6. gaia

1. Ariketa Leire Garcia

ADA* lenpoaian dagokion funtzio emekurtziboa

```
function edk_re (s:[Int]; n,x:Int) return r:[Int] is
{ρ} ≡ {ez_negatiboa(s) ∧ n≥∅ ∧ x≥∅}
if n=∅ then r:=s;                    (#1)
elsif hutsa_da(s) then r:= [-n];     (#2)
elsif azkena(s) = x then r:= edk_re(azkena_kendu(s), n, x) ++ [x];   (#3)
else r:= edk_re(azkena_kendu(s), n-1, x)   (#4)
end if;
{ψ} ≡ {r= edk(s,n,x)}
```

1. urratsa: Errekurrentzia-erlazioa

→ lehenengo kasu emekurtziboa: n /= ∅ && not hutsa_da(s) && azkena(s) == x

edk_re(s,n,x) = edk_re(azkena_kendu(s), n, x) ++ [x]

→ bigarren kasu emekurtziboa: n/=∅ && not hutsa_da(s) && azkena /= x

edk_re(s,n,x) = edk_re(azkena_kendu(s), n-1, x)

2. urratsa: Inbarianteen kalkulatu

edk_re([5,5,8,7,5], 3, 5) = (#3)

(hedatu) = edk_re(azkena_kendu([5,5,8,7,5]), 3, 5) ++ [5] =

(tolestatu) = edk_re([5,5,8,7], 3, 5) ++ [5] = (#4)

(hedatu) = edk_re(azkena_kendu([5,5,8,7]), 3-1, 5) ++ [5] =

(tolestu) = edk_re([5,5,8], 2, 5) ++ [5] = (#4)

(hedatu) = edk_re(azkena_kendu([5,5,8]), 2-1, 5) ++ [5] =

(tolestu) = edk_re([5,5], 1, 5) ++ [5] = (#3)

(hedatu) = edk_re(azkena_kendu([5,5]), 1, 5) ++ [5,5] =

(tolestu) = edk_re([5], 1, 5) ++ [5,5] = (#3)

(hedatu) = edk_re(azkena_kendu([5]), 1, 5) ++ [5,5,5] =

(tolestu) = edk_re([], 1, 5) ++ [5,5,5] = (#2)

6. gaia

1. Ariketa Leire García

⊗ 2. unitatea jarraipena:

(Dinamizko kasua) = [-1] ++ [5,5,5] = [-1,5,5,5]

Orain emelkurrentzia-erlazioa orokortu behar da.

$$= edk\text{-}re([5], 1, 5) ++ [5,5] =$$

$$\underbrace{[5]}_{u},\ \underbrace{1}_{v},\ \underbrace{5}_{w}\qquad \underbrace{[5,5]}_{z}$$

$$INB \equiv edk\text{-}re(s,n,x) = edk\text{-}re(u,v,w) ++ z$$

Garrantzitsua da unbarianteen emelkurrentzia-erlazio bien orokorpena dela ikustea

→ lehenengo kasu emelkurtsiboa: azkena(s) == x

$$edk\text{-}re(s,n,x) = edk\text{-}re(\underbrace{azkena\text{-}kendu(s)}_{u}, \underbrace{n}_{v}, \underbrace{x}_{w}) ++ \underbrace{[x]}_{z}$$

→ bigarren kasu emelkurtsiboa: azkena /= x

$$edk\text{-}re(s,n,x) = edk\text{-}re(\underbrace{azkena\text{-}kendu(s)}_{u}, \underbrace{n-1}_{v}, \underbrace{x}_{w}) ++ \underbrace{[\ ]}_{z}$$

Aipatu beharrekoa da ② aldagaiari dagokion lehuan ezer ez zeposeuz, zerrenda hutsa dagoela konsideratu dela.

Beraz emelkurrentzia-erlazio biek egitura bera dutela kontsidera daiteke eta, ondorioz, unbariante bakar baten bidez orokor daiteke.

6. gaia

1. Ariketa Leire García

⊛ 3. urratsa: Aldagai berrien hasieraketa

Hasieratu ondoren u, v, w, z honako berdintza hau bete behar dute:

edk_re(s, n, x) = edk_re(u, v, w) ++ [z]

Berdintza hori bete dadin u, v, w, z aldagaiak berdintzaren erkemeko aldean posizio bera okupatzen dutu baloiemi hasieratu beharko dira.

edk_re(s, n, x) ++ [ ] = edk_re(u, v, w) ++ z

Beraz, hasieraketak:

u := s;
v := n;
w := x;
z := [];

⊛ 4. urratsa: Bukaera (while-aren baldintza eta bukaerako esleipena)

¬B ≡ "oinarrizko kasuen disjuntzioa aldagai berriek erabiliz".

- Oinarrizko kasuen disjuntzioa hasierako aldagaiekin:

  n = ∅ ∨ hutsa_da(s)

- Oinarrizko kasuen disjuntzioa aldagai berriekin:

  v = ∅ ∨ hutsa_da(u)

Beraz, ¬B ≡ v = ∅ ∨ hutsa_da(u) izango da eta beraz,

B = v ≠ ∅ ∧ ¬hutsa_da(u) izango da.

(*) 4. unitatearen jarraipena:

While-a bukatu ondoren zein esleipen egin behar den erabakitzeko, hau da, hutsoak emaitza berdina zein balio itzuliko duen erabakitzeko, invariantea hartu eta oinarrizko kasu bakoitzean gandela zuposatu beharko da:

- Lehenengo oinarrizko kasua: ($v = \emptyset$)

$$edk_re(s, n, x) = edk_re(u, v, w) ++ z \overset{(\#1)}{=}$$
$$= u ++ z$$

Beraz, while-etik $v = \emptyset$ delako ateratzen bagara, bukaerako esleipena honako hau izango da:

$$r := u ++ z;$$

- Bigarren oinarrizko kasua: (hutsa_da(u))

$$edk_re(s, n, x) = edk_re(u, v, w) ++ z \overset{(\#2)}{=}$$
$$= [-v] ++ z$$

Ondoriozko, while-etik $v \neq \emptyset$ izanda, hutsa_da(u) betetzen delako ateratzen bagara, bukaerako esleipena honako hau izango da:

$$r := [-v] ++ z;$$

Bi oinarrizko kasu daudenez eta kasu bakoitzean bukaerako esleipena desberdina denez, while-a bukatu ondoren if bat ipini beharko da.

Leire García

⊛ 5. urratsa: while-aren baruko agirudiak:

while-aren baruko agirudiak kalkulatzeko 8 baldintza betetzen dela suposatu eta edk_re funtzioaren kasu emelurtziboak hartu beharr dira kontuan.

2 kasu emelurtsibo daudenez, bakoitza bere aldetik aztertu beharko da. Hasteko, inbarianteak hartu eta lehenengo kasu emelurtziboan gaudela kontsideratu beharko da eta hedatze-urrats bat eta tolestatze-urrats bat emango dira. Homela lehenengo kasu emelurtziboan gaudenean u, v, w eta z aldagaiak nola egoeratu beharr ditugun jakingo dugu. Gero, bigaren kasu emelurtziboan gaudela suposatuz gauza bera egingo da.

⊛ <u>Lehenengo kasu emelurtziboa</u>: azkena(u) == w

edk_re(s, n, x) = edk_re(u, v, w) ++ z =

(hedatu) = (edk_re(azkena_kendu(u), v, w) ++ [w]) ++ z =

(tolestatu) = edk_re(azkena_kendu(u), v, w) ++ ([w] ++ z)

Inbarianteak aldagai bakoitza zeporu lehuan tolestatze-urratsen ondoren zer geldituko den kontuan hartuz, honako hauek izango dira egoeraketak:

u := azkena_kendu(u);
v := v;
w := w;
z := [w] ++ z;

Leire Garcia

⊛ Bigarren kasu ematerriboa: azkena(u) |= w

edk_re(s,n,x) = edk_re(u, v, w) ++ z =

(hedatu) = (edk_re(azkena_kendu(u), v-1, w)) ++ z =

(tokstatu) = edk_re(azkena_kendu(u), v-1, w) ++ z

Honako hauek izango dira egunerakotak:

u := azkena_kendu(u);

v := v-1;

w := w;

z := z;

Bi kasu ematerriboa daudenez eta kasu bakoitzean aldagaiak egunerratzeko era desberdina denez, while-aren barnean if bat ipini beharko da.

Funtzio iteratiboa ADA* lengoaiaz idatzita:

```
function edk_it (s:[Int]; n,x:Int) return r:[Int] is
{Ψ} ≡ {ez_negatiboa(s) ∧ n≥∅ ∧ x≥∅}
  u,z:[Int];
  v,w:Int;
  u:=s;
  v:=n;
  w:=x;
  z:=[];
  while {INB} v≠∅ ∧ ¬hutsa_da(u) loop
    if azkena(u)==w then
      u:=azkena_kendu(u);
      v:=v;
      w:=w;
      z:=[w]++z;
    else
      u:=azkena_kendu(u);
      v:=v-1;
      w:=w;
      z:=z;
    end if;
  end loop;
```

Leire Garcia

<u>if</u> $v = \emptyset$ <u>then</u> $r := u ++ z$;

<u>else</u> $r := [-v] ++ z$;

<u>end if</u>;

$\{\Psi\} \equiv \{ r = edk(s, n, x) \}$

Leire Garcia