



HCMUTE

Chapter 11: A Simple Program Whose Proof Isn't

Trần Tuấn Anh - 21110369
Hồ Khánh Đăng - 21110876
Lê Hữu Huy - 21110471
Cấu trúc dữ liệu và Giải thuật
Năm học: 2022-2023

Mục tiêu chính

- Hiểu quá trình chuyển đổi giữa Decimal Fractions và Fixed-Point Binary
- Chứng minh những chương trình con thực hiện chuyển đổi

Chương trình TEX

- TEX có thể xem là ngôn ngữ lập trình do Donald Knuth phát minh.
- Cho phép tạo ra văn bản lượng cao với ít công sức và cung cấp một hệ thống sắp chữ cho ra cùng một kết quả trên mọi máy tính.
- Hoạt động dựa trên bội số của 2^{16} .

Thuật ngữ

- Decimal Fractions: phân số thập phân.
- Fixed-Point Binary: số nhị phân chấm động.

1 Converting Decimal Fractions to Fixed-Point Binary

- Chuyển đổi phân số thập phân thành số nhị phân chấm động.
- Đầu vào** là phần phân số thập phân

$.d_1d_2d_3\dots d_k$

- Đầu ra** là số nhị phân chấm động:

$$n = \left[2^{16} \sum_{j=1}^k \frac{d_j}{10^j} + \frac{1}{2} \right]$$

Mô phỏng chương trình P1 chạy bằng C++:

```
long long fixedPointBinary(int D[],int k)
{
    int l=min(k,17);
    double m=0;
    do{
        m = (131072*D[l-1]+m)/10;
        l--;
    }while(l!=0);
    long long n=(m+1)/2;
    return n;
}
```

2 Converting the Other Way

2.1 Converting Decimal Fractions to Fixed-Point Binary

- Chuyển đổi số nhị phân chấm động sang phần phân số thập phân (ngược lại chương trình P1).
- Đầu vào** là số nguyên $n \in [0, 2^{16}]$
- Đầu ra** là phần phân số thập phân: $.d_1d_2d_3\dots d_k$
- Được xử lý dựa trên thuật toán

$$D = \left\lceil 10^5 \frac{n}{2^{16}} + \frac{1}{2} \right\rceil$$

Mô phỏng chương trình P2 chạy bằng C++:

```
void decimalFraction()
{
    int n; cin>>n;
    int j=0;
    int s = 10*n;
    int t = 10;
    int d[100];
    while (s>=t)
    {
        if (t>65536) s = s + (65536/2);
        j++; d[j] = s/65536;
        s = 10*(s%65536); t*=10;
    }
    int k = j;
    cout<<k<<endl;
    for (int i=1;i<=k;i++) cout<<d[i]<<" ";
}
```

3 Germs of a Proof

- Đây là phần cơ sở của thuật toán.

- Chứng minh chương trình P2
- Cách hoạt động của chương trình P2
- Lý do chọn $k \leq 5$

4 Completion of the Proof

Tại sao lại có câu điều kiện ở P2

If $t > 65536$ then $s = s + 32768 - (t \text{ div } 2)$

- Điều này được thực thi khi $j=4$, cụ thể là khi $t = 10^5$, vì $t = 10^{j+1}$ và vòng lặp không bao giờ được thực thi khi $j > 4$.
- Khi $j=4$ ở đầu vòng lặp thì chúng ta tính d_5 . Và $d_5 > 0$ nên là ta có $s > 65536$. Nên ta có giá trị mới của s trong chương trình P2.

$.d_1d_2d_3\dots d_k$ bằng $\frac{n}{2^{16}}$ sẽ được làm tròn thành 5 chữ số thập phân như mong muốn.

- Mục tiêu là chỉ ra thuật toán P2 đã chứng minh là đúng.

5 Closing Remarks

- Đây là phần mở rộng thuật toán không còn dựa trên trường hợp 16-bit.

Mô phỏng chương trình chạy bằng C++:

```
void extension()
{
    ll n,m; cin>>n>>m;
    ll j=0;
    ll s = 10*n+4;
    ll t = 10;
    ll d[100];
    while (s>=t)
    {
        if (t>m) s = s+1 + (m/2)-t/2;
        j++; d[j] = s/m;
        s = 10*(s%m)+9; t*=10;
    }
    ll k = j;
    cout<<k<<endl;
    for (ll i=1;i<=k;i++) cout<<d[i]<<" ";
}
```

6 Applications of Algorithms

- Ứng dụng để chuyển số thập phân có dấu chấm sang số nhị phân đơn giản nhất, là một cách khá hay ứng dụng cho ngành toán học.
- Ứng dụng thuật toán trong chương trình TEX là tiền thân của chương trình LaTeX sau này.
- Ứng dụng trong lập trình các ngôn ngữ lập trình hay trong tin học về hàm làm tròn: có thể lấy được các chữ số thập phân sau dấu phẩy của một phép tính như ý muốn.
- Có thể ứng dụng rộng rãi cho lập trình máy tính cầm tay casio
- Ứng dụng rộng rãi trong các app để hiện thị ra kết quả vừa ngắn và xấp xỉ gần đúng nhất
- Ứng dụng chuyển một số thập phân sang nhị phân bằng thuật toán trong bài báo