KUNGL
TEKNISKA
HÖGSKOLAN

Institutionen för Numerisk analys och datalogi

Pseudoslumptal

Johan Håstad
Nada, KTH
johanh@nada.kth.se

Användning av slump i datalogi

- Simuleringar.
- Effektivare algoritmer, t.ex. primalitet.
- Kryptografi.
- Konstruktion av kombinatoriska objekt, t.ex. felrättande koder.

Varifrån får vi slumpen

- Människor.
- Fysisk slump, brus från dioder, mikrofoningångar.

Svårt att göra stora mängder högkvalitativ slump.

Pseudoslumptalsgeneratorer

Tillverka mer slump från lite slump

$\boxed{\text{frö}} \longrightarrow \boxed{\text{slumpserie}}$

Fröet är lite bra slump, “slumpserien” skapas deterministiskt från detta.

Klassisk generator

Linjära kongruensgeneratorer.

m, a, b fixa konstanter, x_0 frö.

$$x_{i+1} \equiv a x_i + b \mod m$$

$$b_i = \begin{cases} 1 & x_i \geq m/2 \\ 0 & x_i < m/2 \end{cases}$$

Inbyggd i många programmeringsspråk.

Ett exempel

$a = 14$, $b = 2$, $m = 257$

$x_0 = 1$

i	0	1	2	3	4	5	6
x_i	1	16	226	82	122	168	41
b_i	0	0	1	0	0	1	0

Utserie 0010010...

Problem

Period högst 257, (i detta fall 256).

Korrelationer t.ex. beroende på att

$$x_i + x_{i+12} \equiv x_{i+1} + x_{i+3} \mod 257$$

för alla i.

Kan avhjälpas!?

Lätt att få period $m - 1$ för primtal m, ta m med ≥ 64 bitar.

Lätt att karaktärisera exakta relationer.

Vad händer när vi använder utserien för att simulera ett kärnkraftverk?

Traditionell utvärdering

I klassisk datalogisk litteratur (före 1982) och en del senare beskriver man ett batteri av statistiska test.

Klarar generatorn dessa är den bra!

Duger den då till kärnkraftsverket?

Modern definition

En pseudoslumptalsgenerator (PSG) är bra om den klarar alla statistiska test som kan utföras effektivt.

Klara = Ge (nästan) samma fördelning på utdata som om vi hade använt riktig slump.

Ett statistiskt test

Simulera ett kärnkraftverk och se om vi får härdsmälta.

Om simuleringen kan göras effektivt så duger en bra PSG, om den inte kan göras effektivt så kunde vi inte göra den ens med riktig slump.

Ett annat statistiskt test

Skapa en felrättande kod baserat på slump-serien. Använd den att kommunicera med.

Paradox: Med en PSG får vi en kod som kanske inte har rätt egenskaper, men i så fall hinner vi inte upptäcka det.

Kort sagt

Utdata från en bra PSG är i praktiken lika bra som riktig slump.

Finns de?

Farligt statistiskt test

Antag att fröet har n bitar och utdata m bitar, $m >> n$.

Test: Givet m bitar är det ett möjligt utdata av vår generator?

Alltid "ja" på utdata från generatorn, "ja" med sannolikhet högst 2^{n-m} på riktig slump.

Kalla detta "Möjliga utdata testet".

Slutsats

För att vår generator ska ha en möjlighet att vara bra så kan inte “Möjliga utdata testet” utföras effektivt.

Detta ska vara ett beräkningsmässigt svårt problem.

Effektivt beräkningsbar

Teori: Tid polynomiell i indatas längd, klassen P av effektivt beräkningsbara funktioner. Tid n^2, n^7 etc

Praktik: Kanske 2^{50} operationer, i varje fall mindre än 2^{80}.

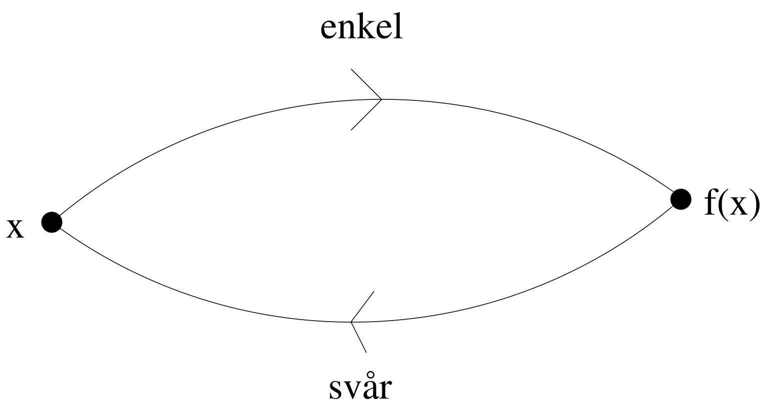
Komplexitetsteorin kan inte visa att “Möjliga utdata testet” är svårt för någon av dessa definitioner för någon effektiv generator.

Komplexitetsteoretiskt krävs att NP$\neq$P i teori fallet.

Bäst möjligt

Visa att en generator är bra baserat på något rimligt antagande.

Enkelriktad funktion



Ta slumpvis x, beräkna $f(x) = y$.

Invertera är att hitta något x' så att $f(x') = y$.

Borde finnas, men vi är långt från att visa detta.

I teorin

Sats: Det finns enkelriktade funktion om och endast om det finns bra PseudoSlumptalsGeneratorer.

Här defineras “effektiv” som att något går att göra i polynomiell tid.

Halvmodern generator

Blum, Micali (1982):

p stort primtal, g generator av Z_p, x_0 frö.

$$x_{i+1} \equiv g^{x_i} \mod p$$

$$b_i = \begin{cases} 1 & x_i \geq p/2 \\ 0 & x_i < p/2 \end{cases}$$

Relaterat beräkningsproblem

Diskreta logaritmer.

Givet primtal p, generator g och tal y, finn x så att

$$g^x \equiv y \bmod p.$$

Exempel $p = 257$, $g = 14$ och $y = 54$ så gäller

$$14^{88} \equiv 54 \bmod 257$$

Känt beräkningsproblem som antas vara svårt.

Hur svårt är diskreta logaritmer

Kusinen från landet till faktorisering av heltal.

Ungefär samma algoritmer fungerar men det blir lite tyngre och det finns mycket mindre praktiska körningar på detta problem.

Om p har 512 bitar, drygt 150 siffror är det nog lösbart för större organisation.

Ingen har idag en changs på 1024 bitar.

Teoretisk sats

Om diskreta logaritmer är ett svårt beräkningsproblem så är BM-generatorn bra.

Praktisk sats

Om vi använder BM-generator på frölängd n för att producera m bitar och har ett statistiskt test som går i tid T och skiljer utdata från riktig slump med fördel ϵ så kan vi lösa diskreta logartimer på instanser av storlek n i tid

$$\approx \frac{nm^3}{\epsilon^3} T$$

Omvändingen

Anta att diskreta logaritment på indata av storlek 10000 kräver 2^{240} operationer.

BM-generatorn som expenderar 10000 bitar till 10^9 bitar kommer, inom $\epsilon = 10^{-6}$, ge samma utdatafördelning som riktig slump på alla statistiska test som gör högst 2^{40} operationer.

Finns alternativ

Andra explicita antaganden på enkelriktade funktioner ger andra bra generatorer.

Man kan t.ex. anta att kända kryptosystem ger bra enkelriktade funktioner.

Bevisbara simuleringar

Om vi använder utdata från BM-generatorn för att köra en simulering och observerar ett visst fenomen så antingen:

- Hade vi extrem otur (som vi kunde haft med riktig slump).
- Är diskreta logaritmer lättare än vi tror.
- Finns fenomenet på riktigt.

Öppen fråga

Är linjära kongruens generatorn definerad i början bra om m är tillräckligt stort?

$$x_{i+1} \equiv ax_i + b \mod m$$

$$b_i = \begin{cases} 1 & x_i \geq m/2 \\ 0 & x_i < m/2 \end{cases}$$

Matar vi ut många bitar per iteration som serie är den inte bra men status för denna variant är okänd.

Slutord

Finns teori för pseudoslump.

Kräver beräkningsmässiga antaganden.