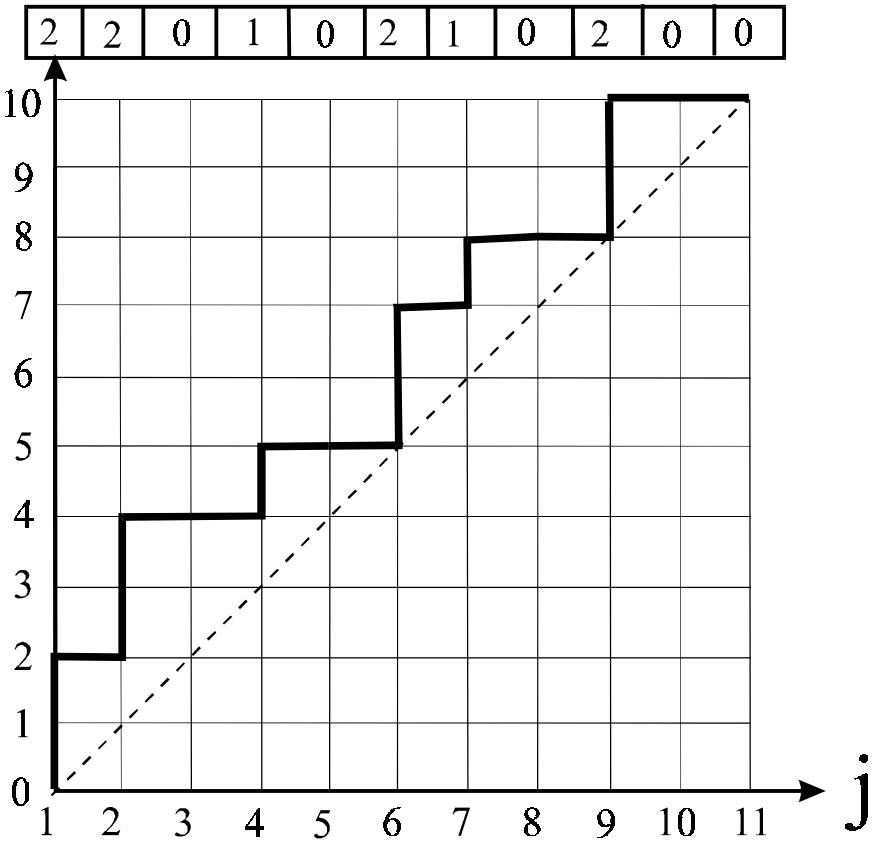
ktoré sú priradené buď valencii koreňa alebo valencii zníženej o jednotku pre ostatné vrcholy



Veta. Nutné a postačujúce podmienky k tomu, aby postupnosť nezáporných celých čísel $(\alpha_1, \alpha_2, ..., \alpha_p) \in \{0,1,2,...\}^p$ bola grafová (t.j. existuje koreňový strom s rovnakým kódom) sú

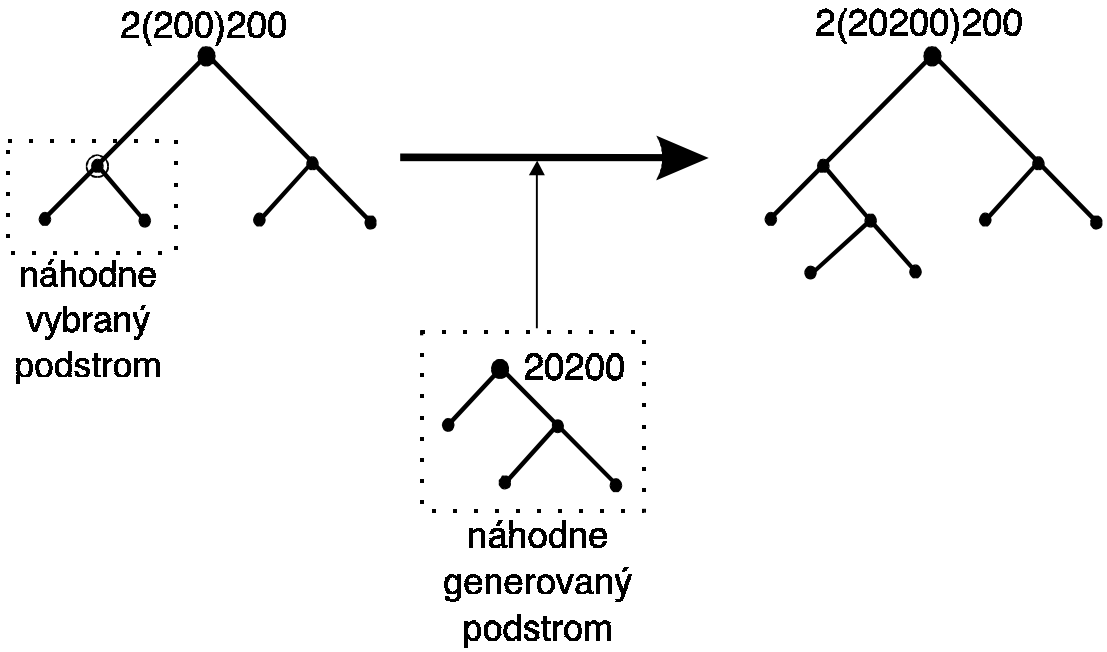
$$\sum_{i=1}^{j} \alpha_i \geq j \quad (j = 1,2,..., p-1)$$

$$\sum_{i=1}^{p} \alpha_i = p - 1$$



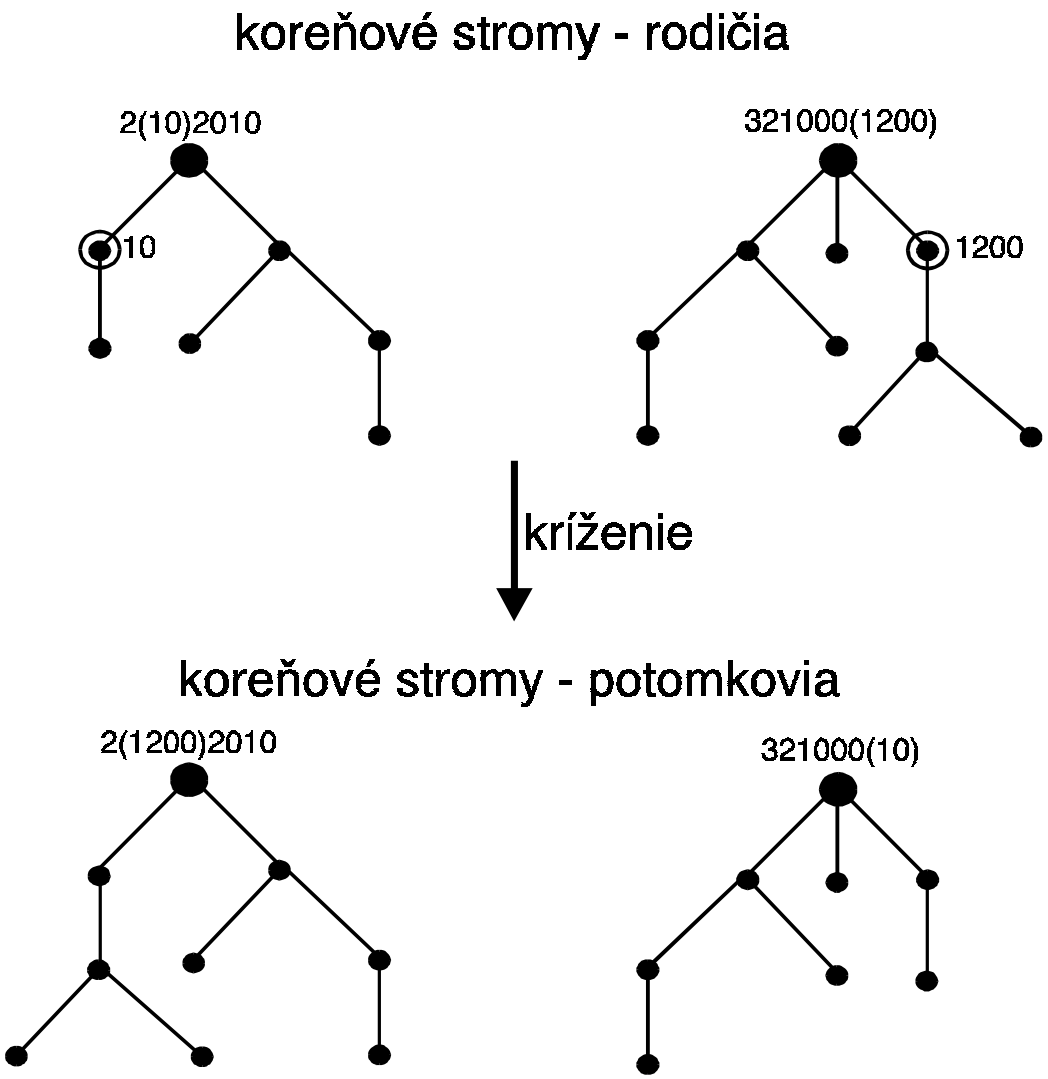
Mutácia Readovho kódu

$$\alpha' = O_{mut}(\alpha)$$



Križemie Readovho kódu

$$(\alpha',\beta') = O_{cross}(\alpha,\beta)$$

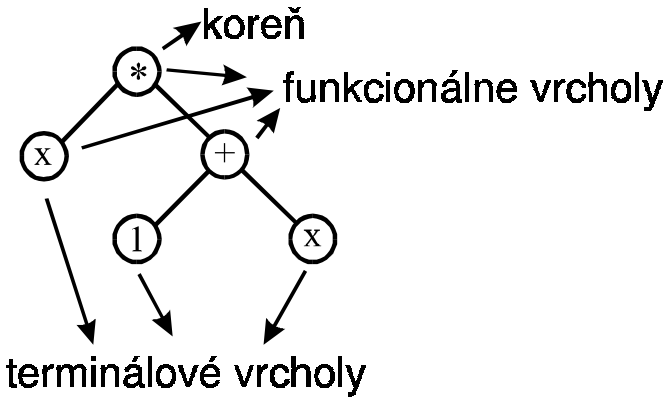


Symbolická regresia

Symbolická regresia patrí medzi základné aplikácie genetického programovania. Ako už bolo spomenuté v úvodnej kapitole 5.1, symbolická regresia spočíva v hľadaní takej funkcie reprezentovanej syntaktickým stromom s predpísanými operáciami, ktorá čo najlepšie aproximuje údaje z treningovej množiny.

Syntaktický strom *t* je koreňový strom *T*, ktorého vrcholy sú ohodnotené symbolmi aritmetických (alebo iných) operácií a celý strom môže byť ohodonotený reálnym číslom reprezentujúcim hodnotu funkcie, ktorá je priradená stromu pre danú vstupnú hodnotu nezávislej premennej (alebo nezávislých premenných).

Klasifikácia vrcholov syntaktického stromu



(1) **Terminálne vrcholy**, tieto vrcholy odpovedajú buď *nezávislým premenným* x,y,... nezáporným celočíselným *konštantám* 0,1,2,....

(2) **Funkcionálne vrcholy** odpovedajú jednoduchým operáciam, ktoré sú unárne, binárne, ternárne,... .

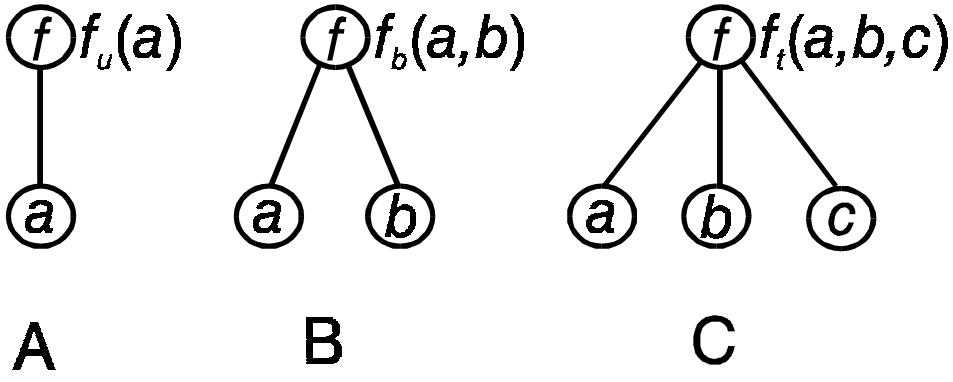
Formálne môžeme syntaktický strom vyjadriť ako usporiadanú dvojicu

$$\boldsymbol{t} = (\boldsymbol{T}, \phi)$$

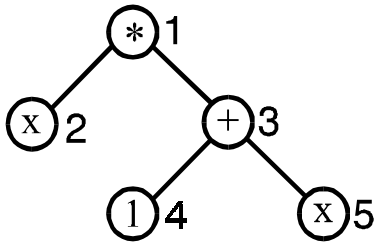
kde $\boldsymbol{T}$ je koreňový strom určujúci štruktúru syntaktického stromu $\boldsymbol{t}$ a ϕ je zobrazenie vrcholov koreňového stroma na "aritmetické" symboly, premenné, alebo konštanty

$$\phi : V \rightarrow \{*, +, -, ..., x, y, ..., 1, 2, ...\}$$

Klasifikácia funkcionálnych vrcholov na unárne (A), binárne (B) a terciárne (D)



Každému syntaktickému stromu ***t*** priradíme funkciu *t(x,y,....)*, kde *x,y,....* sú nezávislé premenné nasledujúcim rekurzívnym spôsobom

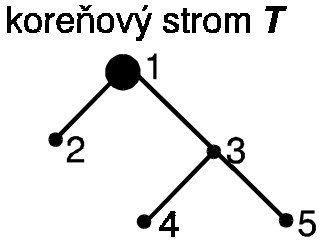


$x_5=x$
$x_4=1$
$x_3=x_4+x_5=1+x$
$x_2=x$
$x_1=t(x)=x_2*x_3=x*(1+x)$

Readov kód syntaktického stromu

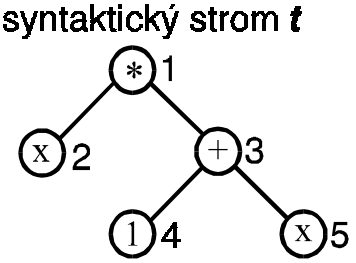
$$\mathrm{code}(t)=\left((\alpha_1,\phi_1),(\alpha_2,\phi_2),...,(\alpha_p,\phi_p)\right)$$

kde ϕ_i je ohodnotenie *i*-teho vertexu buď funkcionálnym alebo terminálovým symbolom.



code(*T*)=(20200)

A



code(*t*)=((2,*),(0,x),(2,+),(0,1),(0,x))

B

Tréningová množina obsahuje n bodov (regresnú tabulku)

$$A_{train} = \{x_i / y_i ; i = 1,2,...,n\}$$

Cieľom štandardne regresnej analýzy je nájsť také optimálne parametre modelovej funkcie $G(x;w)$, kde **w** sú parametre funkcie G, také, že nasledujúca účelová funkcia je minimalizovaná

$$E(w) = \sum_{i=1}^{n} \left| G(x_i; w) - y_i \right|$$

Táto funkcia má minimum v bode

$$w_{opt} = \arg\min_{w} E(w)$$

Hovoríme, že adaptovaná funkcia $G(\boldsymbol{x}, w_{opt})$ modeluje tréningovú množinu A_{train}

Symbolická regresia hladá v množine T takú funkciu, že nasledujúca účelová funkcia (funkcionál) je minimalizovaná

$$E(t) = \sum_{i=1}^{n} \left| t(x_i) - y_i \right|$$

pričom táto účelová funkcia má minimum v "bode"

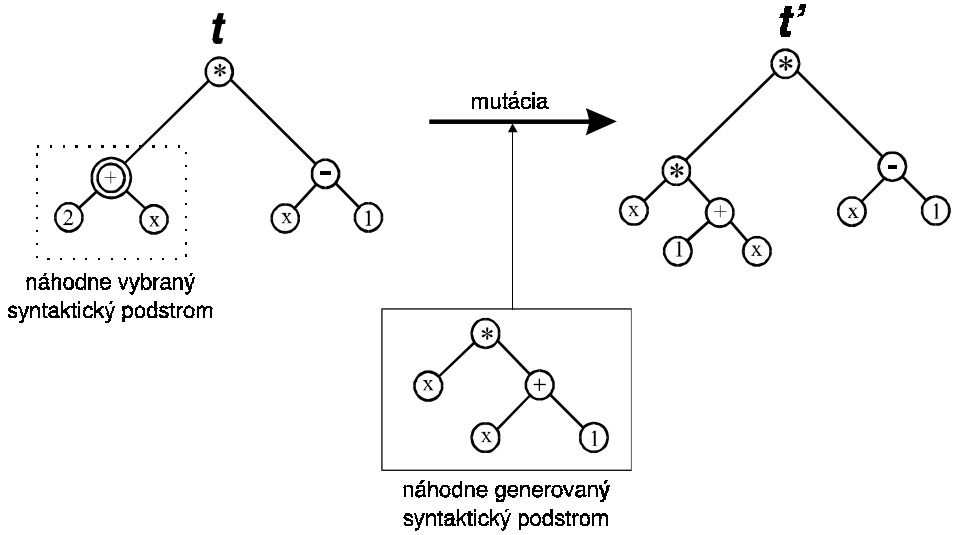
$$t_{opt} = \arg \min_{t \in \tau} E(t)$$

Symbolická regresia je hlavným cielom Kozovho genetického programovania s reštrikciou, že povolené funkcie sú len tie, ktoré môžu byť reprezentované syntaktocým stromom obsahujúcim povolené fukcionálne a terminálové vrcholy.

Riešenie minimalizačného problému sa realizuje pomocou genetického algoritmu. Stochastické operácie mutácie a kríženia sú definované podobným spôsobom ako pre koreňové stromy.

Mutácia

$$t' = O_{mut}(t)$$



Mutáciou sa funkcia *t(x)* priradená pôvodnému stromu zamenila za novú funkciu *t′(x)* .

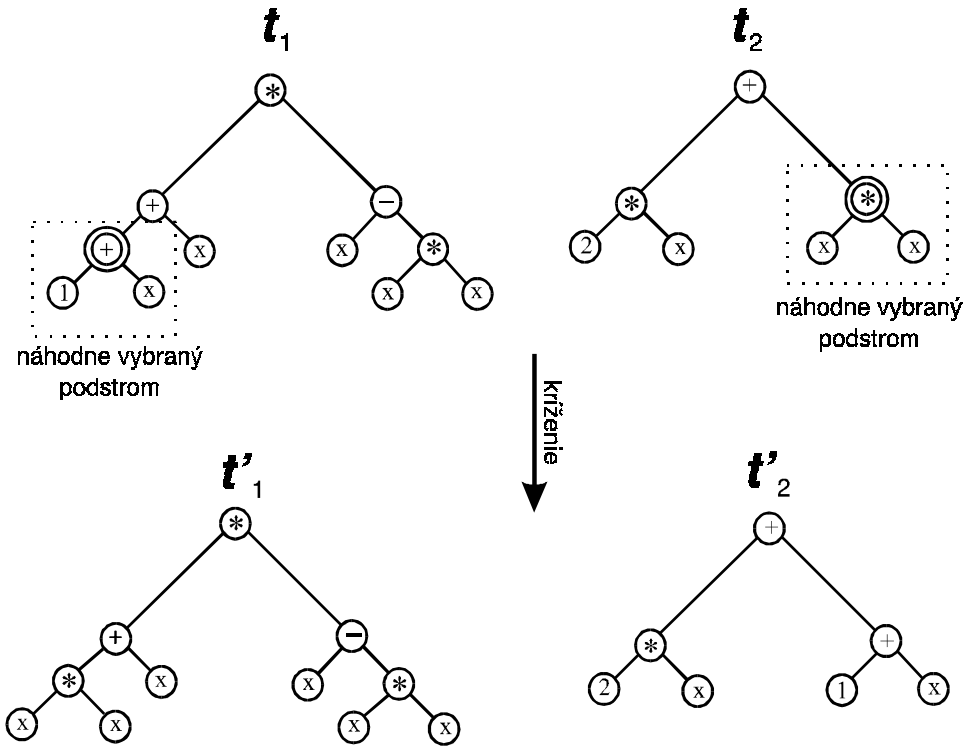
$$t(x) = (2 + x) * (x - 1)$$

$$t'(x) = (x * (1 + x)) * (x - 1)$$

Kríženie

Krížením dva stromy t_1 a t_2 sú transformované na dva nové stromy t_1' a t_2'

$$(t_1', t_2') = O_{cross}(t_1, t_2)$$



$$t_1(x) = (1+2x) * (x - x^2)$$

$$t_2(x) = 2x + x^2$$

$$t_1'(x) = (x^2 + x) * (x - x^2)$$

$$t_2'(x) = 2x + (1+x)$$

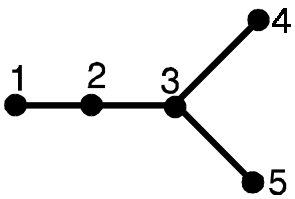
Rekonštrukcia stromov s požadovanou vlastnosťou

Vlastnosti stromov sú popísané topologickými indexami.

Wienerov topologický index

$$\chi_W(\boldsymbol{G}) = \sum_{i<j} d_{ij}$$

kde sumácia obsahuje všetky rôzne dvojice vrcholov a symbol d_{ij} je vzdialenosť medzi i-tým a j-tým vrcholom v grafe $\boldsymbol{G}$



A

$$D = \begin{pmatrix} 0 & & & & \\ 1 & 0 & & & \\ 3 & 2 & 1 & 0 & \\ 3 & 2 & 1 & 2 & 0 \end{pmatrix} \begin{matrix} \ldots 0 \\ \ldots 1 \\ \ldots 6 \\ \ldots 8 \end{matrix}$$

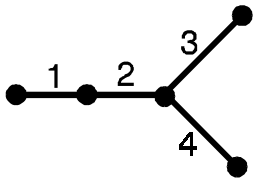
$\chi_W = 18$

B

Randičov topologický index

$$\chi_R(\boldsymbol{G}) = \sum_{[v,v']\in E} \frac{1}{\sqrt{\mathrm{val}(v)\cdot \mathrm{val}(v')}}$$

kde sumácia obsahuje všetky hrany $[v,v']\in E(\boldsymbol{G})$



A

$$\chi_R = \overset{e_1}{\frac{1}{\sqrt{1\cdot 2}}} + \overset{e_2}{\frac{1}{\sqrt{2\cdot 3}}} + \overset{e_3}{\frac{1}{\sqrt{1\cdot 3}}} + \overset{e_4}{\frac{1}{\sqrt{1\cdot 3}}}$$

B

Úloha rekonštrukcie spočíva v tom, že hľadáme v množine prípustných stromov $\mathcal{T}$ taký strom $\boldsymbol{G}$, ktorého vlastnosť $t(\boldsymbol{G})$ je blízka požadovanej vlastnosti t_{req}.

$$E(G) = \left| t(G) - t_{req} \right|$$

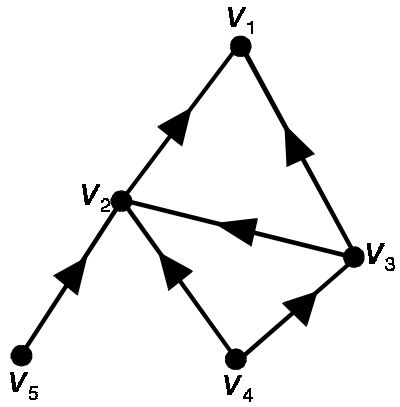
Optimálny strom, ktorého vlastnosť minimalizuje funkcionál $E(G)$ je určený ako riešenie nasledujúceho minimalizačného problému

$$G_{opt} = \arg \min_{\boldsymbol{G} \in \mathcal{G}} E(G)$$

Riešenie tohto optimalizačného problému sa realizuje pomocou genetického algoritmu nad populáciou stromov, ktoré sú kódované pomocou Readovho lineárneho kódu a operácie mutácie a kríženia sa vykonávajú spôsobom popísaným nižšie.

Kódovanie funkcií pomocou acyklických orientovaných grafov

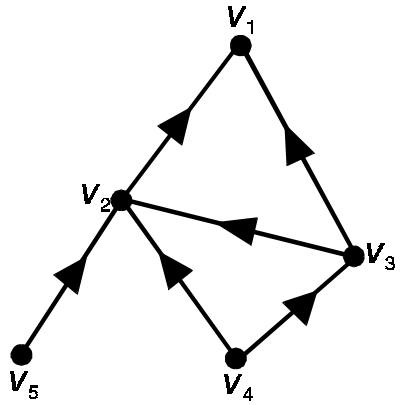
Alternatívny prístup ku kódovaniu funkcií pomocou acyklických orientovaných grafov, ktoré môžu byť chápané ako zovšeobecnenie koreňových stromov.



Orientovaný graf (hrany grafu sú orientované) $\boldsymbol{G}=(V,E)$, kde $V=\{v_1,v_2,...,v_p\}$ je neprázdna množina vrcholov a $E=\{e_1,e_2,...,e_q\}$ je množina hrán. Orientovaný graf sa nazýva *acyklický* vtedy, ak neexistuje orientovaná cyklická cesta obsahujúca postupnosť rovnako orientovaných hrán.

Nech orientovaný graf $\boldsymbol{G}$ je indexovaný, matica susednosti $\boldsymbol{A}=(A_{ij})\in\{0,1\}^{p\times p}$ je určená takto

$$A_{ij}=\begin{cases}1 & \left(\text{pre}\left(v_i,v_j\right)\in E\right)\\ 0 & \left(\text{iná\v{c}}\right)\end{cases}\quad\cdots$$



$$A=\begin{pmatrix}0&0&0&0&0\\1&0&0&0&0\\1&0&0&0&0\\0&1&1&0&0\\0&1&0&0&0\end{pmatrix}$$

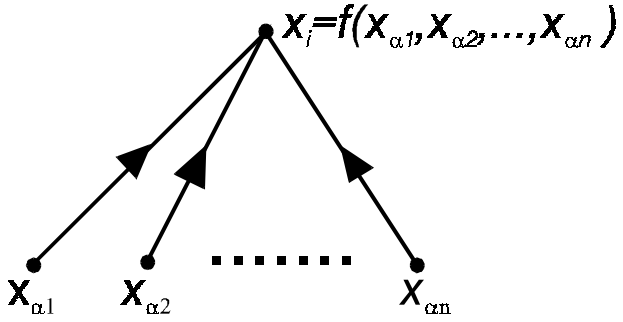
Veta. Orientovaný graf $G=(V,E)$ je acyklický vtedy a len vtedy, ak jeho vrcholy môžu byť kanonicky indexované tak, že platí

$$\forall (v,v') \in E : \varphi(v) > \varphi(v')$$

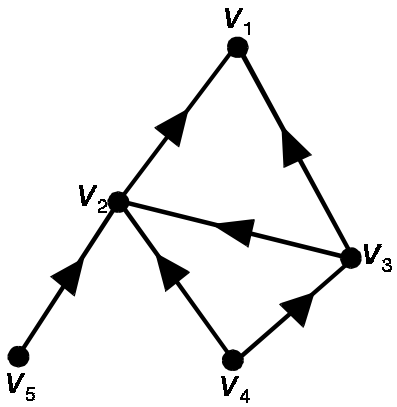
Vrcholy kanonicky indexovaného grafu môžu byť rozdelené na tri disjunktné množiny:

(1) *Vstupné vrcholy*, tieto vrcholy susedia len s vychádzajúcimi hranami,
(2) *Prechodné vrcholy*, tieto vrcholy súčasne susedia tak s vzchádzajúcim, ako aj s vchádzajúcou hranou.
(3) *Výstupné vrcholy*, tieto vrcholy susedia len s vchádzajúcimi hranami.

Kanonicky indexovaný orientovaný acyklický graf, ktorý má jeden výstupný vrchol, sa nazýva **syntaktický graf**.

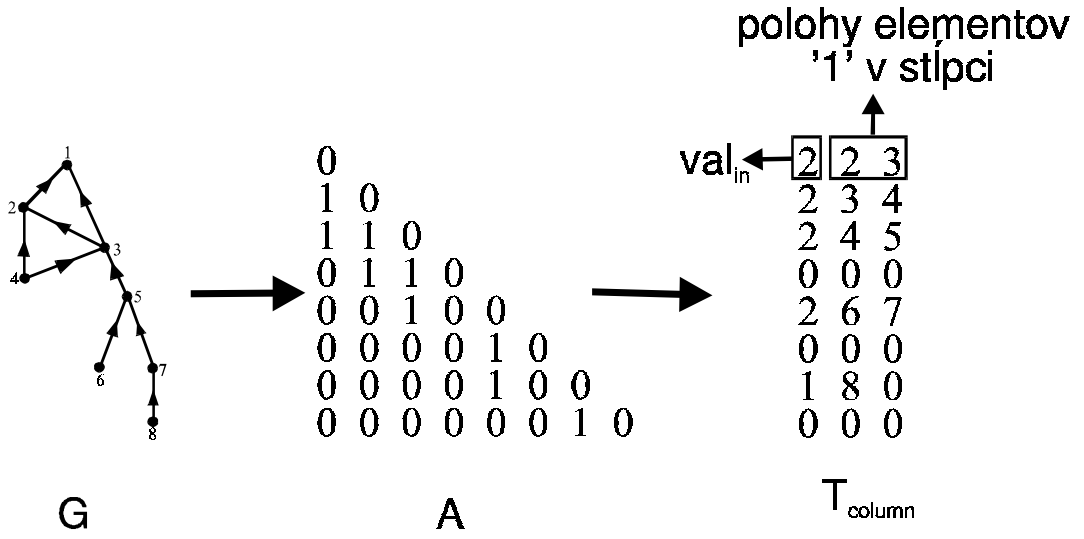


Každý prechodný alebo výstupný vrchol je ohodnotený reálnym číslom, ktoré je určené ako hodnota funkcie priradenej vrcholu, argumenty tejto funkcie sú funkčné hodnoty vrcholov spojené s uvažovaným vrcholom

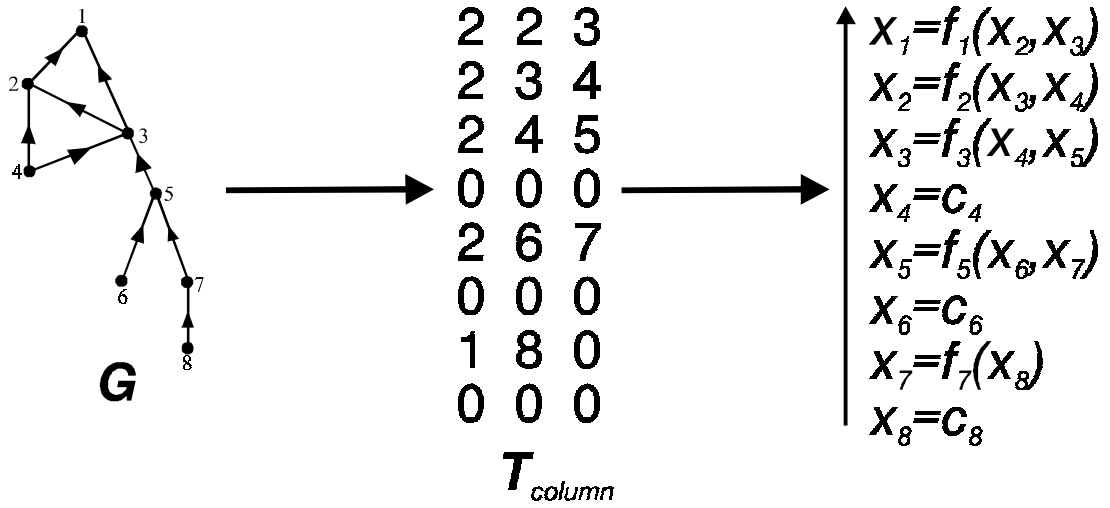


$$x_5 = c_1,\ x_4 = c_2,\ x_3 = f_3\left(x_4\right),\ x_2 = f_2\left(x_3, x_4, x_5\right),$$
$$x_1 = f_1\left(x_2, x_3\right)$$

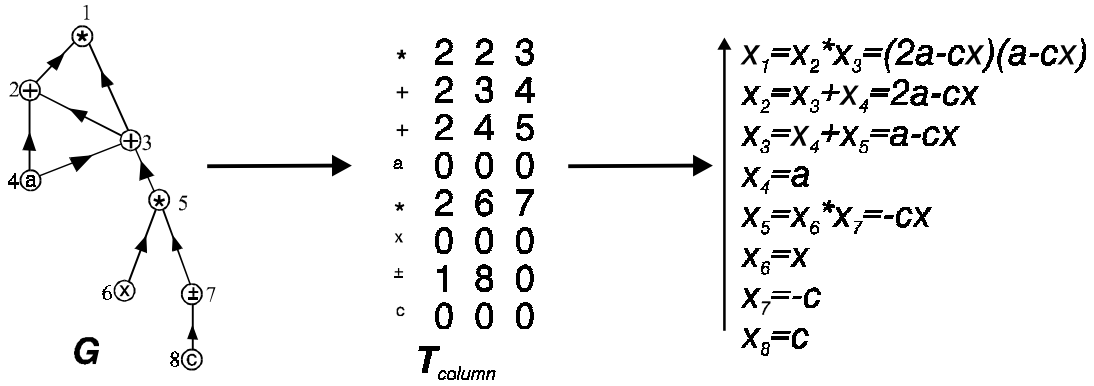
Kódovanie syntaktických grafov



Ilustračný príklad kódovania syntaktického grafu pomocou stĺpcovej tabuľky. Syntaktický graf *G* je kódovaný maticou susednosti *A*, táto matica je v ďalšom kroku "kondenzovaná" do tvaru stĺpcovej tabuľky T_{column}.

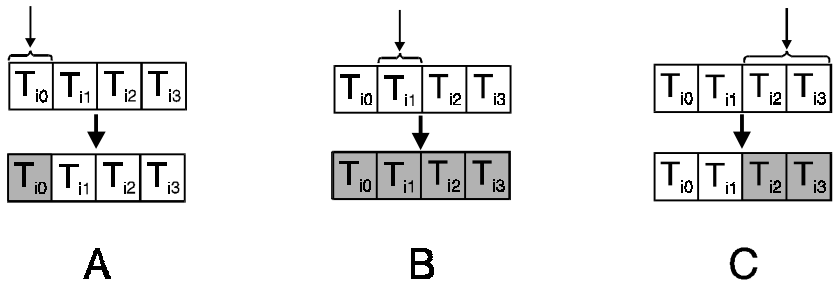


Syntaktický graf ***G*** určený stĺpcovou tabuľkou $\boldsymbol{T}_{column}$ umožňuje jednoduchý rekurentný výpočet funkčnej hodnoty výstupného vrcholu.



Ilustračný príklad výpočtu funkčných hodnôt syntaktického grafu *G* ktorého jednotlivé vrcholy sú ohodnotené algebraickými operáciami. Stĺpcová tabuľka ***T*** v tomto prípade je rozšírená o nový 0-tý stĺpec, ktorý špecifikuje funkcie priradené jednotlivým vrcholom. Výsledné ohodnotenie syntaktického grafu je funkčná hodnota výstupného vrcholu (2*a*-*cx*)(*a*-*cx*).

Mutácia



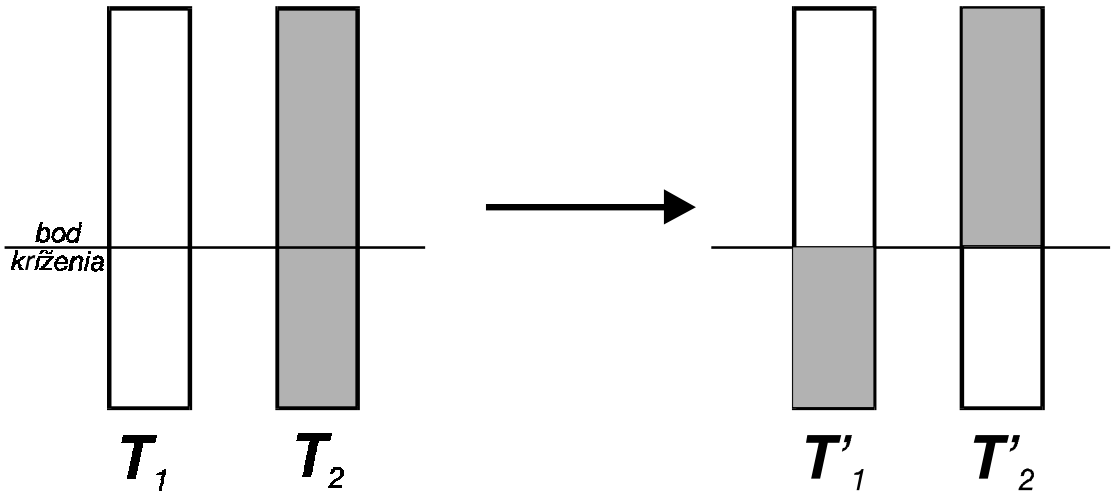
(A) Mutuje sa 0-tý element riadku, ktorý popisuje typ funkcie.
(B) Mutuje sa 1-vý element riadku popisujúci vstupnú valenciu odpovedajúceho vrcholu.
(C) mutujú sa elementy riadku, ktoré popisujú predchodcov vrcholu.

$$\boldsymbol{T}' = O_{mut}\left(\boldsymbol{T}\right)$$

Kríženie

Kríženie dvoch tabuliek T_1 a T_2 je stochastická transformácia, ktorá vytváre dve nové tabuľky T'_1 a T'_2

$$(T_1', T_2') = O_{cross}(T_1, T_2)$$



Ilustratčnýpríklad symbolickej regresie

Racionálny polynóm

Budeme predpokladať, že množiny povolených vrcholov majú toto zloženie:

(1) terminálne vrcholy { celé kladné čísla a premenná x} ,
(2) unárny vrchol { zmena znamienka } ,
(3) binárne vrcholy { súčet, rozdieľ, podieľ a súčin } .

Genetický algoritmus obsahoval populáciu 200 chromozómov (stĺpcových tabuliek, pričom p_{max}=15 a v_{in}^{max}=2).

Regresná tabulka bola generovaná pomocou racionálnej funkcie $f(x)=(1+x-x^2)/(1+x+x^2)$ pre 40 ekvidistančných hodnôt premennej $x\in$ [-10,10], pre Δx=0.5.

Vybrané grafy z priebehu genetického algoritmu sú znázornené na obrázku. Posledný graf na tomto obrázku odpovedá funkcii ((1/*x*+1)-*x*)/((1/*x*+1)+*x*), ktorá po jednoduchých úpravách poskytuje pôvodnú funkciu použitú pre generovanie regresnej tabulky.