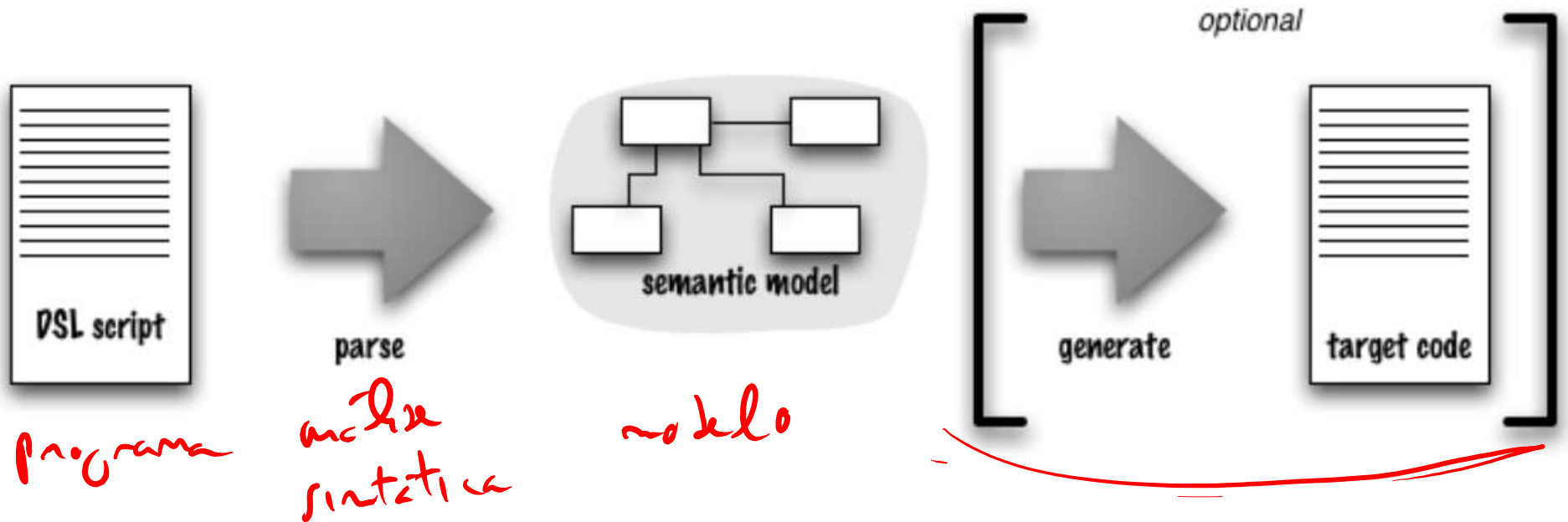


Linguagens de Domínio Específico

Fabio Mascarenhas – 2017.1

<http://www.dcc.ufrj.br/~fabiom/dsl>

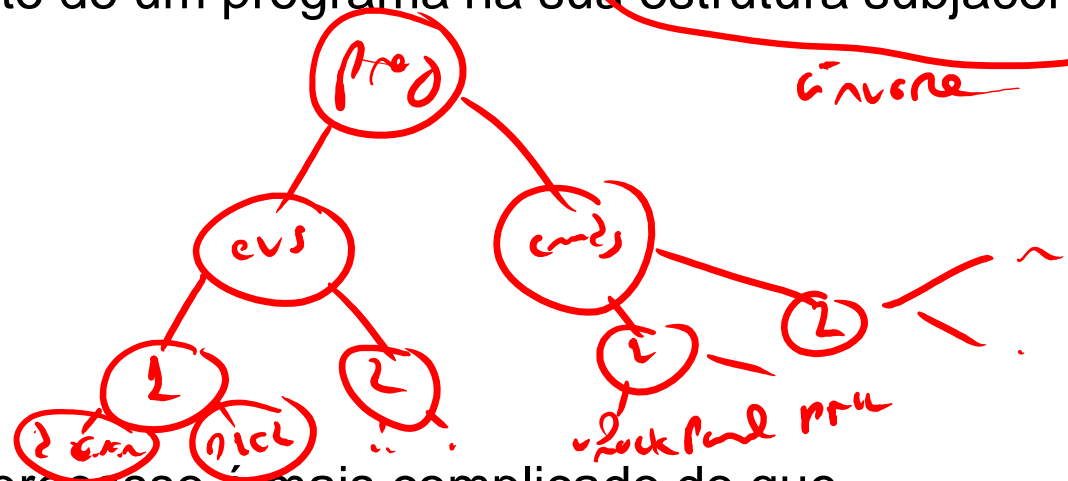
Processamento de uma DSL



Análise sintática

- A análise sintática quebra o texto de um programa na sua estrutura subjacente

```
events
  doorClosed D1CL
  drawerOpened D2OP
end
commands
  unlockPanel PNUL
  lockPanel PNLK
end
```



- Em qualquer linguagem real o processo é mais complicado do que simplesmente quebrar em linhas e palavras e tentar interpretá-las de maneira *ad-hoc*
- O primeiro passo para criar um analisador sintático é definir uma gramática formal para a nossa linguagem

Gramáticas

- Uma gramática é uma descrição formal de como um texto se transforma em uma árvore sintática
- Basicamente é um conjunto de *regras de produção*, onde cada regra possui um termo sintático no lado esquerdo e um modelo de sentença para esse termo no lado direito (formado por outros termos sintáticos)

additionStatement := number ' + ' number

non-terminal / variáveis / termos sintáticos

- Vamos usar uma notação parecida com a de expressões regulares para definir uma gramática, com operadores que mostrados no próximo slide
- Termos entre aspas e termos que não aparecem no lado esquerdo de uma regra são terminais ou tokens *number*

Opcional e repetição

... actions! - ~ ~ ~ ~ ~
~ ~ ~ ~ ~ actions ~ ~

- Em uma gramática, o operador ? indica que o termo imediatamente anterior é opcional

$termo? \rightarrow \langle vazio \rangle$
 $termo? \rightarrow termo$

- O operador * indica que o termo imediatamente anterior pode ser repetido quantas vezes quiser (até mesmo nenhuma), enquanto + exige ao menos uma repetição

(
 $termo^* \rightarrow \langle vazio \rangle$
 $termo^* \rightarrow termo\ termo^*$
 $termo^+ \rightarrow termo\ termo^*$

$termo^* \rightarrow \underbrace{termo \dots termo}_{n \text{ vezes (inclusive } n=0)}$

Escolha

- Também temos uma operação de escolha $|$, que indica que podemos usar tanto a sequência de termos à esquerda do operador quanto a sequência à direita do operador

$termo1 | termo2 \rightarrow termo1$
 $termo1 | termo2 \rightarrow termo2$

$t7 | t4 | t5$

- Podemos também usar parênteses para mudar a forma como esses operadores se associam aos termos
- A precedência normal é $|$, justaposição, operadores unários ($?$, $*$, $+$)

$+$ \times

Gramática da DSL de máquina de estados

```
machine      := events revents? commands state+
events       := "events" event+ "end"
event        := nome codigo
commands    := "commands" command+ "end"
command      := nome codigo
revents     := "resetEvents" nome+ "end"
state       := "state" nome actions? transition* "end"
actions     := "actions" "{" nome+ "}"
transition  := nome "=>" nome
```

↑
non-terminals

terminal