

CS4120/4121/5120/5121—Spring 2026

Homework 2

Syntactic Analysis

Due: Monday, February 9, 11:59PM

0 Updates

- None yet; watch this space.

1 Instructions

1.1 Partners

You may work alone or with *one* partner on this assignment. But remember that the course staff is happy to help with problems you run into. Use Ed for questions, attend office hours, or set up meetings with any course staff member for help.

1.2 Homework structure

All problems are required of all students.

1.3 Tips

You may find the Dot and Graphviz packages helpful for drawing graphs. Your compiler will be required to export a visualization in the [dot](#) format for a later assignment. You can get these packages for multiple OSes from the [Graphviz download page](#). [An example of a DFA drawn using Graphviz](#) may be useful.

1.4 AI Policy

You are not permitted to use AI systems to solve the problems or word your answers. Thanks!

2 Problems

1. Context-free grammars

Consider the following BNF grammar over the alphabet $\{\text{let}, \text{n}, \text{p}, \text{q}, +, -, :, =\}$, where the start symbol is S :

$$\begin{aligned} S &\rightarrow D \mid SD \\ D &\rightarrow \text{let } \text{n} = E; \\ E &\rightarrow T + E \mid T - E \mid T \\ T &\rightarrow V \mid \varepsilon \\ V &\rightarrow \text{p } V \mid \text{q } V \mid \text{p} \mid \text{q} \end{aligned}$$

- Briefly identify two reasons why this grammar is not LL(1).
- Is it possible to refactor this grammar into an equivalent one that is LL(1)? If so, give the new grammar. If not, explain why such a grammar cannot be written.
- Is the string `let n = -pqp;` in the language of the grammar? If so, show the rightmost derivation of the string. If not, explain why it cannot be derived.
- Show the start state for the LR(1) automaton for this grammar. (Hint: remember to take the closure of items.)

2. Designing grammars

Consider the basic symbols needed to represent simple arithmetic operations. The terminals are $\{\text{num}, (,), +, -, *, /, \%, ^\}$ and the operations are as follows (in decreasing order of precedence):

Operator	Associativity
$^$	right
$*, /, \%$	left
$+, -$	left

A restricted first attempt might be

$$\begin{aligned} E &\rightarrow E \text{ Op } E \mid (E) \mid \text{<num>} \\ \text{Op} &\rightarrow + \mid * \mid ^ \end{aligned}$$

This grammar has a few problems: it is ambiguous, it does not enforce operator precedence, and it does not enforce associativity.

- Give an example to show that the grammar is ambiguous.
- Write an LR(1) grammar for the full problem domain, including all the operators in the table with the appropriate precedence relationships and associativity. You don't need to show parse tables.
- Show the bottom-up derivation of $4^5\%2 + 1 * (2 + 2)$ using your grammar.

3. LR grammars

Is the following grammar LALR(1)? What about LR(1)? Explain.

$$\begin{aligned}S &\rightarrow cXq \mid cYp \mid Xp \mid Yq \\X &\rightarrow a \\Y &\rightarrow a\end{aligned}$$

4. Earley parsing

Consider the following grammar:

$$\begin{aligned}S &\rightarrow S \ S \mid A \\A &\rightarrow a\end{aligned}$$

Show the steps of parsing the string “aaa” using an Earley parser, including all the sets I_j .

3 Submission

Submit your solution as a PDF file on CMSX. This file should contain your name, your NetID, all known issues you have with your solution, and the names of anyone you have discussed the homework with.