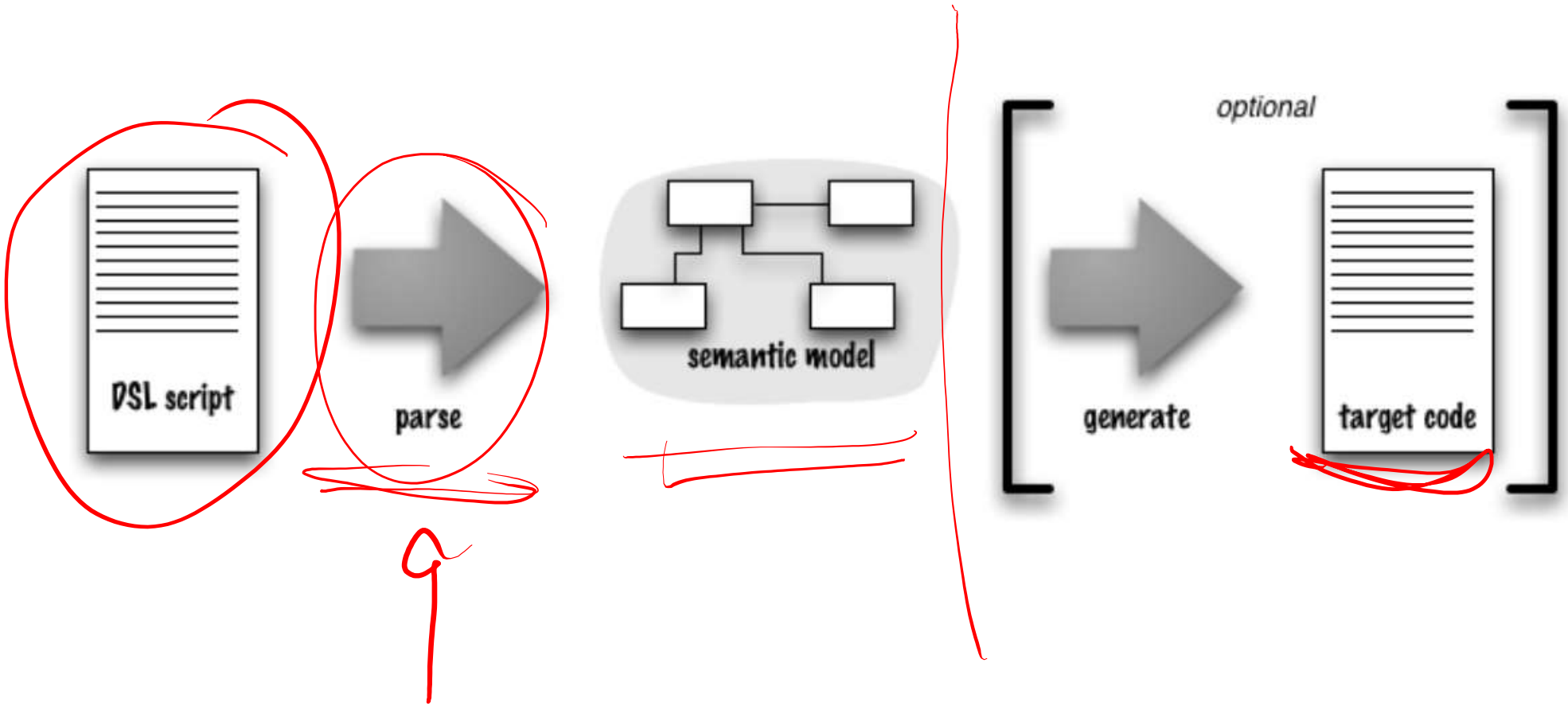
Linguagens de Domínio Específico

Fabio Mascarenhas – 2016.1

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

Processamento de uma DSL



ASTs heterogêneas

- Outra forma de implementar uma AST é usar uma classe diferente para cada tipo de nó que temos

- Classes abstratas ou interfaces implementam estruturas sintáticas que possuem diversos tipos concretos de nó

```
public abstract class AST {
    Token token;
    // missing normalized list of children; subclasses define fields
}
public abstract class ExprNode extends AST { ... }
public class AddNode extends AST {
    ExprNode left, right;          // irregular, named fields
    «fields-specific-to-AddNode»
}
```

- Se estamos escrevendo o código para criar e processar as árvores manualmente essa forma é a ideal

Construindo ASTs (1)

- Construir uma AST heterogênea é mais ad-hoc
- Em um analisador recursivo, cada regra pode retornar o pedaço da AST que ela representa
- O corpo de cada regra compõe os pedaços do jeito que quiser

```
public Exp exp() {
  Exp res = termo();
  while(la.tipo == '+' || la.tipo == '-') {
    if(la.tipo == '+') {
      match('+');
      res = new Soma(res, termo());
    } else {
      match('-');
      res = new Sub(res, termo());
    }
  }
  return res;
}
```

```
public List<Event> events() {
  ArrayList<Event> res =
    new ArrayList<>();
  match(StateMachineLexer.EVENTS);
  do {
    res.add(event());
  } while(la.tipo ==
          StateMachineLexer.NAME);
  match(StateMachineLexer.END);
  return res;
}
```

Construindo ASTs (2)

- Com combinadores, precisamos fazer os combinadores poderem retornar um pedaço da AST, além do sufixo

- Com tipos genéricos podemos manter combinadores bem genéricos, mas que constroem ASTs heterogêneas

```
public interface Parser<A,B> {
  Result<A,B> parse(State<B> in);
}
```

```
public class Result<A,B> {
  public final A res;
  public final State<B> out;
  ...
}
```

- O combinador de sequência fica mais complicado: precisamos passar cada sequência através de uma função responsável por combinar os pedaços da sequência