7th practice

1. Find the longest word in a given text file that contains character 'w'. Words are separated by spaces from each other.

```
Specification:
                                                        Idea:
   S = (x:infile(\mathbb{K}), I:\mathbb{L}, longest:\mathbb{S})
                                                           Enumerate the words along
   Pre = (x = x_0)
                                                           with their length and a boolean
                                                           value indicating whether the
                                                           word contains 'w'.
New specification:
                                                        Conditional maximum search
                                                                         ~ |e.word|
   S = (t:enor(WORD), I:\mathbb{L}, longest:\mathbb{S})
                                                                         ~ e.w
                                                           cond(e)
                 WORD = rec(word: S, w: L)
                                                           H, >
                                                                         ~ N,>
   Pre = (t = t_0)
   Post = ( (I, max, elem) = MAX_{e \in t_0} |e.word | ∧ I → longest =elem.word )
                              e.w
```

Algorithm:

l := false; t.first()					
一t.end()					
t.current().w	t.current().w \wedge I		t.current().w ∧ ¬l		
	\ t.current().word >max		I, max, longest := true,		
_	max, longest := t.current().word , t.current().word		t.current().word , t.current().word		
t.next()					

Enumerator:

t:enor(WORD)	$::enor(WORD) \qquad WORD = rec(word: \mathbb{S}, w: \mathbb{L})$				
WORD *	first()	next()	current() : WORD	end() : $\mathbb L$	
x:infile(K) dx:K sx:Status curr:WORD end:L	sx,dx,x:read next()	see below	return curr	return end	

next() method

$$\begin{split} S &= (x:infile(\mathbb{K}), dx: \mathbb{K}, sx: Status, curr: WORD, end: \mathbb{L}) \\ \textit{Pre} &= (\ x = x' \land dx = dx' \land sx = sx' \) \\ \textit{Post} &= (\ (dx'', (sx'', dx'', x'')) = SELECT_{dx \in (dx', x')} \ (sx=abnorm \lor dx \neq' \ ') \\ &\wedge \ end = (sx''=abnorm) \\ &\wedge \ (\neg end \rightarrow (curr. word, (sx, dx, x)) = \bigoplus_{dx \in (dx'', x'')} < dx > \land (curr. w, (sx, dx, x)) = \bigvee_{dx \in (dx'', x'')} dx = 'w' \) \) \end{split}$$

Remark: the subproblem (decide whether a word contains 'w'), instead of linear search will be determined by OR'ing the boolean values. The reason for this is that the whole current word has to be processed and OR'ing can be merged into the same loop with the concatenation of the word.

Selection

t:enor(E)
$$\sim$$
 x:infile(\mathbb{K}) (sx,dx,x:read)
without first()
cond(e) \sim sx=abnorm \vee dx \neq '

Two summations (concatenation and OR'ing) t:enor(E) \sim x:infile(\mathbb{K}) (sx,dx,x:read) without first(), cond: dx \neq ' 'f(e) \sim (<dx>, dx='w')

s ~ (curr.word, curr.w) H, +, 0 ~ (\mathbb{K}^* , \mathbb{L}), (\bigoplus , \vee), (<>, false)

sx=norm \(dx=' '			
sx,dx,x:read			
end := sx=abnorm			
–end			
curr.word, curr.w := <>,			
sx=norm ∧ dx≠′′	_		
curr.word, curr.w :=			
curr.word \bigoplus <dx>,</dx>			
curr.w \vee (dx='w')			
sx,dx,x:read			

2. Given a file containing data of huntings. Each line of the file consists of the name of the hunter, the date of the hunting, the species and weight of the animal shot by the given hunter at the given hunting. The file is sorted by hunter and then by date.

Decide, whether every hunter has shot a bear at any of his/her hunting.

Specification:

$$S = (x:infile(Trophy), l: \mathbb{L})$$

 $Trophy = rec(name: \mathbb{S}, date: \mathbb{S}, species: \mathbb{S}, weight: \mathbb{N})$
 $Pre = (x = x_0 \land x \nearrow (name, date))$

New specification:

$$S = (t:enor(\mathbb{L}), I:\mathbb{L})$$

 $Pre = (t=t_0)$
 $Post = (I = \forall SEARCH_{e \in t_0} e)$

Optimistic linear search

cond(e) ~ e

Idea:

Enumerate as many boolean values as the number of the hunters, a boolean value is true in case the related hunter has shot a bear.

Algorithm:

	l := true		
t.first()			
	l ∧ ¬t.end()		
	l := t.current()		
	t.next()		

Enumerator:

$t:enor(\mathbb{L})$

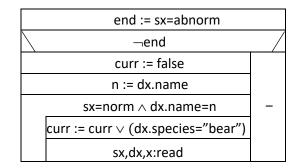
L *	first()	next()	current() : $\mathbb L$	end() : $\mathbb L$
x:infile(Trophy) dx:Trophy sx:Status curr:L end:L	sx,dx,x:read next()	see below	return curr	return end

Trophy = rec(name: \mathbb{S} , date: \mathbb{S} , species: \mathbb{S} , weight: \mathbb{N})

next() method

```
S = (x:infile(Trophy), dx: Trophy, sx:Status, curr: \mathbb{L}, end: \mathbb{L})
Pre = (x = x' \land x \nearrow_{(name,date)} \land dx = dx' \land sx = sx') \qquad dx.name = dx'.name
Post = (end = (sx'=abnorm) \land (\neg end \rightarrow (curr, (sx,dx,x)) = V_{dx \in (dx',x')} curr.species = "bear"))
```

Summation (OR'ing)



3. Given a file containing data of huntings. Each line of the file consists of the name of the hunter, the date of the hunting, the species and weight of the animal shot by the given hunter at the given hunting. The file is sorted by hunter and then by date.

Count the hunters who shot a rabbit at each of his/her huntings and the total weight of animals shot by the hunter is greater than 250 kg.

Specification:

$$S = (x:infile(Trophy), c: \mathbb{N})$$

 $Trophy = rec(name: \mathbb{S}, date: \mathbb{S}, species: \mathbb{S}, weight: \mathbb{N})$
 $Pre = (x = x_0 \land x \nearrow (name, date))$

New specification:

$$S = (\text{t:enor(Hunter)}, \text{c: } \mathbb{N})$$

 $\text{Hunter} = \text{rec(name: } \mathbb{S}, \text{r:} \mathbb{L}, \text{totalw:} \mathbb{N})$
 $Pre = (\text{t=t}_0)$
 $Post = (\text{c} = \sum_{e \in t_0} 1)$
 $e.r \land e.\text{totalw>250}$

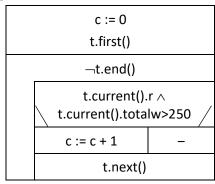
Counting

cond(e) ~ e.r ∧ e.totalw>250

Idea:

Enumerate the hunters (name, boolean value indicating whether the hunter shot a rabbit at each of his/her huntings, total weight of animals shot by the hunter).

Algorithm:



Enumerator:

Idea: In order to create the enumerator t:enor(Hunter) mentioned above, let us suppose that we have an enumerator y:enor(Aggregation) enumerating the total achievement of hunters per huntings. y:enor(Aggregation) is sorted by the name of the hunter and then by the date of the hunting, and also includes whether the hunter has shot any rabbit at the given hunting and the total weight of the hunted animals at the given hunting.

t:enor(Hunter) Hunter = rec(name: S, r:L, totalw:N)

Hunter*	first()	next()	current() : Hunter	end() : $\mathbb L$
y : enor(Aggregation) curr : Hunter end : L	y.first() next()	see below	return curr	return end

Aggregation = rec(name: \mathbb{S} , date: \mathbb{S} , shotr: \mathbb{L} , totalw: \mathbb{N})

next() method of t:enor(Hunter)

$$S = (y:enor(Aggregation), curr:Hunter, end: \mathbb{L})$$

$$Pre = (y = y' \land y \nearrow (name, date))$$

$$Post = (end = y'.end() \land (\neg end \rightarrow curr.name = y'.current().name$$

$$e.name = curr.name$$

$$\land (curr.r, y) = \bigwedge_{e \in (y'.current(),y')} e.shotr$$

e.name = curr.name
$$\land (curr.totalw, y) = \sum_{e \in (y'.current(),y')} e.totalw)))$$

Two summations (AND'ing and addition)

on the same enumerator

t:enor(E) ~ y:enor(Aggregation)

without first()

cond: (y.current().name =

curr.name)

f(e) ~ y.current().shotr

s \sim curr.r H, +, 0 \sim L, \wedge , true

f(e) ~ y.current().totalw

s ~ curr.totalw

 $H, +, 0 \sim \mathbb{N}, +, 0$

end := y.end()	
−end	
curr := (y.current().name, true, 0)	
$\neg y.end() \land \ y.current().name=curr.name$	_
curr.r := curr.r \(\times \) y.current().shotr	
curr.totalw :=	
curr.totalw + y.current().totalw	
y.next()	

Enumerator:

y:enor(Aggregation) Aggregation = rec(name: S, date:S, shotr:L, totalw:N)

Aggregation*	first()	next()	current(): Aggregation	end() : $\mathbb L$
x:infile(Trophy) dx:Trophy sx:Status curr:Aggregation end:L	sx,dx,x:read next()	see below	return curr	return end

Trophy = $rec(name: \mathbb{S}, date: \mathbb{S}, species: \mathbb{S}, weight: \mathbb{N})$

next() method of y:enor(Aggregation)

$$S = (x:infile(Trophy), dx:Trophy, sx:Status, curr: Aggregation, end: L)$$

$$Pre = (x = x' \land x \nearrow (name, date) \land dx = dx' \land sx = sx')$$

Post = (end = (sx'=abnorm)
$$\land$$
 (\neg end \rightarrow curr.name=dx'.name \land curr.date=dx'.date \land

 $dx.name = curr.name \land dx.date = curr.date$

$$\land$$
 (curr.shotr, (sx,dx,x)) = $V_{dx \in (dx',x')}$ dx.species="rabbit" \land

 $dx.name=curr.name \land dx.date=curr.date$

$$\wedge$$
 (curr.totalw, (sx,dx,x)) = $\sum_{dx \in (dx',x')} dx$.weight)

Two summations (OR'ing and addition)

with a common enumerator processing

the same file

t:enor(E) ~ x:infile(Trophy)

sx,dx,x:read, without first(),
cond: dx.name=curr.name

∧ dx.date=curr.date

f(e) ~ dx.species="rabbit"

s ~ curr.shotr H, +, 0 ~ \mathbb{L} , \vee , false

f(e) ~ dx.weight s ~ curr.totalw

 $H, +, 0 \sim \mathbb{N}, +, 0$

end := sx=abnorm				
⊸end				
curr := (dx.name, dx.date, false, 0)				
sx=norm ∧ dx.name=curr.name	-			
\wedge dx.date=curr.date				
curr.shotr :=				
curr.shotr \lor (dx.species="rabbit")				
<pre>curr.totalw := curr.totalw + dx.weight</pre>				
sx,dx,x:read				