**编译原理第六次作业**

**任凯 2020141460080**

Consider the following grammar:



1. Eliminate left recursion using BNF;

**A:**



1. Design an L-SDD to compute the value of the expressions generated by the grammar of (1);

**A：**

|  |  |
| --- | --- |
| **Production** | **Semantic Rules** |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |

1. Convert the SDD of (2) to SDT.

**A:** From (2) the SDT as followed can be obtained:



1. Implement the SDT of (3) as a recursive-descent parser

**A:**

1. procedure exp
2. begin
3. factor();
4. exp'();
5. end exp
6. procedure exp'
7. begin
8. while token = + do
9. match(+);
10. factor();
11. exp'();
12. end while;
13. end exp'
14. procedure factor
15. begin
16. case token of
17. ( : match(();
18. exp();
19. match());
20. number : match(number);
21. else error;
22. end case;
23. end factor