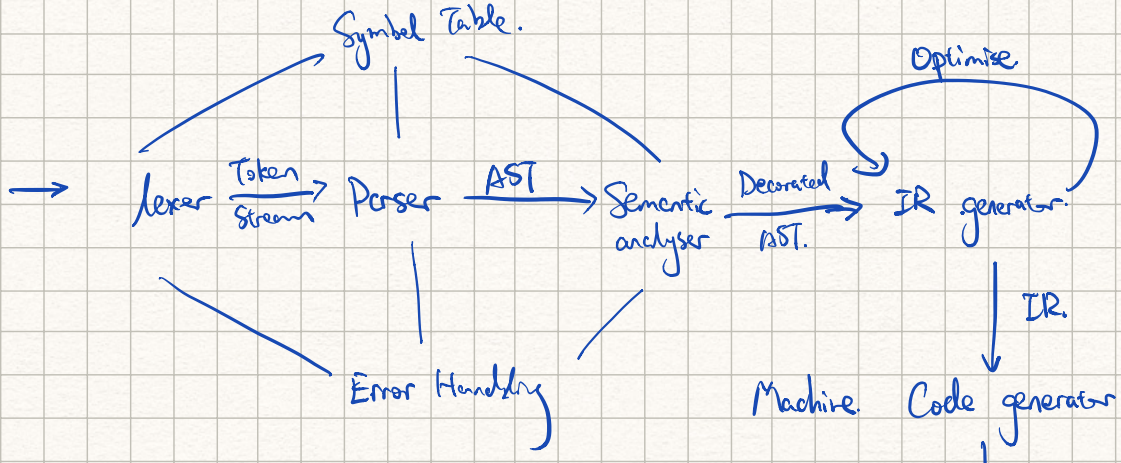
Compiler Exam 2018.

Scan A.

1. (a)



(b) False.

lexer only matches using regex → simply generates token stream.

no contextual info is involved.

(c) $(c|a|b)^* \Leftrightarrow \Sigma^*$ let $\Sigma = \{a, b, c\}$

$\therefore L(r_1) \geq L(r_0)$

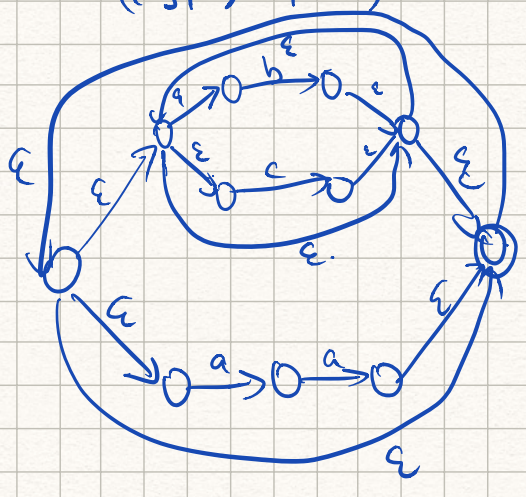
Let ca is not generatable by $L(r_0)$

$\therefore L(r_1) > L(r_0)$

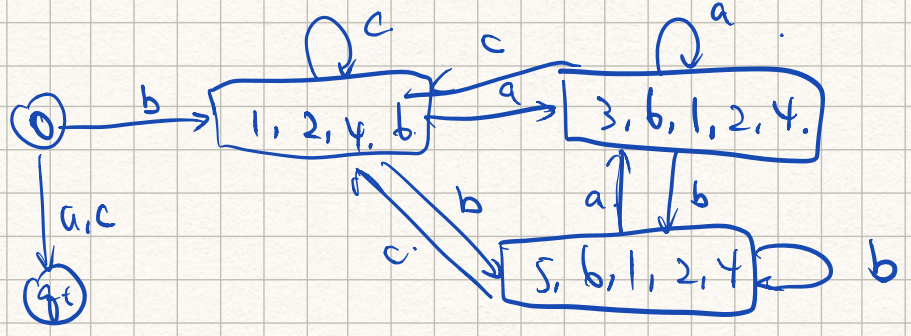
(d) $[^0\text{-}9]\Sigma^*$

(e) $[^0\text{-}9](|Unm\cdots n\,|\,|um\cdots n\,|\,[0\text{-}9]|[a\text{-}zA\text{-}Z]|\_)^*$

(f) $(c|c)^*|aa)^*$



(g).

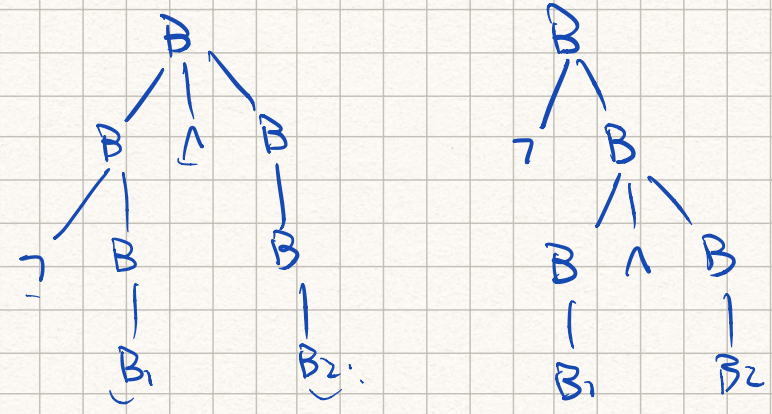


2. (a) [B]

B does not have a right precedence declared.

¬ has a higher precedence than ∨, ∧.

¬B₁∧B₂



(b) True F is not defined.

G = <N, T, R, S>.

Non-terminals no new covered.

terminals ∨.

Rules. every non-terminal is covered.

Starting Symbol is E.

(c). False left recursion → for LL parsing Top ↓ parsing

left factoring → for backtrack parsing

(d) $S \to Aa \mid b \mid B$
$A \to z \mid Cc$
$C \to q \mid Sb$
$B \to Bb \mid a \mid \varepsilon$.

$\to$ $S \to Aa \mid b \mid B$
$A \to z \mid Cc$
$C \to q \mid Sb$
$B \to aB' \mid B'$
$B' \to bB' \mid \varepsilon$.

$\to$ $S \to za \mid Cca \mid b \mid B$
$C \to q \mid Sb$
$B \to aB' \mid B'$
$B' \to bB' \mid \varepsilon$.

$\to$ $S \to za \mid b \mid B \mid qca \mid Sbca$.
$B \to aB' \mid B$
$B' \to bB'' \mid \varepsilon$

$\to$ $S \to zaS' \mid bS' \mid BS' \mid qcaS'$
$S' \to bcaS' \mid \varepsilon$.
$B \to aB' \mid B$.
$B' \to bB' \mid \varepsilon$.

(e). $S \to aAx \mid C$
$A \to aS \mid B \mid q$.
$B \to yC \mid b$.
$C \to zC \mid \varepsilon$

	First.	Follow.
S	a, z, ε.	Fw(A), $ = x, $
A	a, q, y, b.	x
B	y, b.	Fw(A) = x.
C	z, ε.	Fw(B) ∪ Fw(S) = x, $

	a	b	q	x	y	z	$
S	S→aAx			S→C.		S→C	S→ε
A	A→aS	A→B B→b	A→q.		A→B B→yC.		
B							
C				C→ε.		C→zC	C→ε.

(f) LALR.

Create a verbose mode, where only LR(1) parsing is used, report error using LR(1) parser.

3. (a) Syntax → rules language should follow regardless of its context.
Semantics → contextual rules.

(b) $G = \langle N, T, R, S \rangle$

Rules for CFG.	CSG.
$A \to \alpha$.	$\alpha A \beta \to \alpha \gamma \beta$.
$\alpha \in (N \cup T)^*$	$\alpha, \beta \in (N \cup T)^*$

$A \in N$.

$y \in (N \cup T)^*$

(c) type error.

scope error (unidentified vars)

duplicate variables

(d) nested scopes. (need to handle them by numbers)

could be collision in names.

(e). _ —

4. (a) ① During optimisation, since moving code around is common, you don't have to constantly change the code, only a change of pointer is sufficient.

② It saves space compared to quadruples

Since don't have to generate extra redundant temporaries

(b) Inlining.

putting the code of a function directly in where the function is called.

↑: saves function call cost.

↓: expands code size (registry spills) / recursion

↑ large function

Loop unrolling

unroll the loop / expand code in loop.
Certain amount of times (given known loop conditions at compile time)

↑ save expensive jumps / stalls in execution.

↓ code size expand quickly / might cause registry spill

(c) activation record
- control link.
- access link
- function call location
- func. parameters
- local variable.
- return value.
- temps

(d) explicit allocation:

↑ * full control of memory.
* speedy (no extra GC process).

↓ * cause memory leak. (if not careful).
* free collision. (ptr → free → freed again)

ptr2 [illegible]

double free.

G.C.

↑ * automatically deallocates unused references.

* reduces / avoid mem leak ...

↓ * additional resources / performance impact.

* incompatible with manual management