ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC - KỸ THUẬT MÁY TÍNH

Mô hình hóa toán học -Mathematical Modeling 191

Assignment

The minimal-sum section

GVHD: GV. Nguyễn An Khương

Sinh Viên:

1710869 - Nguyễn Tiến Dũng

Mục lục

Danh sách hình vẽ

1 Minimal section

1.1 Định nghĩa

Cho mảng nguyên gồm n phần tử $a[0], a[1], ..., a[n-1]$.Một *section* của a là một phần lên tục $a[i], ..., a[j]$ với $0 \leq i \leq j < n$.Ta viết $S_{i,j}$ là tổng của các phần tử của section : $a[i] + a[i+1] + ... + a[j]$. Một *minimal-sum section* là section $a[i], ..., a[j]$ có $S_{i,j}$ nhỏ hơn hoặc bằng tất cả các tổng $S_{i',j'}$của các section khác $a[i'], ..., a[j']$ của a

1.2 Giải thuật tìm minimal-sum

1.2.1 Giải pháp

Chúng ta hoàn toàn có thể liệt kê hết tất cả các section của mảng và duyệt hết tất cả các section ấy để tìm ra section có tổng nhỏ nhất. Độ phức tạ trong trường hợp xấu nhất của giải thuật này sẽ là n^3, chi phí cho giải thuật này quá cao. Thay vào đó chúng ta có giải thuật *min_sum*,giải thuật cho phép ta có được kết quả bài toán với độ phức tạp chỉ là $O(n)$, ý tưởng giải thuật là sử dụng hai biến ,một biến để lưu tổng nhỏ nhất tìm được sau mỗi lần duyệt , biến thứ 2 dùng để lưu tổng của section nhỏ nhất của các section có điểm kết thúc là điểm đang duyệt mã giả của giải thuật :

```
k=1;
t=a[0];
s=a[0]
while (k!=n){
    t=min(t+a[k],a[k]);
    s=min(s,t);
    k = k+1;
}
```

1.2.2 Hiện thực

Chương trình được hiện thực bằng ngôn ngữ Python (num là tên testcase)

:

```
def min_sum(num) :
    a = load_testcase(num)
    k=1
    t= a[0]
    s= a[0]
    while (k!=len(a)) :
        t= min(t + a[k] , a[k])
        s= min(s,t)
        k = k+1
    return s
```

1.2.3 Chứng minh giải thuật min_sum là partial-corect

Nếu s là minimal sum của mảng thì s phải nhỏ hơn hoặc bằng tất cả các tổng section của mảng và có 1 section có tổng bằng s.
Yêu cầu của giải thuật gồm có 2 phần, viết dưới dạng hoare triples.
$S1 : (|\top|)\ min\_sum\ (|\forall i, j(0 \leq i \leq j < n \rightarrow s \leq S_{i,j}|)$
Có nghĩa là khi chương trình kết thúc, s nhỏ hơn hoặc bằng tổng của bất cứ section nào bên trong mảng
$S2 : (|\top|)\ min\_sum\ (|\exists i, j(0 \leq i \leq j < n \wedge s = S_{i,j}|)$
có nghĩa là có 1 section có tổng là s.

1.2.3.a Chứng minh S1

Đầu tiên chúng ta chứng minh $S1$. Chúng ta tìm 1 invariant phù hợp và invariant trong trường hợp này sẽ là

$$Inv1(s,k) = \forall i, j(0 \leq i \leq j < k \rightarrow s \leq S_{i,j} \tag{1}$$

Dễ thấy Invariant này không đủ mạnh vì đã bỏ qua biến t, biến lưu tổng nhỏ nhất của các section trước $a[k]$,ta tìm thêm invariant biểu dễn cho tính chất này của biến t

$$Inv2 = \forall i(0 \leq i < k \rightarrow t \leq S_i, k-1) \tag{2}$$

Invariant cuối cùng của chúng ta là hội của 2 invariant trên

$$Inv1(s,k) \wedge Inv2(t,k) \tag{3}$$

Ta có Tableau proof cho S1 của `Min_sum` :

$(\|\top\|)$	
$(\|Inv1(a[0],1) \wedge Inv2(a[0],1)\|)$	Implied
`k=1;`	
$(\|Inv1(a[0],k) \wedge Inv2(a[0],k)\|)$	Assignment
`t=a[0];`	
$(\|Inv1(a[0],k) \wedge Inv2(t,k)\|)$	Assignment
`s=a[0]`	
$(\|Inv1(s,k) \wedge Inv2(t,k)\|)$	Assignment
`while (k!= n) {`	
$(\|Inv1(s,k) \wedge Inv2(t,k) \wedge k \neq n\|)$	Invariant Hyp. $\wedge$ guard
$(\|Inv1(min(s,min(t+a[k],a[k]),k+1) \wedge Inv2(min(t+a[k],a[k]),k+1)\|)$	Implied (`Lemma 1`)
`t=min(t+a[k],a[k]);`	
$(\|Inv1(min(s,t),k+1) \wedge Inv2(t,k+1)\|)$	Assignment
`s=min(s,t);`	
$(\|Inv1(s,k+1) \wedge Inv2(t,k+1)\|)$	Assignment
`k=k+1;`	
$(\|Inv1(a[0],k) \wedge Inv2(t,k)\|)$	Assignment
}	
$(\|Inv1(a[0],k) \wedge Inv2(t,k)\| \wedge \neg\neg(k=n)\|)$	Partial-while
$(\|Inv1(s,n)\|)$	Implied

Lemma 1 Cho s và t là 2 số nguyên bất kỳ, n là độ dài của mảng a và k là một chỉ số của mảng đó trong khoảng $0 < k < n$. Khi đó $(|Inv1(s,k) \wedge Inv2(t,k) \wedge k \neq n|)$ dẫn đến :

1. $Inv1(min(s,min(t+a[k],a[k]),k+1)$
2. $Inv2(min(t+a[k],a[k]),k+1)|)$

CHỨNG MINH :

1. chọn bất kỳ i với $0 \leq i < k+1$; nếu $i < k$ thì$S_{i,k} = S_{i,k-1} + a[k]$,vậy ta chỉ cần chứng minh $min(t+a[k],a[k]) < S_{i,k-1} + a[k]$; và ta có thể đạt được điều này bằng cách cộng $a[k]$ vào hai về $t \leq S_{i,k-1}$.Với $i = k$ thì $S_{i,k} = a[k]$
2. chọn bất kỳ i và j với $0 \leq i < k+1$; chúng ta chứng minh $min(s,t+a[k],a[k]) \leq S_{i,j}$, với $i \leq j < k$ thì kết quả là hiển nhiên

1.2.3.b Chứng minh S2 (bài tập số 23)

Tiếp theo chúng ta chứng minh S2 . Invariant đầu tiên của S2 sẽ là

$$Inv3(s,k) = \exists i,j(0 \leq i \leq j < k \wedge s = S_{i,j} \tag{4}$$

Vì invariant này bỏ qua biến t, ta còn có thêm 1 invariant :

$$Inv4(t,k) = \exists i(0 \leq i < k \wedge t = S_{i,k-1} \tag{5}$$

bảng chứng minh :

$(\|\top\|)$	
$(\|Inv3(a[0],1) \wedge Inv4(a[0],1)\|)$	Implied
`k=1;`	
$(\|Inv3(a[0],k) \wedge Inv4(a[0],k)\|)$	Assignment
`t=a[0];`	
$(\|Inv3(a[0],k) \wedge Inv4(t,k)\|)$	Assignment
`s=a[0]`	
$(\|Inv3(s,k) \wedge Inv4(t,k)\|)$	Assignment
`while (k!= n) {`	
$(\|Inv3(s,k) \wedge Inv4(t,k) \wedge k \neq n\|)$	Invariant Hyp. $\wedge$ guard
$(\|Inv3(min(s,min(t+a[k],a[k]),k+1) \wedge Inv4(min(t+a[k],a[k]),k+1)\|)$	Implied (`Lemma 2`)
`t=min(t+a[k],a[k]);`	
$(\|Inv3(min(s,t),k+1) \wedge Inv4(t,k+1)\|)$	Assignment
`s=min(s,t);`	
$(\|Inv3(s,k+1) \wedge Inv4(t,k+1)\|)$	Assignment
`k=k+1;`	
$(\|Inv3(a[0],k) \wedge Inv4(t,k)\|)$	Assignment
}	
$(\|Inv3(a[0],k) \wedge Inv4(t,k)\| \wedge \neg\neg(k=n)\|)$	Partial-while
$(\|Inv3(s,n)\|)$	Implied

Lemma 2 Cho s và t là 2 số nguyên bất kỳ, n là độ dài của mảng a và k là một chỉ số của mảng đó trong khoảng $0 < k < n$. Khi đó $(|Inv1(s,k) \wedge Inv2(t,k) \wedge k \neq n|)$ dẫn đến :

1. $Inv1(min(s, min(t + a[k], a[k]), k + 1)$
2. $Inv2(min(t + a[k], a[k]), k + 1)|)$

CHỨNG MINH :

1. vì hàm $min(s, min(t + a[k], a[k]))$ sẽ trả về s hoặc t+a[k] hoặc a[k].Vì tồn tại tổng $S_{i,j} = s$ với $0 \leq i \leq j < k$ nên chắc chắn sẽ tồn tại $S_{i,j} = s$ với $0 \leq i \leq j < k + 1$,tương tự với$min(s, min(t + a[k], a[k])) = t + a[k]$ vì tồn tại 1 $i, 0 \leq i < k, S_{i,k-1} = t$, cộng a[k] vào 2 về ta sẽ có t+a[k]=$S_{i,k}$ nên có $\exists i, j, 0 \leq i < k + 1, S_{i,j} = t + a[k]$(với j=k). với trường hợp$min(s, min(t + a[k], a[k])) = a[k]$ thì điều cần chứng minh là đúng.
2. như đã chứng minh ở phần trên , $min(t + a[k], a[k])$ trả về $t + a[k]$ hoặc $a[k]$, khi trả về t+a[k], theo phần trên $\exists i, 0 \leq i < k, t + a[k] = S_{i,k}$,và $a[k] = S_{k,k}$. Vậy điều 2 đúng

1.2.3.c Tổng kết lại (bài tập 25c)

Chúng ta sử dụng rule ở bài tập 25 $\dfrac{(|\phi|)\ \mathrm{C}\ (|\psi_1|) \quad (|\phi|)\ \mathrm{C}\ (|\psi_2|)}{(|\phi|)\ \mathrm{C}\ (|\psi_1 \wedge \psi_2|)}$ Conj theo các chứng minh trên ta được :

1. $(|\top|)$ min_sum $(|S1|)$
2. $(|\top|)$ min_sum $(|S2|)$

Áp dụng luật chứng minh trên : $\dfrac{(|\top|)\ \text{min\_sum}\ (|S1|) \quad (|\top|)\ \text{min\_sum}\ (|S2|)}{(|\top|)\ \text{min\_sum}\ (|S1 \wedge S2|)}$ Conj

Và $(\top)$ min_sum $(|S1 \wedge S2|)$ nghĩa là sau khi thực hiện giải thuật min_sum ta có biến s lưu giá trị của tổng của section nhỏ nhất .Giải thuật này đúng với partial correctness.

1.2.4 Chứng minh S1 và S2 là total-correct

Yêu cầu của giải thuật chúng ta đã định nghĩa ở phần chứng minh partial-correctness gồm có 2 phần, viết dưới dạng hoare triples.

$S1 : (|\top|)\ min\_sum\ (|\forall i, j(0 \leq i \leq j < n \rightarrow s \leq S_{i,j}|)$

$S2 : (|\top|)\ min\_sum\ (|\exists i, j(0 \leq i \leq j < n \wedge s = S_{i,j}|)$

1.2.4.a Chứng minh S1

Đầu tiên chúng ta chứng minh $S1$.Invariant của S1 như trên chúng ta đã tìm được ở phần partial-correctness :

$$Inv1(s,k) \wedge Inv2(t,k)$$

với $Inv1(s,k) = \forall i,j(0 \leq i \leq j < k \rightarrow s \leq S_{i,j}$ và $Inv2(t,k) = \forall i(0 \leq i < k \rightarrow t \leq S_i, k-1)$
Gọi P là đoạn chương trình trong vòng lặp while

```
1    t=min(t+a[k],a[k]);
     s=min(s,t);
3    k = k+1;
```

Ta cần tìm một biểu thức E sao cho $(|0 \leq E = E_0|)$ P $(|0 \leq E < E_0|)$ và dễ thấy

$$E = n - k \tag{6}$$

là 1 biểu thức thỏa mãn điều kiện này vì

$$\begin{array}{ll} (|0 \leq n-k = E_0|) & \\ (|0 \leq n-(k+1) < E_0|) & \text{Implied} \\ \ldots & \\ \texttt{k=k+1;} & \\ (|0 \leq n-k < E_0|) & \text{Assignment} \end{array}$$

ta có bảng chứng minh cho S1

$(\|0 < n\|)$	
$(\|Inv1(a[0], 1) \wedge Inv2(a[0], 1) \wedge 0 \leq n - 1\|)$	Implied
`k=1;`	
$(\|Inv1(a[0], k) \wedge Inv2(a[0], k) \wedge 0 \leq n - k\|)$	Assignment
`t=a[0];`	
$(\|Inv1(a[0], k) \wedge Inv2(t, k) \wedge 0 \leq n - k\|)$	Assignment
`s=a[0]`	
$(\|Inv1(s, k) \wedge Inv2(t, k) \wedge 0 \leq n - k\|)$	Assignment
`while (k!= n) {`	
$(\|Inv1(s, k) \wedge Inv2(t, k) \wedge k \neq n \wedge 0 \leq n - k = E_0\|)$	Invariant Hyp. $\wedge$ guard
$(\|Inv1(min(s, min(t + a[k], a[k]), k + 1) \wedge Inv2(min(t + a[k], a[k]), k + 1) \wedge 0 \leq n - (k + 1) < E_0\|)$	Implied (`Lemma 1`)
`t=min(t+a[k],a[k]);`	
$(\|Inv1(min(s, t), k + 1) \wedge Inv2(t, k + 1)\|)$	Assignment
`s=min(s,t);`	
$(\|Inv1(s, k + 1) \wedge Inv2(t, k + 1)\|)$	Assignment
`k=k+1;`	
$(\|Inv1(a[0], k) \wedge Inv2(t, k)\|)$	Assignment
}	
$(\|Inv1(a[0], k) \wedge Inv2(t, k)\| \wedge \neg\neg(k = n)\|)$	Total-while
$(\|Inv1(s, n)\|)$	Implied

Vậy ta đã chứng minh S1 là đúng với total-correctness tiếp theo chúng ta chứng minh S2

1.2.4.b Chứng minh S2

Giống như phần partial-corectness để chứng minh S2 ta có Invariant :

$$Inv3(s, k) = \exists i, j(0 \leq i \leq j < k \wedge s = S_{i,j} \tag{7}$$

và Invariant đủ mạnh ta sử dụng để chứng minh là

$$Inv3(s, k) \wedge Inv4(t, k) \tag{8}$$

với $Inv4(t, k) = \exists i(0 \leq i < k \wedge t = S_{i,k-1}$

Giống với cách chứng minh S1 là total-correct biểu thức E sẽ là

$$E = n - k \tag{9}$$

bảng chứng minh cho S2

$(\|0 < n\|)$	
$(\|Inv3(a[0], 1) \wedge Inv4(a[0], 1) \wedge 0 \leq n - 1\|)$	Implied
`k=1;`	
$(\|Inv3(a[0], k) \wedge Inv4(a[0], k) \wedge 0 \leq n - k\|)$	Assignment
`t=a[0];`	
$(\|Inv3(a[0], k) \wedge Inv4(t, k) \wedge 0 \leq n - k\|)$	Assignment
`s=a[0]`	
$(\|Inv3(s, k) \wedge Inv4(t, k) \wedge 0 \leq n - k\|)$	Assignment
`while (k!= n) {`	
$(\|Inv3(s, k) \wedge Inv4(t, k) \wedge k \neq n \wedge 0 \leq n - k = E_0\|)$	Invariant Hyp. $\wedge$ guard
$(\|Inv3(min(s, min(t + a[k], a[k]), k + 1) \wedge Inv4(min(t + a[k], a[k]), k + 1) \wedge 0 \leq n - (k + 1) < E_0\|)$	Implied (`Lemma 2`)
`t=min(t+a[k],a[k]);`	
$(\|Inv3(min(s, t), k + 1) \wedge Inv4(t, k + 1)\|)$	Assignment
`s=min(s,t);`	
$(\|Inv3(s, k + 1) \wedge Inv4(t, k + 1)\|)$	Assignment
`k=k+1;`	
$(\|Inv3(a[0], k) \wedge Inv4(t, k)\|)$	Assignment
}	
$(\|Inv3(a[0], k) \wedge Inv4(t, k)\| \wedge \neg\neg(k = n)\|)$	Total-while
$(\|Inv3(s, n)\|)$	Implied

Vậy ta đã chứng minh S2 là total-correct

1.3 Các bài tập liên quan

1.3.1 Bài 22

Câu hỏi Nếu ta đảo vị trí 2 lệnh thứ nhất và thứ hai với nhau thì chương trình còn đúng nữa không ? Chứng minh câu trả lời của bạn

Trả lời Chương trình không còn đúng nữa, vì s được gán trước sau đó thì t mới được gán mà s=min(s,t), s phụ thuộc vào giá trị của t và chúng ta bỏ qua giá trị t ở lượt gán sau đó ở lần lặp cuối cùng , nghĩa là chúng ta bỏ qua việc so sánh S với tổng của các section có kết thúc là a[n-1], chương trình sẽ bị sai.

Ta có thể thử với testcase là mảng {1,-2}, dễ thấy minimal-sum ở đây là -2,nhưng chương trình nếu đảo 2 câu lệnh thứ nhất và thứ 2 trong vòng while sẽ ra kết quả là 1 (kết quả sai)

k	s	t
1		
1	1	
1	1	1
1	1	1
1	1	-2
2	1	-2

Bảng 1: lược đồ chương trình chạy testcase {1 ,2 } (thời gian tăng dần từ trên xuống)

1.3.2 Bài 24

Câu hỏi Chương trình `min_sum` không nói cho ta biết ở đâu có thể tìm được minimal-sum section của mảng. Áp dụng `min_sum` để lấy được chúng.Có thể làm việc này với chỉ 1 lần đi qua mảng không ?

Trả lời Có thể đạt được kết quả này ,sử dụng hai biến i , j để lưu vị trí bắt đầu và kết thúc của một minimal-sum section tìm được ,chương trình sẽ như sau

```
k=1;
t=a[0];
s=a[0];
i=0;
j=0;
while (k!=n){
    t=min(t+a[k],a[k]);
    if (s>t){
        if (t==a[k]){
            i=k;
            j=k;
        } else {
            j=k;
        }
        s=t;
    }
    k = k+1;
}
```