Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

Лабораторная работа № 2

по дисциплине: «Конструирование компиляторов»

Тема: «Преобразования грамматик»

Выполнил студент группы ИУ7-21М

Константогло Александр

Москва

30 марта 2020

Цель работы: приобретение практических навыков реализации наиболее важных видов преобразований грамматик, чтобы удовлетворить требованиям алгоритмов синтаксического разбора.

Задачи работы:

1) Принять к сведению соглашения об обозначениях, принятые в литературе по теории формальных языков и грамматик и кратко описанные в приложении.

2) Познакомиться с основными понятиями и определениями теории формальных языков и грамматик.

3) Детально разобраться в алгоритме устранения левой рекурсии.

4) Разработать, тестировать и отладить программу устранения левой рекурсии.

5) Разработать, тестировать и отладить программу преобразования грамматики в соответствии с предложенным вариантом.

Постановка задачи

Устранение левой рекурсии.

Определение. Нетерминал A КС-грамматики $G = (N, \Sigma, P, S)$ называется рекурсивным, если $A =>+ \alpha A \beta$ для некоторых α и β. Если $\alpha = \varepsilon$, то A называется леворекурсивным. Аналогично, если $\beta = \varepsilon$, то A называется праворекурсивным. Грамматика, имеющая хотя бы один леворекурсивный нетерминал, называется леворекурсивной. Аналогично определяется праворекурсивная грамматика. Грамматика, в которой все нетерминалы, кроме, быть может, начального символа, рекурсивные, называется рекурсивной.

Некоторые из алгоритмов разбора не могут работать с леворекурсивными грамматиками. Можно показать, что каждый КС-язык определяется хотя бы одной не леворекурсивной грамматикой.

Построить программу, которая в качестве входа принимает приведенную КС-грамматику $G = (N, \Sigma, P, S)$ и преобразует ее в эквивалентную КС-грамматику G' без левой рекурсии.

Вариант 7. Преобразование к нормальной форме Грейбах 1.

Определение. КС-грамматика $G = (N, \Sigma, P, S)$ называется грамматикой в нормальной форме Грейбах, если в ней нет ε-правил и каждое правило из P, отличное от $S \to \varepsilon$, имеет вид $A \to a\alpha$, где $a \in \Sigma$ и $\alpha \in N^*$.

Постройте программу, которая в качестве входа принимает не леворекурсивную приведенную КС-грамматику $G = (N, \Sigma, P, S)$ и преобразует ее в эквивалентную КС-грамматику G' в нормальной форме Грейбах.

Текст программы, устраняющий левую рекурсию и применяющей левую факторизацию

Программа написана на языке C++.

leftrecursion.h

```
#pragma once
#include <map>
#include <set>
```

```
#include <vector>
#include <string>

class Rhs {
	// true - терминал; false - нетерминал
	std::vector<std::pair<bool, std::string>> rhs;
public:
	Rhs();
	void insert_symbol(bool sym_type, std::string symbol);
	std::vector<std::pair<bool, std::string>> get_rhs();
	friend bool operator<(const Rhs& left, const Rhs& right);
	~Rhs();
};

class Grammar {
	std::set<std::string> nonterminalsymbols;
	std::map<std::string, std::string> terminalsymbols;
	std::map<std::string, std::set<Rhs>> rules;
	std::string startsymbol;
	static std::pair<int, std::vector<std::pair<int, int>>>
max_elements(std::vector<std::vector<int>> matr);
public:
	Grammar();
	void insert_nonterminalsymbol(std::string nonterminalsymbol);
	void insert_terminalsymbol(std::string name, std::string spell);
	void insert_rule(std::string nonterminalsymbol, Rhs rhs);
	void set_rules(std::map<std::string, std::set<Rhs>> rules);
	void set_startsymbol(std::string startsymbol);
	std::set<std::string> get_nonterminalsymbols();
	std::map<std::string, std::string> get_terminalsymbols();
	std::map<std::string, std::set<Rhs>> get_rules();
	std::string get_startsymbol();
	static Grammar remove_left_recursion(Grammar grammar);
	static Grammar left_factorization(Grammar grammar);
	~Grammar();
};

class XmlHandler {
public:
	Grammar read_xml(std::string in_file);
	void write_xml(std::string out_file, Grammar grammar);
};
```

leftrecursion.cpp

```
#include "leftrecursion.h"
#include "tinyxml2.h"
#include <algorithm>
#include <iostream>

using namespace std;
using namespace tinyxml2;

Rhs::Rhs() : rhs(vector<pair<bool, string>>()) { }

void Rhs::insert_symbol(bool sym_type, string symbol)
{
	rhs.push_back(pair<bool, string>(sym_type, symbol));
}

vector<pair<bool, string>> Rhs::get_rhs()
{
```

```
	return rhs;
}

Rhs::~Rhs()
{
	for (int i = 0; i < rhs.size(); i++)
		rhs[i].second.clear();
	rhs.clear();
}

bool operator<(const Rhs & left, const Rhs & right)
{
	if (left.rhs.size() < right.rhs.size())
		return true;
	if (left.rhs.size() > right.rhs.size())
		return false;
	for (int i = 0; i < left.rhs.size(); i++)
	{
		if (left.rhs[i].first < right.rhs[i].first)
			return true;
		if (left.rhs[i].first > right.rhs[i].first)
			return false;
		if (left.rhs[i].second < right.rhs[i].second)
			return true;
		if (left.rhs[i].second > right.rhs[i].second)
			return false;
	}
	return false;
}

pair<int, vector<pair<int, int>>> Grammar::max_elements(vector<vector<int>> matr)
{
	pair<int, vector<pair<int, int>>> positions;
	int maxv = 0, maxx = -1, maxy = -1, size = matr.size();
	for (int i = 0; i < size; i++)
		for (int j = 0; j < size; j++)
			if (matr[i][j] > maxv)
			{
				maxv = matr[i][j];
				maxx = i;
				maxy = j;
			}
	if (maxv == 0)
		return positions;
	for (int i = 0; i < size; i++)
		if (matr[maxx][i] == maxv)
			positions.second.push_back(pair<int, int >(maxx, i));
	positions.first = maxv;
	return positions;
}

Grammar::Grammar() : nonterminalsymbols(set<string>()),
				terminalsymbols(map<string, string>()),
				rules(map<string, set<Rhs>>()),
				startsymbol("") { }
```

```
void Grammar::insert_nonterminalsymbol(string nonterminalsymbol)
{
	nonterminalsymbols.insert(nonterminalsymbol);
}

void Grammar::insert_terminalsymbol(string name, string spell)
{
	terminalsymbols.insert(pair<string, string>(name, spell));
}

void Grammar::insert_rule(string nonterminalsymbol, Rhs rhs)
{
	if (rules.find(nonterminalsymbol) == rules.end())
		rules.insert(pair<string, set<Rhs>>(nonterminalsymbol, set<Rhs>()));
	rules[nonterminalsymbol].insert(rhs);
}

void Grammar::set_rules(map<string, set<Rhs>> rules)
{
	this->rules = rules;
}

void Grammar::set_startsymbol(string startsymbol)
{
	this->startsymbol = startsymbol;
}

set<string> Grammar::get_nonterminalsymbols()
{
	return nonterminalsymbols;
}

map<string, string> Grammar::get_terminalsymbols()
{
	return terminalsymbols;
}

map<string, set<Rhs>> Grammar::get_rules()
{
	return rules;
}

string Grammar::get_startsymbol()
{
	return startsymbol;
}

Grammar Grammar::remove_left_recursion(Grammar grammar)
{
	Grammar g;
	map<string, set<Rhs>> rules = grammar.get_rules();
	set<string> nonterminals = grammar.get_nonterminalsymbols();
	map<string, string> terminals = grammar.get_terminalsymbols();
	set<string> new_nonterminals = set<string>();
	for (string i : nonterminals)
	{
```

```
		if (rules.count(i) == 0)
			continue;
		for (string j : nonterminals)
		{
			if (i <= j)
				break;
			if (rules.count(j) == 0)
				continue;
			set<Rhs> new_rhs;
			for (Rhs r : rules[i])
			{
				if (!(r.get_rhs().begin()->first) && r.get_rhs().begin()->second ==
j)
				{
					for (Rhs r1 : rules[j])
					{
						Rhs rhs;
						for (pair<bool, string> pr : r1.get_rhs())
							rhs.insert_symbol(pr.first, pr.second);
						for (int k = 1; k < r.get_rhs().size(); k++)
							rhs.insert_symbol(r.get_rhs()[k].first,
r.get_rhs()[k].second);
						new_rhs.insert(rhs);
					}
				}
				else
				{
					Rhs rhs;
					for (pair<bool, string> pr : r.get_rhs())
						rhs.insert_symbol(pr.first, pr.second);
					new_rhs.insert(rhs);
				}
			}
			rules[i].clear();
			rules[i] = new_rhs;
		}
		string nonterminal = "";
		for (Rhs r : rules[i])
		{
			if (!(r.get_rhs().begin()->first) && r.get_rhs().begin()->second == i)
			{
				if (nonterminal == "")
				{
					unsigned int k = 0;
					while (nonterminals.find(i + to_string(++k)) !=
nonterminals.end() ||
						new_nonterminals.find(i + to_string(k)) !=
new_nonterminals.end());
					nonterminal = i + to_string(k);
					new_nonterminals.insert(nonterminal);
					//terminals.insert(pair<string, string>("EPSILON", ""));
					rules.insert(pair<string, set<Rhs>>(nonterminal,
set<Rhs>()));
					//Rhs rhs;
					//rhs.insert_symbol(true, "EPSILON");
					//rules[nonterminal].insert(rhs);
```

```
                                    break;
                             }
                      }
               }
               if (nonterminal != "")
               {
                      set<Rhs> chains;
                      for (Rhs r : rules[i])
                      {
                             Rhs rhs;
                             if (!(r.get_rhs().begin()->first) && r.get_rhs().begin()->second ==
i)
                             {
                                    for (int i = 1; i < r.get_rhs().size(); i++)
                                           rhs.insert_symbol(r.get_rhs()[i].first,
r.get_rhs()[i].second);
                                    rules[nonterminal].insert(rhs); // Ликвидация эпсилон-
переходов
                                    rhs.insert_symbol(false, nonterminal);
                                    rules[nonterminal].insert(rhs);
                             }
                             else
                             {
                                    for (pair<bool, string> p : r.get_rhs())
                                           rhs.insert_symbol(p.first, p.second);
                                    chains.insert(rhs); // Ликвидация эпсилон-переходов
                                    rhs.insert_symbol(false, nonterminal);
                                    chains.insert(rhs);
                             }
                      }
                      rules[i].clear();
                      rules[i] = chains;
               }
       }
       for (pair<string, string> term : terminals)
              g.insert_terminalsymbol(term.first, term.second);
       for (string nonterm : nonterminals)
              g.insert_nonterminalsymbol(nonterm);
       for (string nonterm : new_nonterminals)
              g.insert_nonterminalsymbol(nonterm);
       g.set_startsymbol(grammar.get_startsymbol());
       g.set_rules(rules);
       return g;
}

Grammar Grammar::left_factorization(Grammar grammar)
{
       Grammar g;
       map<string, set<Rhs>> rules = grammar.get_rules();
       set<string> nonterminals = grammar.get_nonterminalsymbols();
       map<string, string> terminals = grammar.get_terminalsymbols();
       set<string> new_nonterminals = set<string>();
       for (string nonterminal : nonterminals)
       {
              if (rules.count(nonterminal) == 0)
                     continue;
```

```
		while (true)
		{
			vector<vector<int>> matr;
			vector<Rhs> rs;
			for (Rhs r : rules[nonterminal])
			{
				rs.push_back(r);
				vector<int> str;
				for (Rhs r1 : rules[nonterminal])
				{
					if (!(r < r1) && !(r1 < r))
					{
						str.push_back(-1);
						continue;
					}
					int minsize = min(r.get_rhs().size(), r1.get_rhs().size());
					bool b = true;
					for (int i = 0; i < minsize; i++)
					{
						if (r.get_rhs()[i].first != r1.get_rhs()[i].first ||
r.get_rhs()[i].second != r1.get_rhs()[i].second)
						{
							str.push_back(i);
							b = false;
							break;
						}
					}
					if (b)
						str.push_back(minsize);
				}
				matr.push_back(str);
			}

			pair<int, vector<pair<int, int>>> maxelements = max_elements(matr);
			if (maxelements.first < 1)
				break;
			// Создание нового нетерминала и добавление его в список правил
			string new_nonterminal;
			unsigned int k = 0;
			while (nonterminals.find(nonterminal + to_string(++k)) !=
nonterminals.end() ||
				new_nonterminals.find(nonterminal + to_string(k)) !=
new_nonterminals.end());
			new_nonterminal = nonterminal + to_string(k);
			new_nonterminals.insert(new_nonterminal);
			rules.insert(pair<string, set<Rhs>>(new_nonterminal, set<Rhs>()));
			// Добавление всех правил для нового нетерминала
			Rhs rhs;
			for (int j = maxelements.first; j <
rs[maxelements.second[0].first].get_rhs().size(); j++)

		rhs.insert_symbol(rs[maxelements.second[0].first].get_rhs()[j].first,
rs[maxelements.second[0].first].get_rhs()[j].second);
			if (rhs.get_rhs().size() == 0)
			{
				terminals.insert(pair<string, string>("EPSOLON", ""));
```

```
                        rhs.insert_symbol(true, "EPSOLON");
                    }
                    rules[new_nonterminal].insert(rhs);
                    rhs.~Rhs();
                    for (pair<int, int> p : maxelements.second)
                    {
                        for (int j = maxelements.first; j < rs[p.second].get_rhs().size();
j++)
                            rhs.insert_symbol(rs[p.second].get_rhs()[j].first,
rs[p.second].get_rhs()[j].second);
                        if (rhs.get_rhs().size() == 0)
                        {
                            terminals.insert(pair<string, string>("EPSOLON", ""));
                            rhs.insert_symbol(true, "EPSOLON");
                        }
                        rules[new_nonterminal].insert(rhs);
                        rhs.~Rhs();
                    }
                    // Изменение правил для текущего нетерминала
                    set<Rhs> srhs;
                    set<int> v;
                    v.insert(maxelements.second[0].first);
                    for (pair<int, int> p : maxelements.second)
                        v.insert(p.second);
                    // Изменившаяся правая часть
                    for (int j = 0; j < maxelements.first; j++)

        rhs.insert_symbol(rs[maxelements.second[0].first].get_rhs()[j].first,
rs[maxelements.second[0].first].get_rhs()[j].second);
                    rhs.insert_symbol(false, new_nonterminal);
                    srhs.insert(rhs);
                    rhs.~Rhs();
                    // Остальное без изменений
                    for (int j = 0; j < rs.size(); j++)
                    {
                        if (v.find(j) != v.end())
                            continue;
                        for (pair<bool, string> p : rs[j].get_rhs())
                        {
                            rhs.insert_symbol(p.first, p.second);
                        }
                        srhs.insert(rhs);
                        rhs.~Rhs();
                    }
                    rules[nonterminal].clear();
                    rules[nonterminal] = srhs;
                }
        }
        for (pair<string, string> term : terminals)
            g.insert_terminalsymbol(term.first, term.second);
        for (string nonterm : nonterminals)
            g.insert_nonterminalsymbol(nonterm);
        for (string nonterm : new_nonterminals)
            g.insert_nonterminalsymbol(nonterm);
        g.set_startsymbol(grammar.get_startsymbol());
        g.set_rules(rules);
```

```
	return g;
}

Grammar::~Grammar()
{
	nonterminalsymbols.clear();
	terminalsymbols.clear();
	rules.clear();
	startsymbol.clear();
}

Grammar XmlHandler::read_xml(string in_file)
{
	Grammar grammar;
	XMLDocument doc;
	doc.LoadFile(in_file.c_str());
	if (doc.ErrorID())
		throw runtime_error("Не удалось обработать XML файл\n");
	XMLHandle docHandle(&doc);
	XMLHandle root = docHandle.FirstChildElement("grammar");
	XMLElement* terminalsymbols = root.FirstChildElement("terminalsymbols").ToElement();
	XMLElement* term;
	while (terminalsymbols)
	{
		term = terminalsymbols->FirstChildElement("term");
		while (term) {
			grammar.insert_terminalsymbol(term->Attribute("name"), term-
>Attribute("spell"));
			term = term->NextSiblingElement();
		}
		terminalsymbols = terminalsymbols->NextSiblingElement();
	}
	XMLElement* nonterminalsymbols =
root.FirstChildElement("nonterminalsymbols").ToElement();
	XMLElement* nonterm;
	while (nonterminalsymbols)
	{
		nonterm = nonterminalsymbols->FirstChildElement("nonterm");
		while (nonterm) {
			grammar.insert_nonterminalsymbol(nonterm->Attribute("name"));
			nonterm = nonterm->NextSiblingElement();
		}
		nonterminalsymbols = nonterminalsymbols->NextSiblingElement();
	}
	XMLElement* startsymbol = root.FirstChildElement("startsymbol").ToElement();
	if (!startsymbol)
		throw runtime_error("Отсутствует стартовый символ\n");
	grammar.set_startsymbol(startsymbol->Attribute("name"));
	set<string> nonterminals = grammar.get_nonterminalsymbols();
	if (nonterminals.find(startsymbol->Attribute("name")) == nonterminals.end())
		throw runtime_error("Стартовый символ не найден в списке нетерминалов\n");
	map<string, string> terminals = grammar.get_terminalsymbols();
	XMLElement* productions = root.FirstChildElement("productions").ToElement();
	XMLElement* production, *lhs, *rhs;
	while (productions)
	{
```

```
		production = productions->FirstChildElement("production");
		while (production)
		{
			lhs = production->FirstChildElement("lhs");
			if (!lhs)
				throw runtime_error("Одна из продукций не имеет левой части\n");
			rhs = production->FirstChildElement("rhs");
			if (!rhs)
				throw runtime_error("Одна из продукций не имеет правой части\n");
			string lhs_name;
			lhs_name = lhs->Attribute("name");
			if (nonterminals.find(lhs_name) == nonterminals.end())
				throw runtime_error("В списке правил обнаружен неизвестный
символ\n");
			XMLElement* symbol = rhs->FirstChildElement("symbol");
			Rhs rhs;
			bool sym_type;
			string sym;
			while (symbol)
			{
				if (string(symbol->Attribute("type")) == "nonterm")
					sym_type = false;
				else if (string(symbol->Attribute("type")) == "term")
					sym_type = true;
				else
					throw runtime_error("Неизвестный тип элемента в правой части
продукции\n");
				sym = symbol->Attribute("name");
				if (sym_type)
				{
					if (!terminals.count(sym))
						throw runtime_error("Неизвестный терминал в правой
части продукции\n");
				}
				else
				{
					if (nonterminals.find(sym) == nonterminals.end())
						throw runtime_error("Неизвестный нетерминал в правой
части продукции\n");
				}
				rhs.insert_symbol(sym_type, sym);
				symbol = symbol->NextSiblingElement();
			}
			grammar.insert_rule(lhs_name, rhs);
			production = production->NextSiblingElement();
		}
		productions = productions->NextSiblingElement();
	}
	return grammar;
}

void XmlHandler::write_xml(std::string out_file, Grammar grammar)
{
	static const char* xml =
		"<?xml version=\"1.0\" encoding=\"UTF-8\"?>"
		"<grammar name=\"G1\">"
```

```
		"	<terminalsymbols/>"
		"	<nonterminalsymbols/>"
		"	<productions/>"
		"	<startsymbol/>"
		"</grammar>";
	XMLDocument doc;
	doc.Parse(xml);
	XMLElement* terminalsymbols = doc.FirstChildElement("grammar")-
>FirstChildElement("terminalsymbols");
	map<string, string> terminals = grammar.get_terminalsymbols();
	for (pair<string, string> term : terminals)
	{
		XMLElement* termelement = terminalsymbols->InsertNewChildElement("term");
		termelement->SetAttribute("name", term.first.c_str());
		termelement->SetAttribute("spell", term.second.c_str());
	}
	XMLElement* nonterminalsymbols = doc.FirstChildElement("grammar")-
>FirstChildElement("nonterminalsymbols");
	set<string> nonterminals = grammar.get_nonterminalsymbols();
	for (string nonterm : nonterminals)
		nonterminalsymbols->InsertNewChildElement("nonterm")->SetAttribute("name",
nonterm.c_str());
	map<string, set<Rhs>> rules = grammar.get_rules();
	XMLElement* productions = doc.FirstChildElement("grammar")-
>FirstChildElement("productions");
	for (pair<string, set<Rhs>> term : rules)
		for (Rhs rhs : term.second)
		{
			XMLElement* production = productions->InsertNewChildElement("production");
			production->InsertNewChildElement("lhs")->SetAttribute("name",
term.first.c_str());
			XMLElement* rhselement = production->InsertNewChildElement("rhs");
			vector<pair<bool, string>> chain = rhs.get_rhs();
			for (pair<bool, string> sym : chain)
			{
				XMLElement* symbolelement = rhselement-
>InsertNewChildElement("symbol");
				if (sym.first)
					symbolelement->SetAttribute("type", "term");
				else
					symbolelement->SetAttribute("type", "nonterm");
				symbolelement->SetAttribute("name", sym.second.c_str());
			}
		}
	doc.FirstChildElement("grammar")->FirstChildElement("startsymbol")->SetAttribute("name",
grammar.get_startsymbol().c_str());
	doc.SaveFile(out_file.c_str());
}
```

Source.cpp

```
#include "leftrecursion.h"
#include <iostream>

using namespace std;

int main(int argc, char* argv[])
```

```
{
	setlocale(LC_ALL, "Russian");
	XmlHandler xml_handler;
	try {
		Grammar grammar = xml_handler.read_xml(argv[1]);
		grammar = Grammar::remove_left_recursion(grammar);
		grammar = Grammar::left_factorization(grammar);
		xml_handler.write_xml(argv[2], grammar);
	}
	catch (const exception& err) {
		cerr << err.what() << endl;
	}
	return 0;
}
```

Результаты тестирования

Файл `UnitTests.cpp` содержит тесты для проверки грамматик в XML-файлах после устранения левой рекурсии и левой факторизации. Все тесты пройдены успешно.

Далее приведены примеры тестов. Все тесты имеют схожую структуру, поэтому приведён пример только одной тестовой функции.

UnitTests.cpp

```
#include "pch.h"
#include "CppUnitTest.h"
#include "../leftrecursion/leftrecursion.h"
#include <iostream>
#include <fstream>

using namespace Microsoft::VisualStudio::CppUnitTestFramework;

namespace UnitTests
{
	TEST_CLASS(UnitTests)
	{
	public:

		TEST_METHOD(Factorization1)
		{
			setlocale(LC_ALL, "Russian");
			try {
				// Входная грамматика
				const char* inp = "..\\inps\\inp0.xml";
				// Преобразованная грамматика
				const char* outp = "..\\outps\\outp0.xml";
				// Эталонная грамматика
				const char* standard = "..\\standards\\standard0.xml";
				XmlHandler xml_handler;
				Grammar grammar = xml_handler.read_xml(inp);
				grammar = Grammar::remove_left_recursion(grammar);
				grammar = Grammar::left_factorization(grammar);
				xml_handler.write_xml(outp, grammar);
				std::fstream fin1(outp, std::ios::in | std::ios::ate |
std::ios::binary);
				std::fstream fin2(standard, std::ios::in | std::ios::ate |
std::ios::binary);
				if (!fin1.is_open())
					throw std::runtime_error("Ошибка открытия тестируемого
файла\n");
				if (!fin2.is_open())
```

```
                                throw std::runtime_error("Ошибка открытия эталонного
файла\n");
                        if(fin1.tellg() != fin2.tellg())
                                Assert::IsTrue(false);
                        fin1.seekg(0);
                        fin2.seekg(0);
                        bool result = true;
                        char ch1, ch2;
                        while (fin1.get(ch1) && fin2.get(ch2))
                                if (ch1 != ch2)
                                {
                                        result = false;
                                        break;
                                }
                        fin1.close();
                        fin2.close();
                        Assert::IsTrue(result);
                }
                catch (const std::exception& err) {
                        std::cerr << err.what() << std::endl;
                        Assert::IsTrue(false);
                }
        }

        TEST_METHOD(Factorization2)
        {
        // Тело функции Factorization2
        }
        // Остальные функции
    };
}
```

Содержимое всех XML-файлов приведено в приложении А.

№	Название функции	Входная грамматика	Эталонная грамматика
1	Factorization1	inp0.xml	standard0.xml
2	Factorization2	inp1.xml	standard1.xml
3	Recursion1	inp2.xml	standard2.xml
4	Recursion2	inp3.xml	standard3.xml
5	Recursion3	inp4.xml	standard4.xml
6	Recursion4	inp5.xml	standard5.xml
7	Recursion5	inp6.xml	standard6.xml

Результаты выполнения программы

Программа принимает XML-файл, описывающий грамматику. Результатом является XML-файл, описывающий эту же грамматику после удаления левой рекурсии и левой факторизации.

Текст программы, преобразовывающий приведённую КС-грамматику без левой рекурсии к нормальной форме Грейбах

Программа написана на языке C++.

Greibach.h

```
#pragma once
#include <map>
#include <set>
#include <vector>
```

```
#include <string>

class Rhs {
	// true - терминал; false - нетерминал
	std::vector<std::pair<bool, std::string>> rhs;
public:
	Rhs();
	void insert_symbol(bool sym_type, std::string symbol);
	std::vector<std::pair<bool, std::string>> get_rhs();
	friend bool operator<(const Rhs& left, const Rhs& right);
	~Rhs();
};

class Grammar {
	std::set<std::string> nonterminalsymbols;
	std::map<std::string, std::string> terminalsymbols;
	std::map<std::string, std::set<Rhs>> rules;
	std::string startsymbol;
	bool start_from_termianl(std::set<Rhs> chain);
public:
	Grammar();
	void insert_nonterminalsymbol(std::string nonterminalsymbol);
	void insert_terminalsymbol(std::string name, std::string spell);
	void insert_rule(std::string nonterminalsymbol, Rhs rhs);
	void set_rules(std::map<std::string, std::set<Rhs>> rules);
	void set_startsymbol(std::string startsymbol);
	std::set<std::string> get_nonterminalsymbols();
	std::map<std::string, std::string> get_terminalsymbols();
	std::map<std::string, std::set<Rhs>> get_rules();
	std::string get_startsymbol();
	void greibach_normal_form();
	~Grammar();
};

class XmlHandler {
public:
	Grammar read_xml(std::string in_file);
	void write_xml(std::string out_file, Grammar grammar);
};
```

Greibach.cpp

```
#include "Greibach.h"
#include "tinyxml2.h"
#include <algorithm>
#include <iostream>

using namespace std;
using namespace tinyxml2;

Rhs::Rhs() : rhs(vector<pair<bool, string>>()) { }

void Rhs::insert_symbol(bool sym_type, string symbol)
{
	rhs.push_back(pair<bool, string>(sym_type, symbol));
}

vector<pair<bool, string>> Rhs::get_rhs()
{
	return rhs;
}

Rhs::~Rhs()
{
	for (int i = 0; i < rhs.size(); i++)
```

```
                rhs[i].second.clear();
        rhs.clear();
}

bool operator<(const Rhs & left, const Rhs & right)
{
        if (left.rhs.size() < right.rhs.size())
                return true;
        if (left.rhs.size() > right.rhs.size())
                return false;
        for (int i = 0; i < left.rhs.size(); i++)
        {
                if (left.rhs[i].first < right.rhs[i].first)
                        return true;
                if (left.rhs[i].first > right.rhs[i].first)
                        return false;
                if (left.rhs[i].second < right.rhs[i].second)
                        return true;
                if (left.rhs[i].second > right.rhs[i].second)
                        return false;
        }
        return false;
}

bool Grammar::start_from_termianl(set<Rhs> chain)
{
        for (Rhs r : chain)
                if (!r.get_rhs()[0].first)
                        return false;
        return true;
}

Grammar::Grammar() : nonterminalsymbols(set<string>()),
                                        terminalsymbols(map<string, string>()),
                                        rules(map<string, set<Rhs>>()),
                                        startsymbol("") { }

void Grammar::insert_nonterminalsymbol(string nonterminalsymbol)
{
        nonterminalsymbols.insert(nonterminalsymbol);
}

void Grammar::insert_terminalsymbol(string name, string spell)
{
        terminalsymbols.insert(pair<string, string>(name, spell));
}

void Grammar::insert_rule(string nonterminalsymbol, Rhs rhs)
{
        if (rules.find(nonterminalsymbol) == rules.end())
                rules.insert(pair<string, set<Rhs>>(nonterminalsymbol, set<Rhs>()));
        rules[nonterminalsymbol].insert(rhs);
}

void Grammar::set_rules(map<string, set<Rhs>> rules)
{
        this->rules = rules;
}

void Grammar::set_startsymbol(string startsymbol)
{
        this->startsymbol = startsymbol;
}
```

```
set<string> Grammar::get_nonterminalsymbols()
{
	return nonterminalsymbols;
}

map<string, string> Grammar::get_terminalsymbols()
{
	return terminalsymbols;
}

map<string, set<Rhs>> Grammar::get_rules()
{
	return rules;
}

string Grammar::get_startsymbol()
{
	return startsymbol;
}

void Grammar::greibach_normal_form()
{
	Grammar g;

	for (string nonterminal : nonterminalsymbols)
	{
		if (rules.count(nonterminal) == 0)
			continue;
		while (!start_from_termianl(rules[nonterminal]))
		{
			set<Rhs> new_rhs;
			for (Rhs rhs : rules[nonterminal])
				if (rhs.get_rhs()[0].first == true)
					new_rhs.insert(rhs);
				else
				{

					string sym = rhs.get_rhs()[0].second;
					for (Rhs rhs2 : rules[sym])
					{
						Rhs r;
						r = rhs2;
						int i = 0;
						for (pair<bool, string> s : rhs.get_rhs())
							if (i++ != 0)
								r.insert_symbol(s.first, s.second);
						new_rhs.insert(r);
					}

				}
			rules[nonterminal].clear();
			rules[nonterminal] = new_rhs;
		}
	}
	set<string> additional_nonterminals;
	map<string, set<Rhs>> additional_rules;
	for (string nonterminal : nonterminalsymbols)
	{
		if (rules.count(nonterminal) == 0)
			continue;
		set<Rhs> new_rhs;
		for (Rhs rhs : rules[nonterminal])
		{
			Rhs r;
```

```
                    r.insert_symbol(rhs.get_rhs()[0].first, rhs.get_rhs()[0].second);
                    for (int i = 1; i < rhs.get_rhs().size(); i++)
                    {
                           if (!rhs.get_rhs()[i].first)
                                  r.insert_symbol(rhs.get_rhs()[i].first,
rhs.get_rhs()[i].second);
                           else
                           {
                                  string sym;
                                  bool b = false;
                                  for (pair<string, set<Rhs>> pr : additional_rules)
                                  {
                                         for (Rhs rr: pr.second) // 1 элемент
                                                if (rr.get_rhs()[0].second ==
rhs.get_rhs()[i].second)
                                                {
                                                       sym = pr.first;
                                                       b = true;
                                                }
                                         if (b)
                                                break;
                                  }
                                  if (!b)
                                  {
                                         int k = 0;
                                         sym = rhs.get_rhs()[i].second;
                                         while (nonterminalsymbols.find(sym + to_string(++k))
!= nonterminalsymbols.end() ||
                                                additional_nonterminals.find(sym + to_string(k))
!= additional_nonterminals.end());
                                         sym = sym + to_string(k);
                                         additional_nonterminals.insert(sym);
                                         Rhs t;
                                         set<Rhs> st;
                                         t.insert_symbol(true, rhs.get_rhs()[i].second);
                                         st.insert(t);
                                         additional_rules.insert(pair<string, set<Rhs>>(sym,
st));
                                  }
                                  r.insert_symbol(false, sym);
                           }
                    }
                    new_rhs.insert(r);
             }
             rules[nonterminal].clear();
             rules[nonterminal] = new_rhs;
      }
      for (string s : additional_nonterminals)
             nonterminalsymbols.insert(s);
      for (pair<string, set<Rhs>> p : additional_rules)
             rules.insert(p);
}

Grammar::~Grammar()
{
      nonterminalsymbols.clear();
      terminalsymbols.clear();
      rules.clear();
      startsymbol.clear();
}

Grammar XmlHandler::read_xml(string in_file)
{
      Grammar grammar;
```

```
	XMLDocument doc;
	doc.LoadFile(in_file.c_str());
	if (doc.ErrorID())
		throw runtime_error("Не удалось обработать XML файл\n");
	XMLHandle docHandle(&doc);
	XMLHandle root = docHandle.FirstChildElement("grammar");
	XMLElement* terminalsymbols = root.FirstChildElement("terminalsymbols").ToElement();
	XMLElement* term;
	while (terminalsymbols)
	{
		term = terminalsymbols->FirstChildElement("term");
		while (term) {
			grammar.insert_terminalsymbol(term->Attribute("name"), term-
>Attribute("spell"));
			term = term->NextSiblingElement();
		}
		terminalsymbols = terminalsymbols->NextSiblingElement();
	}
	XMLElement* nonterminalsymbols =
root.FirstChildElement("nonterminalsymbols").ToElement();
	XMLElement* nonterm;
	while (nonterminalsymbols)
	{
		nonterm = nonterminalsymbols->FirstChildElement("nonterm");
		while (nonterm) {
			grammar.insert_nonterminalsymbol(nonterm->Attribute("name"));
			nonterm = nonterm->NextSiblingElement();
		}
		nonterminalsymbols = nonterminalsymbols->NextSiblingElement();
	}
	XMLElement* startsymbol = root.FirstChildElement("startsymbol").ToElement();
	if (!startsymbol)
		throw runtime_error("Отсутствует стартовый символ\n");
	grammar.set_startsymbol(startsymbol->Attribute("name"));
	set<string> nonterminals = grammar.get_nonterminalsymbols();
	if (nonterminals.find(startsymbol->Attribute("name")) == nonterminals.end())
		throw runtime_error("Стартовый символ не найден в списке нетерминалов\n");
	map<string, string> terminals = grammar.get_terminalsymbols();
	XMLElement* productions = root.FirstChildElement("productions").ToElement();
	XMLElement* production, *lhs, *rhs;
	while (productions)
	{
		production = productions->FirstChildElement("production");
		while (production)
		{
			lhs = production->FirstChildElement("lhs");
			if (!lhs)
				throw runtime_error("Одна из продукций не имеет левой части\n");
			rhs = production->FirstChildElement("rhs");
			if (!rhs)
				throw runtime_error("Одна из продукций не имеет правой части\n");
			string lhs_name;
			lhs_name = lhs->Attribute("name");
			if (nonterminals.find(lhs_name) == nonterminals.end())
				throw runtime_error("В списке правил обнаружен неизвестный
символ\n");
			XMLElement* symbol = rhs->FirstChildElement("symbol");
			Rhs rhs;
			bool sym_type;
			string sym;
			while (symbol)
			{
				if (string(symbol->Attribute("type")) == "nonterm")
					sym_type = false;
```

```
                        else if (string(symbol->Attribute("type")) == "term")
                                sym_type = true;
                        else
                                throw runtime_error("Неизвестный тип элемента в правой части
продукции\n");
                        sym = symbol->Attribute("name");
                        if (sym_type)
                        {
                                if (!terminals.count(sym))
                                        throw runtime_error("Неизвестный терминал в правой
части продукции\n");
                        }
                        else
                        {
                                if (nonterminals.find(sym) == nonterminals.end())
                                        throw runtime_error("Неизвестный нетерминал в правой
части продукции\n");
                        }
                        rhs.insert_symbol(sym_type, sym);
                        symbol = symbol->NextSiblingElement();
                }
                grammar.insert_rule(lhs_name, rhs);
                production = production->NextSiblingElement();
        }
        productions = productions->NextSiblingElement();
    }
    return grammar;
}

void XmlHandler::write_xml(std::string out_file, Grammar grammar)
{
    static const char* xml =
        "<?xml version=\"1.0\" encoding=\"UTF-8\"?>"
        "<grammar name=\"G1\">"
        "    <terminalsymbols/>"
        "    <nonterminalsymbols/>"
        "    <productions/>"
        "    <startsymbol/>"
        "</grammar>";
    XMLDocument doc;
    doc.Parse(xml);
    XMLElement* terminalsymbols = doc.FirstChildElement("grammar")-
>FirstChildElement("terminalsymbols");
    map<string, string> terminals = grammar.get_terminalsymbols();
    for (pair<string, string> term : terminals)
    {
        XMLElement* termelement = terminalsymbols->InsertNewChildElement("term");
        termelement->SetAttribute("name", term.first.c_str());
        termelement->SetAttribute("spell", term.second.c_str());
    }
    XMLElement* nonterminalsymbols = doc.FirstChildElement("grammar")-
>FirstChildElement("nonterminalsymbols");
    set<string> nonterminals = grammar.get_nonterminalsymbols();
    for (string nonterm : nonterminals)
        nonterminalsymbols->InsertNewChildElement("nonterm")->SetAttribute("name",
nonterm.c_str());
    map<string, set<Rhs>> rules = grammar.get_rules();
    XMLElement* productions = doc.FirstChildElement("grammar")-
>FirstChildElement("productions");
    for (pair<string, set<Rhs>> term : rules)
        for (Rhs rhs : term.second)
        {
            XMLElement* production = productions->InsertNewChildElement("production");
```

```
                    production->InsertNewChildElement("lhs")->SetAttribute("name",
term.first.c_str());
                    XMLElement* rhselement = production->InsertNewChildElement("rhs");
                    vector<pair<bool, string>> chain = rhs.get_rhs();
                    for (pair<bool, string> sym : chain)
                    {
                           XMLElement* symbolelement = rhselement-
>InsertNewChildElement("symbol");
                           if (sym.first)
                                  symbolelement->SetAttribute("type", "term");
                           else
                                  symbolelement->SetAttribute("type", "nonterm");
                           symbolelement->SetAttribute("name", sym.second.c_str());
                    }
             }
      doc.FirstChildElement("grammar")->FirstChildElement("startsymbol")->SetAttribute("name",
grammar.get_startsymbol().c_str());
      doc.SaveFile(out_file.c_str());
}
```

Source.cpp

```
#include "Greibach.h"
#include <iostream>

using namespace std;

int main(int argc, char* argv[])
{
      setlocale(LC_ALL, "Russian");
      XmlHandler xml_handler;
      try {
             Grammar grammar = xml_handler.read_xml(argv[1]);
             grammar.greibach_normal_form();
             xml_handler.write_xml(argv[2], grammar);
      }
      catch (const exception& err) {
             cerr << err.what() << endl;
      }
      return 0;
}
```

Результаты тестирования

Файл `UnitTests.cpp` содержит тесты для проверки грамматик в XML-файлах после их преобразования к нормальной форме Грейбах. Все тесты пройдены успешно.

UnitTests.cpp

```
#include "pch.h"
#include "CppUnitTest.h"
#include "../Greibach/Greibach.h"
#include <iostream>
#include <fstream>

using namespace Microsoft::VisualStudio::CppUnitTestFramework;

namespace UnitTests
{
      TEST_CLASS(UnitTests)
      {
      public:
```

```
		TEST_METHOD(Greibach1)
		{
			setlocale(LC_ALL, "Russian");
			try {
				// Входная грамматика
				const char* inp = "..\\inps\\inp0.xml";
				// Преобразованная грамматика
				const char* outp = "..\\outps\\outp0.xml";
				// Эталонная грамматика
				const char* standard = "..\\standards\\standard0.xml";
				XmlHandler xml_handler;
				Grammar grammar = xml_handler.read_xml(inp);
				grammar.greibach_normal_form();
				xml_handler.write_xml(outp, grammar);
				std::fstream fin1(outp, std::ios::in | std::ios::ate |
std::ios::binary);
				std::fstream fin2(standard, std::ios::in | std::ios::ate |
std::ios::binary);
				if (!fin1.is_open())
					throw std::runtime_error("Ошибка открытия тестируемого
файла\n");
				if (!fin2.is_open())
					throw std::runtime_error("Ошибка открытия эталонного
файла\n");
				if(fin1.tellg() != fin2.tellg())
					Assert::IsTrue(false);
				fin1.seekg(0);
				fin2.seekg(0);
				bool result = true;
				char ch1, ch2;
				while (fin1.get(ch1) && fin2.get(ch2))
					if (ch1 != ch2)
					{
						result = false;
						break;
					}
				fin1.close();
				fin2.close();
				Assert::IsTrue(result);
			}
			catch (const std::exception& err) {
				std::cerr << err.what() << std::endl;
				Assert::IsTrue(false);
			}
		}

		TEST_METHOD(Greibach2)
		{
			setlocale(LC_ALL, "Russian");
			try {
				const char* inp = "..\\inps\\inp1.xml";
				const char* outp = "..\\outps\\outp1.xml";
				const char* standard = "..\\standards\\standard1.xml";
				XmlHandler xml_handler;
				Grammar grammar = xml_handler.read_xml(inp);
				grammar.greibach_normal_form();
				xml_handler.write_xml(outp, grammar);
				std::fstream fin1(outp, std::ios::in | std::ios::ate |
std::ios::binary);
				std::fstream fin2(standard, std::ios::in | std::ios::ate |
std::ios::binary);
				if (!fin1.is_open())
					throw std::runtime_error("Ошибка открытия тестируемого
файла\n");
```

```
                        if (!fin2.is_open())
                                throw std::runtime_error("Ошибка открытия эталонного
файла\n");
                        if (fin1.tellg() != fin2.tellg())
                                Assert::IsTrue(false);
                        fin1.seekg(0);
                        fin2.seekg(0);
                        bool result = true;
                        char ch1, ch2;
                        while (fin1.get(ch1) && fin2.get(ch2))
                                if (ch1 != ch2)
                                {
                                        result = false;
                                        break;
                                }
                        fin1.close();
                        fin2.close();
                        Assert::IsTrue(result);
                }
                catch (const std::exception& err) {
                        std::cerr << err.what() << std::endl;
                        Assert::IsTrue(false);
                }
        }
    };
}
```

Содержимое всех XML-файлов приведено в приложении Б.

№	Название функции	Входная грамматика	Эталонная грамматика
1	Greibach1	inp0.xml	standard0.xml
2	Greibach2	inp1.xml	standard1.xml

Результаты выполнения программы

Программа принимает XML-файл, описывающий приведённую КС-грамматику без левой рекурсии. Результатом является XML-файл, описывающий эту же грамматику после её преобразования к нормальной форме Грейбах.

Выводы

В результате выполнения лабораторной работы было выполнено следующее:

1. написаны функции для чтения грамматики из XML-файла и записи её в XML-файл;
2. составлен преобразователь грамматики из леворекурсивной в нелеворекурсивную;
3. написана функция, проводящая левую факторизация грамматики;
4. написана функция, преобразовывающая приведённую КС-грамматику без левой рекурсии к нормальной форме Грейбах.

Приложение А

inp0.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<grammar name="G0">
	<terminalsymbols>
		<term name="a" spell="a" />
		<term name="b" spell="b" />
	</terminalsymbols>
	<nonterminalsymbols>
		<nonterm name="S" />
		<nonterm name="X" />
	</nonterminalsymbols>
	<productions>
		<production>
			<lhs name="S" />
			<rhs>
				<symbol type="nonterm" name="X" />
				<symbol type="nonterm" name="S" />
			</rhs>
		</production>
		<production>
			<lhs name="S" />
			<rhs>
				<symbol type="term" name="a" />
			</rhs>
		</production>
		<production>
			<lhs name="X" />
			<rhs>
				<symbol type="term" name="b" />
			</rhs>
		</production>
		<production>
			<lhs name="X" />
			<rhs>
				<symbol type="term" name="a" />
			</rhs>
		</production>
		<production>
			<lhs name="X" />
			<rhs>
				<symbol type="term" name="b" />
				<symbol type="nonterm" name="S" />
			</rhs>
		</production>
		<production>
			<lhs name="X" />
			<rhs>
			<symbol type="term" name="b" />
				<symbol type="nonterm" name="S" />
				<symbol type="term" name="a" />
			</rhs>
		</production>
		<production>
			<lhs name="X" />
			<rhs>
			<symbol type="term" name="b" />
				<symbol type="nonterm" name="X" />
				<symbol type="term" name="b" />
			</rhs>
		</production>
	</productions>
	<startsymbol name="S" />
```

```
</grammar>
```

inp1.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<grammar name="G0">
	<terminalsymbols>
		<term name="IDENT" spell="a" />
		<term name="ADD" spell="+" />
		<term name="MUL" spell="*" />
		<term name="LPAREN" spell="(" />
		<term name="RPAREN" spell=")" />
	</terminalsymbols>
	<nonterminalsymbols>
		<nonterm name="E" />
		<nonterm name="T" />
		<nonterm name="F" />
	</nonterminalsymbols>
	<productions>
		<production>
			<lhs name="E" />
			<rhs>
				<symbol type="nonterm" name="E" />
				<symbol type="term" name="ADD" />
				<symbol type="nonterm" name="T" />
			</rhs>
		</production>
		<production>
			<lhs name="E" />
			<rhs>
				<symbol type="nonterm" name="T" />
			</rhs>
		</production>
		<production>
			<lhs name="T" />
			<rhs>
				<symbol type="nonterm" name="T" />
				<symbol type="term" name="MUL" />
				<symbol type="nonterm" name="F" />
			</rhs>
		</production>
		<production>
			<lhs name="T" />
			<rhs>
				<symbol type="nonterm" name="F" />
			</rhs>
		</production>
		<production>
			<lhs name="F" />
			<rhs>
				<symbol type="term" name="IDENT" />
			</rhs>
		</production>
		<production>
			<lhs name="F" />
			<rhs>
				<symbol type="term" name="LPAREN" />
				<symbol type="nonterm" name="E" />
				<symbol type="term" name="RPAREN" />
			</rhs>
		</production>
	</productions>
	<startsymbol name="E" />
</grammar>
```

inp2.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<grammar name="G0">
	<terminalsymbols>
		<term name="a" spell="a" />
		<term name="b" spell="b" />
	</terminalsymbols>
	<nonterminalsymbols>
		<nonterm name="S" />
		<nonterm name="X" />
	</nonterminalsymbols>
	<productions>
		<production>
			<lhs name="S" />
			<rhs>
				<symbol type="nonterm" name="S" />
				<symbol type="nonterm" name="X" />
			</rhs>
		</production>
		<production>
			<lhs name="S" />
			<rhs>
				<symbol type="nonterm" name="S" />
				<symbol type="nonterm" name="S" />
				<symbol type="term" name="b" />
			</rhs>
		</production>
		<production>
			<lhs name="S" />
			<rhs>
				<symbol type="nonterm" name="X" />
				<symbol type="nonterm" name="S" />
			</rhs>
		</production>
		<production>
			<lhs name="S" />
			<rhs>
				<symbol type="term" name="a" />
			</rhs>
		</production>
		<production>
			<lhs name="X" />
			<rhs>
				<symbol type="nonterm" name="S" />
				<symbol type="term" name="a" />
			</rhs>
		</production>
		<production>
			<lhs name="X" />
			<rhs>
				<symbol type="nonterm" name="X" />
				<symbol type="term" name="b" />
			</rhs>
		</production>
	</productions>
	<startsymbol name="S" />
</grammar>
```

inp3.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<grammar name="G0">
	<terminalsymbols>
```

```
        <term name="+" spell="+"/>
        <term name="-" spell="-"/>
        <term name="*" spell="*"/>
        <term name="/" spell="/"/>
        <term name="(" spell="("/>
        <term name=")" spell=")"/>
        <term name="id" spell="id"/>
    </terminalsymbols>
    <nonterminalsymbols>
        <nonterm name="E" />
        <nonterm name="T" />
        <nonterm name="F" />
    </nonterminalsymbols>
    <productions>
        <production>
            <lhs name="E" />
            <rhs>
                <symbol type="nonterm" name="E" />
                <symbol type="term" name="+" />
                <symbol type="nonterm" name="T" />
            </rhs>
        </production>
        <production>
            <lhs name="E" />
            <rhs>
                <symbol type="nonterm" name="E" />
                <symbol type="term" name="-" />
                <symbol type="nonterm" name="T" />
            </rhs>
        </production>
        <production>
            <lhs name="E" />
            <rhs>
                <symbol type="nonterm" name="T" />
            </rhs>
        </production>
        <production>
            <lhs name="T" />
            <rhs>
                <symbol type="nonterm" name="T" />
                <symbol type="term" name="*" />
                <symbol type="nonterm" name="F" />
            </rhs>
        </production>
        <production>
            <lhs name="T" />
            <rhs>
                <symbol type="nonterm" name="T" />
                <symbol type="term" name="/" />
                <symbol type="nonterm" name="F" />
            </rhs>
        </production>
        <production>
            <lhs name="T" />
            <rhs>
                <symbol type="nonterm" name="F" />
            </rhs>
        </production>
        <production>
            <lhs name="F" />
            <rhs>
                <symbol type="term" name="id" />
            </rhs>
        </production>
```

```
		<production>
			<lhs name="F" />
			<rhs>
				<symbol type="term" name="(" />
				<symbol type="nonterm" name="E" />
				<symbol type="term" name=")" />
			</rhs>
		</production>
	</productions>
	<startsymbol name="E" />
</grammar>
```

inp4.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<grammar name="G0">
	<terminalsymbols>
		<term name="a" spell="a" />
		<term name="b" spell="b" />
		<term name="c" spell="c" />
		<term name="d" spell="d" />
		<term name="EPSILON" spell="" />
	</terminalsymbols>
	<nonterminalsymbols>
		<nonterm name="A" />
		<nonterm name="S" />
	</nonterminalsymbols>
	<productions>
		<production>
			<lhs name="S" />
			<rhs>
				<symbol type="nonterm" name="A" />
				<symbol type="term" name="a" />
			</rhs>
		</production>
		<production>
			<lhs name="S" />
			<rhs>
				<symbol type="term" name="b" />
			</rhs>
		</production>
		<production>
			<lhs name="A" />
			<rhs>
				<symbol type="nonterm" name="A" />
				<symbol type="term" name="c" />
			</rhs>
		</production>
		<production>
			<lhs name="A" />
			<rhs>
				<symbol type="nonterm" name="S" />
				<symbol type="term" name="d" />
			</rhs>
		</production>
		<production>
			<lhs name="A" />
			<rhs>
				<symbol type="term" name="EPSILON" />
			</rhs>
		</production>
	</productions>
	<startsymbol name="S" />
</grammar>
```

inp5.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<grammar name="G0">
	<terminalsymbols>
		<term name="a" spell="a" />
		<term name="b" spell="b" />
		<term name="i" spell="i" />
		<term name="e" spell="e" />
		<term name="t" spell="t" />
	</terminalsymbols>
	<nonterminalsymbols>
		<nonterm name="E" />
		<nonterm name="S" />
	</nonterminalsymbols>
	<productions>
		<production>
			<lhs name="S" />
			<rhs>
				<symbol type="term" name="i" />
				<symbol type="nonterm" name="E" />
				<symbol type="term" name="t" />
				<symbol type="nonterm" name="S" />
			</rhs>
		</production>
		<production>
			<lhs name="S" />
			<rhs>
				<symbol type="term" name="a" />
			</rhs>
		</production>
		<production>
			<lhs name="S" />
			<rhs>
				<symbol type="term" name="i" />
				<symbol type="nonterm" name="E" />
				<symbol type="term" name="t" />
				<symbol type="nonterm" name="S" />
				<symbol type="term" name="e" />
				<symbol type="nonterm" name="S" />
			</rhs>
		</production>
		<production>
			<lhs name="E" />
			<rhs>
				<symbol type="term" name="b" />
			</rhs>
		</production>
	</productions>
	<startsymbol name="S" />
</grammar>
```

inp6.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<grammar name="G0">
	<terminalsymbols>
		<term name="a" spell="a" />
		<term name="b" spell="b" />
	</terminalsymbols>
	<nonterminalsymbols>
		<nonterm name="A" />
		<nonterm name="B" />
		<nonterm name="S" />
```

```
	</nonterminalsymbols>
	<productions>
		<production>
			<lhs name="S" />
			<rhs>
				<symbol type="nonterm" name="B" />
				<symbol type="term" name="a" />
			</rhs>
		</production>
		<production>
			<lhs name="S" />
			<rhs>
				<symbol type="nonterm" name="A" />
				<symbol type="term" name="b" />
			</rhs>
		</production>
		<production>
			<lhs name="A" />
			<rhs>
				<symbol type="nonterm" name="A" />
				<symbol type="nonterm" name="A" />
				<symbol type="term" name="b" />
			</rhs>
		</production>
		<production>
			<lhs name="A" />
			<rhs>
				<symbol type="term" name="a" />
			</rhs>
		</production>
		<production>
			<lhs name="A" />
			<rhs>
				<symbol type="nonterm" name="S" />
				<symbol type="term" name="a" />
			</rhs>
		</production>
		<production>
			<lhs name="B" />
			<rhs>
				<symbol type="nonterm" name="B" />
				<symbol type="nonterm" name="B" />
				<symbol type="term" name="a" />
			</rhs>
		</production>
		<production>
			<lhs name="B" />
			<rhs>
				<symbol type="term" name="b" />
			</rhs>
		</production>
		<production>
			<lhs name="B" />
			<rhs>
				<symbol type="nonterm" name="S" />
				<symbol type="term" name="b" />
			</rhs>
		</production>
	</productions>
	<startsymbol name="S" />
</grammar>
```

standard0.xml

```
<?xml version="1.0" encoding="UTF-8"?>
```

```
<grammar name="G1">
    <terminalsymbols>
        <term name="EPSOLON" spell=""/>
        <term name="a" spell="a"/>
        <term name="b" spell="b"/>
    </terminalsymbols>
    <nonterminalsymbols>
        <nonterm name="S"/>
        <nonterm name="X"/>
        <nonterm name="X1"/>
        <nonterm name="X2"/>
    </nonterminalsymbols>
    <productions>
        <production>
            <lhs name="S"/>
            <rhs>
                <symbol type="term" name="a"/>
            </rhs>
        </production>
        <production>
            <lhs name="S"/>
            <rhs>
                <symbol type="nonterm" name="X"/>
                <symbol type="nonterm" name="S"/>
            </rhs>
        </production>
        <production>
            <lhs name="X"/>
            <rhs>
                <symbol type="term" name="a"/>
            </rhs>
        </production>
        <production>
            <lhs name="X"/>
            <rhs>
                <symbol type="term" name="b"/>
                <symbol type="nonterm" name="X2"/>
            </rhs>
        </production>
        <production>
            <lhs name="X1"/>
            <rhs>
                <symbol type="term" name="EPSOLON"/>
            </rhs>
        </production>
        <production>
            <lhs name="X1"/>
            <rhs>
                <symbol type="term" name="a"/>
            </rhs>
        </production>
        <production>
            <lhs name="X2"/>
            <rhs>
                <symbol type="term" name="EPSOLON"/>
            </rhs>
        </production>
        <production>
            <lhs name="X2"/>
            <rhs>
                <symbol type="nonterm" name="S"/>
                <symbol type="nonterm" name="X1"/>
            </rhs>
        </production>
```

```
        <production>
            <lhs name="X2"/>
            <rhs>
                <symbol type="nonterm" name="X"/>
                <symbol type="term" name="b"/>
            </rhs>
        </production>
    </productions>
    <startsymbol name="S"/>
</grammar>
```

standard1.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<grammar name="G1">
    <terminalsymbols>
        <term name="ADD" spell="+"/>
        <term name="EPSOLON" spell=""/>
        <term name="IDENT" spell="a"/>
        <term name="LPAREN" spell="("/>
        <term name="MUL" spell="*"/>
        <term name="RPAREN" spell=")"/>
    </terminalsymbols>
    <nonterminalsymbols>
        <nonterm name="E"/>
        <nonterm name="E1"/>
        <nonterm name="E11"/>
        <nonterm name="E2"/>
        <nonterm name="F"/>
        <nonterm name="T"/>
        <nonterm name="T1"/>
        <nonterm name="T11"/>
        <nonterm name="T2"/>
        <nonterm name="T3"/>
    </nonterminalsymbols>
    <productions>
        <production>
            <lhs name="E"/>
            <rhs>
                <symbol type="nonterm" name="T"/>
                <symbol type="nonterm" name="E2"/>
            </rhs>
        </production>
        <production>
            <lhs name="E1"/>
            <rhs>
                <symbol type="term" name="ADD"/>
                <symbol type="nonterm" name="T"/>
                <symbol type="nonterm" name="E11"/>
            </rhs>
        </production>
        <production>
            <lhs name="E11"/>
            <rhs>
                <symbol type="nonterm" name="E1"/>
            </rhs>
        </production>
        <production>
            <lhs name="E11"/>
            <rhs>
                <symbol type="term" name="EPSOLON"/>
            </rhs>
        </production>
        <production>
            <lhs name="E2"/>
```

```
            <rhs>
                <symbol type="nonterm" name="E1"/>
            </rhs>
        </production>
        <production>
            <lhs name="E2"/>
            <rhs>
                <symbol type="term" name="EPSOLON"/>
            </rhs>
        </production>
        <production>
            <lhs name="F"/>
            <rhs>
                <symbol type="term" name="IDENT"/>
            </rhs>
        </production>
        <production>
            <lhs name="F"/>
            <rhs>
                <symbol type="term" name="LPAREN"/>
                <symbol type="nonterm" name="E"/>
                <symbol type="term" name="RPAREN"/>
            </rhs>
        </production>
        <production>
            <lhs name="T"/>
            <rhs>
                <symbol type="term" name="IDENT"/>
                <symbol type="nonterm" name="T3"/>
            </rhs>
        </production>
        <production>
            <lhs name="T"/>
            <rhs>
                <symbol type="term" name="LPAREN"/>
                <symbol type="nonterm" name="E"/>
                <symbol type="term" name="RPAREN"/>
                <symbol type="nonterm" name="T2"/>
            </rhs>
        </production>
        <production>
            <lhs name="T1"/>
            <rhs>
                <symbol type="term" name="MUL"/>
                <symbol type="nonterm" name="F"/>
                <symbol type="nonterm" name="T11"/>
            </rhs>
        </production>
        <production>
            <lhs name="T11"/>
            <rhs>
                <symbol type="nonterm" name="T1"/>
            </rhs>
        </production>
        <production>
            <lhs name="T11"/>
            <rhs>
                <symbol type="term" name="EPSOLON"/>
            </rhs>
        </production>
        <production>
            <lhs name="T2"/>
            <rhs>
                <symbol type="nonterm" name="T1"/>
```

```
            </rhs>
        </production>
        <production>
            <lhs name="T2"/>
            <rhs>
                <symbol type="term" name="EPSOLON"/>
            </rhs>
        </production>
        <production>
            <lhs name="T3"/>
            <rhs>
                <symbol type="nonterm" name="T1"/>
            </rhs>
        </production>
        <production>
            <lhs name="T3"/>
            <rhs>
                <symbol type="term" name="EPSOLON"/>
            </rhs>
        </production>
    </productions>
    <startsymbol name="E"/>
</grammar>
```

standard2.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<grammar name="G1">
    <terminalsymbols>
        <term name="EPSOLON" spell=""/>
        <term name="a" spell="a"/>
        <term name="b" spell="b"/>
    </terminalsymbols>
    <nonterminalsymbols>
        <nonterm name="S"/>
        <nonterm name="S1"/>
        <nonterm name="S11"/>
        <nonterm name="S12"/>
        <nonterm name="S2"/>
        <nonterm name="S3"/>
        <nonterm name="X"/>
        <nonterm name="X1"/>
        <nonterm name="X11"/>
        <nonterm name="X12"/>
        <nonterm name="X13"/>
        <nonterm name="X14"/>
        <nonterm name="X2"/>
        <nonterm name="X3"/>
        <nonterm name="X4"/>
    </nonterminalsymbols>
    <productions>
        <production>
            <lhs name="S"/>
            <rhs>
                <symbol type="term" name="a"/>
                <symbol type="nonterm" name="S3"/>
            </rhs>
        </production>
        <production>
            <lhs name="S"/>
            <rhs>
                <symbol type="nonterm" name="X"/>
                <symbol type="nonterm" name="S"/>
                <symbol type="nonterm" name="S2"/>
            </rhs>
```

```
        </production>
        <production>
            <lhs name="S1"/>
            <rhs>
                <symbol type="nonterm" name="X"/>
                <symbol type="nonterm" name="S12"/>
            </rhs>
        </production>
        <production>
            <lhs name="S1"/>
            <rhs>
                <symbol type="nonterm" name="S"/>
                <symbol type="term" name="b"/>
                <symbol type="nonterm" name="S11"/>
            </rhs>
        </production>
        <production>
            <lhs name="S11"/>
            <rhs>
                <symbol type="nonterm" name="S1"/>
            </rhs>
        </production>
        <production>
            <lhs name="S11"/>
            <rhs>
                <symbol type="term" name="EPSOLON"/>
            </rhs>
        </production>
        <production>
            <lhs name="S12"/>
            <rhs>
                <symbol type="nonterm" name="S1"/>
            </rhs>
        </production>
        <production>
            <lhs name="S12"/>
            <rhs>
                <symbol type="term" name="EPSOLON"/>
            </rhs>
        </production>
        <production>
            <lhs name="S2"/>
            <rhs>
                <symbol type="nonterm" name="S1"/>
            </rhs>
        </production>
        <production>
            <lhs name="S2"/>
            <rhs>
                <symbol type="term" name="EPSOLON"/>
            </rhs>
        </production>
        <production>
            <lhs name="S3"/>
            <rhs>
                <symbol type="nonterm" name="S1"/>
            </rhs>
        </production>
        <production>
            <lhs name="S3"/>
            <rhs>
                <symbol type="term" name="EPSOLON"/>
            </rhs>
        </production>
```

```
        <production>
            <lhs name="X"/>
            <rhs>
                <symbol type="term" name="a"/>
                <symbol type="nonterm" name="X4"/>
            </rhs>
        </production>
        <production>
            <lhs name="X1"/>
            <rhs>
                <symbol type="nonterm" name="S"/>
                <symbol type="nonterm" name="X14"/>
            </rhs>
        </production>
        <production>
            <lhs name="X1"/>
            <rhs>
                <symbol type="term" name="b"/>
                <symbol type="nonterm" name="X13"/>
            </rhs>
        </production>
        <production>
            <lhs name="X11"/>
            <rhs>
                <symbol type="nonterm" name="X1"/>
            </rhs>
        </production>
        <production>
            <lhs name="X11"/>
            <rhs>
                <symbol type="term" name="EPSOLON"/>
            </rhs>
        </production>
        <production>
            <lhs name="X12"/>
            <rhs>
                <symbol type="nonterm" name="X1"/>
            </rhs>
        </production>
        <production>
            <lhs name="X12"/>
            <rhs>
                <symbol type="term" name="EPSOLON"/>
            </rhs>
        </production>
        <production>
            <lhs name="X13"/>
            <rhs>
                <symbol type="nonterm" name="X1"/>
            </rhs>
        </production>
        <production>
            <lhs name="X13"/>
            <rhs>
                <symbol type="term" name="EPSOLON"/>
            </rhs>
        </production>
        <production>
            <lhs name="X14"/>
            <rhs>
                <symbol type="term" name="a"/>
                <symbol type="nonterm" name="X12"/>
            </rhs>
        </production>
```

```
        <production>
            <lhs name="X14"/>
            <rhs>
                <symbol type="nonterm" name="S1"/>
                <symbol type="term" name="a"/>
                <symbol type="nonterm" name="X11"/>
            </rhs>
        </production>
        <production>
            <lhs name="X2"/>
            <rhs>
                <symbol type="nonterm" name="X1"/>
            </rhs>
        </production>
        <production>
            <lhs name="X2"/>
            <rhs>
                <symbol type="term" name="EPSOLON"/>
            </rhs>
        </production>
        <production>
            <lhs name="X3"/>
            <rhs>
                <symbol type="nonterm" name="X1"/>
            </rhs>
        </production>
        <production>
            <lhs name="X3"/>
            <rhs>
                <symbol type="term" name="EPSOLON"/>
            </rhs>
        </production>
        <production>
            <lhs name="X4"/>
            <rhs>
                <symbol type="term" name="a"/>
                <symbol type="nonterm" name="X3"/>
            </rhs>
        </production>
        <production>
            <lhs name="X4"/>
            <rhs>
                <symbol type="nonterm" name="S1"/>
                <symbol type="term" name="a"/>
                <symbol type="nonterm" name="X2"/>
            </rhs>
        </production>
    </productions>
    <startsymbol name="S"/>
</grammar>
```

standard3.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<grammar name="G1">
    <terminalsymbols>
        <term name="(" spell="("/>
        <term name=")" spell=")"/>
        <term name="*" spell="*"/>
        <term name="+" spell="+"/>
        <term name="-" spell="-"/>
        <term name="/" spell="/"/>
        <term name="EPSOLON" spell=""/>
        <term name="id" spell="id"/>
    </terminalsymbols>
```

```
    <nonterminalsymbols>
        <nonterm name="E"/>
        <nonterm name="E1"/>
        <nonterm name="E11"/>
        <nonterm name="E12"/>
        <nonterm name="E2"/>
        <nonterm name="F"/>
        <nonterm name="T"/>
        <nonterm name="T1"/>
        <nonterm name="T11"/>
        <nonterm name="T12"/>
        <nonterm name="T2"/>
        <nonterm name="T3"/>
    </nonterminalsymbols>
    <productions>
        <production>
            <lhs name="E"/>
            <rhs>
                <symbol type="nonterm" name="T"/>
                <symbol type="nonterm" name="E2"/>
            </rhs>
        </production>
        <production>
            <lhs name="E1"/>
            <rhs>
                <symbol type="term" name="+"/>
                <symbol type="nonterm" name="T"/>
                <symbol type="nonterm" name="E11"/>
            </rhs>
        </production>
        <production>
            <lhs name="E1"/>
            <rhs>
                <symbol type="term" name="-"/>
                <symbol type="nonterm" name="T"/>
                <symbol type="nonterm" name="E12"/>
            </rhs>
        </production>
        <production>
            <lhs name="E11"/>
            <rhs>
                <symbol type="nonterm" name="E1"/>
            </rhs>
        </production>
        <production>
            <lhs name="E11"/>
            <rhs>
                <symbol type="term" name="EPSOLON"/>
            </rhs>
        </production>
        <production>
            <lhs name="E12"/>
            <rhs>
                <symbol type="nonterm" name="E1"/>
            </rhs>
        </production>
        <production>
            <lhs name="E12"/>
            <rhs>
                <symbol type="term" name="EPSOLON"/>
            </rhs>
        </production>
        <production>
            <lhs name="E2"/>
```

```
            <rhs>
                <symbol type="nonterm" name="E1"/>
            </rhs>
        </production>
        <production>
            <lhs name="E2"/>
            <rhs>
                <symbol type="term" name="EPSOLON"/>
            </rhs>
        </production>
        <production>
            <lhs name="F"/>
            <rhs>
                <symbol type="term" name="id"/>
            </rhs>
        </production>
        <production>
            <lhs name="F"/>
            <rhs>
                <symbol type="term" name="("/>
                <symbol type="nonterm" name="E"/>
                <symbol type="term" name=")"/>
            </rhs>
        </production>
        <production>
            <lhs name="T"/>
            <rhs>
                <symbol type="term" name="id"/>
                <symbol type="nonterm" name="T3"/>
            </rhs>
        </production>
        <production>
            <lhs name="T"/>
            <rhs>
                <symbol type="term" name="("/>
                <symbol type="nonterm" name="E"/>
                <symbol type="term" name=")"/>
                <symbol type="nonterm" name="T2"/>
            </rhs>
        </production>
        <production>
            <lhs name="T1"/>
            <rhs>
                <symbol type="term" name="*"/>
                <symbol type="nonterm" name="F"/>
                <symbol type="nonterm" name="T11"/>
            </rhs>
        </production>
        <production>
            <lhs name="T1"/>
            <rhs>
                <symbol type="term" name="/"/>
                <symbol type="nonterm" name="F"/>
                <symbol type="nonterm" name="T12"/>
            </rhs>
        </production>
        <production>
            <lhs name="T11"/>
            <rhs>
                <symbol type="nonterm" name="T1"/>
            </rhs>
        </production>
        <production>
            <lhs name="T11"/>
```

```
            <rhs>
                <symbol type="term" name="EPSOLON"/>
            </rhs>
        </production>
        <production>
            <lhs name="T12"/>
            <rhs>
                <symbol type="nonterm" name="T1"/>
            </rhs>
        </production>
        <production>
            <lhs name="T12"/>
            <rhs>
                <symbol type="term" name="EPSOLON"/>
            </rhs>
        </production>
        <production>
            <lhs name="T2"/>
            <rhs>
                <symbol type="nonterm" name="T1"/>
            </rhs>
        </production>
        <production>
            <lhs name="T2"/>
            <rhs>
                <symbol type="term" name="EPSOLON"/>
            </rhs>
        </production>
        <production>
            <lhs name="T3"/>
            <rhs>
                <symbol type="nonterm" name="T1"/>
            </rhs>
        </production>
        <production>
            <lhs name="T3"/>
            <rhs>
                <symbol type="term" name="EPSOLON"/>
            </rhs>
        </production>
    </productions>
    <startsymbol name="E"/>
</grammar>
```

standard4.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<grammar name="G1">
    <terminalsymbols>
        <term name="EPSILON" spell=""/>
        <term name="EPSOLON" spell=""/>
        <term name="a" spell="a"/>
        <term name="b" spell="b"/>
        <term name="c" spell="c"/>
        <term name="d" spell="d"/>
    </terminalsymbols>
    <nonterminalsymbols>
        <nonterm name="A"/>
        <nonterm name="A1"/>
        <nonterm name="A11"/>
        <nonterm name="A2"/>
        <nonterm name="A3"/>
        <nonterm name="S"/>
        <nonterm name="S1"/>
        <nonterm name="S11"/>
```

```
        <nonterm name="S12"/>
        <nonterm name="S13"/>
        <nonterm name="S2"/>
        <nonterm name="S3"/>
        <nonterm name="S4"/>
        <nonterm name="S5"/>
    </nonterminalsymbols>
    <productions>
        <production>
            <lhs name="A"/>
            <rhs>
                <symbol type="term" name="EPSILON"/>
                <symbol type="nonterm" name="A3"/>
            </rhs>
        </production>
        <production>
            <lhs name="A"/>
            <rhs>
                <symbol type="nonterm" name="S"/>
                <symbol type="term" name="d"/>
                <symbol type="nonterm" name="A2"/>
            </rhs>
        </production>
        <production>
            <lhs name="A1"/>
            <rhs>
                <symbol type="term" name="c"/>
                <symbol type="nonterm" name="A11"/>
            </rhs>
        </production>
        <production>
            <lhs name="A11"/>
            <rhs>
                <symbol type="nonterm" name="A1"/>
            </rhs>
        </production>
        <production>
            <lhs name="A11"/>
            <rhs>
                <symbol type="term" name="EPSOLON"/>
            </rhs>
        </production>
        <production>
            <lhs name="A2"/>
            <rhs>
                <symbol type="nonterm" name="A1"/>
            </rhs>
        </production>
        <production>
            <lhs name="A2"/>
            <rhs>
                <symbol type="term" name="EPSOLON"/>
            </rhs>
        </production>
        <production>
            <lhs name="A3"/>
            <rhs>
                <symbol type="nonterm" name="A1"/>
            </rhs>
        </production>
        <production>
            <lhs name="A3"/>
            <rhs>
                <symbol type="term" name="EPSOLON"/>
```

```
            </rhs>
        </production>
        <production>
            <lhs name="S"/>
            <rhs>
                <symbol type="term" name="EPSILON"/>
                <symbol type="nonterm" name="S5"/>
            </rhs>
        </production>
        <production>
            <lhs name="S"/>
            <rhs>
                <symbol type="term" name="b"/>
                <symbol type="nonterm" name="S4"/>
            </rhs>
        </production>
        <production>
            <lhs name="S1"/>
            <rhs>
                <symbol type="term" name="d"/>
                <symbol type="nonterm" name="S13"/>
            </rhs>
        </production>
        <production>
            <lhs name="S11"/>
            <rhs>
                <symbol type="nonterm" name="S1"/>
            </rhs>
        </production>
        <production>
            <lhs name="S11"/>
            <rhs>
                <symbol type="term" name="EPSOLON"/>
            </rhs>
        </production>
        <production>
            <lhs name="S12"/>
            <rhs>
                <symbol type="nonterm" name="S1"/>
            </rhs>
        </production>
        <production>
            <lhs name="S12"/>
            <rhs>
                <symbol type="term" name="EPSOLON"/>
            </rhs>
        </production>
        <production>
            <lhs name="S13"/>
            <rhs>
                <symbol type="term" name="a"/>
                <symbol type="nonterm" name="S12"/>
            </rhs>
        </production>
        <production>
            <lhs name="S13"/>
            <rhs>
                <symbol type="nonterm" name="A1"/>
                <symbol type="term" name="a"/>
                <symbol type="nonterm" name="S11"/>
            </rhs>
        </production>
        <production>
            <lhs name="S2"/>
```

```
            <rhs>
                <symbol type="nonterm" name="S1"/>
            </rhs>
        </production>
        <production>
            <lhs name="S2"/>
            <rhs>
                <symbol type="term" name="EPSOLON"/>
            </rhs>
        </production>
        <production>
            <lhs name="S3"/>
            <rhs>
                <symbol type="nonterm" name="S1"/>
            </rhs>
        </production>
        <production>
            <lhs name="S3"/>
            <rhs>
                <symbol type="term" name="EPSOLON"/>
            </rhs>
        </production>
        <production>
            <lhs name="S4"/>
            <rhs>
                <symbol type="nonterm" name="S1"/>
            </rhs>
        </production>
        <production>
            <lhs name="S4"/>
            <rhs>
                <symbol type="term" name="EPSOLON"/>
            </rhs>
        </production>
        <production>
            <lhs name="S5"/>
            <rhs>
                <symbol type="term" name="a"/>
                <symbol type="nonterm" name="S3"/>
            </rhs>
        </production>
        <production>
            <lhs name="S5"/>
            <rhs>
                <symbol type="nonterm" name="A1"/>
                <symbol type="term" name="a"/>
                <symbol type="nonterm" name="S2"/>
            </rhs>
        </production>
    </productions>
    <startsymbol name="S"/>
</grammar>
```

standard5.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<grammar name="G1">
    <terminalsymbols>
        <term name="EPSOLON" spell=""/>
        <term name="a" spell="a"/>
        <term name="b" spell="b"/>
        <term name="e" spell="e"/>
        <term name="i" spell="i"/>
        <term name="t" spell="t"/>
    </terminalsymbols>
```

```
    <nonterminalsymbols>
        <nonterm name="E"/>
        <nonterm name="S"/>
        <nonterm name="S1"/>
    </nonterminalsymbols>
    <productions>
        <production>
            <lhs name="E"/>
            <rhs>
                <symbol type="term" name="b"/>
            </rhs>
        </production>
        <production>
            <lhs name="S"/>
            <rhs>
                <symbol type="term" name="a"/>
            </rhs>
        </production>
        <production>
            <lhs name="S"/>
            <rhs>
                <symbol type="term" name="i"/>
                <symbol type="nonterm" name="E"/>
                <symbol type="term" name="t"/>
                <symbol type="nonterm" name="S"/>
                <symbol type="nonterm" name="S1"/>
            </rhs>
        </production>
        <production>
            <lhs name="S1"/>
            <rhs>
                <symbol type="term" name="EPSOLON"/>
            </rhs>
        </production>
        <production>
            <lhs name="S1"/>
            <rhs>
                <symbol type="term" name="e"/>
                <symbol type="nonterm" name="S"/>
            </rhs>
        </production>
    </productions>
    <startsymbol name="S"/>
</grammar>
```

standard6.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<grammar name="G1">
    <terminalsymbols>
        <term name="EPSOLON" spell=""/>
        <term name="a" spell="a"/>
        <term name="b" spell="b"/>
    </terminalsymbols>
    <nonterminalsymbols>
        <nonterm name="A"/>
        <nonterm name="A1"/>
        <nonterm name="A11"/>
        <nonterm name="A2"/>
        <nonterm name="A3"/>
        <nonterm name="B"/>
        <nonterm name="B1"/>
        <nonterm name="B11"/>
        <nonterm name="B2"/>
        <nonterm name="B3"/>
```

```
        <nonterm name="S"/>
        <nonterm name="S1"/>
        <nonterm name="S11"/>
        <nonterm name="S12"/>
        <nonterm name="S13"/>
        <nonterm name="S14"/>
        <nonterm name="S15"/>
        <nonterm name="S16"/>
        <nonterm name="S2"/>
        <nonterm name="S3"/>
        <nonterm name="S4"/>
        <nonterm name="S5"/>
        <nonterm name="S6"/>
        <nonterm name="S7"/>
    </nonterminalsymbols>
    <productions>
        <production>
            <lhs name="A"/>
            <rhs>
                <symbol type="term" name="a"/>
                <symbol type="nonterm" name="A3"/>
            </rhs>
        </production>
        <production>
            <lhs name="A"/>
            <rhs>
                <symbol type="nonterm" name="S"/>
                <symbol type="term" name="a"/>
                <symbol type="nonterm" name="A2"/>
            </rhs>
        </production>
        <production>
            <lhs name="A1"/>
            <rhs>
                <symbol type="nonterm" name="A"/>
                <symbol type="term" name="b"/>
                <symbol type="nonterm" name="A11"/>
            </rhs>
        </production>
        <production>
            <lhs name="A11"/>
            <rhs>
                <symbol type="nonterm" name="A1"/>
            </rhs>
        </production>
        <production>
            <lhs name="A11"/>
            <rhs>
                <symbol type="term" name="EPSOLON"/>
            </rhs>
        </production>
        <production>
            <lhs name="A2"/>
            <rhs>
                <symbol type="nonterm" name="A1"/>
            </rhs>
        </production>
        <production>
            <lhs name="A2"/>
            <rhs>
                <symbol type="term" name="EPSOLON"/>
            </rhs>
        </production>
        <production>
```

```
            <lhs name="A3"/>
            <rhs>
                <symbol type="nonterm" name="A1"/>
            </rhs>
        </production>
        <production>
            <lhs name="A3"/>
            <rhs>
                <symbol type="term" name="EPSOLON"/>
            </rhs>
        </production>
        <production>
            <lhs name="B"/>
            <rhs>
                <symbol type="term" name="b"/>
                <symbol type="nonterm" name="B3"/>
            </rhs>
        </production>
        <production>
            <lhs name="B"/>
            <rhs>
                <symbol type="nonterm" name="S"/>
                <symbol type="term" name="b"/>
                <symbol type="nonterm" name="B2"/>
            </rhs>
        </production>
        <production>
            <lhs name="B1"/>
            <rhs>
                <symbol type="nonterm" name="B"/>
                <symbol type="term" name="a"/>
                <symbol type="nonterm" name="B11"/>
            </rhs>
        </production>
        <production>
            <lhs name="B11"/>
            <rhs>
                <symbol type="nonterm" name="B1"/>
            </rhs>
        </production>
        <production>
            <lhs name="B11"/>
            <rhs>
                <symbol type="term" name="EPSOLON"/>
            </rhs>
        </production>
        <production>
            <lhs name="B2"/>
            <rhs>
                <symbol type="nonterm" name="B1"/>
            </rhs>
        </production>
        <production>
            <lhs name="B2"/>
            <rhs>
                <symbol type="term" name="EPSOLON"/>
            </rhs>
        </production>
        <production>
            <lhs name="B3"/>
            <rhs>
                <symbol type="nonterm" name="B1"/>
            </rhs>
        </production>
```

```
        <production>
            <lhs name="B3"/>
            <rhs>
                <symbol type="term" name="EPSOLON"/>
            </rhs>
        </production>
        <production>
            <lhs name="S"/>
            <rhs>
                <symbol type="term" name="a"/>
                <symbol type="nonterm" name="S6"/>
            </rhs>
        </production>
        <production>
            <lhs name="S"/>
            <rhs>
                <symbol type="term" name="b"/>
                <symbol type="nonterm" name="S7"/>
            </rhs>
        </production>
        <production>
            <lhs name="S1"/>
            <rhs>
                <symbol type="term" name="a"/>
                <symbol type="nonterm" name="S15"/>
            </rhs>
        </production>
        <production>
            <lhs name="S1"/>
            <rhs>
                <symbol type="term" name="b"/>
                <symbol type="nonterm" name="S16"/>
            </rhs>
        </production>
        <production>
            <lhs name="S11"/>
            <rhs>
                <symbol type="nonterm" name="S1"/>
            </rhs>
        </production>
        <production>
            <lhs name="S11"/>
            <rhs>
                <symbol type="term" name="EPSOLON"/>
            </rhs>
        </production>
        <production>
            <lhs name="S12"/>
            <rhs>
                <symbol type="nonterm" name="S1"/>
            </rhs>
        </production>
        <production>
            <lhs name="S12"/>
            <rhs>
                <symbol type="term" name="EPSOLON"/>
            </rhs>
        </production>
        <production>
            <lhs name="S13"/>
            <rhs>
                <symbol type="nonterm" name="S1"/>
            </rhs>
        </production>
```

```
        <production>
            <lhs name="S13"/>
            <rhs>
                <symbol type="term" name="EPSOLON"/>
            </rhs>
        </production>
        <production>
            <lhs name="S14"/>
            <rhs>
                <symbol type="nonterm" name="S1"/>
            </rhs>
        </production>
        <production>
            <lhs name="S14"/>
            <rhs>
                <symbol type="term" name="EPSOLON"/>
            </rhs>
        </production>
        <production>
            <lhs name="S15"/>
            <rhs>
                <symbol type="term" name="b"/>
                <symbol type="nonterm" name="S13"/>
            </rhs>
        </production>
        <production>
            <lhs name="S15"/>
            <rhs>
                <symbol type="nonterm" name="A1"/>
                <symbol type="term" name="b"/>
                <symbol type="nonterm" name="S11"/>
            </rhs>
        </production>
        <production>
            <lhs name="S16"/>
            <rhs>
                <symbol type="term" name="a"/>
                <symbol type="nonterm" name="S14"/>
            </rhs>
        </production>
        <production>
            <lhs name="S16"/>
            <rhs>
                <symbol type="nonterm" name="B1"/>
                <symbol type="term" name="a"/>
                <symbol type="nonterm" name="S12"/>
            </rhs>
        </production>
        <production>
            <lhs name="S2"/>
            <rhs>
                <symbol type="nonterm" name="S1"/>
            </rhs>
        </production>
        <production>
            <lhs name="S2"/>
            <rhs>
                <symbol type="term" name="EPSOLON"/>
            </rhs>
        </production>
        <production>
            <lhs name="S3"/>
            <rhs>
                <symbol type="nonterm" name="S1"/>
```

```
            </rhs>
        </production>
        <production>
            <lhs name="S3"/>
            <rhs>
                <symbol type="term" name="EPSOLON"/>
            </rhs>
        </production>
        <production>
            <lhs name="S4"/>
            <rhs>
                <symbol type="nonterm" name="S1"/>
            </rhs>
        </production>
        <production>
            <lhs name="S4"/>
            <rhs>
                <symbol type="term" name="EPSOLON"/>
            </rhs>
        </production>
        <production>
            <lhs name="S5"/>
            <rhs>
                <symbol type="nonterm" name="S1"/>
            </rhs>
        </production>
        <production>
            <lhs name="S5"/>
            <rhs>
                <symbol type="term" name="EPSOLON"/>
            </rhs>
        </production>
        <production>
            <lhs name="S6"/>
            <rhs>
                <symbol type="term" name="b"/>
                <symbol type="nonterm" name="S4"/>
            </rhs>
        </production>
        <production>
            <lhs name="S6"/>
            <rhs>
                <symbol type="nonterm" name="A1"/>
                <symbol type="term" name="b"/>
                <symbol type="nonterm" name="S2"/>
            </rhs>
        </production>
        <production>
            <lhs name="S7"/>
            <rhs>
                <symbol type="term" name="a"/>
                <symbol type="nonterm" name="S5"/>
            </rhs>
        </production>
        <production>
            <lhs name="S7"/>
            <rhs>
                <symbol type="nonterm" name="B1"/>
                <symbol type="term" name="a"/>
                <symbol type="nonterm" name="S3"/>
            </rhs>
        </production>
    </productions>
    <startsymbol name="S"/>
```

```
</grammar>
```

Приложение Б

inp0.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<grammar name="G0">
	<terminalsymbols>
		<term name="a" spell="a" />
		<term name="(" spell="(" />
		<term name=")" spell=")" />
		<term name="+" spell="+" />
		<term name="*" spell="*" />
	</terminalsymbols>
	<nonterminalsymbols>
		<nonterm name="E" />
		<nonterm name="T" />
		<nonterm name="E1" />
		<nonterm name="T1" />
		<nonterm name="F" />
	</nonterminalsymbols>
	<productions>
		<production>
			<lhs name="E" />
			<rhs>
				<symbol type="nonterm" name="T" />
			</rhs>
		</production>
		<production>
			<lhs name="E" />
			<rhs>
				<symbol type="nonterm" name="T" />
				<symbol type="nonterm" name="E1" />
			</rhs>
		</production>
		<production>
			<lhs name="E1" />
			<rhs>
				<symbol type="term" name="+" />
				<symbol type="nonterm" name="T" />
			</rhs>
		</production>
		<production>
			<lhs name="E1" />
			<rhs>
				<symbol type="term" name="+" />
				<symbol type="nonterm" name="T" />
				<symbol type="nonterm" name="E1" />
			</rhs>
		</production>
		<production>
			<lhs name="T" />
			<rhs>
				<symbol type="nonterm" name="F" />
			</rhs>
		</production>
		<production>
			<lhs name="T" />
			<rhs>
				<symbol type="nonterm" name="F" />
				<symbol type="nonterm" name="T1" />
			</rhs>
		</production>
		<production>
```

```
                    <lhs name="T1" />
                    <rhs>
                           <symbol type="term" name="*" />
                           <symbol type="nonterm" name="F" />
                    </rhs>
             </production>
             <production>
                    <lhs name="T1" />
                    <rhs>
                           <symbol type="term" name="*" />
                           <symbol type="nonterm" name="F" />
                           <symbol type="nonterm" name="T1" />
                    </rhs>
             </production>
             <production>
                    <lhs name="F" />
                    <rhs>
                           <symbol type="term" name="(" />
                           <symbol type="nonterm" name="E" />
                           <symbol type="term" name=")" />
                    </rhs>
             </production>
             <production>
                    <lhs name="F" />
                    <rhs>
                           <symbol type="term" name="a" />
                    </rhs>
             </production>
       </productions>
       <startsymbol name="E" />
</grammar>
```

inp1.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<grammar name="G1">
    <terminalsymbols>
        <term name="a" spell="a"/>
        <term name="b" spell="b"/>
    </terminalsymbols>
    <nonterminalsymbols>
        <nonterm name="A"/>
        <nonterm name="A1"/>
        <nonterm name="B"/>
        <nonterm name="B1"/>
        <nonterm name="S"/>
        <nonterm name="S1"/>
    </nonterminalsymbols>
    <productions>
        <production>
            <lhs name="A"/>
            <rhs>
                <symbol type="term" name="a"/>
            </rhs>
        </production>
        <production>
            <lhs name="A"/>
            <rhs>
                <symbol type="nonterm" name="S"/>
                <symbol type="term" name="a"/>
            </rhs>
        </production>
        <production>
            <lhs name="A"/>
            <rhs>
```

```
            <symbol type="term" name="a"/>
            <symbol type="nonterm" name="A1"/>
        </rhs>
    </production>
    <production>
        <lhs name="A"/>
        <rhs>
            <symbol type="nonterm" name="S"/>
            <symbol type="term" name="a"/>
            <symbol type="nonterm" name="A1"/>
        </rhs>
    </production>
    <production>
        <lhs name="A1"/>
        <rhs>
            <symbol type="nonterm" name="A"/>
            <symbol type="term" name="b"/>
        </rhs>
    </production>
    <production>
        <lhs name="A1"/>
        <rhs>
            <symbol type="nonterm" name="A"/>
            <symbol type="term" name="b"/>
            <symbol type="nonterm" name="A1"/>
        </rhs>
    </production>
    <production>
        <lhs name="B"/>
        <rhs>
            <symbol type="term" name="b"/>
        </rhs>
    </production>
    <production>
        <lhs name="B"/>
        <rhs>
            <symbol type="nonterm" name="S"/>
            <symbol type="term" name="b"/>
        </rhs>
    </production>
    <production>
        <lhs name="B"/>
        <rhs>
            <symbol type="term" name="b"/>
            <symbol type="nonterm" name="B1"/>
        </rhs>
    </production>
    <production>
        <lhs name="B"/>
        <rhs>
            <symbol type="nonterm" name="S"/>
            <symbol type="term" name="b"/>
            <symbol type="nonterm" name="B1"/>
        </rhs>
    </production>
    <production>
        <lhs name="B1"/>
        <rhs>
            <symbol type="nonterm" name="B"/>
            <symbol type="term" name="a"/>
        </rhs>
    </production>
    <production>
        <lhs name="B1"/>
```

```
            <rhs>
                <symbol type="nonterm" name="B"/>
                <symbol type="term" name="a"/>
                <symbol type="nonterm" name="B1"/>
            </rhs>
        </production>
        <production>
            <lhs name="S"/>
            <rhs>
                <symbol type="term" name="a"/>
                <symbol type="term" name="b"/>
            </rhs>
        </production>
        <production>
            <lhs name="S"/>
            <rhs>
                <symbol type="term" name="b"/>
                <symbol type="term" name="a"/>
            </rhs>
        </production>
        <production>
            <lhs name="S"/>
            <rhs>
                <symbol type="term" name="a"/>
                <symbol type="nonterm" name="A1"/>
                <symbol type="term" name="b"/>
            </rhs>
        </production>
        <production>
            <lhs name="S"/>
            <rhs>
                <symbol type="term" name="a"/>
                <symbol type="term" name="b"/>
                <symbol type="nonterm" name="S1"/>
            </rhs>
        </production>
        <production>
            <lhs name="S"/>
            <rhs>
                <symbol type="term" name="b"/>
                <symbol type="nonterm" name="B1"/>
                <symbol type="term" name="a"/>
            </rhs>
        </production>
        <production>
            <lhs name="S"/>
            <rhs>
                <symbol type="term" name="b"/>
                <symbol type="term" name="a"/>
                <symbol type="nonterm" name="S1"/>
            </rhs>
        </production>
        <production>
            <lhs name="S"/>
            <rhs>
                <symbol type="term" name="a"/>
                <symbol type="nonterm" name="A1"/>
                <symbol type="term" name="b"/>
                <symbol type="nonterm" name="S1"/>
            </rhs>
        </production>
        <production>
            <lhs name="S"/>
            <rhs>
```

```
            <symbol type="term" name="b"/>
            <symbol type="nonterm" name="B1"/>
            <symbol type="term" name="a"/>
            <symbol type="nonterm" name="S1"/>
        </rhs>
    </production>
    <production>
        <lhs name="S1"/>
        <rhs>
            <symbol type="term" name="a"/>
            <symbol type="term" name="b"/>
        </rhs>
    </production>
    <production>
        <lhs name="S1"/>
        <rhs>
            <symbol type="term" name="b"/>
            <symbol type="term" name="a"/>
        </rhs>
    </production>
    <production>
        <lhs name="S1"/>
        <rhs>
            <symbol type="term" name="a"/>
            <symbol type="nonterm" name="A1"/>
            <symbol type="term" name="b"/>
        </rhs>
    </production>
    <production>
        <lhs name="S1"/>
        <rhs>
            <symbol type="term" name="a"/>
            <symbol type="term" name="b"/>
            <symbol type="nonterm" name="S1"/>
        </rhs>
    </production>
    <production>
        <lhs name="S1"/>
        <rhs>
            <symbol type="term" name="b"/>
            <symbol type="nonterm" name="B1"/>
            <symbol type="term" name="a"/>
        </rhs>
    </production>
    <production>
        <lhs name="S1"/>
        <rhs>
            <symbol type="term" name="b"/>
            <symbol type="term" name="a"/>
            <symbol type="nonterm" name="S1"/>
        </rhs>
    </production>
    <production>
        <lhs name="S1"/>
        <rhs>
            <symbol type="term" name="a"/>
            <symbol type="nonterm" name="A1"/>
            <symbol type="term" name="b"/>
            <symbol type="nonterm" name="S1"/>
        </rhs>
    </production>
    <production>
        <lhs name="S1"/>
        <rhs>
```

```
                <symbol type="term" name="b"/>
                <symbol type="nonterm" name="B1"/>
                <symbol type="term" name="a"/>
                <symbol type="nonterm" name="S1"/>
            </rhs>
        </production>
    </productions>
    <startsymbol name="S"/>
</grammar>
```

standard0.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<grammar name="G1">
    <terminalsymbols>
        <term name="(" spell="("/>
        <term name=")" spell=")"/>
        <term name="*" spell="*"/>
        <term name="+" spell="+"/>
        <term name="a" spell="a"/>
    </terminalsymbols>
    <nonterminalsymbols>
        <nonterm name=")1"/>
        <nonterm name="E"/>
        <nonterm name="E1"/>
        <nonterm name="F"/>
        <nonterm name="T"/>
        <nonterm name="T1"/>
    </nonterminalsymbols>
    <productions>
        <production>
            <lhs name=")1"/>
            <rhs>
                <symbol type="term" name=")"/>
            </rhs>
        </production>
        <production>
            <lhs name="E"/>
            <rhs>
                <symbol type="term" name="a"/>
            </rhs>
        </production>
        <production>
            <lhs name="E"/>
            <rhs>
                <symbol type="term" name="a"/>
                <symbol type="nonterm" name="E1"/>
            </rhs>
        </production>
        <production>
            <lhs name="E"/>
            <rhs>
                <symbol type="term" name="a"/>
                <symbol type="nonterm" name="T1"/>
            </rhs>
        </production>
        <production>
            <lhs name="E"/>
            <rhs>
                <symbol type="term" name="("/>
                <symbol type="nonterm" name="E"/>
                <symbol type="nonterm" name=")1"/>
            </rhs>
        </production>
        <production>
```

```
            <lhs name="E"/>
            <rhs>
                <symbol type="term" name="a"/>
                <symbol type="nonterm" name="T1"/>
                <symbol type="nonterm" name="E1"/>
            </rhs>
        </production>
        <production>
            <lhs name="E"/>
            <rhs>
                <symbol type="term" name="("/>
                <symbol type="nonterm" name="E"/>
                <symbol type="nonterm" name=")1"/>
                <symbol type="nonterm" name="E1"/>
            </rhs>
        </production>
        <production>
            <lhs name="E"/>
            <rhs>
                <symbol type="term" name="("/>
                <symbol type="nonterm" name="E"/>
                <symbol type="nonterm" name=")1"/>
                <symbol type="nonterm" name="T1"/>
            </rhs>
        </production>
        <production>
            <lhs name="E"/>
            <rhs>
                <symbol type="term" name="("/>
                <symbol type="nonterm" name="E"/>
                <symbol type="nonterm" name=")1"/>
                <symbol type="nonterm" name="T1"/>
                <symbol type="nonterm" name="E1"/>
            </rhs>
        </production>
        <production>
            <lhs name="E1"/>
            <rhs>
                <symbol type="term" name="+"/>
                <symbol type="nonterm" name="T"/>
            </rhs>
        </production>
        <production>
            <lhs name="E1"/>
            <rhs>
                <symbol type="term" name="+"/>
                <symbol type="nonterm" name="T"/>
                <symbol type="nonterm" name="E1"/>
            </rhs>
        </production>
        <production>
            <lhs name="F"/>
            <rhs>
                <symbol type="term" name="a"/>
            </rhs>
        </production>
        <production>
            <lhs name="F"/>
            <rhs>
                <symbol type="term" name="("/>
                <symbol type="nonterm" name="E"/>
                <symbol type="nonterm" name=")1"/>
            </rhs>
        </production>
```

```
        <production>
            <lhs name="T"/>
            <rhs>
                <symbol type="term" name="a"/>
            </rhs>
        </production>
        <production>
            <lhs name="T"/>
            <rhs>
                <symbol type="term" name="a"/>
                <symbol type="nonterm" name="T1"/>
            </rhs>
        </production>
        <production>
            <lhs name="T"/>
            <rhs>
                <symbol type="term" name="("/>
                <symbol type="nonterm" name="E"/>
                <symbol type="nonterm" name=")1"/>
            </rhs>
        </production>
        <production>
            <lhs name="T"/>
            <rhs>
                <symbol type="term" name="("/>
                <symbol type="nonterm" name="E"/>
                <symbol type="nonterm" name=")1"/>
                <symbol type="nonterm" name="T1"/>
            </rhs>
        </production>
        <production>
            <lhs name="T1"/>
            <rhs>
                <symbol type="term" name="*"/>
                <symbol type="nonterm" name="F"/>
            </rhs>
        </production>
        <production>
            <lhs name="T1"/>
            <rhs>
                <symbol type="term" name="*"/>
                <symbol type="nonterm" name="F"/>
                <symbol type="nonterm" name="T1"/>
            </rhs>
        </production>
    </productions>
    <startsymbol name="E"/>
</grammar>
```

standard1.xml

```
<?xml version="1.0" encoding="UTF-8"?>
<grammar name="G1">
    <terminalsymbols>
        <term name="a" spell="a"/>
        <term name="b" spell="b"/>
    </terminalsymbols>
    <nonterminalsymbols>
        <nonterm name="A"/>
        <nonterm name="A1"/>
        <nonterm name="B"/>
        <nonterm name="B1"/>
        <nonterm name="S"/>
        <nonterm name="S1"/>
        <nonterm name="a1"/>
```

```
        <nonterm name="b1"/>
    </nonterminalsymbols>
    <productions>
        <production>
            <lhs name="A"/>
            <rhs>
                <symbol type="term" name="a"/>
            </rhs>
        </production>
        <production>
            <lhs name="A"/>
            <rhs>
                <symbol type="term" name="a"/>
                <symbol type="nonterm" name="A1"/>
            </rhs>
        </production>
        <production>
            <lhs name="A"/>
            <rhs>
                <symbol type="term" name="a"/>
                <symbol type="nonterm" name="b1"/>
                <symbol type="nonterm" name="a1"/>
            </rhs>
        </production>
        <production>
            <lhs name="A"/>
            <rhs>
                <symbol type="term" name="b"/>
                <symbol type="nonterm" name="a1"/>
                <symbol type="nonterm" name="a1"/>
            </rhs>
        </production>
        <production>
            <lhs name="A"/>
            <rhs>
                <symbol type="term" name="a"/>
                <symbol type="nonterm" name="A1"/>
                <symbol type="nonterm" name="b1"/>
                <symbol type="nonterm" name="a1"/>
            </rhs>
        </production>
        <production>
            <lhs name="A"/>
            <rhs>
                <symbol type="term" name="a"/>
                <symbol type="nonterm" name="b1"/>
                <symbol type="nonterm" name="S1"/>
                <symbol type="nonterm" name="a1"/>
            </rhs>
        </production>
        <production>
            <lhs name="A"/>
            <rhs>
                <symbol type="term" name="a"/>
                <symbol type="nonterm" name="b1"/>
                <symbol type="nonterm" name="a1"/>
                <symbol type="nonterm" name="A1"/>
            </rhs>
        </production>
        <production>
            <lhs name="A"/>
            <rhs>
                <symbol type="term" name="b"/>
                <symbol type="nonterm" name="B1"/>
```

```
            <symbol type="nonterm" name="a1"/>
            <symbol type="nonterm" name="a1"/>
        </rhs>
    </production>
    <production>
        <lhs name="A"/>
        <rhs>
            <symbol type="term" name="b"/>
            <symbol type="nonterm" name="a1"/>
            <symbol type="nonterm" name="S1"/>
            <symbol type="nonterm" name="a1"/>
        </rhs>
    </production>
    <production>
        <lhs name="A"/>
        <rhs>
            <symbol type="term" name="b"/>
            <symbol type="nonterm" name="a1"/>
            <symbol type="nonterm" name="a1"/>
            <symbol type="nonterm" name="A1"/>
        </rhs>
    </production>
    <production>
        <lhs name="A"/>
        <rhs>
            <symbol type="term" name="a"/>
            <symbol type="nonterm" name="A1"/>
            <symbol type="nonterm" name="b1"/>
            <symbol type="nonterm" name="S1"/>
            <symbol type="nonterm" name="a1"/>
        </rhs>
    </production>
    <production>
        <lhs name="A"/>
        <rhs>
            <symbol type="term" name="a"/>
            <symbol type="nonterm" name="A1"/>
            <symbol type="nonterm" name="b1"/>
            <symbol type="nonterm" name="a1"/>
            <symbol type="nonterm" name="A1"/>
        </rhs>
    </production>
    <production>
        <lhs name="A"/>
        <rhs>
            <symbol type="term" name="a"/>
            <symbol type="nonterm" name="b1"/>
            <symbol type="nonterm" name="S1"/>
            <symbol type="nonterm" name="a1"/>
            <symbol type="nonterm" name="A1"/>
        </rhs>
    </production>
    <production>
        <lhs name="A"/>
        <rhs>
            <symbol type="term" name="b"/>
            <symbol type="nonterm" name="B1"/>
            <symbol type="nonterm" name="a1"/>
            <symbol type="nonterm" name="S1"/>
            <symbol type="nonterm" name="a1"/>
        </rhs>
    </production>
    <production>
        <lhs name="A"/>
```

```
            <rhs>
                <symbol type="term" name="b"/>
                <symbol type="nonterm" name="B1"/>
                <symbol type="nonterm" name="a1"/>
                <symbol type="nonterm" name="a1"/>
                <symbol type="nonterm" name="A1"/>
            </rhs>
        </production>
        <production>
            <lhs name="A"/>
            <rhs>
                <symbol type="term" name="b"/>
                <symbol type="nonterm" name="a1"/>
                <symbol type="nonterm" name="S1"/>
                <symbol type="nonterm" name="a1"/>
                <symbol type="nonterm" name="A1"/>
            </rhs>
        </production>
        <production>
            <lhs name="A"/>
            <rhs>
                <symbol type="term" name="a"/>
                <symbol type="nonterm" name="A1"/>
                <symbol type="nonterm" name="b1"/>
                <symbol type="nonterm" name="S1"/>
                <symbol type="nonterm" name="a1"/>
                <symbol type="nonterm" name="A1"/>
            </rhs>
        </production>
        <production>
            <lhs name="A"/>
            <rhs>
                <symbol type="term" name="b"/>
                <symbol type="nonterm" name="B1"/>
                <symbol type="nonterm" name="a1"/>
                <symbol type="nonterm" name="S1"/>
                <symbol type="nonterm" name="a1"/>
                <symbol type="nonterm" name="A1"/>
            </rhs>
        </production>
        <production>
            <lhs name="A1"/>
            <rhs>
                <symbol type="term" name="a"/>
                <symbol type="nonterm" name="b1"/>
            </rhs>
        </production>
        <production>
            <lhs name="A1"/>
            <rhs>
                <symbol type="term" name="a"/>
                <symbol type="nonterm" name="A1"/>
                <symbol type="nonterm" name="b1"/>
            </rhs>
        </production>
        <production>
            <lhs name="A1"/>
            <rhs>
                <symbol type="term" name="a"/>
                <symbol type="nonterm" name="b1"/>
                <symbol type="nonterm" name="A1"/>
            </rhs>
        </production>
        <production>
```

```
            <lhs name="A1"/>
            <rhs>
                <symbol type="term" name="a"/>
                <symbol type="nonterm" name="A1"/>
                <symbol type="nonterm" name="b1"/>
                <symbol type="nonterm" name="A1"/>
            </rhs>
        </production>
        <production>
            <lhs name="A1"/>
            <rhs>
                <symbol type="term" name="a"/>
                <symbol type="nonterm" name="b1"/>
                <symbol type="nonterm" name="a1"/>
                <symbol type="nonterm" name="b1"/>
            </rhs>
        </production>
        <production>
            <lhs name="A1"/>
            <rhs>
                <symbol type="term" name="b"/>
                <symbol type="nonterm" name="a1"/>
                <symbol type="nonterm" name="a1"/>
                <symbol type="nonterm" name="b1"/>
            </rhs>
        </production>
        <production>
            <lhs name="A1"/>
            <rhs>
                <symbol type="term" name="a"/>
                <symbol type="nonterm" name="A1"/>
                <symbol type="nonterm" name="b1"/>
                <symbol type="nonterm" name="a1"/>
                <symbol type="nonterm" name="b1"/>
            </rhs>
        </production>
        <production>
            <lhs name="A1"/>
            <rhs>
                <symbol type="term" name="a"/>
                <symbol type="nonterm" name="b1"/>
                <symbol type="nonterm" name="S1"/>
                <symbol type="nonterm" name="a1"/>
                <symbol type="nonterm" name="b1"/>
            </rhs>
        </production>
        <production>
            <lhs name="A1"/>
            <rhs>
                <symbol type="term" name="a"/>
                <symbol type="nonterm" name="b1"/>
                <symbol type="nonterm" name="a1"/>
                <symbol type="nonterm" name="A1"/>
                <symbol type="nonterm" name="b1"/>
            </rhs>
        </production>
        <production>
            <lhs name="A1"/>
            <rhs>
                <symbol type="term" name="a"/>
                <symbol type="nonterm" name="b1"/>
                <symbol type="nonterm" name="a1"/>
                <symbol type="nonterm" name="b1"/>
                <symbol type="nonterm" name="A1"/>
```

```
            </rhs>
        </production>
        <production>
            <lhs name="A1"/>
            <rhs>
                <symbol type="term" name="b"/>
                <symbol type="nonterm" name="B1"/>
                <symbol type="nonterm" name="a1"/>
                <symbol type="nonterm" name="a1"/>
                <symbol type="nonterm" name="b1"/>
            </rhs>
        </production>
        <production>
            <lhs name="A1"/>
            <rhs>
                <symbol type="term" name="b"/>
                <symbol type="nonterm" name="a1"/>
                <symbol type="nonterm" name="S1"/>
                <symbol type="nonterm" name="a1"/>
                <symbol type="nonterm" name="b1"/>
            </rhs>
        </production>
        <production>
            <lhs name="A1"/>
            <rhs>
                <symbol type="term" name="b"/>
                <symbol type="nonterm" name="a1"/>
                <symbol type="nonterm" name="a1"/>
                <symbol type="nonterm" name="A1"/>
                <symbol type="nonterm" name="b1"/>
            </rhs>
        </production>
        <production>
            <lhs name="A1"/>
            <rhs>
                <symbol type="term" name="b"/>
                <symbol type="nonterm" name="a1"/>
                <symbol type="nonterm" name="a1"/>
                <symbol type="nonterm" name="b1"/>
                <symbol type="nonterm" name="A1"/>
            </rhs>
        </production>
        <production>
            <lhs name="A1"/>
            <rhs>
                <symbol type="term" name="a"/>
                <symbol type="nonterm" name="A1"/>
                <symbol type="nonterm" name="b1"/>
                <symbol type="nonterm" name="S1"/>
                <symbol type="nonterm" name="a1"/>
                <symbol type="nonterm" name="b1"/>
            </rhs>
        </production>
        <production>
            <lhs name="A1"/>
            <rhs>
                <symbol type="term" name="a"/>
                <symbol type="nonterm" name="A1"/>
                <symbol type="nonterm" name="b1"/>
                <symbol type="nonterm" name="a1"/>
                <symbol type="nonterm" name="A1"/>
                <symbol type="nonterm" name="b1"/>
            </rhs>
        </production>
```

```
        <production>
            <lhs name="A1"/>
            <rhs>
                <symbol type="term" name="a"/>
                <symbol type="nonterm" name="A1"/>
                <symbol type="nonterm" name="b1"/>
                <symbol type="nonterm" name="a1"/>
                <symbol type="nonterm" name="b1"/>
                <symbol type="nonterm" name="A1"/>
            </rhs>
        </production>
        <production>
            <lhs name="A1"/>
            <rhs>
                <symbol type="term" name="a"/>
                <symbol type="nonterm" name="b1"/>
                <symbol type="nonterm" name="S1"/>
                <symbol type="nonterm" name="a1"/>
                <symbol type="nonterm" name="A1"/>
                <symbol type="nonterm" name="b1"/>
            </rhs>
        </production>
        <production>
            <lhs name="A1"/>
            <rhs>
                <symbol type="term" name="a"/>
                <symbol type="nonterm" name="b1"/>
                <symbol type="nonterm" name="S1"/>
                <symbol type="nonterm" name="a1"/>
                <symbol type="nonterm" name="b1"/>
                <symbol type="nonterm" name="A1"/>
            </rhs>
        </production>
        <production>
            <lhs name="A1"/>
            <rhs>
                <symbol type="term" name="a"/>
                <symbol type="nonterm" name="b1"/>
                <symbol type="nonterm" name="a1"/>
                <symbol type="nonterm" name="A1"/>
                <symbol type="nonterm" name="b1"/>
                <symbol type="nonterm" name="A1"/>
            </rhs>
        </production>
        <production>
            <lhs name="A1"/>
            <rhs>
                <symbol type="term" name="b"/>
                <symbol type="nonterm" name="B1"/>
                <symbol type="nonterm" name="a1"/>
                <symbol type="nonterm" name="S1"/>
                <symbol type="nonterm" name="a1"/>
                <symbol type="nonterm" name="b1"/>
            </rhs>
        </production>
        <production>
            <lhs name="A1"/>
            <rhs>
                <symbol type="term" name="b"/>
                <symbol type="nonterm" name="B1"/>
                <symbol type="nonterm" name="a1"/>
                <symbol type="nonterm" name="a1"/>
                <symbol type="nonterm" name="A1"/>
                <symbol type="nonterm" name="b1"/>
```

```
            </rhs>
        </production>
        <production>
            <lhs name="A1"/>
            <rhs>
                <symbol type="term" name="b"/>
                <symbol type="nonterm" name="B1"/>
                <symbol type="nonterm" name="a1"/>
                <symbol type="nonterm" name="a1"/>
                <symbol type="nonterm" name="b1"/>
                <symbol type="nonterm" name="A1"/>
            </rhs>
        </production>
        <production>
            <lhs name="A1"/>
            <rhs>
                <symbol type="term" name="b"/>
                <symbol type="nonterm" name="a1"/>
                <symbol type="nonterm" name="S1"/>
                <symbol type="nonterm" name="a1"/>
                <symbol type="nonterm" name="A1"/>
                <symbol type="nonterm" name="b1"/>
            </rhs>
        </production>
        <production>
            <lhs name="A1"/>
            <rhs>
                <symbol type="term" name="b"/>
                <symbol type="nonterm" name="a1"/>
                <symbol type="nonterm" name="S1"/>
                <symbol type="nonterm" name="a1"/>
                <symbol type="nonterm" name="b1"/>
                <symbol type="nonterm" name="A1"/>
            </rhs>
        </production>
        <production>
            <lhs name="A1"/>
            <rhs>
                <symbol type="term" name="b"/>
                <symbol type="nonterm" name="a1"/>
                <symbol type="nonterm" name="a1"/>
                <symbol type="nonterm" name="A1"/>
                <symbol type="nonterm" name="b1"/>
                <symbol type="nonterm" name="A1"/>
            </rhs>
        </production>
        <production>
            <lhs name="A1"/>
            <rhs>
                <symbol type="term" name="a"/>
                <symbol type="nonterm" name="A1"/>
                <symbol type="nonterm" name="b1"/>
                <symbol type="nonterm" name="S1"/>
                <symbol type="nonterm" name="a1"/>
                <symbol type="nonterm" name="A1"/>
                <symbol type="nonterm" name="b1"/>
            </rhs>
        </production>
        <production>
            <lhs name="A1"/>
            <rhs>
                <symbol type="term" name="a"/>
                <symbol type="nonterm" name="A1"/>
                <symbol type="nonterm" name="b1"/>
```

```
            <symbol type="nonterm" name="S1"/>
            <symbol type="nonterm" name="a1"/>
            <symbol type="nonterm" name="b1"/>
            <symbol type="nonterm" name="A1"/>
        </rhs>
    </production>
    <production>
        <lhs name="A1"/>
        <rhs>
            <symbol type="term" name="a"/>
            <symbol type="nonterm" name="A1"/>
            <symbol type="nonterm" name="b1"/>
            <symbol type="nonterm" name="a1"/>
            <symbol type="nonterm" name="A1"/>
            <symbol type="nonterm" name="b1"/>
            <symbol type="nonterm" name="A1"/>
        </rhs>
    </production>
    <production>
        <lhs name="A1"/>
        <rhs>
            <symbol type="term" name="a"/>
            <symbol type="nonterm" name="b1"/>
            <symbol type="nonterm" name="S1"/>
            <symbol type="nonterm" name="a1"/>
            <symbol type="nonterm" name="A1"/>
            <symbol type="nonterm" name="b1"/>
            <symbol type="nonterm" name="A1"/>
        </rhs>
    </production>
    <production>
        <lhs name="A1"/>
        <rhs>
            <symbol type="term" name="b"/>
            <symbol type="nonterm" name="B1"/>
            <symbol type="nonterm" name="a1"/>
            <symbol type="nonterm" name="S1"/>
            <symbol type="nonterm" name="a1"/>
            <symbol type="nonterm" name="A1"/>
            <symbol type="nonterm" name="b1"/>
        </rhs>
    </production>
    <production>
        <lhs name="A1"/>
        <rhs>
            <symbol type="term" name="b"/>
            <symbol type="nonterm" name="B1"/>
            <symbol type="nonterm" name="a1"/>
            <symbol type="nonterm" name="S1"/>
            <symbol type="nonterm" name="a1"/>
            <symbol type="nonterm" name="b1"/>
            <symbol type="nonterm" name="A1"/>
        </rhs>
    </production>
    <production>
        <lhs name="A1"/>
        <rhs>
            <symbol type="term" name="b"/>
            <symbol type="nonterm" name="B1"/>
            <symbol type="nonterm" name="a1"/>
            <symbol type="nonterm" name="a1"/>
            <symbol type="nonterm" name="A1"/>
            <symbol type="nonterm" name="b1"/>
            <symbol type="nonterm" name="A1"/>
```

```
            </rhs>
        </production>
        <production>
            <lhs name="A1"/>
            <rhs>
                <symbol type="term" name="b"/>
                <symbol type="nonterm" name="a1"/>
                <symbol type="nonterm" name="S1"/>
                <symbol type="nonterm" name="a1"/>
                <symbol type="nonterm" name="A1"/>
                <symbol type="nonterm" name="b1"/>
                <symbol type="nonterm" name="A1"/>
            </rhs>
        </production>
        <production>
            <lhs name="A1"/>
            <rhs>
                <symbol type="term" name="a"/>
                <symbol type="nonterm" name="A1"/>
                <symbol type="nonterm" name="b1"/>
                <symbol type="nonterm" name="S1"/>
                <symbol type="nonterm" name="a1"/>
                <symbol type="nonterm" name="A1"/>
                <symbol type="nonterm" name="b1"/>
                <symbol type="nonterm" name="A1"/>
            </rhs>
        </production>
        <production>
            <lhs name="A1"/>
            <rhs>
                <symbol type="term" name="b"/>
                <symbol type="nonterm" name="B1"/>
                <symbol type="nonterm" name="a1"/>
                <symbol type="nonterm" name="S1"/>
                <symbol type="nonterm" name="a1"/>
                <symbol type="nonterm" name="A1"/>
                <symbol type="nonterm" name="b1"/>
                <symbol type="nonterm" name="A1"/>
            </rhs>
        </production>
        <production>
            <lhs name="B"/>
            <rhs>
                <symbol type="term" name="b"/>
            </rhs>
        </production>
        <production>
            <lhs name="B"/>
            <rhs>
                <symbol type="term" name="b"/>
                <symbol type="nonterm" name="B1"/>
            </rhs>
        </production>
        <production>
            <lhs name="B"/>
            <rhs>
                <symbol type="term" name="a"/>
                <symbol type="nonterm" name="b1"/>
                <symbol type="nonterm" name="b1"/>
            </rhs>
        </production>
        <production>
            <lhs name="B"/>
            <rhs>
```

```
            <symbol type="term" name="b"/>
            <symbol type="nonterm" name="a1"/>
            <symbol type="nonterm" name="b1"/>
        </rhs>
    </production>
    <production>
        <lhs name="B"/>
        <rhs>
            <symbol type="term" name="a"/>
            <symbol type="nonterm" name="A1"/>
            <symbol type="nonterm" name="b1"/>
            <symbol type="nonterm" name="b1"/>
        </rhs>
    </production>
    <production>
        <lhs name="B"/>
        <rhs>
            <symbol type="term" name="a"/>
            <symbol type="nonterm" name="b1"/>
            <symbol type="nonterm" name="S1"/>
            <symbol type="nonterm" name="b1"/>
        </rhs>
    </production>
    <production>
        <lhs name="B"/>
        <rhs>
            <symbol type="term" name="a"/>
            <symbol type="nonterm" name="b1"/>
            <symbol type="nonterm" name="b1"/>
            <symbol type="nonterm" name="B1"/>
        </rhs>
    </production>
    <production>
        <lhs name="B"/>
        <rhs>
            <symbol type="term" name="b"/>
            <symbol type="nonterm" name="B1"/>
            <symbol type="nonterm" name="a1"/>
            <symbol type="nonterm" name="b1"/>
        </rhs>
    </production>
    <production>
        <lhs name="B"/>
        <rhs>
            <symbol type="term" name="b"/>
            <symbol type="nonterm" name="a1"/>
            <symbol type="nonterm" name="S1"/>
            <symbol type="nonterm" name="b1"/>
        </rhs>
    </production>
    <production>
        <lhs name="B"/>
        <rhs>
            <symbol type="term" name="b"/>
            <symbol type="nonterm" name="a1"/>
            <symbol type="nonterm" name="b1"/>
            <symbol type="nonterm" name="B1"/>
        </rhs>
    </production>
    <production>
        <lhs name="B"/>
        <rhs>
            <symbol type="term" name="a"/>
            <symbol type="nonterm" name="A1"/>
```

```
            <symbol type="nonterm" name="b1"/>
            <symbol type="nonterm" name="S1"/>
            <symbol type="nonterm" name="b1"/>
        </rhs>
    </production>
    <production>
        <lhs name="B"/>
        <rhs>
            <symbol type="term" name="a"/>
            <symbol type="nonterm" name="A1"/>
            <symbol type="nonterm" name="b1"/>
            <symbol type="nonterm" name="b1"/>
            <symbol type="nonterm" name="B1"/>
        </rhs>
    </production>
    <production>
        <lhs name="B"/>
        <rhs>
            <symbol type="term" name="a"/>
            <symbol type="nonterm" name="b1"/>
            <symbol type="nonterm" name="S1"/>
            <symbol type="nonterm" name="b1"/>
            <symbol type="nonterm" name="B1"/>
        </rhs>
    </production>
    <production>
        <lhs name="B"/>
        <rhs>
            <symbol type="term" name="b"/>
            <symbol type="nonterm" name="B1"/>
            <symbol type="nonterm" name="a1"/>
            <symbol type="nonterm" name="S1"/>
            <symbol type="nonterm" name="b1"/>
        </rhs>
    </production>
    <production>
        <lhs name="B"/>
        <rhs>
            <symbol type="term" name="b"/>
            <symbol type="nonterm" name="B1"/>
            <symbol type="nonterm" name="a1"/>
            <symbol type="nonterm" name="b1"/>
            <symbol type="nonterm" name="B1"/>
        </rhs>
    </production>
    <production>
        <lhs name="B"/>
        <rhs>
            <symbol type="term" name="b"/>
            <symbol type="nonterm" name="a1"/>
            <symbol type="nonterm" name="S1"/>
            <symbol type="nonterm" name="b1"/>
            <symbol type="nonterm" name="B1"/>
        </rhs>
    </production>
    <production>
        <lhs name="B"/>
        <rhs>
            <symbol type="term" name="a"/>
            <symbol type="nonterm" name="A1"/>
            <symbol type="nonterm" name="b1"/>
            <symbol type="nonterm" name="S1"/>
            <symbol type="nonterm" name="b1"/>
            <symbol type="nonterm" name="B1"/>
```

```
            </rhs>
        </production>
        <production>
            <lhs name="B"/>
            <rhs>
                <symbol type="term" name="b"/>
                <symbol type="nonterm" name="B1"/>
                <symbol type="nonterm" name="a1"/>
                <symbol type="nonterm" name="S1"/>
                <symbol type="nonterm" name="b1"/>
                <symbol type="nonterm" name="B1"/>
            </rhs>
        </production>
        <production>
            <lhs name="B1"/>
            <rhs>
                <symbol type="term" name="b"/>
                <symbol type="nonterm" name="a1"/>
            </rhs>
        </production>
        <production>
            <lhs name="B1"/>
            <rhs>
                <symbol type="term" name="b"/>
                <symbol type="nonterm" name="B1"/>
                <symbol type="nonterm" name="a1"/>
            </rhs>
        </production>
        <production>
            <lhs name="B1"/>
            <rhs>
                <symbol type="term" name="b"/>
                <symbol type="nonterm" name="a1"/>
                <symbol type="nonterm" name="B1"/>
            </rhs>
        </production>
        <production>
            <lhs name="B1"/>
            <rhs>
                <symbol type="term" name="a"/>
                <symbol type="nonterm" name="b1"/>
                <symbol type="nonterm" name="b1"/>
                <symbol type="nonterm" name="a1"/>
            </rhs>
        </production>
        <production>
            <lhs name="B1"/>
            <rhs>
                <symbol type="term" name="b"/>
                <symbol type="nonterm" name="B1"/>
                <symbol type="nonterm" name="a1"/>
                <symbol type="nonterm" name="B1"/>
            </rhs>
        </production>
        <production>
            <lhs name="B1"/>
            <rhs>
                <symbol type="term" name="b"/>
                <symbol type="nonterm" name="a1"/>
                <symbol type="nonterm" name="b1"/>
                <symbol type="nonterm" name="a1"/>
            </rhs>
        </production>
        <production>
```

```
            <lhs name="B1"/>
            <rhs>
                <symbol type="term" name="a"/>
                <symbol type="nonterm" name="A1"/>
                <symbol type="nonterm" name="b1"/>
                <symbol type="nonterm" name="b1"/>
                <symbol type="nonterm" name="a1"/>
            </rhs>
        </production>
        <production>
            <lhs name="B1"/>
            <rhs>
                <symbol type="term" name="a"/>
                <symbol type="nonterm" name="b1"/>
                <symbol type="nonterm" name="S1"/>
                <symbol type="nonterm" name="b1"/>
                <symbol type="nonterm" name="a1"/>
            </rhs>
        </production>
        <production>
            <lhs name="B1"/>
            <rhs>
                <symbol type="term" name="a"/>
                <symbol type="nonterm" name="b1"/>
                <symbol type="nonterm" name="b1"/>
                <symbol type="nonterm" name="B1"/>
                <symbol type="nonterm" name="a1"/>
            </rhs>
        </production>
        <production>
            <lhs name="B1"/>
            <rhs>
                <symbol type="term" name="a"/>
                <symbol type="nonterm" name="b1"/>
                <symbol type="nonterm" name="b1"/>
                <symbol type="nonterm" name="a1"/>
                <symbol type="nonterm" name="B1"/>
            </rhs>
        </production>
        <production>
            <lhs name="B1"/>
            <rhs>
                <symbol type="term" name="b"/>
                <symbol type="nonterm" name="B1"/>
                <symbol type="nonterm" name="a1"/>
                <symbol type="nonterm" name="b1"/>
                <symbol type="nonterm" name="a1"/>
            </rhs>
        </production>
        <production>
            <lhs name="B1"/>
            <rhs>
                <symbol type="term" name="b"/>
                <symbol type="nonterm" name="a1"/>
                <symbol type="nonterm" name="S1"/>
                <symbol type="nonterm" name="b1"/>
                <symbol type="nonterm" name="a1"/>
            </rhs>
        </production>
        <production>
            <lhs name="B1"/>
            <rhs>
                <symbol type="term" name="b"/>
                <symbol type="nonterm" name="a1"/>
```

```
            <symbol type="nonterm" name="b1"/>
            <symbol type="nonterm" name="B1"/>
            <symbol type="nonterm" name="a1"/>
        </rhs>
    </production>
    <production>
        <lhs name="B1"/>
        <rhs>
            <symbol type="term" name="b"/>
            <symbol type="nonterm" name="a1"/>
            <symbol type="nonterm" name="b1"/>
            <symbol type="nonterm" name="a1"/>
            <symbol type="nonterm" name="B1"/>
        </rhs>
    </production>
    <production>
        <lhs name="B1"/>
        <rhs>
            <symbol type="term" name="a"/>
            <symbol type="nonterm" name="A1"/>
            <symbol type="nonterm" name="b1"/>
            <symbol type="nonterm" name="S1"/>
            <symbol type="nonterm" name="b1"/>
            <symbol type="nonterm" name="a1"/>
        </rhs>
    </production>
    <production>
        <lhs name="B1"/>
        <rhs>
            <symbol type="term" name="a"/>
            <symbol type="nonterm" name="A1"/>
            <symbol type="nonterm" name="b1"/>
            <symbol type="nonterm" name="b1"/>
            <symbol type="nonterm" name="B1"/>
            <symbol type="nonterm" name="a1"/>
        </rhs>
    </production>
    <production>
        <lhs name="B1"/>
        <rhs>
            <symbol type="term" name="a"/>
            <symbol type="nonterm" name="A1"/>
            <symbol type="nonterm" name="b1"/>
            <symbol type="nonterm" name="b1"/>
            <symbol type="nonterm" name="a1"/>
            <symbol type="nonterm" name="B1"/>
        </rhs>
    </production>
    <production>
        <lhs name="B1"/>
        <rhs>
            <symbol type="term" name="a"/>
            <symbol type="nonterm" name="b1"/>
            <symbol type="nonterm" name="S1"/>
            <symbol type="nonterm" name="b1"/>
            <symbol type="nonterm" name="B1"/>
            <symbol type="nonterm" name="a1"/>
        </rhs>
    </production>
    <production>
        <lhs name="B1"/>
        <rhs>
            <symbol type="term" name="a"/>
            <symbol type="nonterm" name="b1"/>
```

```
            <symbol type="nonterm" name="S1"/>
            <symbol type="nonterm" name="b1"/>
            <symbol type="nonterm" name="a1"/>
            <symbol type="nonterm" name="B1"/>
        </rhs>
    </production>
    <production>
        <lhs name="B1"/>
        <rhs>
            <symbol type="term" name="a"/>
            <symbol type="nonterm" name="b1"/>
            <symbol type="nonterm" name="b1"/>
            <symbol type="nonterm" name="B1"/>
            <symbol type="nonterm" name="a1"/>
            <symbol type="nonterm" name="B1"/>
        </rhs>
    </production>
    <production>
        <lhs name="B1"/>
        <rhs>
            <symbol type="term" name="b"/>
            <symbol type="nonterm" name="B1"/>
            <symbol type="nonterm" name="a1"/>
            <symbol type="nonterm" name="S1"/>
            <symbol type="nonterm" name="b1"/>
            <symbol type="nonterm" name="a1"/>
        </rhs>
    </production>
    <production>
        <lhs name="B1"/>
        <rhs>
            <symbol type="term" name="b"/>
            <symbol type="nonterm" name="B1"/>
            <symbol type="nonterm" name="a1"/>
            <symbol type="nonterm" name="b1"/>
            <symbol type="nonterm" name="B1"/>
            <symbol type="nonterm" name="a1"/>
        </rhs>
    </production>
    <production>
        <lhs name="B1"/>
        <rhs>
            <symbol type="term" name="b"/>
            <symbol type="nonterm" name="B1"/>
            <symbol type="nonterm" name="a1"/>
            <symbol type="nonterm" name="b1"/>
            <symbol type="nonterm" name="a1"/>
            <symbol type="nonterm" name="B1"/>
        </rhs>
    </production>
    <production>
        <lhs name="B1"/>
        <rhs>
            <symbol type="term" name="b"/>
            <symbol type="nonterm" name="a1"/>
            <symbol type="nonterm" name="S1"/>
            <symbol type="nonterm" name="b1"/>
            <symbol type="nonterm" name="B1"/>
            <symbol type="nonterm" name="a1"/>
        </rhs>
    </production>
    <production>
        <lhs name="B1"/>
        <rhs>
```

```
            <symbol type="term" name="b"/>
            <symbol type="nonterm" name="a1"/>
            <symbol type="nonterm" name="S1"/>
            <symbol type="nonterm" name="b1"/>
            <symbol type="nonterm" name="a1"/>
            <symbol type="nonterm" name="B1"/>
        </rhs>
    </production>
    <production>
        <lhs name="B1"/>
        <rhs>
            <symbol type="term" name="b"/>
            <symbol type="nonterm" name="a1"/>
            <symbol type="nonterm" name="b1"/>
            <symbol type="nonterm" name="B1"/>
            <symbol type="nonterm" name="a1"/>
            <symbol type="nonterm" name="B1"/>
        </rhs>
    </production>
    <production>
        <lhs name="B1"/>
        <rhs>
            <symbol type="term" name="a"/>
            <symbol type="nonterm" name="A1"/>
            <symbol type="nonterm" name="b1"/>
            <symbol type="nonterm" name="S1"/>
            <symbol type="nonterm" name="b1"/>
            <symbol type="nonterm" name="B1"/>
            <symbol type="nonterm" name="a1"/>
        </rhs>
    </production>
    <production>
        <lhs name="B1"/>
        <rhs>
            <symbol type="term" name="a"/>
            <symbol type="nonterm" name="A1"/>
            <symbol type="nonterm" name="b1"/>
            <symbol type="nonterm" name="S1"/>
            <symbol type="nonterm" name="b1"/>
            <symbol type="nonterm" name="a1"/>
            <symbol type="nonterm" name="B1"/>
        </rhs>
    </production>
    <production>
        <lhs name="B1"/>
        <rhs>
            <symbol type="term" name="a"/>
            <symbol type="nonterm" name="A1"/>
            <symbol type="nonterm" name="b1"/>
            <symbol type="nonterm" name="b1"/>
            <symbol type="nonterm" name="B1"/>
            <symbol type="nonterm" name="a1"/>
            <symbol type="nonterm" name="B1"/>
        </rhs>
    </production>
    <production>
        <lhs name="B1"/>
        <rhs>
            <symbol type="term" name="a"/>
            <symbol type="nonterm" name="b1"/>
            <symbol type="nonterm" name="S1"/>
            <symbol type="nonterm" name="b1"/>
            <symbol type="nonterm" name="B1"/>
            <symbol type="nonterm" name="a1"/>
```

```
            <symbol type="nonterm" name="B1"/>
        </rhs>
    </production>
    <production>
        <lhs name="B1"/>
        <rhs>
            <symbol type="term" name="b"/>
            <symbol type="nonterm" name="B1"/>
            <symbol type="nonterm" name="a1"/>
            <symbol type="nonterm" name="S1"/>
            <symbol type="nonterm" name="b1"/>
            <symbol type="nonterm" name="B1"/>
            <symbol type="nonterm" name="a1"/>
        </rhs>
    </production>
    <production>
        <lhs name="B1"/>
        <rhs>
            <symbol type="term" name="b"/>
            <symbol type="nonterm" name="B1"/>
            <symbol type="nonterm" name="a1"/>
            <symbol type="nonterm" name="S1"/>
            <symbol type="nonterm" name="b1"/>
            <symbol type="nonterm" name="a1"/>
            <symbol type="nonterm" name="B1"/>
        </rhs>
    </production>
    <production>
        <lhs name="B1"/>
        <rhs>
            <symbol type="term" name="b"/>
            <symbol type="nonterm" name="B1"/>
            <symbol type="nonterm" name="a1"/>
            <symbol type="nonterm" name="b1"/>
            <symbol type="nonterm" name="B1"/>
            <symbol type="nonterm" name="a1"/>
            <symbol type="nonterm" name="B1"/>
        </rhs>
    </production>
    <production>
        <lhs name="B1"/>
        <rhs>
            <symbol type="term" name="b"/>
            <symbol type="nonterm" name="a1"/>
            <symbol type="nonterm" name="S1"/>
            <symbol type="nonterm" name="b1"/>
            <symbol type="nonterm" name="B1"/>
            <symbol type="nonterm" name="a1"/>
            <symbol type="nonterm" name="B1"/>
        </rhs>
    </production>
    <production>
        <lhs name="B1"/>
        <rhs>
            <symbol type="term" name="a"/>
            <symbol type="nonterm" name="A1"/>
            <symbol type="nonterm" name="b1"/>
            <symbol type="nonterm" name="S1"/>
            <symbol type="nonterm" name="b1"/>
            <symbol type="nonterm" name="B1"/>
            <symbol type="nonterm" name="a1"/>
            <symbol type="nonterm" name="B1"/>
        </rhs>
    </production>
```

```
        <production>
            <lhs name="B1"/>
            <rhs>
                <symbol type="term" name="b"/>
                <symbol type="nonterm" name="B1"/>
                <symbol type="nonterm" name="a1"/>
                <symbol type="nonterm" name="S1"/>
                <symbol type="nonterm" name="b1"/>
                <symbol type="nonterm" name="B1"/>
                <symbol type="nonterm" name="a1"/>
                <symbol type="nonterm" name="B1"/>
            </rhs>
        </production>
        <production>
            <lhs name="S"/>
            <rhs>
                <symbol type="term" name="a"/>
                <symbol type="nonterm" name="b1"/>
            </rhs>
        </production>
        <production>
            <lhs name="S"/>
            <rhs>
                <symbol type="term" name="b"/>
                <symbol type="nonterm" name="a1"/>
            </rhs>
        </production>
        <production>
            <lhs name="S"/>
            <rhs>
                <symbol type="term" name="a"/>
                <symbol type="nonterm" name="A1"/>
                <symbol type="nonterm" name="b1"/>
            </rhs>
        </production>
        <production>
            <lhs name="S"/>
            <rhs>
                <symbol type="term" name="a"/>
                <symbol type="nonterm" name="b1"/>
                <symbol type="nonterm" name="S1"/>
            </rhs>
        </production>
        <production>
            <lhs name="S"/>
            <rhs>
                <symbol type="term" name="b"/>
                <symbol type="nonterm" name="B1"/>
                <symbol type="nonterm" name="a1"/>
            </rhs>
        </production>
        <production>
            <lhs name="S"/>
            <rhs>
                <symbol type="term" name="b"/>
                <symbol type="nonterm" name="a1"/>
                <symbol type="nonterm" name="S1"/>
            </rhs>
        </production>
        <production>
            <lhs name="S"/>
            <rhs>
                <symbol type="term" name="a"/>
                <symbol type="nonterm" name="A1"/>
```

```
            <symbol type="nonterm" name="b1"/>
            <symbol type="nonterm" name="S1"/>
        </rhs>
    </production>
    <production>
        <lhs name="S"/>
        <rhs>
            <symbol type="term" name="b"/>
            <symbol type="nonterm" name="B1"/>
            <symbol type="nonterm" name="a1"/>
            <symbol type="nonterm" name="S1"/>
        </rhs>
    </production>
    <production>
        <lhs name="S1"/>
        <rhs>
            <symbol type="term" name="a"/>
            <symbol type="nonterm" name="b1"/>
        </rhs>
    </production>
    <production>
        <lhs name="S1"/>
        <rhs>
            <symbol type="term" name="b"/>
            <symbol type="nonterm" name="a1"/>
        </rhs>
    </production>
    <production>
        <lhs name="S1"/>
        <rhs>
            <symbol type="term" name="a"/>
            <symbol type="nonterm" name="A1"/>
            <symbol type="nonterm" name="b1"/>
        </rhs>
    </production>
    <production>
        <lhs name="S1"/>
        <rhs>
            <symbol type="term" name="a"/>
            <symbol type="nonterm" name="b1"/>
            <symbol type="nonterm" name="S1"/>
        </rhs>
    </production>
    <production>
        <lhs name="S1"/>
        <rhs>
            <symbol type="term" name="b"/>
            <symbol type="nonterm" name="B1"/>
            <symbol type="nonterm" name="a1"/>
        </rhs>
    </production>
    <production>
        <lhs name="S1"/>
        <rhs>
            <symbol type="term" name="b"/>
            <symbol type="nonterm" name="a1"/>
            <symbol type="nonterm" name="S1"/>
        </rhs>
    </production>
    <production>
        <lhs name="S1"/>
        <rhs>
            <symbol type="term" name="a"/>
            <symbol type="nonterm" name="A1"/>
```

```
                <symbol type="nonterm" name="b1"/>
                <symbol type="nonterm" name="S1"/>
            </rhs>
        </production>
        <production>
            <lhs name="S1"/>
            <rhs>
                <symbol type="term" name="b"/>
                <symbol type="nonterm" name="B1"/>
                <symbol type="nonterm" name="a1"/>
                <symbol type="nonterm" name="S1"/>
            </rhs>
        </production>
        <production>
            <lhs name="a1"/>
            <rhs>
                <symbol type="term" name="a"/>
            </rhs>
        </production>
        <production>
            <lhs name="b1"/>
            <rhs>
                <symbol type="term" name="b"/>
            </rhs>
        </production>
    </productions>
    <startsymbol name="S"/>
</grammar>
```