Problema Ghinion

Propusă de: Tamio-Vesa Nakajima

În primul rând, mai degrabă decât să numărăm efectiv câte șiruri de caractere valide există, vom calcula probabilitatea că șirul de caractere este unul valid dacă fiecare caracter este ales uniform aleator dintre `0`, `1`, `^` și `=`. La final vom înmulți cu 4^n.

Să fixăm mai întâi pentru fiecare caracter dacă el va fi o cifră sau nu. Condiționând asupra acestei fixări, care este probabilitatea ca șirul să fie bun? De exemplu, putem fixa că șirul nostru este de forma `???^????=??^?`, unde `?` reprezintă un 0 sau un 1. În acest caz, în realitat primele trei cifre și ultima cifră *nu contează*: indiferent cum ar fi ea fixată, probabilitatea ca șirul să fie valid este același. Mai mult, tot ce vine după egal *nu contează* — singurele cifre care contează sunt cele patru `?`-uri consecutive: oricum ar fi fixate celelalte cifre, există exact un mod de a le fixa pe acestea care să facă șirul într-o ecuație validă.

Aceasta idee se poate generaliza la secvențe de caractere cu mai multe `=`-uri. De exemplu, să presupunem că fixăm că doar `?`-urile din următorul șir pot deveni cifre: `??^???=??=????^?^???`. Singurele cifre care contează sunt cele care formează cel mai lung număr dintr-o expresie formată din numere și `^`-uri, mai puțin cel mai scurt număr de acest fel. Astfel aici contează următoarele: `??^???=??=????^?^???`. Oricum ar fi fixate celelalte cifre, există un mod unic de a fixa cifrele cu roșu care duc la un șir de caractere valid.

Astfel, dacă fixăm un șablon de `=`-uri, `^`-uri și cifre, putem calcula probabilitatea ca, dacă setăm cifrele uniform aleator, să creăm un șir valid de caractere: aceasta va fi 2^{-c} unde c este suma din lungimile șirurilor maxime de cifre consecutive din fiecare expresie conținând doar cifre si `^`-uri, mai puțin cea mai scurtă lungime. Astfel, rezultatul va fi suma, pentru fiecare șablon, din aceasta putere a lui 2 înmulțită cu probabilitatea ca acel șablon să apară (care este $2^{-\text{numărul de cifre}}4^{-\text{numărul de non-cifre}}$). Partea dificilă a problemei acum este calcularea acestei sume. Acest lucru îl vom face cu ajutorul programării dinamice.

Fie p_{ij} probabilitatea ca un șir de i `0`-uri, `1`-uri și `^`-uri alese uniform aleator să aibă cea mai lunga secvență de cifre consecutive de lungimea j. Observăm că $p_{ii} = (2/3)^i$, iar, pentru $j \leq i-2$,

$$p_{ij} = (2/3)^j \sum_{k \leq j} p_{i-j-1,k} + \sum_{k<j} (2/3)^k p_{i-k-1,j}.$$

Primul termen din sumă reprezintă cazul în care șirul se termină cu j `0`-uri sau `1`-uri consecutive, iar al doilea când se termină cu mai puțin de j astfel de caractere. Observăm că aceste valori se pot calcula în timp cubic. (Toate valorile pe care nu le-am menționat până acum sunt 0.)

Acum, fie q_{ij} probabilitatea ca un șir de lungime i să fie valid, și ca cea mai scurtă lungime maximă de cifre consecutive dintr-o expresie formată doar din cifre și `^` sa fie j.[1] Observăm că

$$q_{ij} = (3/4)^i p_{ij} + \sum_{k=1}^{i-2} (1/4)(3/4)^k \left(p_{kj} \left(\sum_{\ell \leq j} (1/2)^\ell q_{i-k-1,\ell} \right) + \left(\sum_{\ell < j} (1/2)^\ell p_{k\ell} \right) q_{i-k-1,j} \right)$$

Primul termen reprezintă cazul în care șirul nu conține `=`. În suma de mai sus, k reprezintă lungimea ultimei secvențe formate doar din `0`, `1` si `^`. Primul termen din interiorul sumei este cazul în care lungimea cea mai scurtă maximă este atinsă în ultima secvență formată doar din `0`, `1` și `^`, iar al doilea este cazul când acest lucru nu se întâmplă. Observăm că și această sumă se poate calcula în timp cubic, cu ajutorul unor sume pe prefix ce ne permit să aflăm $\sum_{\ell \leq j} (1/2)^\ell q_{i-k-1,\ell}$ și $\sum_{\ell<j} (1/2)^\ell p_{k\ell}$ în timp constant. Rezultatul este apoi $q_{n1} + \ldots + q_{nn}$.

[1] Atenție, aici nu este vorba de probabilitatea ca șirul să fie valid *condiționat pe* acea lungime.