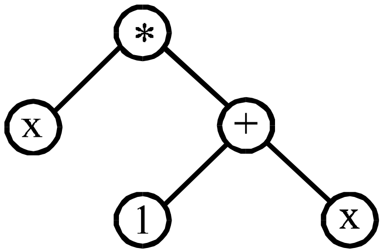
Genetické programovanie

Genetické programovanie (GP)

- Na prelome 80-tých a 90-tých rokov americký informatik John Koza (Stanford University) navrhol originálnu modifikáciu genetického algoritmu, ktorú nazval *genetické programovanie*.
- V tomto prístupe sú chromozómy - znakové reťazce nahradené zložitejšími štruktúrami - funkciami.
- Čo sa rozumie pod pojmom "funkcia"? V najjednoduchšej verzii genetického programovania **sa funkcie rovnajú výrazom obsahujúcim premenné, konštanty, základné aritmetické operácie a elementárne funkcie**.

Reprezentácia funkcie

Jednoduchá funkcia "$x*(1+x)$" je reprezentovaná pomocou syntaktického stromu (parse tree)



kde horný vrchol (priradený operácii násobenia) sa nazýva koreň stromu. Interpretácia tohto syntaktického stromu je jednoduchá, postupujúc zdola - nahor, od koncových vrcholov ku koreňu stromu, postupne sa vykonáva **výpočet funkcie "$x*(1+x)$"**.

Reprezentácia funkcie

- Prvý problém, ktorý musíme riešiť v rámci genetického programovania, je **spôsob vyhodnotenia chromozómu** - stromu.

- Toto vyhodnotenie je formálne chápané ako **funkcia $y=t(x)$** priraďujúca nezávislej premennej (vstup) x závislú premennú (výstup) y. V našom príklade je táto funkcia určená výrazom $t(x)=x*(1+x)$.

- Nech $\boldsymbol{A=\{(x_i, y_i); i=1,2,...,p\}}$ je tréningová množina obsahujúca $\boldsymbol{p}$ bodov $\boldsymbol{(x_i, y_i)}$.

- **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.**

Symbolická regresia

- V prípade, že je tento súčet nulový, funkcia ***t*(*x*) presne vystihuje body z tréningovej množiny**. Nájsť takú funkciu sa nám pravdepodobne nepodarí, no môžeme sa pokúsiť nájsť takú funkciu, pre ktorú bude tento súčet chýb čo najmenší.

- Hľadanú funkciu vyjadrujeme pomocou syntaktických stromov obsahujúcich predpísané typy vrcholov.

- Takto formulovaná úloha je veľmi blízka regresnej analýze, kde pre danú tréningovú množinu bodov a pre daný typ funkcie hľadáme také parametre - koeficienty funkcie, ktoré minimalizujú účelovú funkciu.

- V tomto prípade je "priestor" funkcií podstatne ohraničený

Príklad:

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

- Pohyb robota v prostredí môže byť určený funkciou $y=t(x)$, kde x sú vstupné informácie (napr. z TV kamery) o jeho blízkom okolí a y sú príkazy pre jeho motorické zariadenie.

- Naším cieľom je zostrojiť takú funkciu **$t(x)$, ktorá čo najlepšie popisuje pohyb robota** (umelého živočícha) a jeho úspešnosť v prostredí. V tomto prípade už nemôžeme hovoriť o tréningovej množine A ako o množine bodov vopred zadanej.

- Úspešnosť danej funkcie $t(x)$ je daná úspešnosťou robota v prostredí po určitom vopred danom počte krokov (napr. počtom kúskov potravy, ktoré nazbieral pri pohybe v prostredí po n krokoch).

Záver

- Vyššie uvedený postup pre ohodnotenie funkcie $f(x)$ je vhodný pre riešenie širokej triedy problémov adaptácie pomocou genetického programovania. Môžeme teda konštatovať, že prístup genetického programovania je vhodný k riešeniu dvoch tried problémov, **a to**
  a. **symbolickej regresie a**
  b. **adaptácie (alebo učenia).**
- Zaujímajú nás možnosti práce so stromami a ich možné reprezentácie.

Koreňové stromy a Readov lineárny kód

- Nech ***G***=(*V*,*E*) je *strom* (súvislý acyklický graf), kde $V=\{v_1, v_2, ..., v_p\}$ je neprázdna *množina vrcholov* a $E=\{e_1, e_2, ..., e_q\}$ je *množina hrán*.
$$|V| = |E| + 1$$
- Nech $v \in V$ je vrchol stromu ***G***, valentnosť tohto vrcholu, označená *val*(*v*), je nezáporné celé číslo, ktoré určuje počet hrán incidentných s vrcholom *v*. Suma valencií všetkých vrcholov spĺňa podmienku
$$\sum_{v \in V} val(v) = 2q = 2(p-1)$$
- ***Koreňový strom*** je strom, ktorý ma jeden vrchol špeciálne odlíšený od ostatných vrcholov, tento vrchol sa nazýva ***koreň.*** Koreňový strom je formálne určený ako usporiadaná trojica kde $v \in V$ je koreň.
$$T = (V, E, v)$$

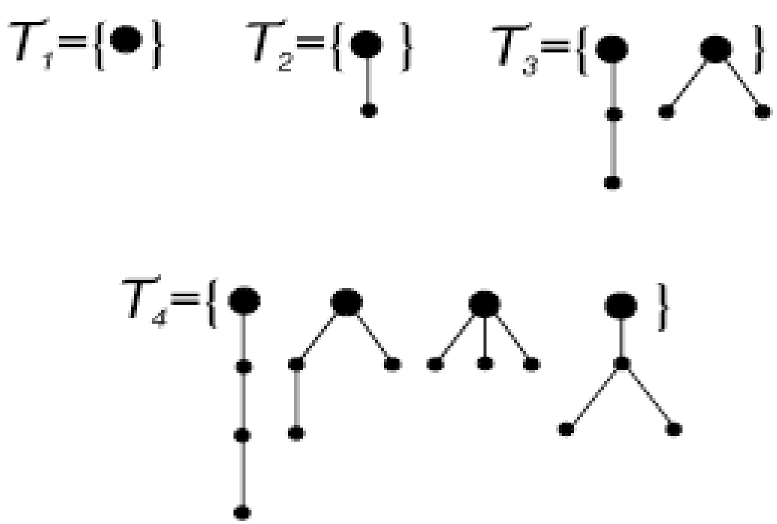
Koreňové stromy a Readov lineárny kód

- Dva koreňové stromy ***T*** a ***T'*** sú izomorfné vtedy a len vtedy, ak existuje také 1-1 zobrazenie ϕ, ktoré zachováva susedstvo vrcholov a zobrazuje koreň stromu ***T*** na koreň stromu ***T'***, t.j. $\phi(v)=v'$.

- Nech T_i je množina obsahujúca všetky možné koreňové stromy, ktoré sú zložené z *i* vrcholov. Zjednotenie týchto množín
$$T = T_1 \cup T_2 \cup \ldots$$

obsahuje všetky možné neizomorfné koreňové stromy.

Koreňové stromy



Prvé štyri množiny T1-T4 , ktoré obsahujú všetky možné neizomorfné koreňové stromy s počtom vrcholov 1 až 4.

Koreňové stromy a Readov lineárny kód

- Nech každý koreňový strom z T je ohodnotený reálnym číslom $t: T \to R$
- Táto funkcia ohodnotí každý koreňový strom reálnym číslom, ktoré nazývame ***index.*** Vyjadruje v určitom priblížení "topológiu" koreňového stromu. Požadovaná hodnota indexu nech je označená t_{req}, potom môžeme definovať účelovú funkciu takto
$$f(\mathbf{T}) = |\, t(\mathbf{T}) - t_{req} \,|$$
- Účelová funkcia určuje odchýlku vypočítaného indexu $t(\boldsymbol{T})$ od požadovanej hodnoty t_{req} . Ak je hodnota účelovej funkcie nulová, potom index $t(\boldsymbol{T})$ je rovný požadovanej hodnote.
- Nulová hodnota účelovej funkcie odpovedá globálnemu minimu nad priestorom všetkých možných koreňových stromov, odpovedajúci koreňový strom ako riešenie nasledujúceho problému minimalizácie
$$\mathbf{T}_{opt} = \arg\min_{\mathbf{T} \in T} f(\mathbf{T})$$

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

Readov lineárny kód

- Readov kód pre koreňové stromy je reťazec (sekvencia) celých čísel, ktoré odpovedajú buď valencii koreňa alebo valencii zníženej o jednotku pre ostatné vrcholy.

- Lineárny kód koreňového stromu T bude označený code(T); ak strom T obsahuje jeden vrchol, potom jeho kód je code(T)= '0'.
- Študujme koreňový strom obsahujúci dva alebo viac vrcholov a predpokladajme, že jeho koreň je incidentný s r hranami, odstránením koreňa zo stromu dostaneme tzv. stromy prvej generácie.

Readov lineárny kód

- Tieto stromy prvej generácie sú znovu uvažované ako koreňové stromy, koreň je identifikovaný s vrcholom, ktorý bol priamym potomkom koreňa pôvodného stromu.
- Nech vzniknuté koreňové stromy sú označené $\boldsymbol{T}_1$, $\boldsymbol{T}_2$, ..., $\boldsymbol{T}_r$, potom Readov kód pôvodného stromu ***T*** je definovaný vzťahom

$$\mathrm{code}(\mathbf{T}) = 'r' + \mathrm{code}(\mathbf{T}_1) + \mathrm{code}(\mathbf{T}_2) + \ldots + \mathrm{code}(\mathbf{T}_r)$$

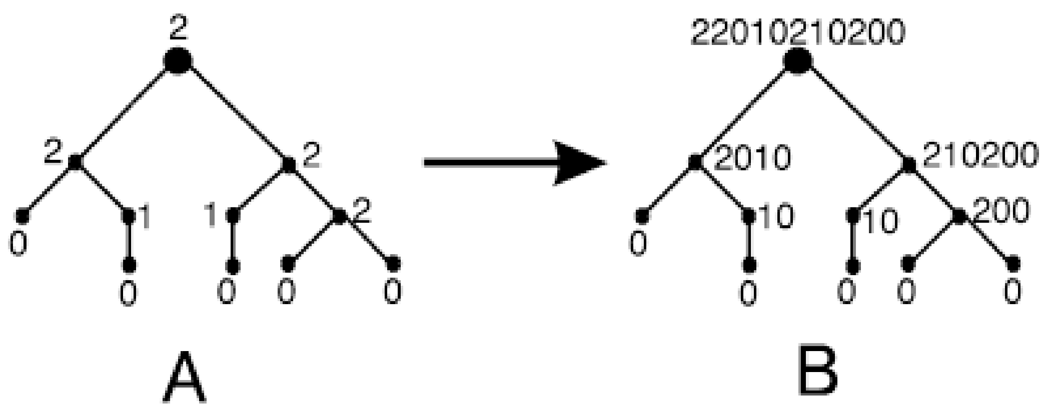
kde '+' reprezentuje operáciu zreťazenia.

Readov lineárny kód

- Výsledný kód code(***T***) má tvar reťazca - postupnosti

  $$a = (\alpha_1, \alpha_2, \ldots, \alpha_p)$$

- jeho dĺžka je označená $|\boldsymbol{\alpha}| = p$.
- Príklad:



Readov lineárny kód

- **Niekoľko poznámok k jednoznačnosti Readovho kódu**. Ako vyplýva z vyššie uvedenej konštrukcie kódu, táto môže byť realizovaná mnohými spôsobmi, t.j. koreňový strom ***T*** je reprezentovaný mnohými kódmi. Ináč povedané, **dva rôzne kódy môžu reprezentovať koreňové stromy, ktoré sú izomorfné**.

- Toto ohraničenie Readovho kódu je jednoducho odstrániteľné tak, že formula je aplikovaná tak, že príslušné "**podkódy**" sú usporiadané pred ich zreťazením do nerastúcej alebo neklesajúcej postupnosti.

- Týmto jednoduchým spôsobom získame **tzv. *kanonický Readov kód,*** ktorý má tú vlastnosť, že ak **dva kanonické kódy sú rôzne**, potom odpovedajúce koreňové stromy nie sú izomorfné.
- Pre potreby genetického programovania je plne postačujúci obyčajný Readov kód.

Analýza Readovho lineárneho kódu

```
procedure Parsing_Code(input:α; output:V,E);
begin E:=∅; V:={1};
      index_0:=1; branch_0:=α_1; d:=0; i:=1;
      while d≥0 do
      if branch_d>0 then
      begin branch_d:=branch_d-1; i:=i+1; d:=d+1;
            branch_d:=α_i; index_d:=i;
            V:=V∪{index_d};
            E:=E∪{[index_d-1,index_d]};
      end else d:=d-1;
end;
```

Pseudopascalovská implementácia analýzy Readovho lineárneho kódu, ktorý vyhovuje podmienkam, t.j. k analyzovanému kódu existuje koreňový strom. Výstupné parametre V a E sú vrcholová resp. hranová množina. Hĺbka backtrack algoritmu je určená celočíselnou premennou d. Algoritmus postupne navštívi všetky vrcholy koreňového stromu. Po ukončení algoritmu množiny V a E obsahujú všetky vrcholy resp. hrany, pričom koreň je indexovaný 1.

Analýza Readovho lineárneho kódu

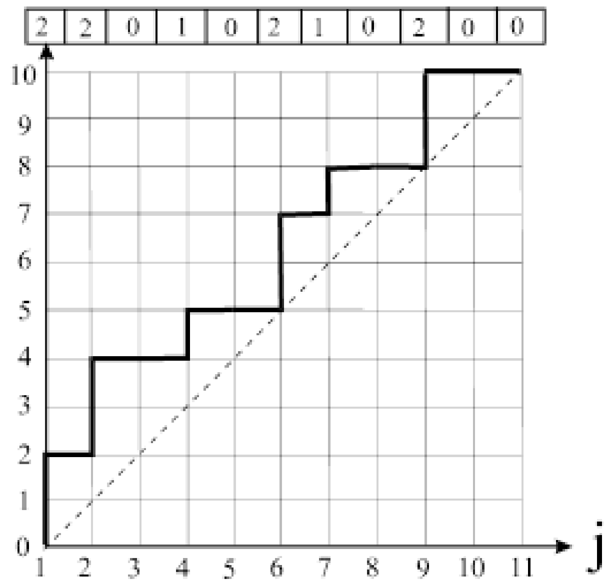
- Postupnosť čísel sa nazýva ***grafová***, ak existuje koreňový strom obsahujúci *p* vrcholov, ktorého Readov kód je totožný s postupnosťou.

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

$$\sum_{i=1}^{j} \alpha_i \geq j \left(j = 1,2,\ldots, p-1 \right) \qquad \sum_{i=1}^{p} \alpha_i = p-1$$

Táto veta má veľký význam pre náhodné generovanie koreňových stromov a pre hľadanie podstromov v koreňových stromoch, čo sú dôležité operácie pre jednoduchú implementáciu mutácia a kríženia nad stromami.

Náhodné generovanie koreňových stromov



Diagramatické znázornenie podmienok, jednotlivé komponenty kódu (22010210200) sú interpretované ako dĺžky vertikálnych častí hrubej lomenej čiary. Prerušovaná čiara vyjadruje nerovnosti, hrubá čiara musí ležať nad touto čiarou.

Náhodné generovanie Readovho lineárneho kódu dĺžky p

```
Procedure Generation_Code(output:p,α);
begin p:=pmin+random(pmax-pmin+1);
      α1:=1+random(p-1); d1:=p-1;
      for j:=2 to p-1 do
      begin dj:=dj-1-αi=1;
            if dj=p-j then  αj:=1+random(dj)
            else αj:=random(dj+1);
      end;
      αp:=0;
end
```

Pseudopascalovská implementácia náhodnej generácie Readovho lineárneho kódu dĺžky p, ktorý je automaticky grafový. Funkcia random(l) je náhodný generátor celých čísel z uzavretého intervalu [0,l-1]. Algoritmus je založený na podmienkach. Dĺžka generovaného kódu je ohraničená podmienkami $p_{min} \leq p \leq p_{max}$.

Podmienky pre náhodné generovanie Readových kódov

- Definujme tieto veličiny rekurentne

$$d_1 = p - 1$$

$$d_j = d_{j-1} - \alpha_{j-1} \left(j = 2,3,\dots, p \right)$$

ktoré ohraničujú zhora komponenty kódu $\left(\alpha_1, \alpha_2, \dots, \alpha_p\right)$

$$1 \leq \alpha_j \leq d_j \left(\mathrm{if}\, d_j = p - j \right)$$

$$0 \leq \alpha_j \leq d_j \left(\mathrm{if}\, d_j < p - j \right)$$

pre *j*=1,2,...,*p*. *Tieto vzťahy sú použiteľné pri generovaní Readovho kódu.*

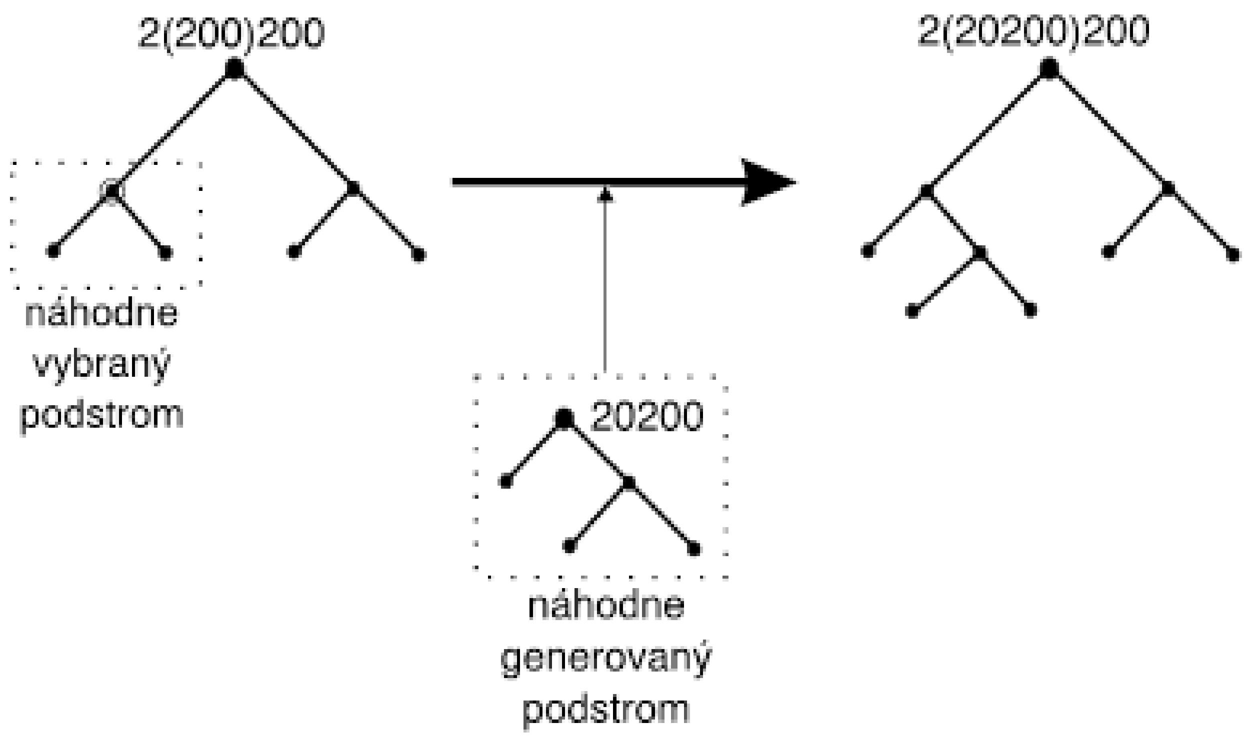
Operácie mutácie a kríženia

- Operácia mutácie priradí stochasticky koreňovému stromu iný koreňový strom, pričom oba stromy môžu mať rôzny počet vrcholov (obvykle ohraničené intervalom $[p_{min},p_{max}]$). Ak starý a nový lineárny kód sú označené α a α', potom mutáciu formálne píšeme ako operátor transformujúci reťazec α na reťazec α'

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

- kde $|\alpha|,|\alpha'| \in [p_{min},p_{max}]$.
- Podľa Kozu mutácia koreňového stromu je realizovaná tak, že v prvom kroku sa náhodne vyberie nekoreňový vrchol a príslušný podstrom sa nahradí iným podstromom náhodne generovaným.

Operácia mutácie



Operácia mutácie

- Alternatívna reprezentácia procesu mutácie môže byť sformulovaná nasledujúcim spôsobom. Nech kód α je rozdelený na tri podkódy, $a = (a_1, a_2, a_3)$, kde podkód α_2 odpovedá nejakému podstromu stromu vyjadrenému kódom α.
- Podkód α_2 je nahradený novým náhodné generovaným podkódom , potom dostaneme nový kód interpretovaný ako mutácia kódu α, $a' = (a_1, a_2', a_3)$. Dĺžka podkódov α_2 a α_2' je ohraničená nerovnosťami
- $$p_{min} - |\alpha| + |\alpha_2| \leq |\alpha_2'| \leq p_{max} - |\alpha| + |\alpha_2|$$

Operácia mutácie

- Podkód α_2 kódu α je určený tak, že najprv náhodne určíme tzv. mutačný bod $2 \le \tau \le |\alpha| - 1$, určuje tú polohu v kóde α od ktorej začína podkód α_2 .
- Ako efektívne stanovíme dĺžku podkódu α_2 ? K tomuto účelu je vhodné použiť uvedené nerovnosti , hovoríme, že podkód α_2 má dĺžku *r* ak nasledujúce podmienky sú splnené

$$\sum_{i=1}^{j} \alpha_{i+\tau-1} \ge j \left(j = 1, 2, \dots, r-1 \right)$$

$$\sum_{i=1}^{r} \alpha_{i+\tau-1} = r - 1$$

Implementácia mutácie Readovho lineárneho kódu

```
procedure Mutation(input:p,α;output:p',α');
begin τ:=2+random(p-1);
      r:=length of a subcode α₂ of α that
         starts at τ;
      r':=p_min-p+r+random(p_max-p_min+1);
      Generation_Code(r',α'₂);
      a'=(a₁,a'₂,a₃); p':=p-r+r';
end;
```

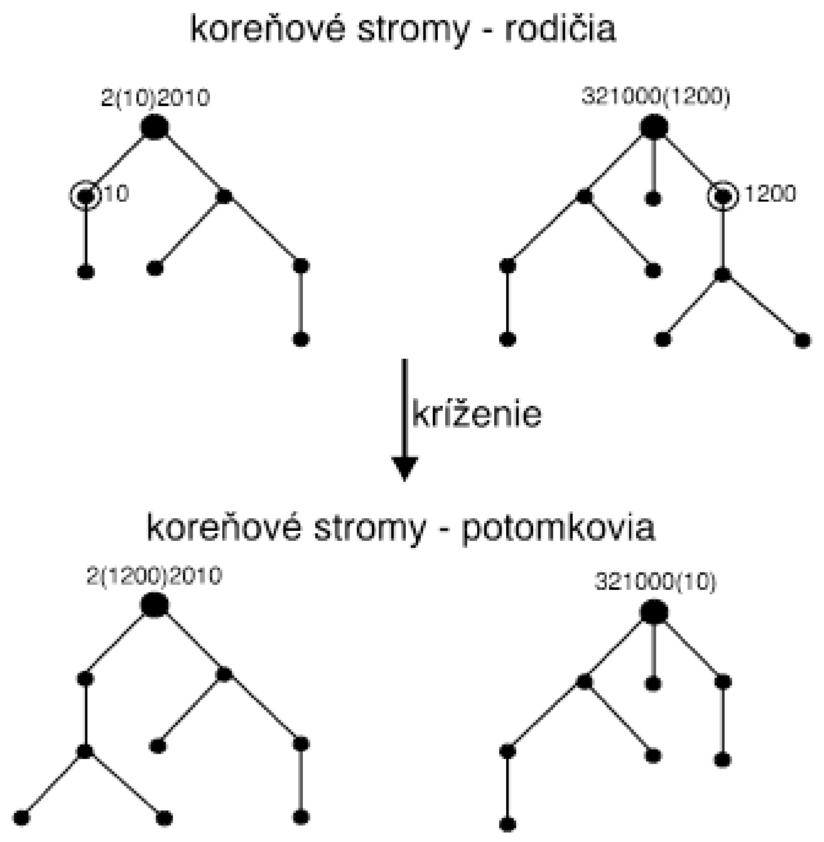
Implementácia mutačného procesu aplikovaného na kód α, pričom výsledný kód je α'. Algoritmus je inicializovaný náhodným generovaním mutačného bodu τ, dĺžka podkódu, ktorý začína v bode τ je vyjadrená premennou r. Dĺžka nového náhodne generovaného podkódu je označená r'.

Operácia prekríženia

- Nech α a β sú dva (rodičovské) lineárne kódy koreňových stromov, aplikovaním operácie kríženie dostaneme dva nové lineárne kódy (potomkov) α' a β'
$$(a',b')=O_{cross}(a,b)$$
- Táto operácia je podľa Kozu realizovaná tak, že v rodičovských kódoch - chromozómoch α and β náhodne vyberieme dva body kríženia a príslušné podkódy, ktoré začínajú v týchto bodoch sa vymenia.
- Alternatívna reprezentácia operátora kríženia je nasledovná. Nech rodičovské kódy α and β koreňových stromov majú tvar $a=(a_1,a_2,a_3)$ resp. $b=(b_1,b_2,b_3)$, kde počiatočné body podkódov α_2 a β_2 boli náhodne generované.
- Aplikácia operácie kríženia na rodičovské kódy α and β spočíva vo výmene podkódov α_2 and β_2 , potomkovia majú potom tvar

$$a'=(a_1,b_2,a_3) \qquad b'=(b_1,a_2,b_3)$$

Príklad:



koreňové stromy - rodičia
2(10)2010
10
321000(1200)
1200
kríženie
koreňové stromy - potomkovia
2(1200)2010
321000(10)

Operácia prekríženia

Procedure Crossover(input:p, $\tilde{p}$, $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$; output p', $\tilde{p}$', $\boldsymbol{\alpha}$', $\boldsymbol{\beta}$');

begin τ:=2+random(p-1); $\tilde{\tau}:=2+\text{random}(\tilde{p}-1)$

$a'=(a_1,b_2,a_3)$; $b'=(b_1,a_2,b_3)$;

p':=p-$|\boldsymbol{\alpha}_2|+|\boldsymbol{\beta}_2|$; $\tilde{p}$':=$\tilde{p}$-$|\boldsymbol{\beta}_2|+|\boldsymbol{\alpha}_2|$;

end;

Implementácia operátora kríženia dvoch rodičovských lineárnych kódov α a β , ktoré sú modifikované na nové lineárne kódy - potomkov α' a β'. Náhodný výber bodov kríženia τ a $\tilde{\tau}$ je realizovaný tak, že dĺžky nových lineárnych kódov vyhovujú nerovnostiam $p_{min} \le |\alpha'| \le p_{max}$ a $p_{min} \le |\beta'| \le p_{max}$.

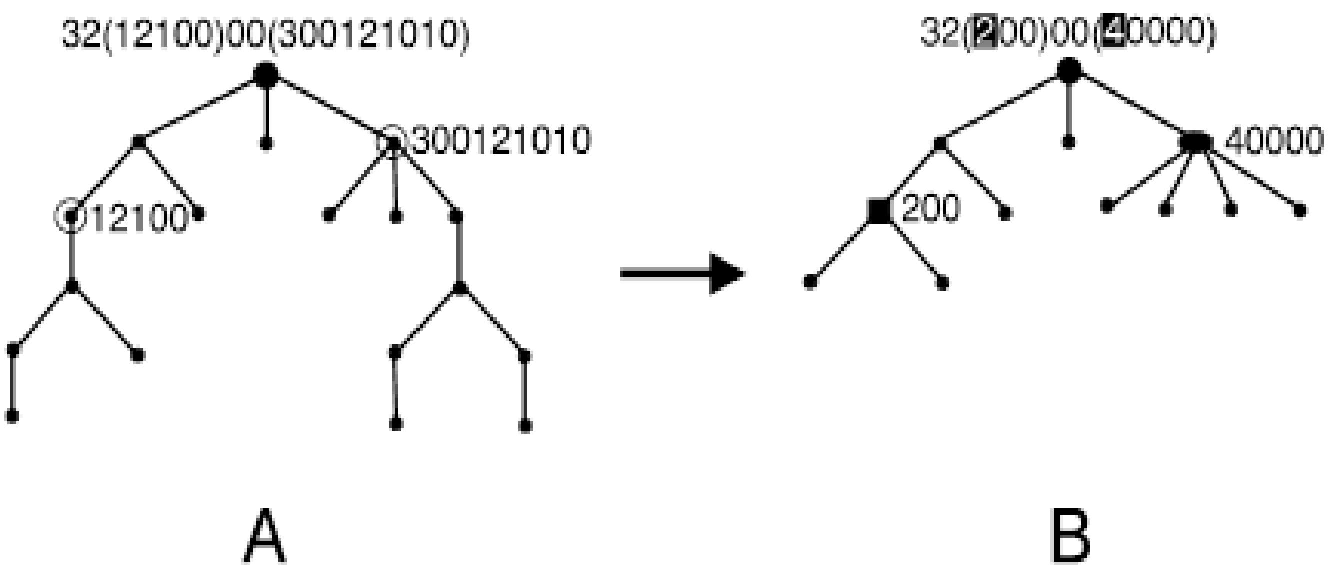
Kompresia Readovho kódu

- **Proces kompresie** spočíva v substitúcii podkódu α_2 tzv. komprimovaným kódom typu (*n*00...0), kde *n* je počet terminálových vrcholov v podstrome (alebo počet komponent '0' v podkóde α_2).

$$a = (a_1, a_2, a_3) \xrightarrow{kompresia} a_{compress} = \left(a_1, a_{2,compress} = (n00...0), a_3\right)$$

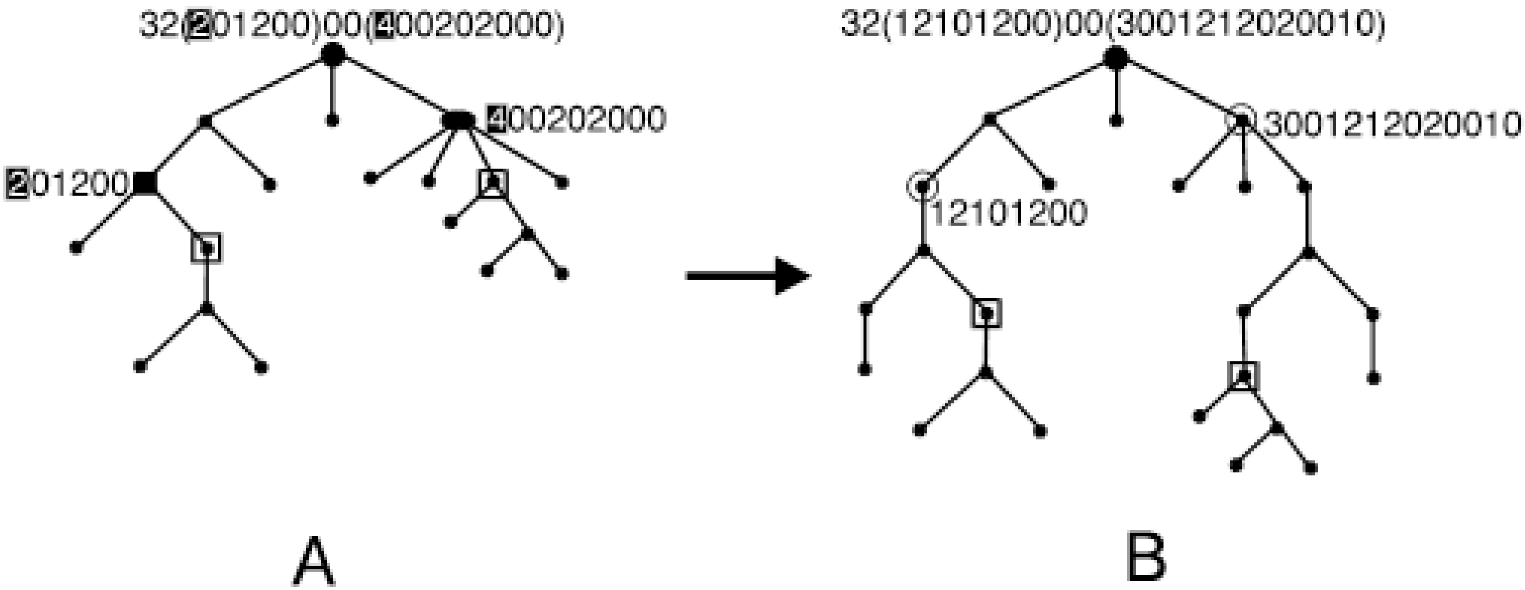
- Ako realizovať inverzný proces ku kompresii ? Nech kód β obsahuje vrchol interpretovaný ako "komprimovaný" vrchol. Potom kód β môže byť dekomprimovaný tak, že odpovedajúci podstrom priradený komprimovanému vrcholu je nahradený koreňovým podstromom, ktorý indukuje daný komprimovaný vrchol, pričom niektoré terminalové vrcholy môžu byť substituované podstromami.

Kompresia Readovho kódu



Koreňový strom reprezentovaný diagramom A je komprimovaný na menší koreňový strom reprezentovaný diagramom B. Tento proces sa vykoná tak, že vybrané podstromy (ich koreňové vrcholy sú zakrúžkované) v diagrame A sú nahradené novými vrcholmi (štvorcovým alebo oválnym), ktoré sú incidentné len s terminálovými vrcholmi, pričom ich počet je rovnaký ako počet terminálových vrcholov v pôvodných podstromoch.

Dekompresia Readovho kódu

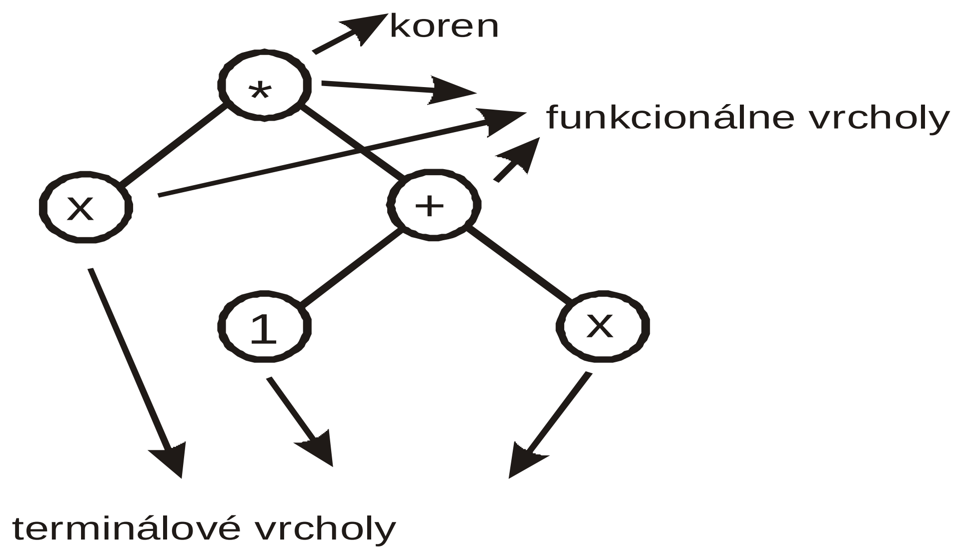


Koreňový strom A obsahuje dva komprimované vrcholy reprezentované štvorcovým a oválnym vrcholom, pričom vrcholy označené štvorcom (pôvodne terminalové) boli substituované reťazcami 1200 resp. 20200. Ak komprimované vrcholy sú "dekomprimované" odpovedajúcimi koreňovými podstromami, potom pôvodný strom A je dekomprimovaný na väčší strom B, ktorý už obsahuje len štandardné vrcholy.

Symbolická regresia

- Symbolická regresia patrí medzi základné aplikácie genetického programovania.
- Symbolická regresia spočíva v hľadaní takej funkcie reprezentovanej syntaktickým stromom s predpísanými operáciami, ktorá čo najlepšie aproximuje údaje z tréningovej množiny.
- Pod **syntaktickým stromom *t*** budeme rozumieť koreňový strom ***T***, ktorého vrcholy sú ohodnotené symbolmi aritmetických (alebo iných) operácií a celý strom môže byť ohodnotený reálnym číslom reprezentujúcim hodnotu funkcie, ktorá je priradená stromu pre danú vstupnú hodnotu nezávislej premennej (alebo nezávislých premenných).

Vrcholy v syntaktických stromoch



(1) *Terminálne vrcholy*, tieto vrcholy odpovedajú buď *nezávislým premenným* x, y,... nezáporným celočíselným *konštantám* 0, 1, 2,....
(2) *Funkcionálne vrcholy* odpovedajú jednoduchým operáciám, ktoré sú unárne, binárne, ternárne,... .

Formálne vyjadrenie

- Formálne môžeme syntaktický strom vyjadriť ako usporiadanú dvojicu

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

kde ***T*** je koreňový strom určujúci štruktúru syntaktického stromu ***t*** a ϕ je zobrazenie vrcholov koreňového stromu na "aritmetické" symboly, premenné, alebo konštanty

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

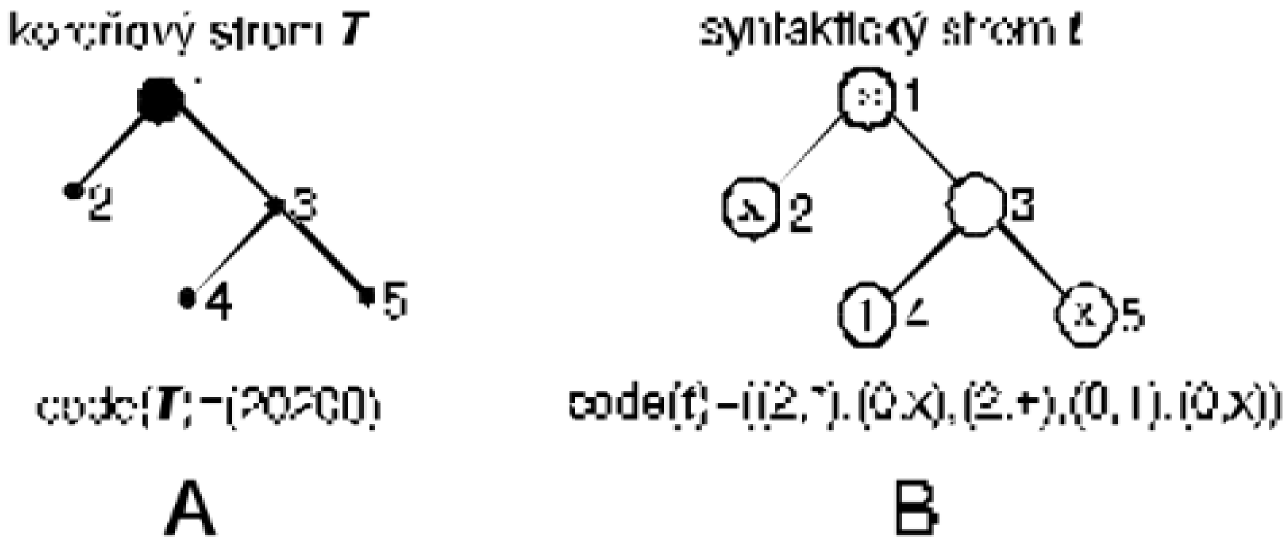
Kódovanie syntaktických stromov

- Nech Readov lineárny kód stromu ***T*** je code(***T***)=**α**=(α_1,α_2,...,α_p), potom lineárny kód syntaktického stromu ***t*** môže byť zostrojený tak, že Readov kód koreňového stromu je rozšírený o zobrazenie ϕ

$$\mathrm{code}(\mathbf{t}) = ((\alpha_1, \phi_1), (\alpha_2, \phi_2), \ldots, (\alpha_p, \phi_p))$$

- kde ϕ_i je ohodnotenie *i*-teho vrcholu buď funkcionálnym alebo terminálnym symbolom.

Kódovanie syntaktických stromov



Úloha regresnej analýzy

- Majme tréningovú množinu obsahujúcu *n* bodov

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

- 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

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

Úloha

- *Implementuje genetický algoritmus tak, že chromozómy budú určené ako Readove lineárne kódy s ohodnotením vrcholov. Toto ohodnotenie nech je vykonané tak, že syntakticé stromy budú odpovedať buď racionálnym polynómom (t.j. funkčné vrcholy sú +,-,*, a /, terminálové vrcholy sú celočíselné konštanty alebo premenná x) alebo Boolovským funkciám (t.j. funkčné vrcholy sú operácia and, or, xor, a not, a terminálové vrcholy sú jednotlivé Boolovské premenné).* ***Účelová funkcia*** *nech je rozšírená o pokutový člen, ktorý bude preferovať syntaktické stromy s menším počtom vrcholov*

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

- *kde α je malá kladná konštanta a symbol |t| vyjadruje počet vrcholov v strome t. Modifikujete napísanú implementáciu tak že bude odpovedať horolezeckému algoritmu alebo evolučnému programovaniu. Porovnajte efektívnosť týchto troch rozdielnych metód pri konštrukcii funkcií, ktoré reprodukujú predpísanú regresnú tabuľku.*

Spracované podľa knihy V. Kvasničku a J. Pospíchala

Ďakujem za pozornosť