Heman Bagul

Assignment 2

1. [7 points] What two features would you focus on when designing a programming language? Why?

⇒ In my opinion, I would focus on making a programming language that can handle parallel computations better. In today's language, dealing with lock, thread or callback functions for asynchronous operations can be very difficult. So, If I have to design a programming language then I will make sure it helps the programmer to express parallel operations in a clear way without any unexpected race condition, side effects or deadlocks. Another feature would be to have a high speed compilation.

2. [7 points] Discuss some examples of efficiency in the Java programming language.

⇒ Java is efficiet in regard to execution time, memory consumption and energy consumption is much lower for high processing jobs. It has an automatic memory management and much better garbage collection so it does space utilization much efficiently. Also, tasks are very optimally handled by the Java Virtual machine.

3. [7 points] Discuss some examples of inefficiency in the Java programming language.

⇒ Java consumes lot of memory compared to other programming language such as C and C++. It takes a lot of time to compile large programming files. It requires lots of code to write small things where we can develope same in less in other languages.

4. [7 points] Discuss some examples of regularity in the Java programming language.

⇒ Using Java we can organize our software as a combination of different types of objects that integrate both behavior and data. Java also follows the case of no explicat pointer which help to make Java easy to use and easy to understand.

5. [7 points] Discuss some examples of irregularity in the Java programming language.

=> Some of the irregularity of Java are, it is very slow in performance, since it consumes lots of memory. Java is limited to latency critical tuning and there is no backup in Java.

6. [10 pts] Should a language require the declaration of variables? Languages such as Lisp and Python allow variables names to be used without declarations, while C, Java, and Ada require all variables to be declared. Discuss the requirement that variables should be declared from the point of view of readability, writability, efficiency, and security.

=> Yes, a language should definetly require declaration of variable. Whenever you declare a variable it makes things more organized. As a programmer declaration variable is also important because we have several storage classer and scope of variable that defines the scope and lifetime of a variable. Suppose in C, extern key word is used to delare that variable in somewhere and have to used in current source file. If extern varible is not declared anywhere then it will come as syntax errors and it is similar with Java.

7. [5 points] Describe some strings that are represented by the following regular expressions:
   - [-+]?[0-9]+\.?[0-9]*
   - [a-z]+ and ([a-z]+|\.\.\.)

a) [-+]? once or not at all
[0-9]+ once or more
\.? once or not at all
[0-9]* zero or more

Strings: +12.12
-13.13
1234

(b) [a-z]+ once or more

strings: cats and dog
eggs and chicken

8. [10 points] Build the regular expression for the following:
   a) • Identifiers in a language that must start with an underscore character and must end in a numeric digit. The length can be any size and values in between can be any alphanumeric character.
   b) • A phone number with either the following formats: (888) 888-8888 or 888-888-8888
   c) • The VCU V Number

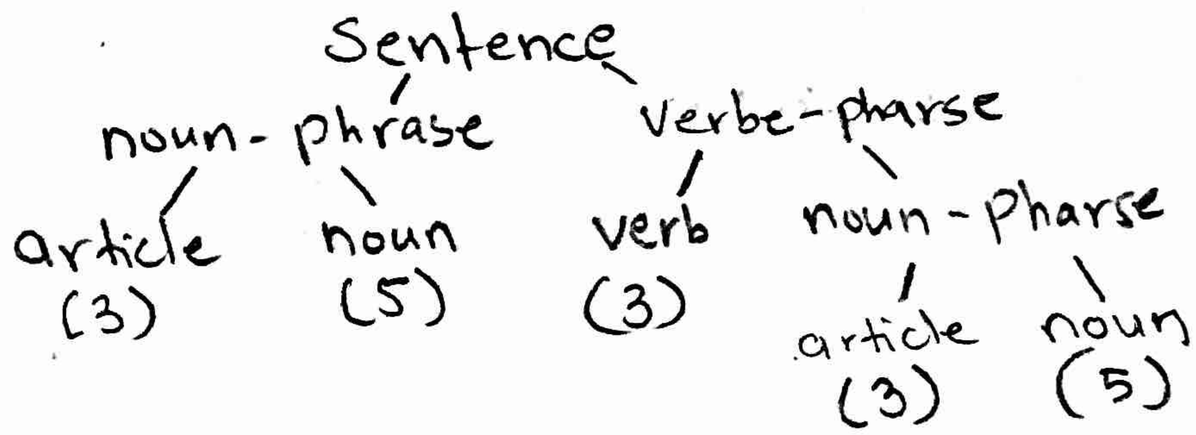
a) ^_[A-Z a-z 0-9]* [0-9]$

b) ^\(?\d{3}\)? [-]? \d{3} [-]? \d{4}$

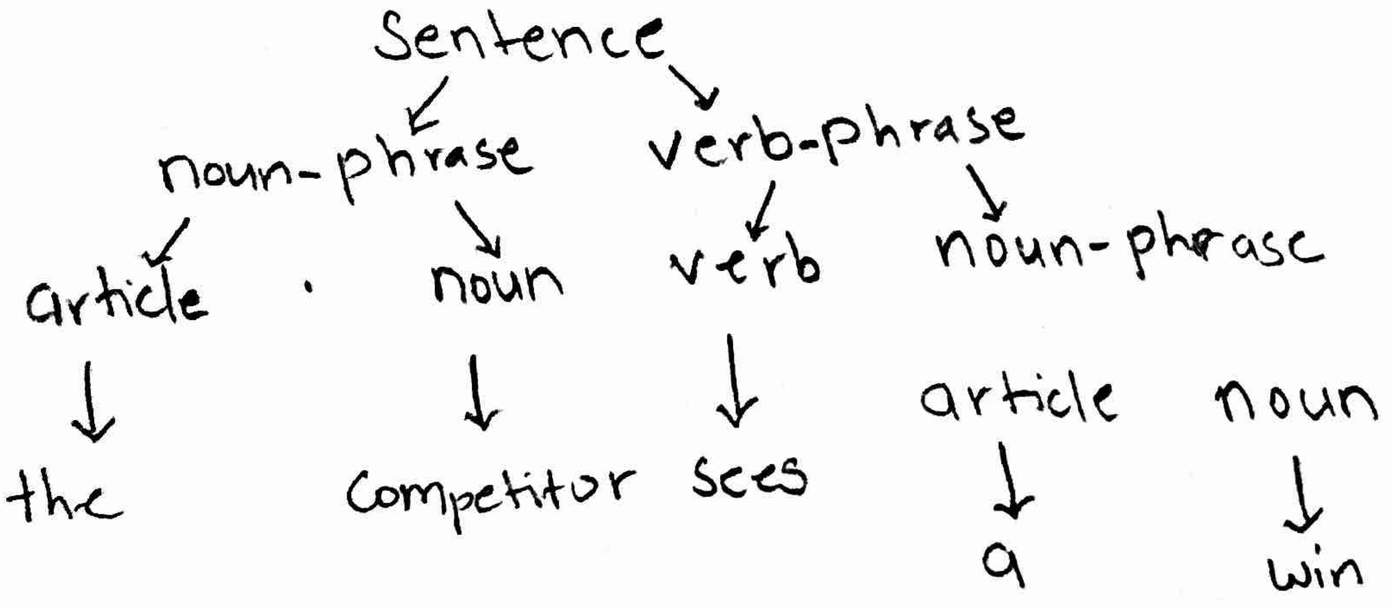
c) (v|V)-\d{8}

9. [10 points] Using the grammar below, how many legal sentences are there? Why is that? Suppose white space was completely ignored in the grammar so that sentences could be written as "thecompetitorseesawin." Can this grammar still be parsed? Explain

```
1) sentence → noun-phrase verb-phrase
2) noun-phrase → article noun
3) article → a | and | the
4) noun → girl | competitor | win | dog | comp
5) verb-phrase → verb noun-phrase
6) verb → sees | permits | objects
```



∴ The number of legal sentence is $3 \times 5 \times 3 \times 3 \times 5 = 675$



∴ The sentence "thecompetitorseesawin" can be parsed.

10. [15 points] Add the following four operations in the proper location for the order of operations to apply to the EBNF grammar below.
    - subtraction,
    - division,
    - integer modulus division,
    - exponents

```
expr → term { + term }
term → factor { * factor }
factor → ( expr ) | number
number → digit { digit }
digit → 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
```

Figure 6.18 EBNF rules for simple integer arithmetic expressions

expr → term {+|-term}
term → mod {% mod}
mod → factor {*|/ factor}
factor → (expr) | number
number → digit {digit}
digit → 0|1|2|3|4|5|6|7|8|9

expr → term { + term }

term → factor { * factor }

factor → (expr) | number

number → digit {digit}

digit → 0 - 9

expr → expr + term | term

term → term * factor | factor

factor → (expr) | number

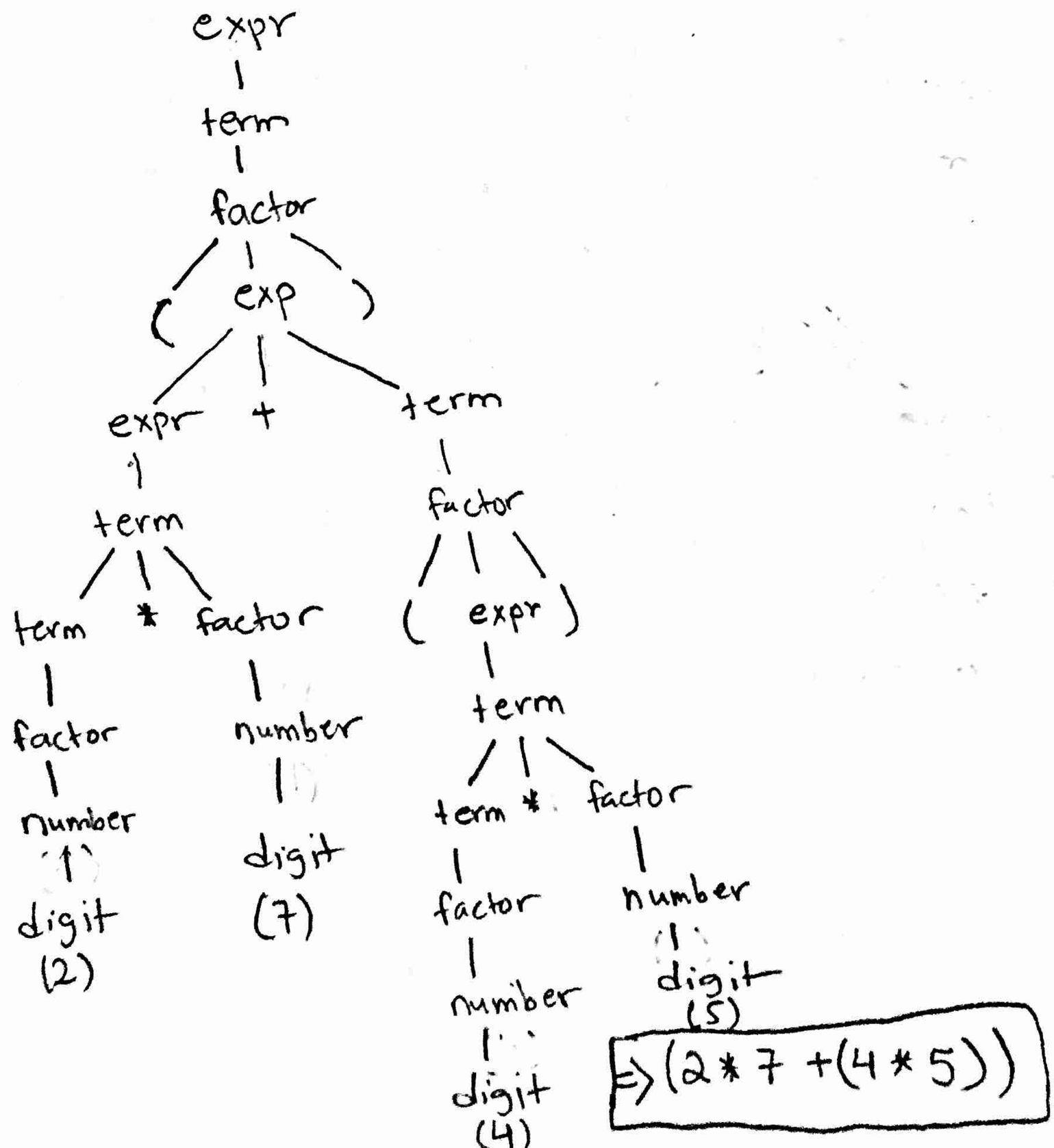
number → number digit | digit

digit → 1 - 9

11. [15 points] Use the grammar above to draw the following:
    a) • (2 * 7 + (4 * 5)) – parse tree
    b) • 3 * (4 * 5) + (6 + 7) – abstract syntax tree
    c) • 3 + (4 + 2) * 6 + (7 * 8) – parse tree

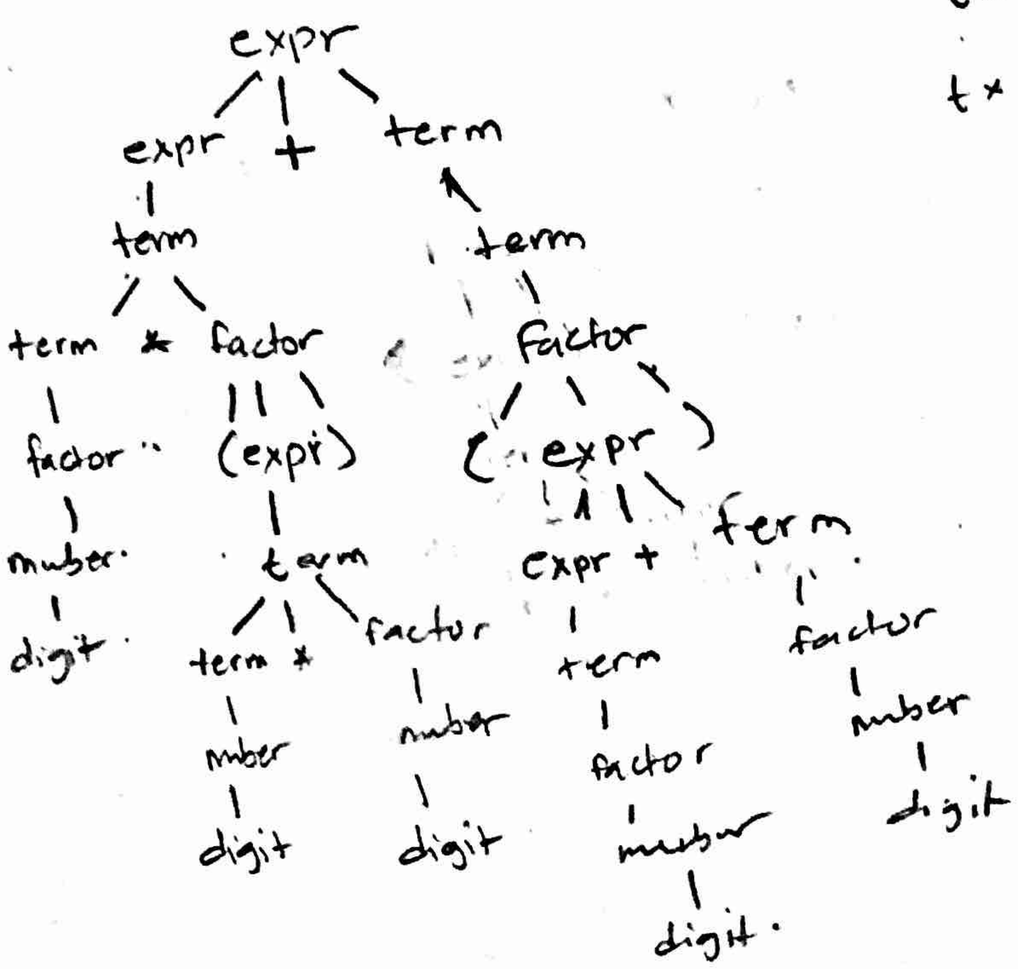
expr → term → factor → (expr) → (expr + term) → (term + term)
→ (term * factor + term) → (factor * factor + term) → (factor * factor + factor)
→ (factor * factor + (expr)) → (factor * factor + (term * factor)) →
(factor * factor + (factor * factor)) → (number * number + (number * number)
→ (2 * 7 + (4 * 5))

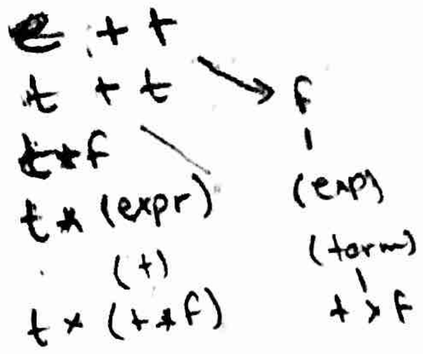
a)



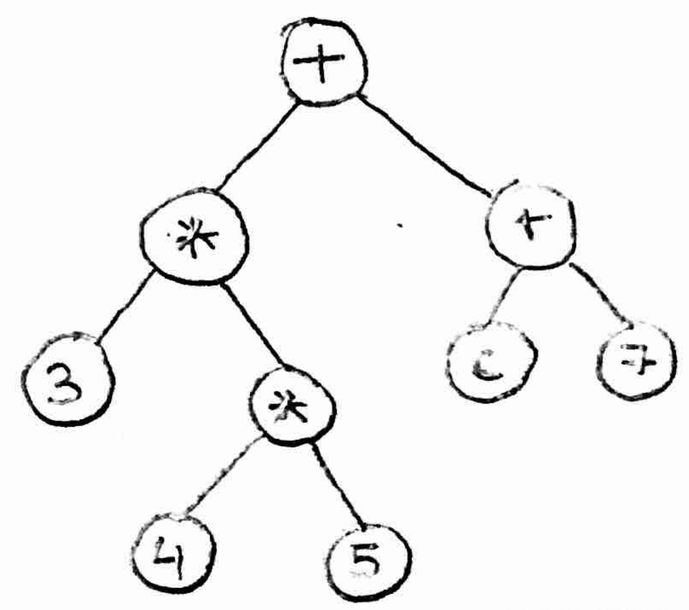
b) $3*(4*5)+(6+7)$

Parse tree



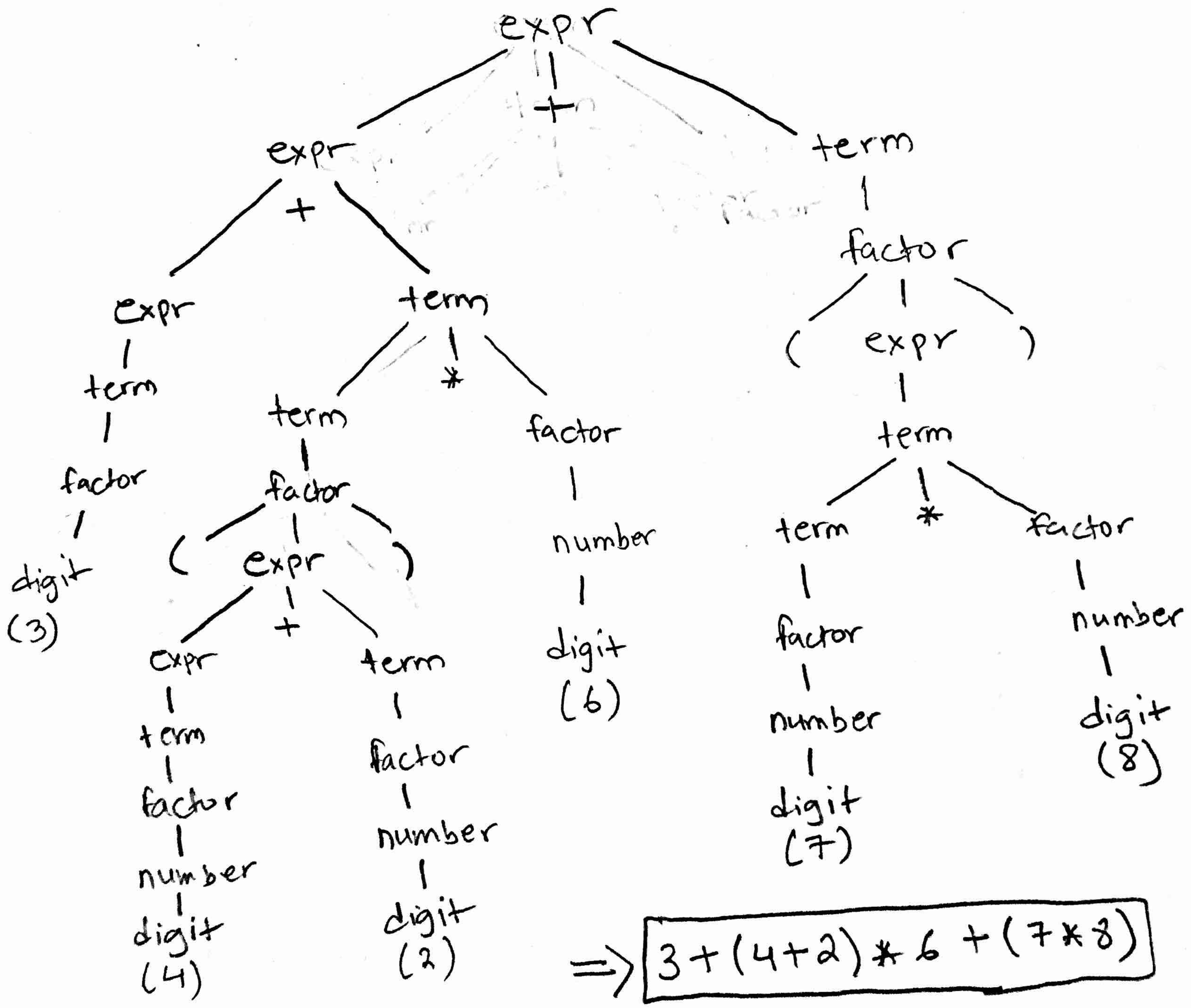


AST



$3*(4*5)+(6+7)$

c) $3+(4+2)*6+(7*8)$



$\Rightarrow$ $3+(4+2)*6+(7*8)$