Department of Computer Science
Ashoka University

Programming Language Design and Implementation (PLDI): CS-1319-1

Quiz 2 Solutions

Marks: 100

Date: *November 07, 2023*

Time: *16:40 - 18:10*

Instructions:

1. The quiz will be physical, on paper, and in the classroom.
2. Write your name and Ashoka ID on the answer-script and additional papers.
3. The quiz comprises one question (totalling 100 marks) and one bonus question (for 10 marks). Each question has multiple parts with marks shown for each.
4. The test is entirely closed book.
5. Any copy from peers will be dealt with zero tolerance - both to get zero in the question. Consultations or chats with others will lead to zero score for the entire quiz.
6. No question or doubt will be entertained. If you have any query, make your own assumptions, state them clearly in your answer and proceed.
7. Write in clear handwriting and in an unambiguous manner. If TAs have difficulty reading / understanding your answer, they will make assumptions at their best capacity to evaluate. You would not get an opportunity for explanation or rebuttal.

1. You have to develop a **Set Calculator** Cantor for set algebra with lower case letters as elements. For this you need to write Flex & Bison specifications. The calculator uses character literals, special symbols and a number of binary and unary operators as listed in the grammar below.

   Cantor reads a set expression involving character literals, the operators, and the special symbols; and it displays the resultant set on the console. The expression is terminated by a $.

Sr#	Rule	Operator Semantics	
		Name	**Remarks**
1:	C → S $		*Displays the set S when $ is in the input*
2:	$S \rightarrow S + S$	Union[a]	*Set Union*: Elements in either A or B
3:	$S \rightarrow S - S$	Difference[a]	*Set Difference*: $A - B = A * -B$
4:	$S \rightarrow S * S$	Intersection[a]	*Set Intersection*: Elements in both A and B
5:	$S \rightarrow S / S$	Disjunctive Union[a]	*Symmetric Difference*: $A/B = (A - B) + (B - A)$
6:	$S \rightarrow + S$	Identity[a]	*No change in the set*
7:	$S \rightarrow - S$	Complement[a]	*Set Complement*: $A + -A = U$
8:	$S \rightarrow (S)$	Parenthesis	*Supersede precedence & associativity*
9:	$S \rightarrow \{\ \}$	Empty Set	*Set with no element*: $\{\ \} = -U$
10:	$S \rightarrow U$	Universal Set[b]	*Set with all elements*: $U = \{$`a, b, ..., z`$\} = -\{\ \}$
11:	$S \rightarrow \{\ L\ \}$	Set Builder	*Builds a set from comma separated list of elements*
12:	$L \rightarrow L\ ,\ E$	List	*Comma separated list of elements*
13:	$L \rightarrow E$	Element	*An element of a set – a single lower case letter*
14:	$E \rightarrow$ `chr`	Character[c]	*Char literal – lower case letters*: `a, b, ..., z`

[a]: *Operators* `+, -, *, /, unary+, and unary-` *follow the precedence and associativity of their C counterparts. We have attached a table below for the same.*
[b]: *U is a special terminal denoting universal set.*
[c]: `chr` *is the token class of lower case letters.*

Precedence	Associativity
`unary+, unary-`	Right to Left
`*,/`	Left to Right
`+,-`	Left to Right

(a) Fully parenthesize and evaluate the following expressions according to the rules of Cantor.
[2 + 3 + 3 + 2 = 10]

i. `{c, f, h, k, r, w} - {a, f, r} * {c, d, h, k} - {h}`

   Ans. `((({c, f, h, k, r, w}) - (({a, f, r}) * ({c, d, h, k}))) - ({h})) = {c, f, k, r, w}`

ii. `{b, e, q, v, x} + +{a, d, e, v, x} - -{f, l, o, q, r, t, u, x, y}`

   Ans. `((({b, e, q, v, x}) + (+({a, d, e, v, x}))) - (-({f, l, o, q, r, t, u, x, y}))) = {q, x}`

iii. `(U - -U) * (-{ } - { })`

   Ans. `((({a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z}) - (-({a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z}))) * ((-({ })) - ({ })))`
   `= {a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y, z}`
   `= U`

iv. `{a, b, c, d} / {b, c, d, e} / {c, d, e, f}`

   Ans. `((({a, b, c, d}) / ({b, c, d, e})) / ({c, d, e, f})) = {a, c, d, f}`

(b) Write the Flex specifications for Cantor. Write them as per the format you would need for your bison specifications in the next question. For your convenience we have attached a template sheet at the end. **[10]**

Please see attached zip file.

(c) Write the Bison specifications for Cantor. For your convenience we have attached a template sheet at the end. **[20]**

Please see attached zip file.

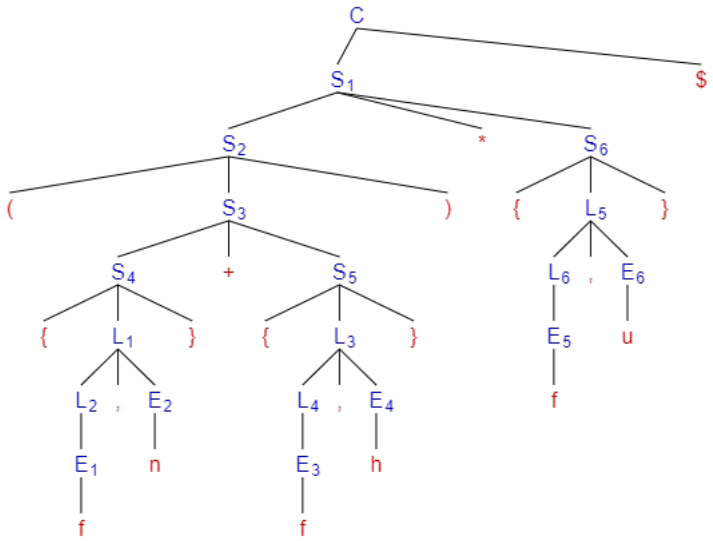
You may assume auxiliary functions for writing the bison actions. Specify the functions clearly with input, output, and behaviors; however you do not need to write the codes of these functions.

(d) Consider an input

`A` ≡ `({f, n} + {f, h}) * {f, u}$`

i. Draw the parse tree for `A` using the grammar of Cantor. **[10]**

Ans.



ii. Show the working of Cantor for `A` using the symbol and the value stacks. Show the state of the stacks for every reduction and mark the production rule used in reduction. You do not need to show the state of the stacks for shifts. **[30]**

E	f
{	
(	

$E \rightarrow$chr

L	f
{	
(	

$L \rightarrow E$

E	n
,	
L	f
{	
(	

$E \rightarrow$chr

L	f, n
{	
(	

$L \rightarrow L, E$

S	{f, n}
(	

$S \rightarrow \{L\}$

E	f
{	
+	
S	{f, n}
(	

$E \rightarrow$ chr

L	f
{	
+	
S	{f, n}
(	

$L \rightarrow E$

E	h
,	
L	f
{	
+	
S	{f, n}
(	

$E \rightarrow$ chr

L	f, h
{	
+	
S	{f, n}
(	

$L \rightarrow L, E$

S	{f, h}
+	
S	{f, n}
(	

$S \rightarrow \{L\}$

S	{f, h, n}
(	

$S \rightarrow S + S$

S	({f,h,n})

$S \rightarrow (S)$

E	f
{	
*	
S	({f,h,n})

$E \rightarrow$ chr

L	f
{	
*	
S	({f,h,n})

$L \rightarrow E$

E	n
,	
L	f
{	
*	
S	({f,h,n})

$E \rightarrow$ chr

L	f, n
{	
*	
S	({f,h,n})

$L \rightarrow L, E$

S	{f, n}
*	
S	({f,h,n})

$S \rightarrow \{L\}$

S	{f}

$S \rightarrow S * S$

C	{f}$

$C \rightarrow S\$$

Evaluation: {f}

(e) We want to add the power set computation capability to Cantor. Show the required changes in the grammar, Flex and Bison. Assume that your power set operator is `!`. It is unary, postfix, and right associative. **[10]**

Please see attached zip file.

2. Consider the following grammar, **[10]**

$$
\begin{aligned}
S &\rightarrow AD \mid BE \mid CF \\
A &\rightarrow aA \mid E \mid \epsilon \\
B &\rightarrow bB \mid D \mid \epsilon \\
C &\rightarrow cC \mid \epsilon \\
D &\rightarrow d \mid dD \\
E &\rightarrow e \mid Sa \\
F &\rightarrow fF \mid \epsilon
\end{aligned}
$$

Find,

- FIRST(S) = {a,b,c,d,e,f,ϵ} **[5]**

- FOLLOW(C) = {f, a, $} **[5]**

3. **Bonus Problem**: Consider the following grammar, **[10]**

$$
\begin{aligned}
S &\rightarrow Ae \mid Bf \mid Cg \\
A &\rightarrow D \mid E \mid \epsilon \\
B &\rightarrow b \mid \epsilon \\
C &\rightarrow c \\
D &\rightarrow d \mid \epsilon \\
E &\rightarrow e
\end{aligned}
$$

Is this grammar LL(1)? If yes, justify and if not, explain what conditions does it fail. (An answer with no/incorrect explanation will not receive points)

Ans. No, it is not for the following three reasons (each is sufficient):

(a) **Ambiguous:** $S \rightarrow Ae, A \rightarrow D, D \rightarrow \epsilon$ and $S \rightarrow Ae, A \rightarrow \epsilon$ both derive e.

(b) **FIRST(A)** $\cap$ **FIRST(B)** $\neq \phi$ (and same violation for B,C)

(c) We see that $D \xrightarrow{*} \epsilon$ and **FIRST(E)** $\cap$ **FOLLOW(A)** $\neq \phi$.

*The credit for a bonus problem (***10** *here) is not counted in the total of* **100**. *However, marks scored in a bonus problem will be added to total score (capped, of course, at* **100***).*