Wstęp do programowania

Całkowita poprawność programów

Całkowita poprawność

- Powiemy, że program P jest *całkowicie poprawny* względem warunków α,β wtedy i tylko wtedy, gdy
  - {α} P {β}
  - Program P dla danych spełniających α zawsze się zatrzyma
  - Program P dla danych spełniających α nie wygeneruje błędu

Wyszukiwanie binarne

- To jest chyba najważniejszy algorytm w całej informatyce.
- Chodzi o znalezienie konkretnego elementu w posortowanej tablicy
- Ogólna idea: w każdym kroku porównujemy ze środkowym elementem badanego segmentu i odrzucamy jedną z jego połówek (lub prawie połówek).

Binsearch

```
l:=1; p:=n;
{A[1]≤A[2]≤...≤A[n]}
while l<p do
  begin
    s:=(l+p) div 2;
    if x>A[s] then l:=s+1
                  else p:=s
  end;
jest:=A[l]=x
{jest↔∃1≤k≤n: A[k]=x}
```

Zbiory dobrze ufundowane

- Łańcuchem relacji $r \subseteq A \times A$ nazwiemy każdy skończony lub nieskończony ciąg $a=a_1,a_2,a_3,\ldots$ taki, że dla każdego naturalnego i jeśli a_i oraz a_{i+1} należą do ciągu a, to $(a_i,a_{i+1}) \in r$.
- Powiemy, że relacja r jest dobrze ufundowana, jeśli nie ma nieskończonych łancuchów.

Przykłady

- $(N,>)$
- $(N^n,>)$, gdzie $(x_1,...,x_n)>(y_1,...,y_n)$ wtedy i tylko wtedy, gdy dla każdego $i=1,...,n$: $x_i \geq y_i$ oraz $(x_1,...,x_n)\neq(y_1,...,y_n)$.
- (A,r), gdzie A=zbiór pozycji w grze w reversi (albo dowolnej innej skończonej grze), $r=\{(a,b)\in A\times A$: z pozycji a można w jednym ruchu dojść do b$\}$
- (D,r), gdzie D - zbiór poddrzew danego (być może nieskończonego) drzewa d, a relacja r jest relacją „bycia ojcem”.

Metoda Floyda

- Aby udowodnić to, że pętla zakończy działanie, stosujemy odwzorowanie f wartości zmiennych w jakiś zbiór dobrze ufundowany.
- Jeżeli pokażemy, że znaleziona funkcja f odwzorowuje w kolejnych obrotach pętli wartości zmiennych w elementy pozostające ze sobą w relacji dobrze ufundowanej, to będzie to oznaczać, że pętla zakończy swoje działanie.
- Typowym zbiorem dobrze ufundowanym wykorzystywanym przy dowodach jest zbiór (N, >)

Uzyteczny lemat

- $l<p \leftrightarrow l \leq (l+p) \text{ div } 2 < p$
- Dowód. Operacja div jest słabo monotoniczna, gdyż jeśli $x \leq y$, to $x \text{ div } 2 \leq y \text{ div } 2$.
  - "$\rightarrow$". Jeśli $l<p$, to istnieje $k>0$ takie, że $l+k=p$, lub równoważnie $p-k=l$. Zatem:
    - $(l+p) \text{ div } 2 = (2l+k) \text{ div } 2 \geq (2l) \text{ div } 2 = l$
    - $(l+p) \text{ div } 2 = (2p-k) \text{ div } 2 < p$, gdyż $2p$ jest najmniejszą liczbą, która po podzieleniu przez 2 daje p.
  - "$\leftarrow$" Jeśli $(l+p) \text{ div } 2 \geq p = (p+p) \text{ div } 2$, to $l \geq p$.

... i jego dualna wersja

- $l<p \leftrightarrow l < (l+p+1) \text{ div } 2 \leq p$
- Dowód: analogiczny

Dowód własności stopu dla binsearch

- f(l,s,p)=p-l
- Wiemy, że s=(l+p)div 2
- W każdym obrocie pętli albo wykonujemy instrukcję l:=s+1 albo p:=s.
- Ze względu na to, że $l \leq s < p$, różnica p-l musi się zmniejszyć, a jednocześnie po przypisaniu mamy $l \leq p$, więc wartości p-l są naturalne.

Dobry Binsearch

```
l:=1; p:=n;
{A[1]≤A[2]≤...≤A[n], l=1,p=n}
while l<p do {(∃1≤k≤n: A[k]=x)↔(∃l≤k≤p: A[k]=x), 1≤l≤p≤n}
  begin
    s:=(l+p) div 2;
    if x>A[s] then   l :=  s+1
                  else    p := s
  end;
jest := A[l]=x
{jest↔∃1≤k≤n: A[k]=x }
```

Zły Binsearch

```
l:=1; p:=n;
{A[1]≤A[2]≤...≤A[n]}
while l<p do {(∃1≤k≤n: A[k]=x)↔(∃l≤k≤p: A[k]=x), 1≤l≤p≤n}
  begin
    s:=(l+p) div 2;
    if x≥A[s] then   l :=  s
                  else    p := s-1
  end;
jest := A[l]=x
{jest↔∃1≤k≤n: A[k]=x  NIEPRAWDA!}
```

Inne typowe funkcje dobrze ufundowane

- Dla pętli for i:=1 to n f(i,n)=n-i
- Dla algorytmów Euklidesa f(m,n)=m+n
- Dla algorytmów przechodzenia grafu: f(G)=liczba nieodwiedzonych węzłów

Dowodzenie poprawności procedur i funkcji rekurencyjnych

- Dowodząc poprawność procedur i funkcji rekurencyjnych stosujemy bezpośrednio metodę indukcji – najlepiej ogólnie indukcji noetherowskiej, czyli w zbiorze dobrze ufundowanym.

Indukcja noetherowska

- Załóżmy, że zbiór A jest dobrze ufundowany za pomocą relacji r.
- Niech $Next(x)=\{y\in A: (x,y)\in r\}$
- Jeśli dla formuły jednej zmiennej $\varphi(x)$ określonej w zbiorze dobrze ufundowanym (A,r) zachodzi dla każdego elementu $x\in A$ następujący warunek

$$(\forall y\in Next(x): \varphi(y))\rightarrow \varphi(x) \qquad (*)$$

to $\forall x\in A: \varphi(x)$

Dowód twierdzenia o indukcji noetherowskiej

- Załóżmy, że w zbiorze dobrze ufundowanym (A,r) zachodzi warunek (*), a jednak istnieje $x \in A$ takie, że $\sim\varphi(x)$. Zatem musi istnieć $x_1 \in Next(x)$ takie, że $\sim\varphi(x_1)$. Ale wtedy istnieje też $x_2 \in Next(x_1)$ takie, że $\sim\varphi(x_2)$, ...itd. Konstruujemy nieskończony ciąg $x=x_0, x_1, x_2, \ldots$ taki, że dla każdego $i>0$ $x_i \in Next(x_{i-1})$ oraz $\sim\varphi(x_i)$. Pal licho to, że $\sim\varphi(x_i)$. Istotne jest to, że skonstruowaliśmy nieskończony łańcuch relacji r. Sprzeczność.