

EÖTVÖS LORÁND
UNIVERSITY | BUDAPEST

Custom Enumerators

Solving Concrete Problems

Object Oriented Programming | 2024 Spring
Practice 7

Presented by Tarlan Ahadli
Supervised by Prof. Teréz Anna Várkonyi

Problem 1

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

Specification:

$A = (x:\text{infile}(\mathbb{K}), l:\mathbb{L}, \text{longest}:\mathbb{S})$

$Pre = (x = x_0)$

Problem 1 | Enumerator Based Specification

Idea:

Enumerate the words along with their length and a boolean value indicating whether the word contains 'w'.

Conditional maximum search

$f(e)$	\sim	$ e.word $
$cond(e)$	\sim	$e.w$
$H, >$	\sim	$\mathbb{N}, >$

New specification:

$A = (t: \text{enor}(\text{WORD}), l: \mathbb{L}, \text{longest}: \mathbb{S})$
 $\text{WORD} = \text{rec}(\text{word}: \mathbb{S}, w: \mathbb{L})$

$Pre = (t = t_0)$

$Post = ((l, \text{max}, \text{elem}) = \mathbf{MAX}_{e \in t_0} |e.word| \wedge l \rightarrow \text{longest} = \text{elem.word})$
 $e.w$

Problem 1 | Structogram

Algorithm:

l := false; t.first()			
¬t.end()			
¬ t.current().w	t.current().w ∧ l		t.current().w ∧ ¬l
—	t.current().word > max		l, max, longest := true, t.current().word , t.current().word
	max, longest := t.current().word , t.current().word	—	
t.next()			

Problem 1 | Enumerator

Enumerator:

t:enor(WORD)

WORD = rec(word:S, w:L)

WORD *	first()	next()	current() : WORD	end() : L
x : infile(K) dx : K sx : Status curr : WORD end : L	sx,dx,x:read next()	see below	return curr	return end

Problem 1 | Enumerator | Next

next() method

$A = (x:\text{infile}(\mathbb{K}), dx:\mathbb{K}, sx:\text{Status}, \text{curr}:\text{WORD}, \text{end}:\mathbb{L})$

$Pre = (x = x' \wedge dx = dx' \wedge sx = sx')$

$Post = ((dx'', (sx'', dx'', x'')) = \text{SELECT}_{dx \in (dx', x')} (sx = \text{abnorm} \vee dx \neq ' '))$

$\wedge \text{end} = (sx'' = \text{abnorm})$

$dx \neq ' '$

$dx \neq ' '$

$\wedge (\neg \text{end} \rightarrow (\text{curr.word}, (sx, dx, x)) = \bigoplus_{dx \in (dx'', x'')} \langle dx \rangle \wedge (\text{curr.w}, (sx, dx, x)) = \bigvee_{dx \in (dx'', x'')} dx = 'w'))$

Problem 1 | Enumerator | Next | Algorithm

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(),
 as long as: $dx \neq ''$
 $f(e) \sim (<dx>, dx='w')$
 $s \sim (curr.word, curr.w)$
 $H, +, 0 \sim (\mathbb{K}^*, \mathbb{L}), (\oplus, \vee), (<>, false)$

$sx=norm \wedge dx=''$	
$sx, dx, x:read$	
$end := sx=abnorm$	
$\neg end$	
$curr.word, curr.w := <>, false$	$-$
$sx=norm \wedge dx \neq ''$	
$curr.word, curr.w :=$ $curr.word \oplus <dx> ,$ $curr.w \vee (dx='w')$	
$sx, dx, x:read$	

Problem 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:

$$A = (x:\text{infile}(\text{Trophy}), i:\mathbb{L})$$
$$\text{Trophy} = \text{rec}(\text{name}:\mathbb{S}, \text{date}:\mathbb{S}, \\ \text{species}:\mathbb{S}, \text{weight}:\mathbb{N})$$
$$\text{Pre} = (x=x_0 \wedge x \nearrow (\text{name}, \text{date}))$$

Problem 2 | Enumerator Based Specification

New specification:

$A = (t: \text{enor}(\mathbb{L}), l: \mathbb{L})$

$Pre = (t = t_0)$

$Post = (l = \forall \text{SEARCH}_{e \in t_0} e)$

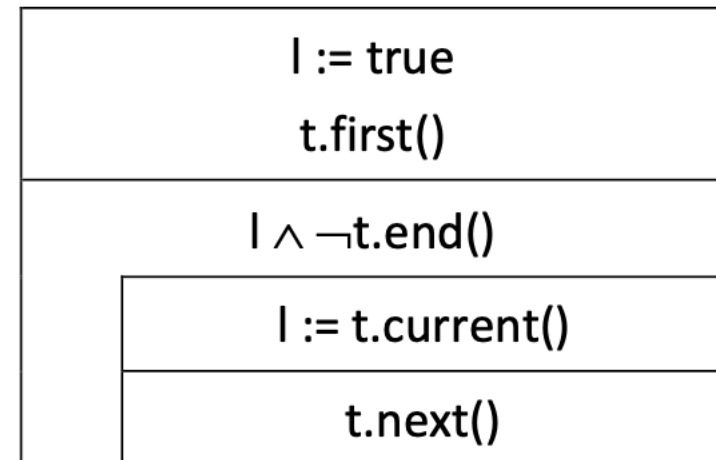
linear search

$\text{cond}(e) \sim e$

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:



Problem 2 | Enumerator

Enumerator:

t:enor(\mathbb{L})

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

Trophy = rec(name:\$, date:\$, species:\$, weight:N)

Problem 2 | Enumerator | Next

next() method

$A = (x:\text{infile}(\text{Trophy}), dx:\text{Trophy}, sx:\text{Status}, \text{curr}:\mathbb{L}, \text{end}:\mathbb{L})$

$Pre = (x = x' \wedge x \nearrow (\text{name}, \text{date}) \wedge dx = dx' \wedge sx = sx') \quad dx.\text{name} = dx'.\text{name}$

$Post = (\text{end} = (sx' = \text{abnorm}) \wedge (\neg \text{end} \rightarrow (\text{curr}, (sx, dx, x)) = \bigvee_{dx \in (dx', x')} \text{curr}.\text{species} = \text{"bear"}))$

Summation (OR'ing)

$t:\text{enor}(E) \sim x:\text{infile}(\text{Trophy}) (sx, dx, x:\text{read})$
without first(),

as long as: $dx.\text{name} = dx'.\text{name}$

$f(e) \sim dx.\text{species} = \text{"bear"}$

$s \sim \text{curr}$

$H, +, 0 \sim \mathbb{L}, \vee, \text{false}$

end := sx=abnorm	
¬end	
curr := false	-
n := dx.name	
sx=norm ∧ dx.name=n	
curr := curr ∨ (dx.species="bear")	
sx,dx,x:read	