Genetsko programiranje

Marko Đurasević

marko.durasevic@fer.hr

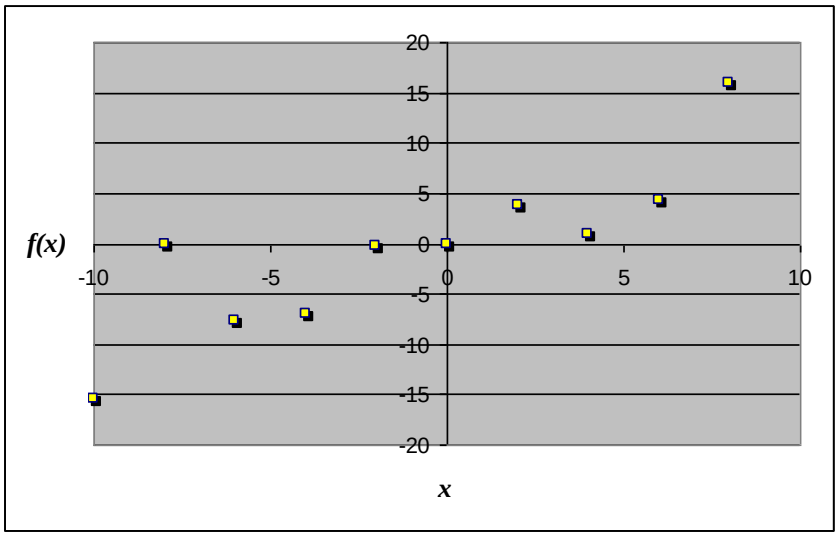
Motivacija

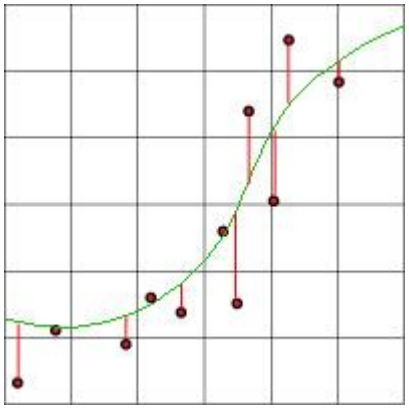
- Problem pronalaženja odgovarajućih koeficijenata (regresija):
  - $a * x_1^2 * \sin(b * x_2) + \ln(c * x_1 + d * x_2)$

- Što ako oblik funkcije nije poznat?
  - Simbolička regresija – tražimo oblik funkcije

Motivacija - simbolička regresija

- zadatak: otkriti *simbolički* oblik modela
  - nemamo pretpostavki (predznanja) o nepoznatoj funkciji





Motivacija

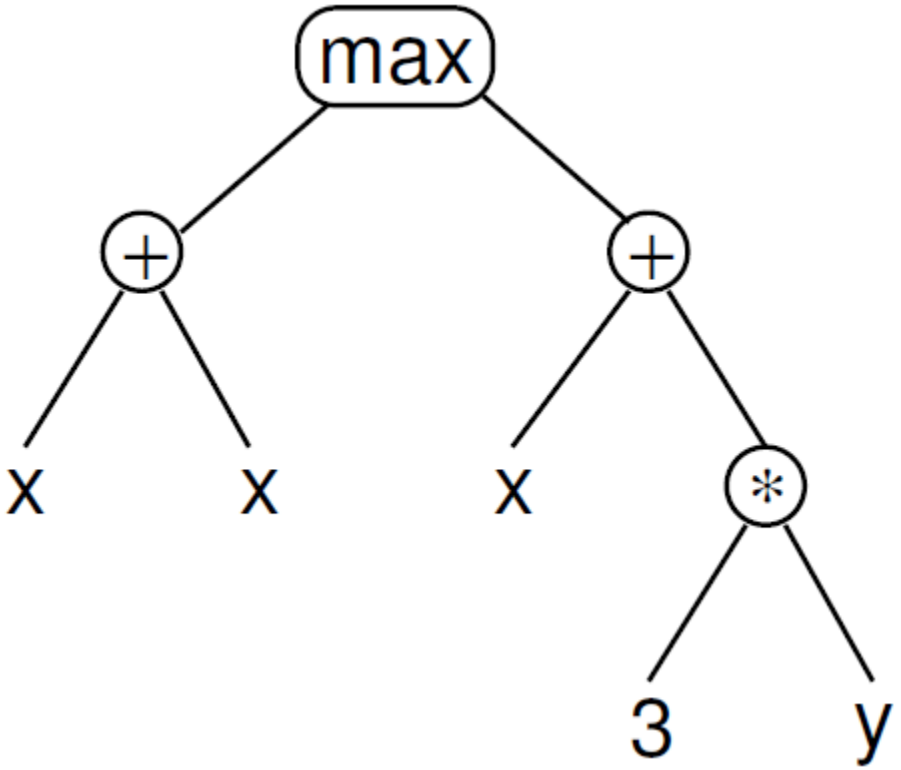
- Problemi klasifikacije – izrada klasifikatora
- Izrada upravljača (programa)
- Pravila za trgovinu dionicama (kriptovalutama)

Osnove genetskog programiranja

Genetsko programiranje

- Genetski algoritam s drugačijim prikazom rješenja
- Jedinke predstavljaju matematičke izraze ili programe
  - Najčešće zapisane u obliku stabla
- Prilagođeni operatori mutacije i križanja
- Parametri: veličina populacije, vjerojatnost operatora, kriterij zaustavljanja...

Prikaz rješenja



$$\max(x + x, x + 3 * y)$$

Prikaz rješenja

- Odabrati skup primitiva (operatora i operanada)
- Operatori (unutarnji čvorovi)
  - Aritmetički (+, -, *, /, sin, cos, log, exp, pow, sqrt), logički (AND, OR, NOT), uvjetni (IF, IFGTE), petlje
- Operandi (terminali ili vanjski čvorovi)
  - Ulazne varijable (x, y), konstante (0, 1, 3.14), funkcije bez argumenata (rand)

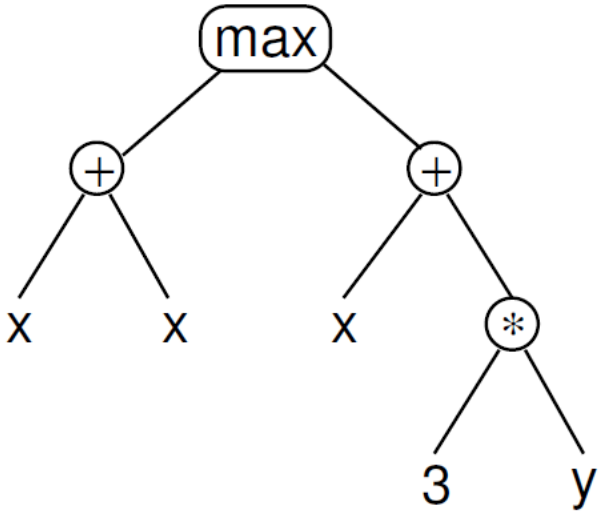
Prikaz rješenja

- Potrebna svojstva skupa primitiva:
  - Potpunost (*sufficiency*) – skupom moguće riješiti problem
  - Zatvorenost *(closure)* – definirane sve kombinacije operacija-operand
- Ograničiti veličinu rješenja:
  - Maksimalna dubina stabla
  - Maksimalni broj čvorova u stablu

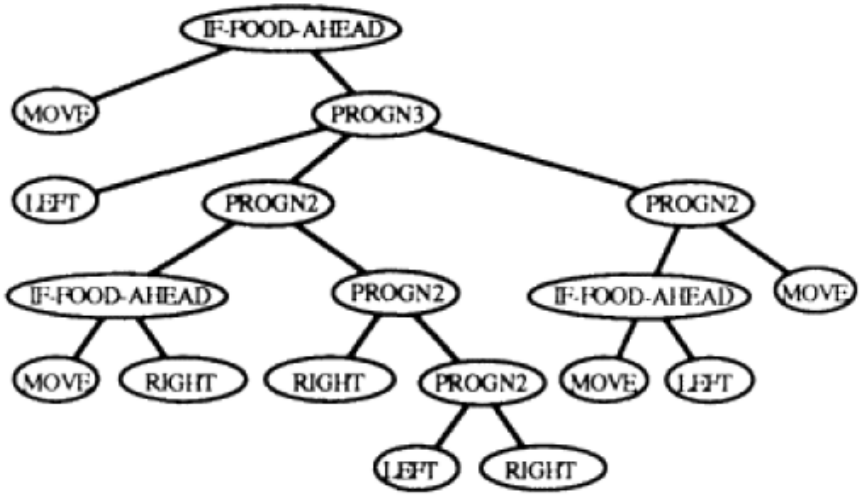
Interpretacija rješenja

- Ovisno o vrsti problema, stablo se može interpretirati na različite načine

Simbolička regresija - računanje izlaza za pojedine vrijednosti ulaznih varijabli



Upravljanje agentom - prolazak kroz stablo i izvođenje čvorova redom

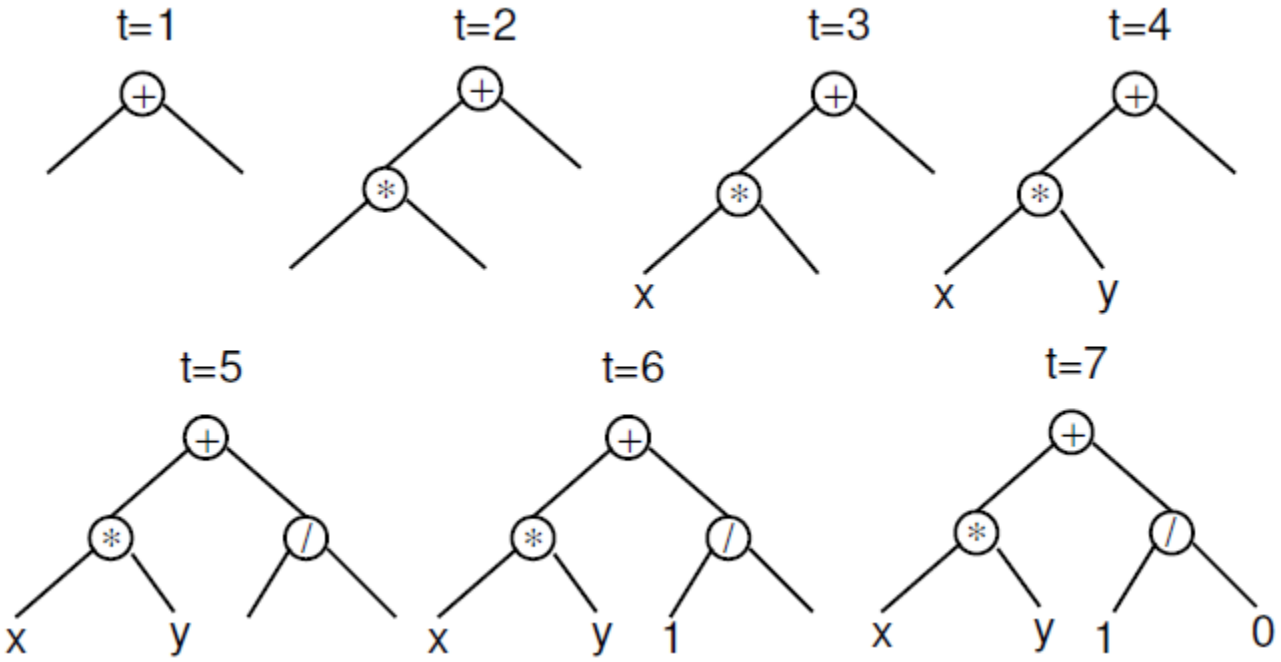


Inicijalizacija početne populacije

- Obično se zada maksimalna početna dubina
- Metode izgradnje jedinki:
  - *Full* – sva stabla su potpuna s maksimalnom dubinom
  - *Grow* – ne moraju svi terminali biti na maksimalnoj dubini
  - *Ramped half-and-half* – najčešće korišten
    - 50% full, 50% grow uz različite maksimalne dubine

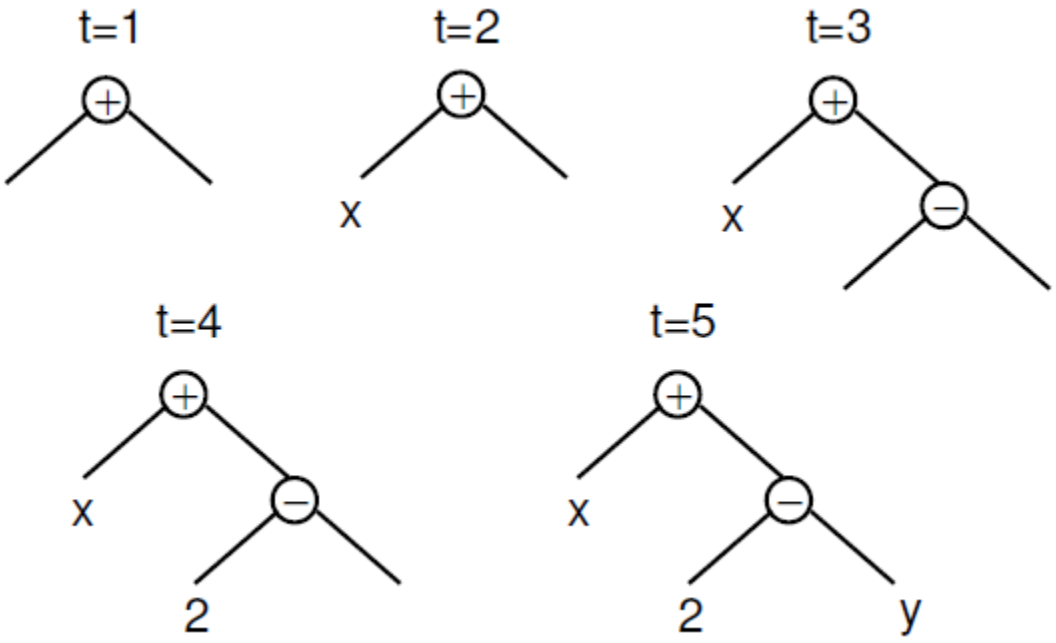
Inicijalizacija početne populacije

- Full metoda – stablo maksimalne dubine 2



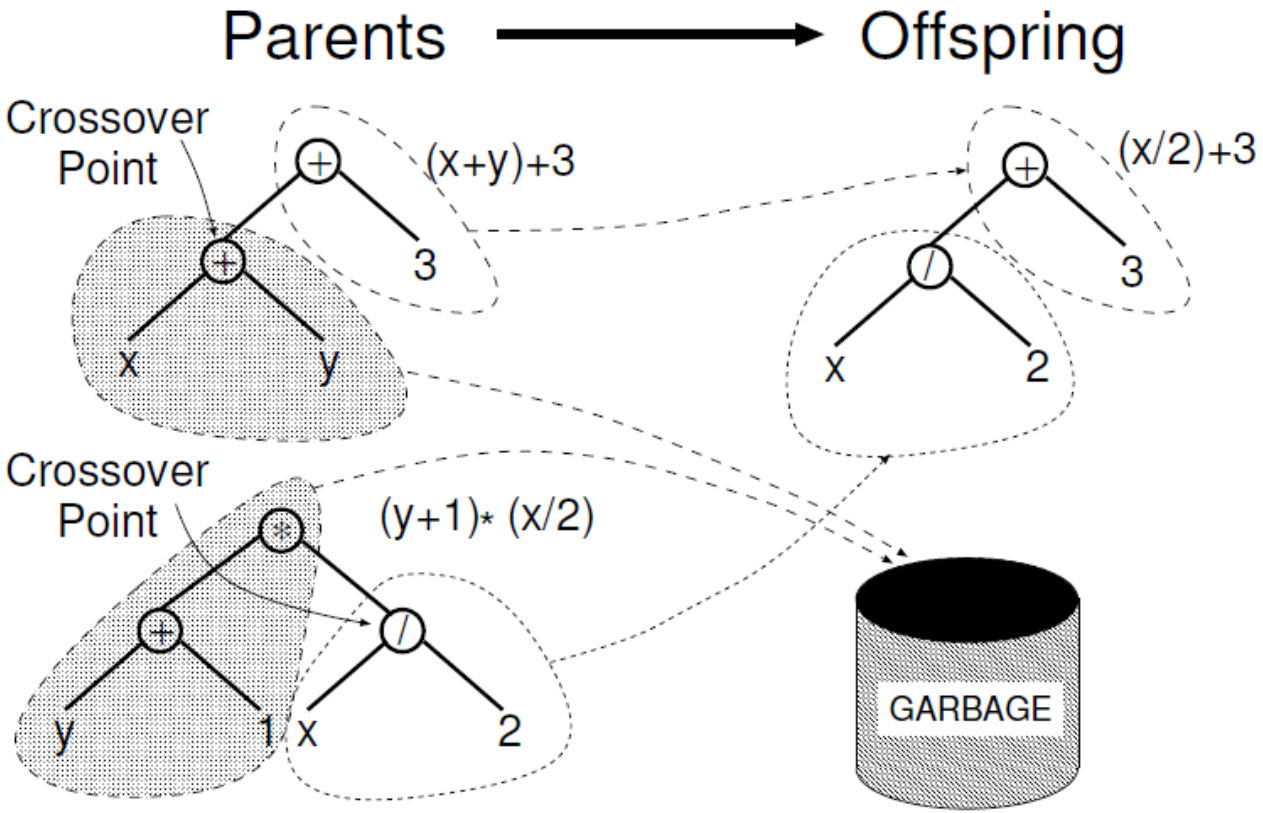
Inicijalizacija početne populacije

- Grow metoda – stablo maksimalne dubine 2



Križanje

- Subtree crossover

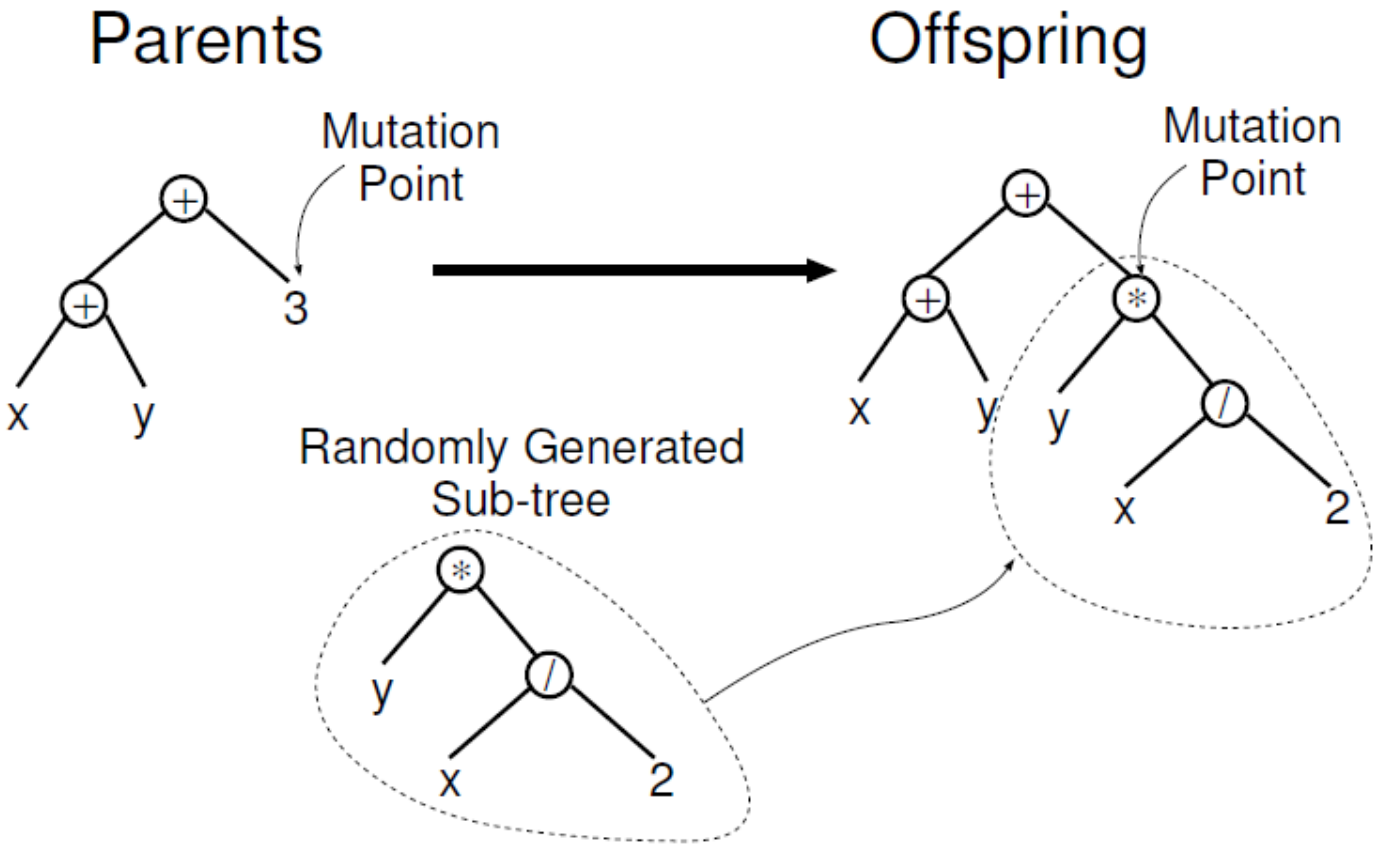


Križanje

- One-point crossover
- Uniform crossover
- Context-preserving crossover
- Size-fair crossover

Mutacija

- Subtree mutation



Mutacija

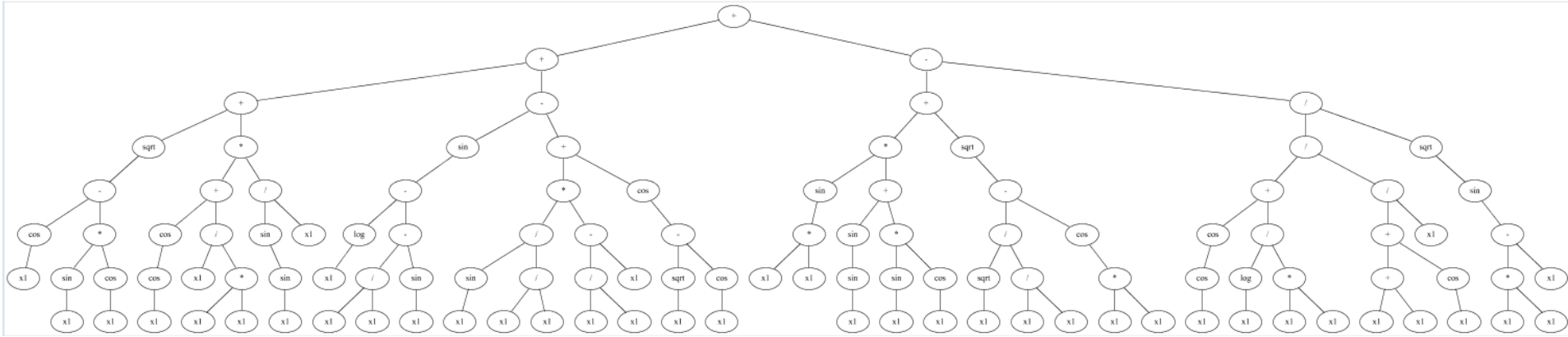
- Subtree mutation
- Size-fair subtree mutation
- Node replacement mutation
- Hoist mutation
- Shrink mutation
- Permutation mutation
- Mutating constants at random
- Mutating constants systematically

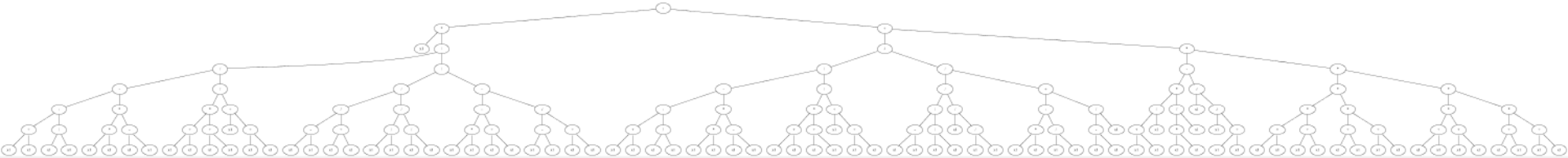
Napredni koncepti

Bloat

- povećanje stabala tijekom evolucije
- ponašanje za gotovo svaki problem
- puno filozofije o
  - uzrocima (fitness, introni, broj ispitnih primjera...)
  - mjerama (parsimony pressure – više cijenimo manje jedinke)
  - izbjegavanju (posebni operatori, prilagođena selekcija, višekriterijska optimizacija)
- pitanje: koja je veza veličine stabla i sposobnosti generalizacije? (*Occam's razor*)

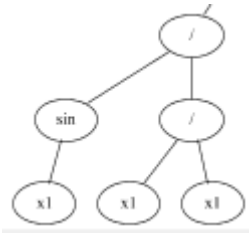
Bloat

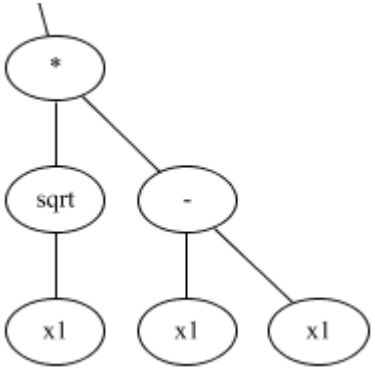




Bloat

- Metode sprječavanja ili popravljanja:
  - *Parsimony pressure*
  - Mnogokriterijska optimizacija – jedan kriterij veličina stabla
  - Posebni operatori križanja i mutacije
  - Editing – uklanjanje nebitnih dijelova u stablu





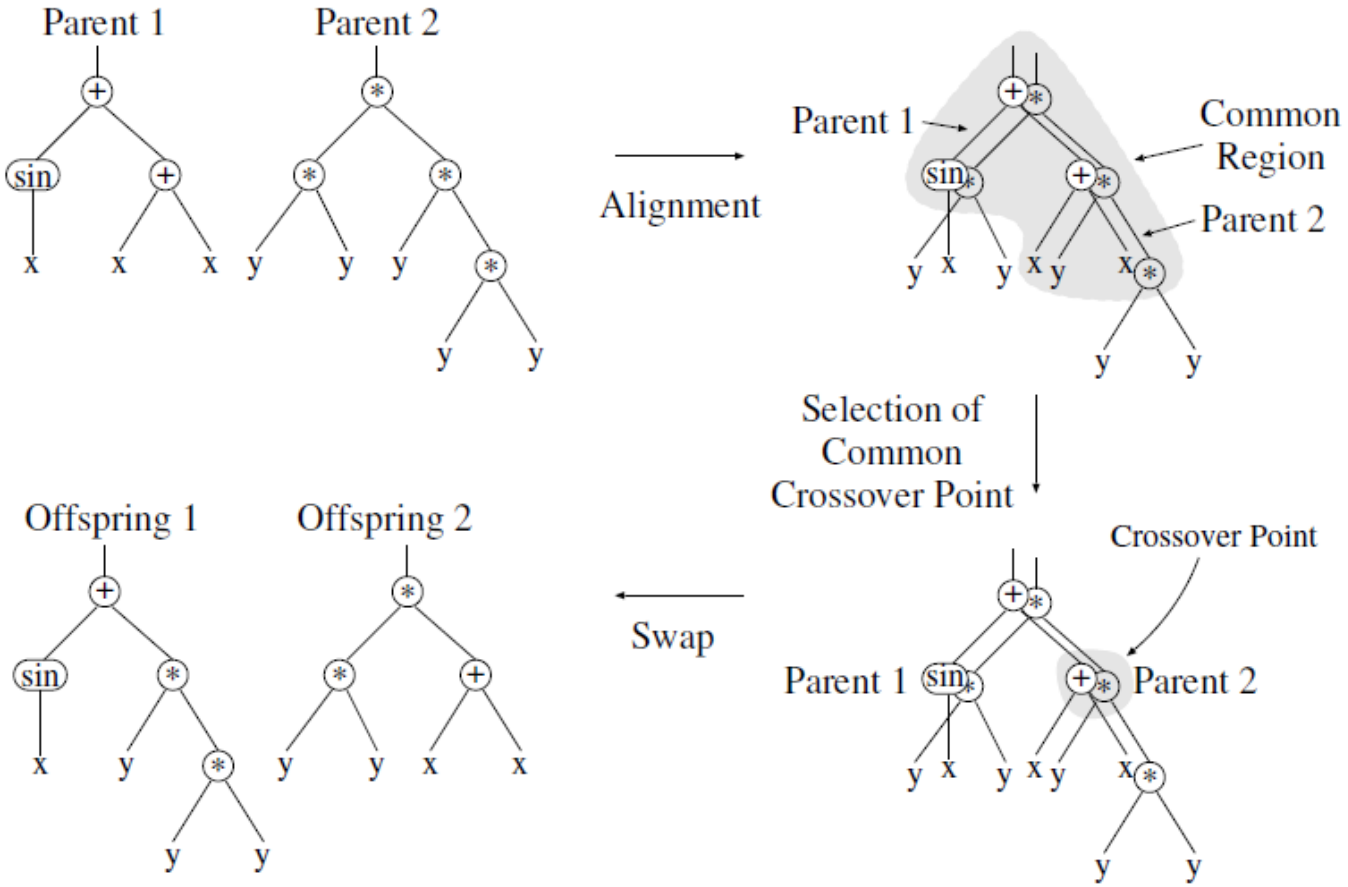
Ephemereal random constants

- Tražimo $f(x) = 0.5 * x^2 + 1.3$
- Slučajna početna vrijednost (prilikom inicijalizacije populacije ili prilikom mutacije)
- Kasnije nepromjenjive, osim za prilagođenu mutaciju (npr. Gaussova)
- Korisno u simboličkoj regresiji!

Linearno skaliranje

- Problem u simboličkoj regresiji: funkcija ima dobar oblik, ali je pomaknuta u odnosu na podatke
- Tražimo $f(x) = 0.5 * x^2 + 1.3$
- Teško pogoditi konstante
- pojednostavimo kao: $f'(x) = a * f(x) + b$
- konstante **a** i **b** tražimo minimizacijom srednje kvadratne pogreške (izvan GP-a)
  - tada GP treba samo skužiti (x^2)

Homologna križanja



Tipovi u GP-u

STGP: Strongly typed GP

- podijeliti terminale i funkcije po tipu (double, int, bool, vektor, matrica...)
- dozvoliti samo sintaksno ispravne programe!

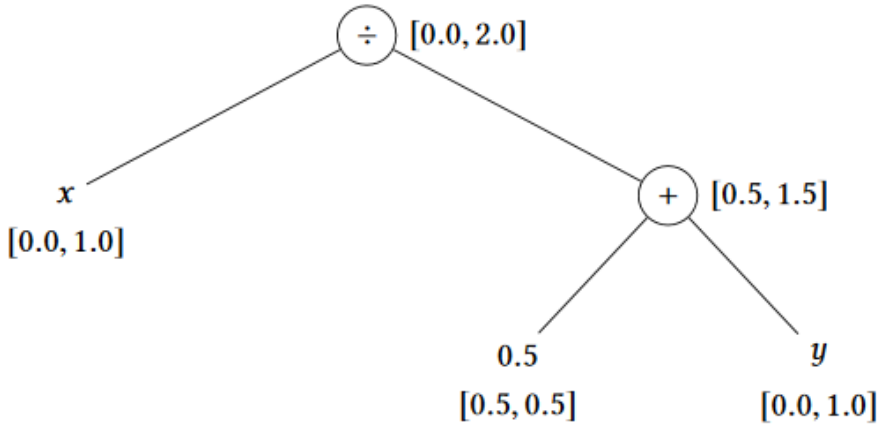
- većina GP aplikacija ipak ‘typeless’

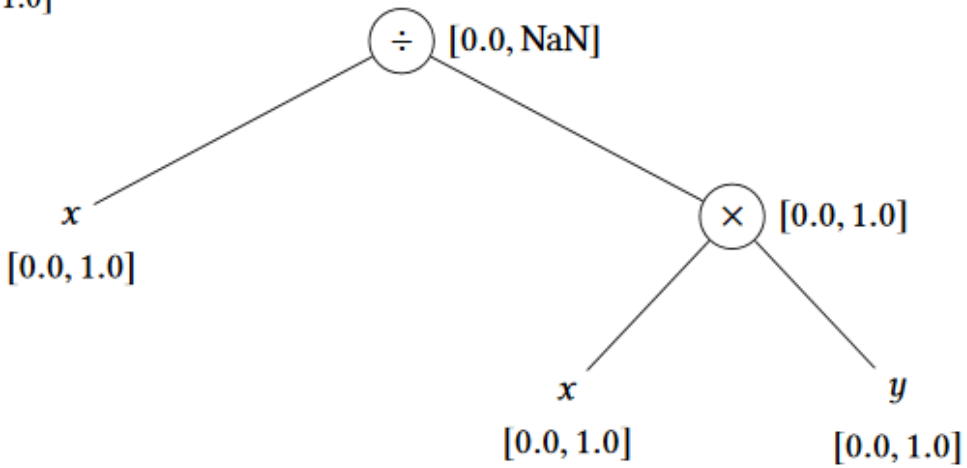
- srodna ideja: semantička (a ne sintaksna) ispravnost!
- DAGP: *dimensionally aware GP*
  - npr: ako je t vrijeme, ne zbrajati t i t^2!
  - studentski rad: http://bib.irb.hr/prikazi-rad?&rad=408164

Intervalna aritmetika

- uobičajeno: ‘zaštićeno’ dijeljenje (x % y)
  - division = fabs(second) > MIN ? first / second : 1.;
  - division = fabs(second) > MIN ? first / second : first;
- bolje: provjera mogućeg raspona vrijednosti svakog čvora (podstabla)
  - samo za simboličku regresiju
  - potrebno poznavati skup ispitnih primjera
  - podstabla nedefinirane vrijednosti se ne evaluiraju

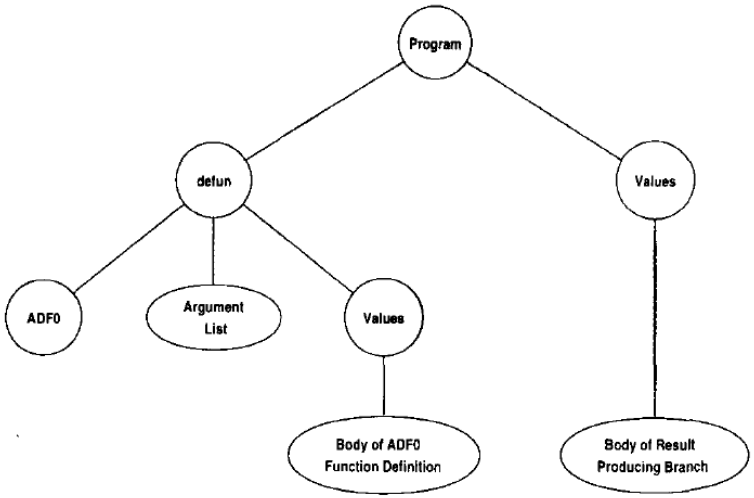
Intervalna aritmetika

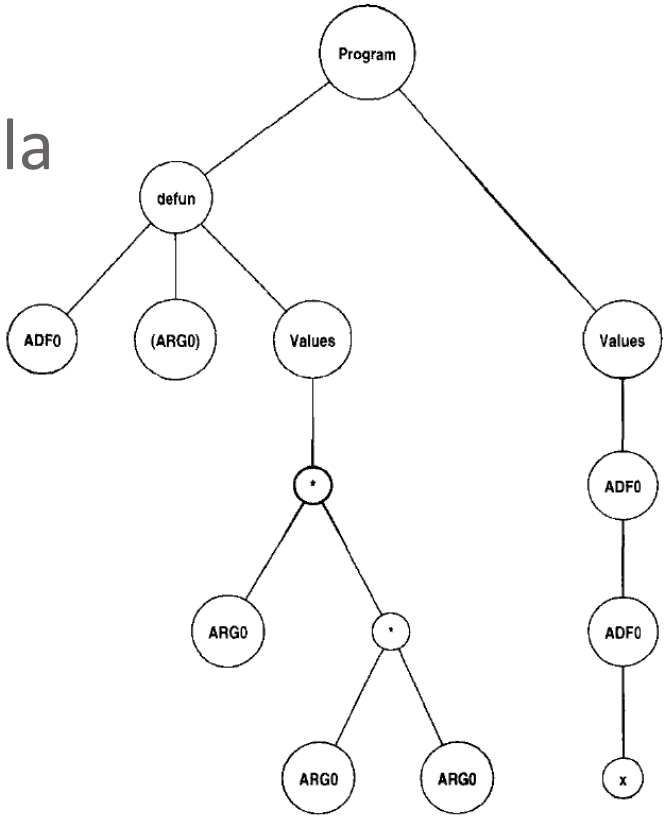




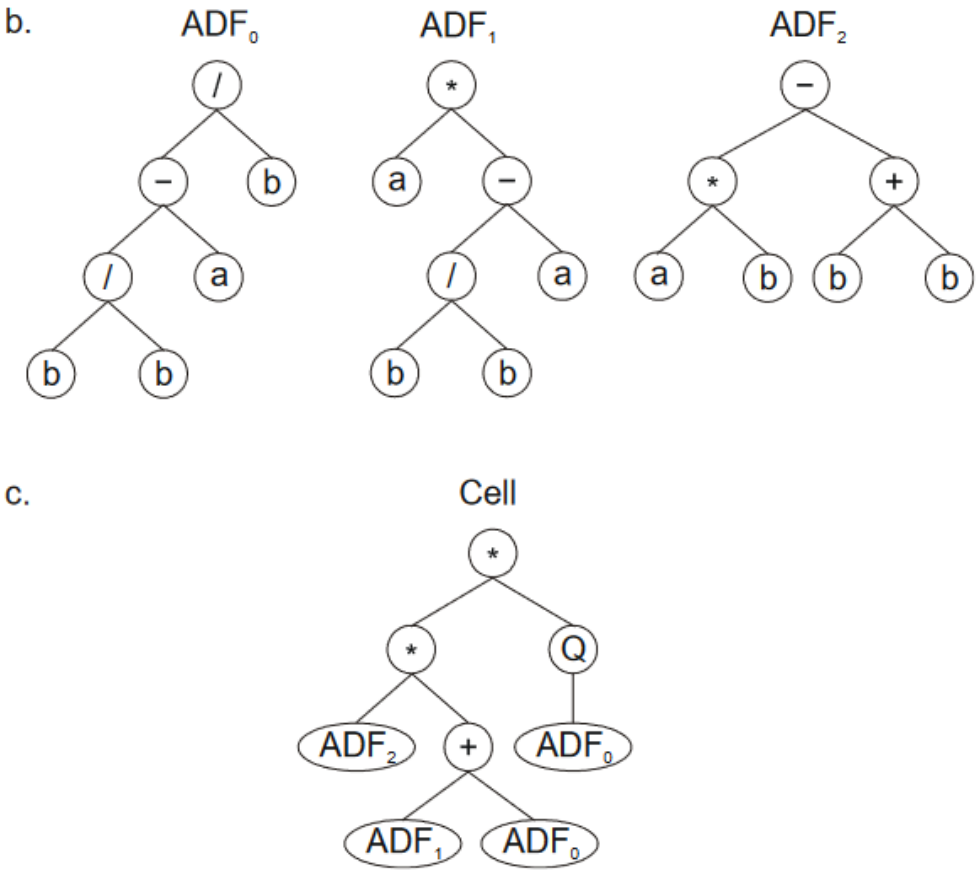
Automatski definirane funkcije

- Potprogrami implementirani kao dodatna stabla
- Posebni skup čvorova
- Poziva se kao funkcija glavnog stabla





Automatski definirane funkcije



Prikazi u genetskom programiranju

Drugi oblici prikaza rješenja

- CGP – Cartesian GP
  - studentski rad: http://bib.irb.hr/prikazi-rad?&rad=519001
- GEP – gene expression programming
  - niz znakova
  - glava niza: funkcije i terminali (zadajemo duljinu)
  - rep niza: do maksimalne duljine (ovisno o broju i n-arnosti funkcija)
  - križanje, mutacija, *transpozicija*
  - studentski rad: http://bib.irb.hr/prikazi-rad?&rad=518985
- grammatical evolution

CGP

- Čvorovi raspoređeni rešetku dimenzija n*m (proizvoljan broj stupaca i redaka)
- Može biti više izlaza
- Čvorovi međusobno povezani
- Neki čvorovi se ne moraju uzimati u obzir pri računanju izlaza
- Rješenje zapisano kao niz cijelih brojeva
  - Brojevi određuju ulaze u pojedini čvor i operaciju koju čvor predstavlja

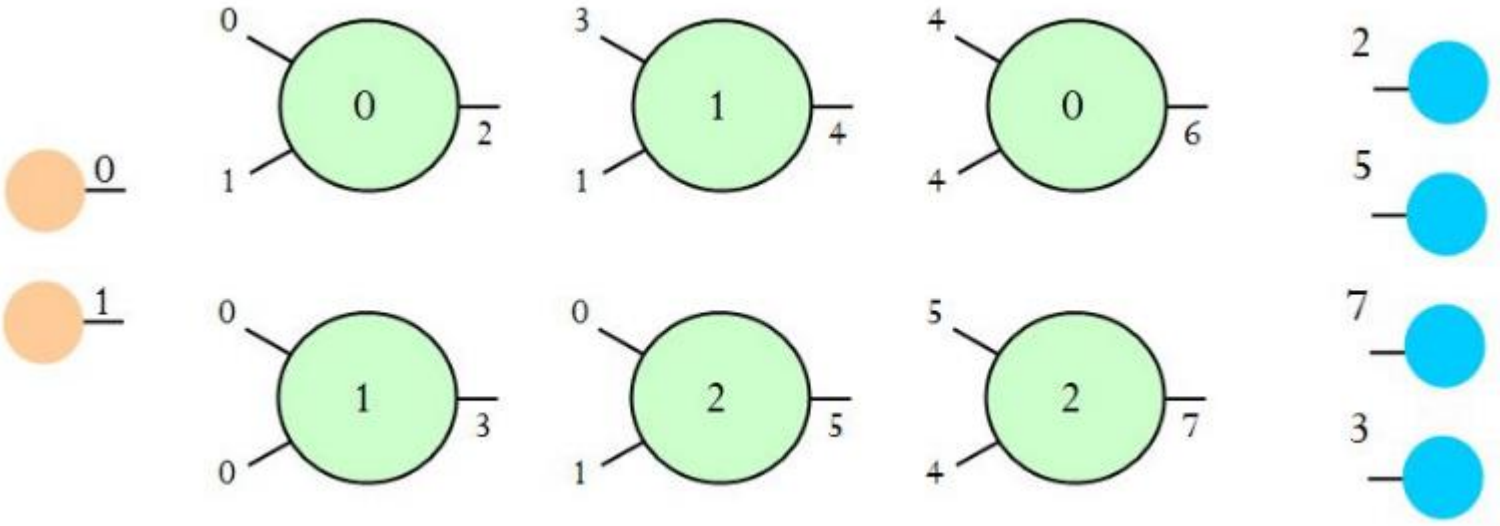
CGP

Indeks	Operator
0	+
1	-
2	*

Genotip

0 0 1 1 0 0 1 3 1 2 0 1 0 4 4 2 5 4 2 5 7 3

Fenotip



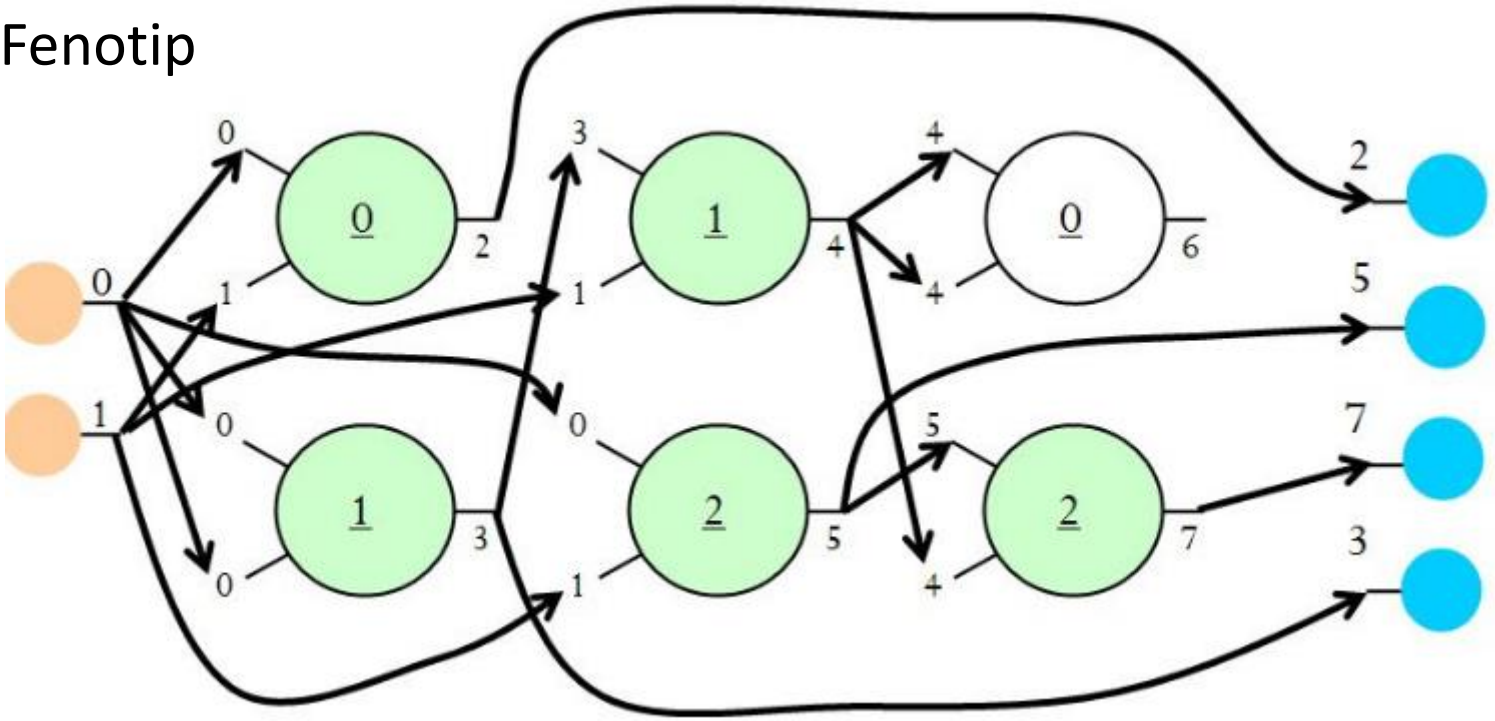
CGP

Indeks	Operator
0	+
1	-
2	*

Genotip

0 0 1 1 0 0 1 3 1 2 0 1 0 4 4 2 5 4 2 5 7 3

Fenotip



CGP - operatori

- Mutacija simbola
- Križanje
  - Jednom točkom prekida
  - Često se uopće ne koristi
- Koriste se male populacije i 1+λ

CGP – primjena

- Razvoj sklopova
- Evolucija neuronskih mreža
- Sve što se je rješavalo GP-om

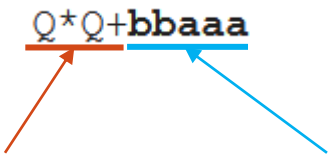
GEP

- Čvorovi zapisani u obliku niza
- Niz se sastoji od više dijelova koje zovemo geni
- Svaki gen se sastoji od glave i repa
  - U glavi se nalaze bilo koji čvorovi
  - U repu se nalaze samo terminalni čvorovi
- Iz niza se gradi stablo koje predstavlja izraz

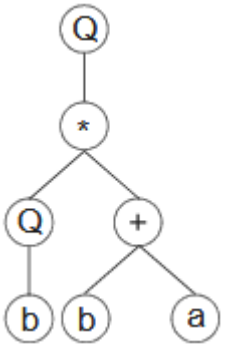
GEP

- Veličinu glave definira korisnik: *h*
- Veličina repa računa se kao: *t=h*(n-1)+1*
  - *n* maksimalni broj argumenata svih funkcijskih čvorova
- Osigurano da se ne može izgraditi neispravni izraz

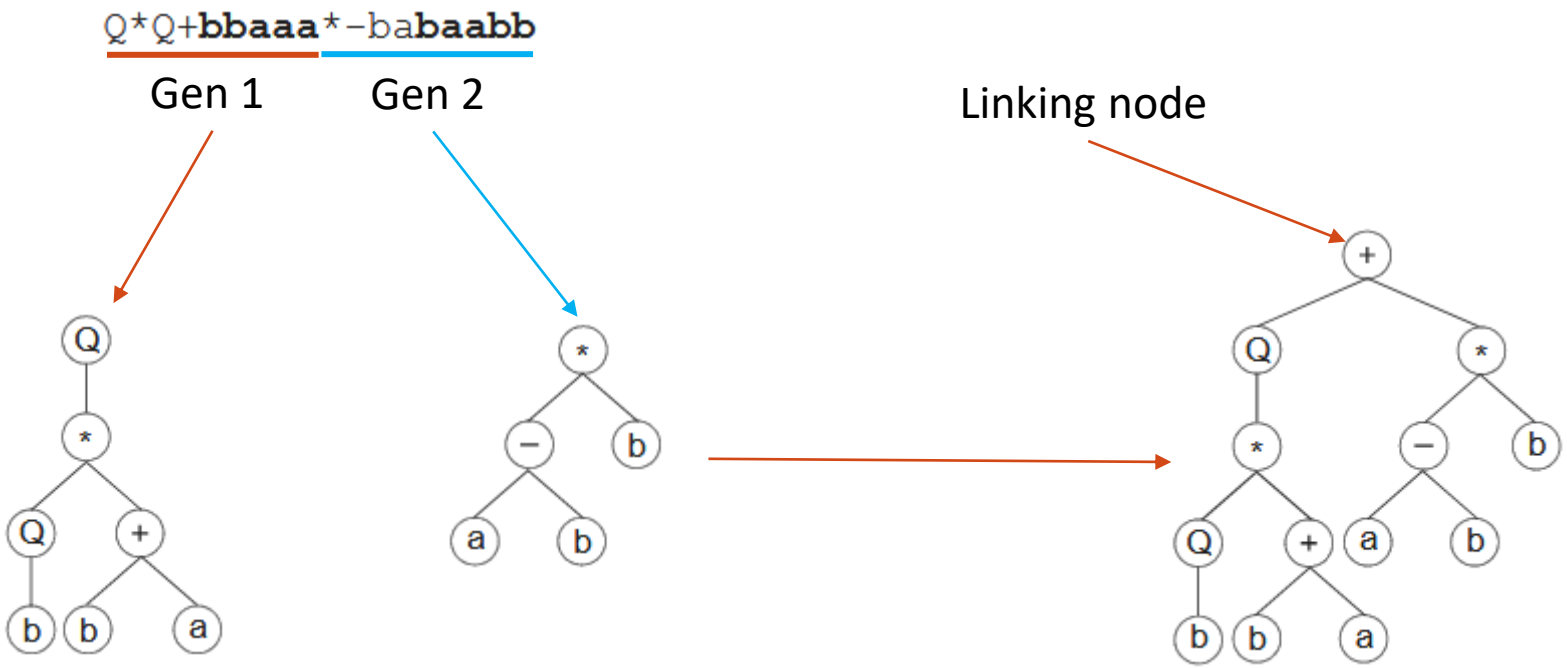
GEP



Glava gena Rep gena



GEP



GEP - operatori

- Mutacija – promjena jednog ili više elemenata u nizu
- Križanje:
  - Jednom točkom prekida
  - Dvije točke prekida
  - Križanje gena
- Transpozicija
  - IS
  - RIS
  - Genska transpozicija

GEP - transpozicije

- IS

Prije transpozicije

+ - pt w MR age w | * PAT w w dd SL pt | w + / pt SL dd age

gen 1 gen 2 gen 3

Nakon transpozicije

+ dd age w MR age w | * PAT w w dd SL pt | w + / pt SL dd age

gen 1 gen 2 gen 3

GEP - transpozicije

- RIS

Prije transpozicije

+ - pt w MR age w | * PAT w w dd SL pt | w + / pt SL dd age

gen 1 gen 2 gen 3

Nakon transpozicije

+ - pt w MR age w | * PAT w w dd SL pt | - pt w pt SL dd age

gen 1 gen 2 gen 3

GEP - transpozicije

- Transpozicija gena

Prije transpozicije

+ - pt w MR age w | * PAT w w dd SL pt | w + / pt SL dd age

gen 1 gen 2 gen 3

Nakon transpozicije

w + / pt SL dd age | + - pt w MR age w | * PAT w w dd SL pt

gen 1 gen 2 gen 3

Grammatical evolution

- Gramatika koja definira program ili izraz koji se treba generirati
  - Završni i nezavršni znakovi, produkcijska pravila, početni nezavršni znak
- Prikaz u obliku niza cijelih brojeva
  - Brojevi definiraju koja se produkcijska pravila koriste

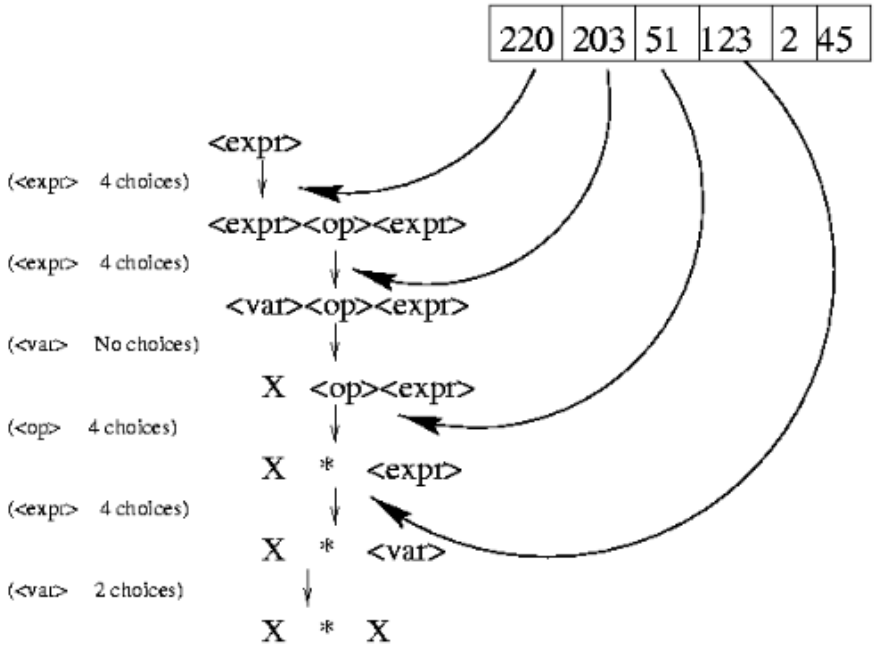
Grammatical evolution

```
(1) <expr> ::= <expr> <op> <expr> (A)
             | ( <expr> <op> <expr> ) (B)
             | <pre-op> ( <expr> ) (C)
             | <var> (D)

(2) <op> ::= + (A)
           | - (B)
           | / (C)
           | * (D)

(3) <pre-op> ::= Sin (A)
               | Cos (B)
               | Tan (C)

(4) <var> ::= X (A)
```



Implementacija

Kako zapisati jedinke?

- Bitne informacije: dubina čvora, broj djece čvora, broj potomaka čvora, maksimalna dubina stabla, dubina stabla u čvoru
- **Kao stablo** (čvorovi s pokazivačima na roditelje/djecu):
  - Predizračunati sve informacije i pospremiti u čvorove
  - On-line računati informacije po potrebi
- Kao polje:
  - Prefiksni zapis
  - Indeksi pozicije djece

Kako ostvariti križanje?

- Kako odabrati točku prekida?
  - Random između 1 i ukupnog broja čvorova
  - Želimo favorizirati funkcijske čvorove (odrediti broj terminala!)
- Izrada nove jedinke:
  - Obavezno kopirati čvorove
  - Paziti na uvjet maksimalne dubine stabla:
    - Dubina trenutnog čvora + dubina stabla u novom čvoru < maksimalna dubina stabla!
    - Ako ne vrijedi probati ponovo odabrati točke (nekoliko puta) ili probati ponovo odabrati roditelje

Kako ostvariti mutaciju?

- Odabrati nasumično čvor (isto kao kod križanja)
- Generirati novo stablo (full ili grow metodom)
  - Paziti da se generira podstablo koje neće narušiti ograničenje maksimalne dubine cijelog stabla!
- Zamijeniti odabrano podstablo staviti novogenerirano podstablo

Zaključak

Zaključak

- GP je primjenjiv na širok spektar problema
- Dosta otvorenih problema i mogućnosti za poboljšanje
- Različiti prikazi rješenja
- Područje koje se aktivno istražuje

Što smo na FER-u radili s GP-om

- razni oblici raspoređivanja
- konstrukcija kombinatoričkih fja za kriptografiju
- problem usmjeravanja vozila (VRP)
- oblikovanje kombinatoričkih sklopova
- upravljanje robotom
- detekcija malignih tvorevina na RTG snimkama pluća
- automatsko rezanje
- dijagnosticiranje plućne embolije
- automatska paralelizacija
- primjena GP u strojnom učenju (status posla u grozdu)
- trgovanje dionicama
- strategije zamjene stranica u straničenju

Reference

- Knjige/stranice
  - Field guide to GP
  - GP Notebook
  - Koza GP site
- Alati
  - ECJ, Watchmaker (Java)
  - ECF, ECF_LAB, ECF_SRM (C++)
  - HeuristicLab (C#)
  - EO, OpenBEAGLE (C++)
  - PyEvolve, DEAP (Phyton)