

Ferramentas didáticas de análise léxica

Gustavo Scaloni Vendramini
Guilherme José Henrique
Sean Carlisto de Alvarenga
Vinícius Fernandes de Jesus

25 de agosto de 2013

<i>SUMÁRIO</i>	2
----------------	---

Sumário

1	resumo	3
2	C-gen	4
2.1	Considerações	7
3	Sintelo	9
3.1	Exemplo	9
3.2	Considerações	10

1 resumo

2 C-gen

A ferramenta C-gen foi desenvolvida como trabalho de conclusão de curso (TCC) por Jerônimo Backes, pela Universidade de Santa Cruz do Sul (UNISC) em 2005. O propósito da ferramenta é de auxiliar o estudo das três fases de construção de um compilador, sendo elas a análise léxica, análise sintática e análise semântica. Segundo [1], no curso de compiladores é comum a implementação de um pequeno compilador, ou partes deles, para aplicar o conhecimento teórico aprendido em sala de aula. Porém, em cada fase, podem ser aplicados diversos métodos bastante complexos, o que dificulta o aprendizado do aluno.

Existem ferramentas que auxiliam na construção de um compilador, mas a maioria delas são usadas em apenas uma fase do compilador e são destinadas ao uso profissional [1], sendo assim ferramentas nada didáticas e de fácil uso.

A ferramenta C-gen foi desenvolvida após a análise de características de três ferramentas profissionais, *Flex*¹, *Bison*² e *COCO/R*³, levando em conta seus pontos positivos e negativos sobre a ótica educacional.



Figura 1: A ferramenta C-gen.

¹<http://flex.sourceforge.net/>

²<http://www.gnu.org/software/bison/>

³<http://www.ssw.uni-linz.ac.at/coco/>

C-gen possui interface gráfica bastante fácil e intuitiva, não sendo necessário programar a entrada como em outras ferramentas, o que sem dúvidas facilita a interatividade com o aluno, e portanto, o aprendizado. Apesar da ferramenta realizar as fases de análise léxica, análise sintática e análise semântica, o escopo desse trabalho se limitou apenas na fase de análise léxica.

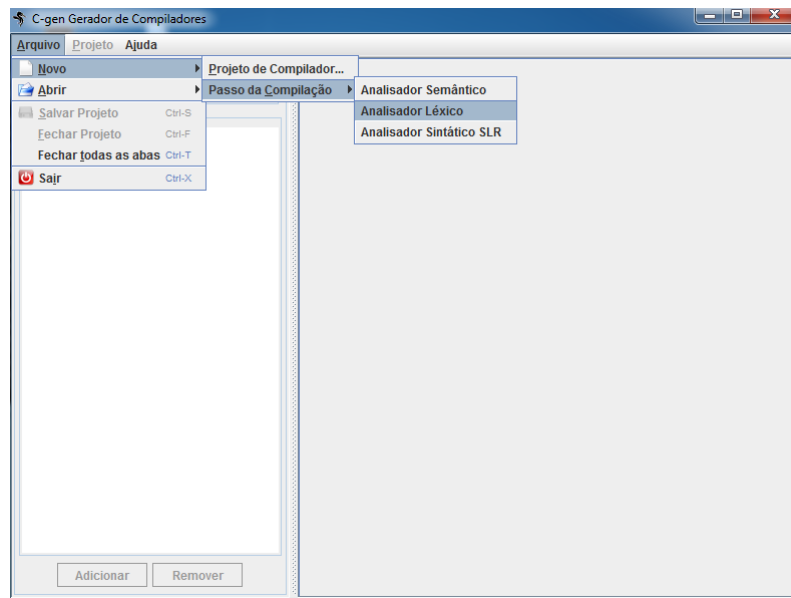


Figura 2: Criando um projeto para a análise léxica.

Após criar um projeto para a fase de análise léxica, a ferramenta abre um editor de autômatos como mostra a figura 4, que será utilizado para o reconhecimento dos *tokens*. O editor é de fácil uso e nele é configurado os estados, as transições entre eles, qual é o estado inicial, quais são os estados finais e quais são os *tokens* reconhecidos por cada estado final. Nas transições dos estados, é possível escolher quais caracteres serão consumidos, onde “>” representa “o que estiver entre”.

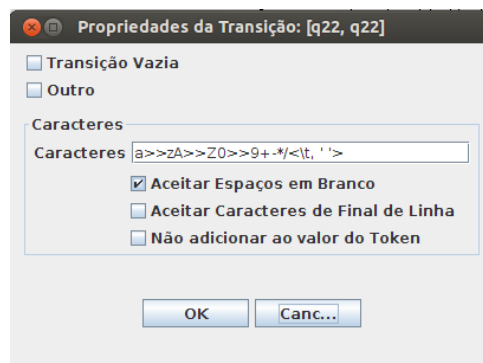


Figura 3: Exemplo de transição que consome qualquer caractere de a até z, A até Z, 0 até 9 entre outros.

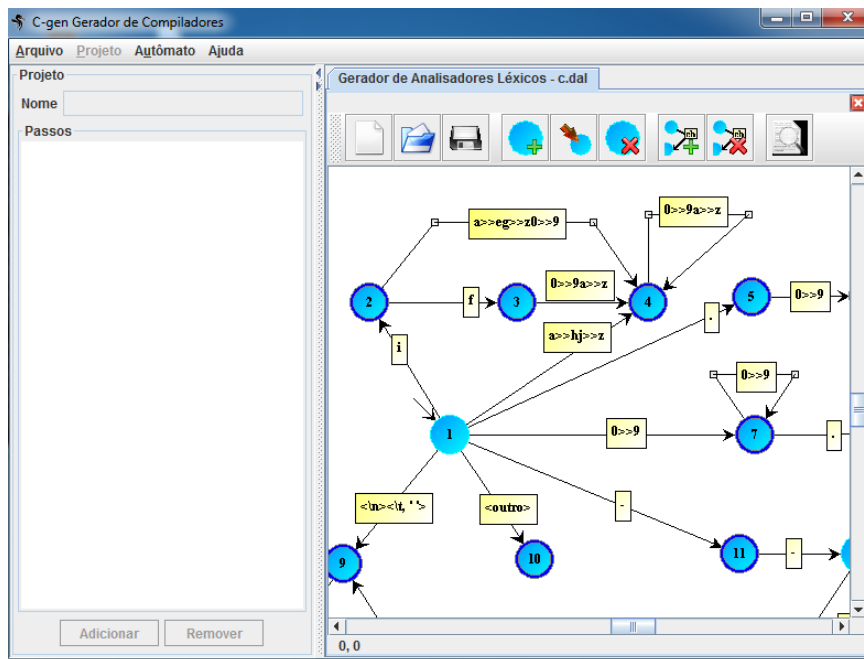


Figura 4: Editor de autômatos da ferramenta C-gen.

A ferramenta também possui opções para inserir as palavras reservadas da linguagem como mostra a figura 5.

Palavras Reservadas			
Lexema	Nome do identific...	Valor	Classe
End	End	String	
File	File	String	
For	Foward	String	
Goto	Goto	String	
In	In	String	
Label	Label	String	
and	and	String	
array	array	String	
asm	asm	String	
begin	begin	String	
byte	byte	String	
case	case	String	
comp	comp	String	
const	const	String	
constructor	constructor	String	
destructor	destructor	String	

Buttons: Adici..., Remo..., Classes, OK, Canc...

Figura 5: Inclusão de palavras reservadas.

Ao término de toda configuração, é executado a análise léxica para o reconhecimento dos *tokens*. Nessa parte, é possível visualizar a matriz de transição dos estados como mostra a figura 6. Nesse momento, a ferramenta possui diversas opções como executar a análise caractere a caractere, ou por *token* e até por um intervalo de tempo predeterminado.

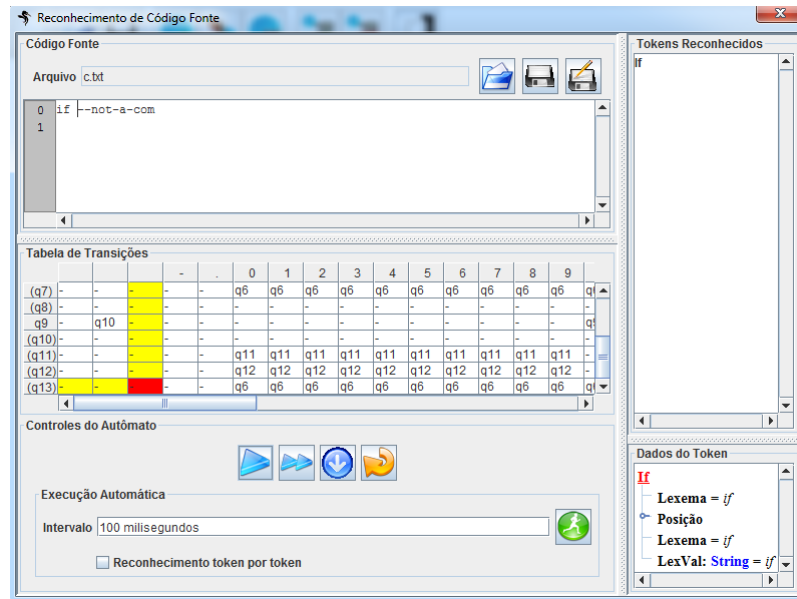


Figura 6: Matriz de transições.

2.1 Considerações

Tabela 1: Considerações de algumas características das ferramentas

Ferramenta	Design da interface com o usuário	Facilidade de uso da ferramenta	Interação com o usuário	Facilidade de aplicação da fundamentação teórica	Capacidade de aplicação das ramificações teóricas	Relação entre uso e aprendizado
C-gen	Bom	Bom	Bom	Bom	—	Bom

A ferramenta C-gen, na parte de análise léxica (o escopo desse trabalho) é uma ferramenta de fácil uso, onde não é necessário programar a entrada, pois possui um editor de autômato junto a ferramenta. Sua interface gráfica é intuitiva e amigável ao usuário e possui diversas opções para visualizar o reconhecimento dos *tokens*, mostrando passo a passo o que está acontecendo na análise. A ferramenta porém, possui um ponto negativo ao entrar no processo de reconhecimento. O autômato de entrada é convertido para um autômato finito determinístico, onde os identificadores dos estados no autômato mudam após a conversão apenas na matriz de transição. Isso, dificulta acompanhar o reconhecimento pela matriz de transições junto ao autômato. Além disso, caso a conversão seja executada mesmo que o autômato já seja determinístico, o autômato resultante da conversão pode apresentar configurações inúteis. A ferramenta possui fácil aplicação da fundamentação teórica uma vez que possui opções para a análise léxica, sintática e semântica. Como a parte de análise léxica não possui diversas abordagens (como por exemplo, na análise sintática,

abordagens *bottom-up* e *top-down*), sobre o aspecto da capacidade de aplicação das ramificações teóricas não se pode dar um julgamento. Por fim, a ferramenta C-gen possui um manual que explica qual a função de cada passo do compilador, com exemplos de como utilizar a ferramenta, o que mesmo para um aluno que não domina o conteúdo teórico em sua totalidade, ainda assim pode utilizar a ferramenta com certa facilidade.

3 Sintelo

A ferramenta sintelo é uma ferramenta didática para o ensino de compiladores. A mesma foi criada por Karina Kieling Dos Santos e Philipe Marcon Dos Reis, no ano de 2008 como Trabalho de Conclusão de Curso pela Universidade do Sul de Santa Catarina e foi publicada em [2].

Embora a ferramenta em questão possua mais funcionalidades, esse trabalho se limitou a parte de análise léxica, permanecendo em seu escopo.

Para que possamos realizar a análise léxica, primeiramente devemos informar ao sintelo os tokens de nossa linguagem, através de expressões regulares. Essas expressões regulares devem ser informadas na “caixa” chamada léxico, conforme ilustrado na figura 7.

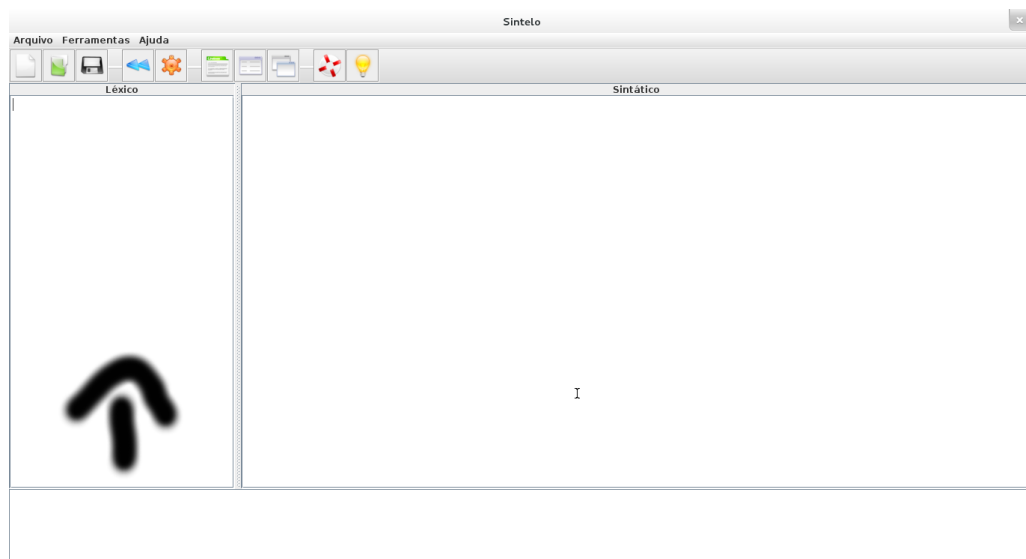


Figura 7: Tela inicial do sintelo.

A notação utilizada para representar as expressões regulares são descritas em [2] ou no manual da ferramenta, que pode ser acessado através do menu “Ajuda”, item “Manual” e seção “Especificações Léxicas”. Desta forma este artigo não explicará a notação.

3.1 Exemplo

Para exibir a análise léxica, o exemplo de operações aritméticas de [3] será adaptado. Esse exemplo reconhece operações aritméticas que podem conter números (inteiros e/ou reais), identificadores (variáveis), operações binárias e também parêntesis; ignorando espaços em branco, quebra de linha e tabulação.

A seguir, a especificação dos tokens que deve ser passada para o sintelo:

```
ID: [a-zA-Z_] [_a-zA-Z0-9]*
INT: [0-9]+
REAL: ([0-9]\.[0-9]+) | ([0-9]+\.[0-9])
" ("
```

```
"")
binop: "+" | \- | \* | "/"
:[\s\n\t]+
```

Com os tokens da linguagem especificados, basta realizar a simulação da análise léxica. Para fazer isso, primeiramente vá ao menu “Ferramentas”, item “Léxico”, opção “Simular”.

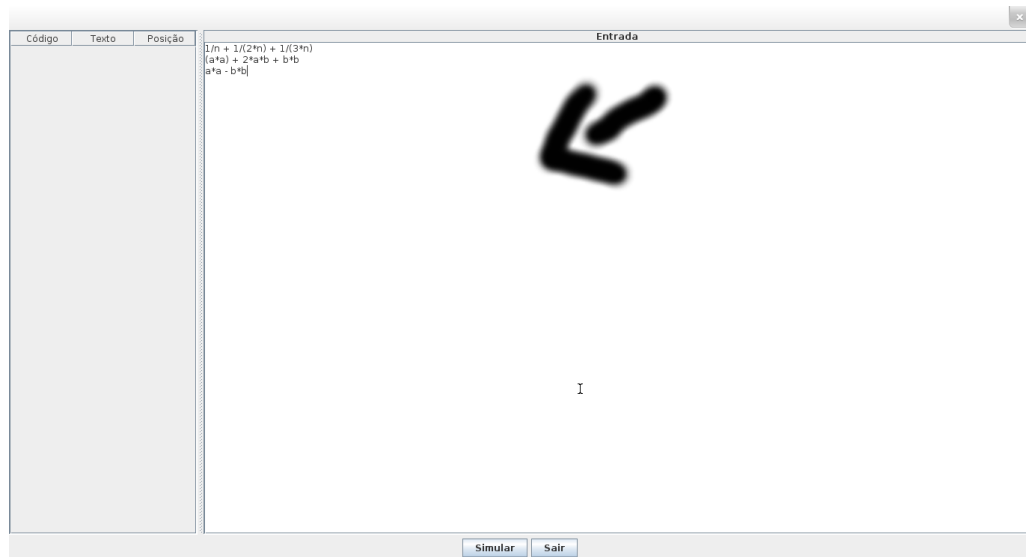


Figura 8: Tela inicial da simulação da análise léxica.

A seguir, é necessário informar a entrada da linguagem, no local informado pela figura 8. Como exemplo de entrada para o analisador léxico, temos a seguinte:

```
1/n + 1/(2*n) + 1/(3*n)
(a*a) + 2*a*b + b*b
a*a - b*b
```

Para efetuar a simulação basta clicar no botão “Simular”. A figura 9 exibe o resultado da simulação. Note que a ferramenta exibe os tokens reconhecidos e seus respectivos lugares no código de entrada.

Agora considere a seguinte entrada:

```
a^a - b*b
```

Como a operação $^$ não está definida na linguagem em questão, um erro será obtido com a simulação dessa entrada, conforme ilustrado na figura 10.

No sintelo, também podemos visualizar o autômato gerado a partir das expressões regulares. Basta ir ao menu “Ferramentas”, item “Léxico”, opção “Tabela de Transição”. A figura 11 ilustra o automato do exemplo utilizado.

3.2 Considerações

O sintelo...

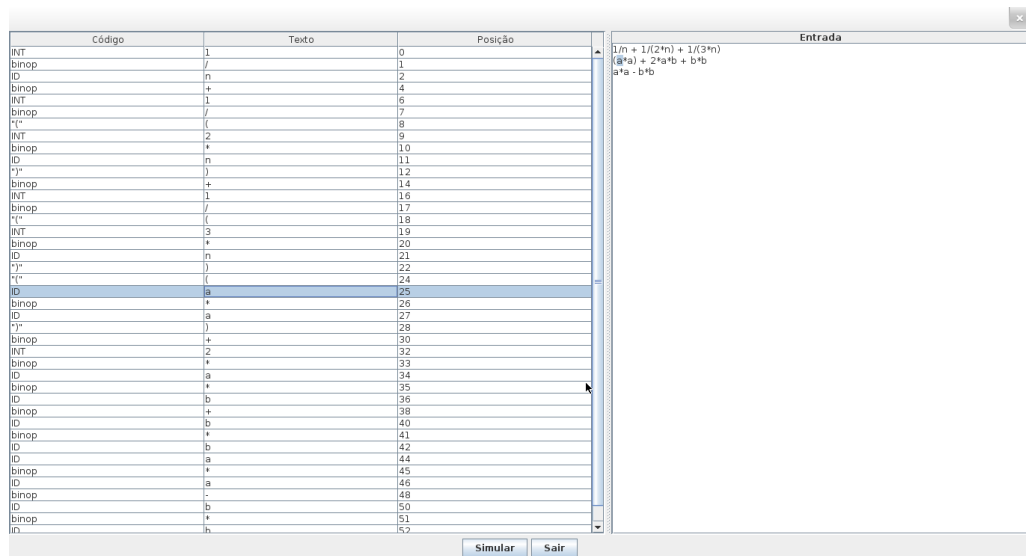


Figura 9: Tela após a simulação da análise léxica.

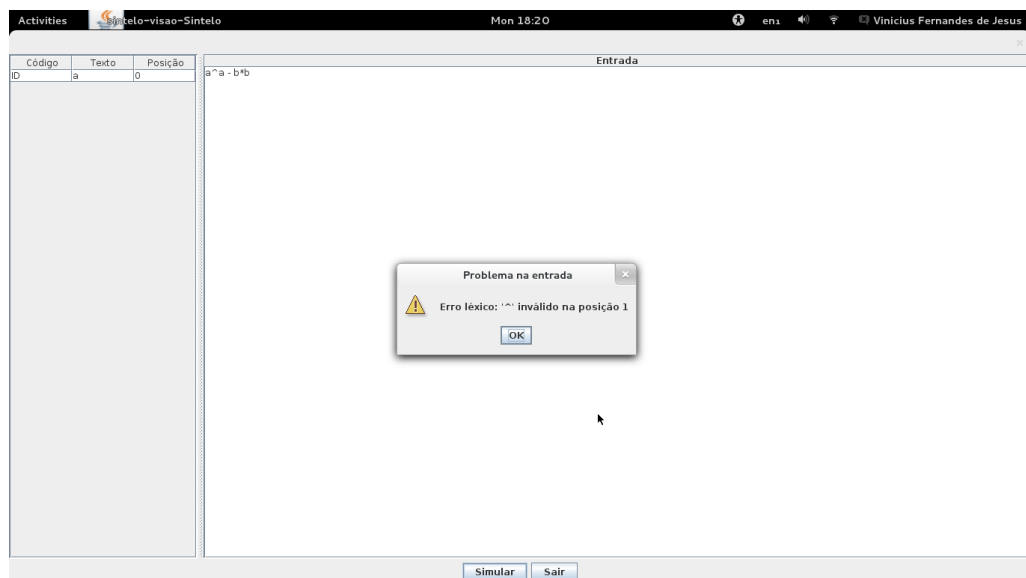


Figura 10: Tela de erro léxico.

Tabela de transição																			
	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q		
q0	q1	q1	q1	q1	q1	q1	q1	q1	q1	q1	q1	q1	q1	q1	q1	q1	q1	q1	q1
q1(1)	q1	q1	q1	q1	q1	q1	q1	q1	q1	q1	q1	q1	q1	q1	q1	q1	q1	q1	q1
q2(3)																			
q3																			
q4(3)																			
q5(4)																			
q6(5)																			
q7(6)																			
q8(7)																			
q9(8)																			
q10(2)																			
q11(2)																			
q12																			

Figura 11: Automato do exemplo em questão.

Referências

- [1] Jerônimo Backes. *Ferramenta de Apoio ao Estudo de Compiladores*. Universidade de Santa Cruz do Sul, 2006.
- [2] Karina Kieling Dos Santos e Philipe Marcon Dos Reis. *Ambiente de Apoio ao Ensino de Análise Sintática em Disciplinas de Compiladores*. Universidade Do Sul De Santa Catarina, 2008.
- [3] Robert W. Sebesta. *Concepts of Programming Languages (10th Edition)*. Addison-Wesley, 10 edition, 1 2012.