

A Strafunski Application Letter

From: Ralf Lammel, Joost Visser

O que é o Srafunski?

Pacote de software baseado em Haskell para implementação de componentes para processamento de linguagens, principalmente análise e transformações.

- Biblioteca para *generic programming*
- integração de componentes externos (parsers, pretty printers, etc..).

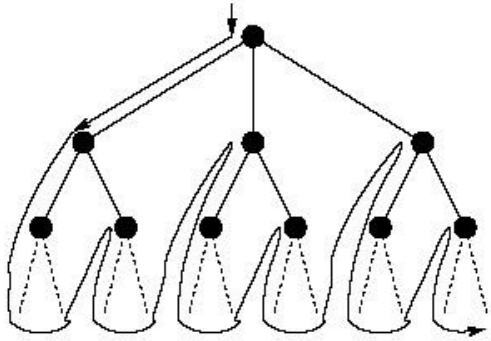
Generic Traversall (Travessia genérica)

As travessias são baseadas em estratégias, definidas através de combinações de funções.

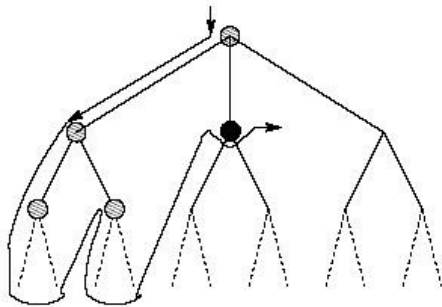
As travessias comportam-se como uma mistura de comportamento genérico e comportamento dependente do tipo de dados.

- StrategyLib – combinadores de estratégias: adhocTP, adhocTU, allTU, etc...
- DrIFT – gerador de instancias. Recebe um tipo de dados e gera instâncias para esse tipo de dados, das classes Term, ATerm e Xml.

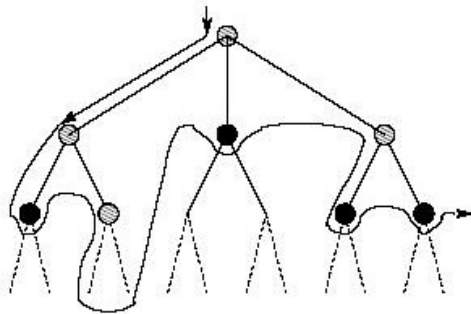
Travesias Genéricas



full_td: processa todos os nodos da árvore, de cima para baixo, da esquerda para a direita.



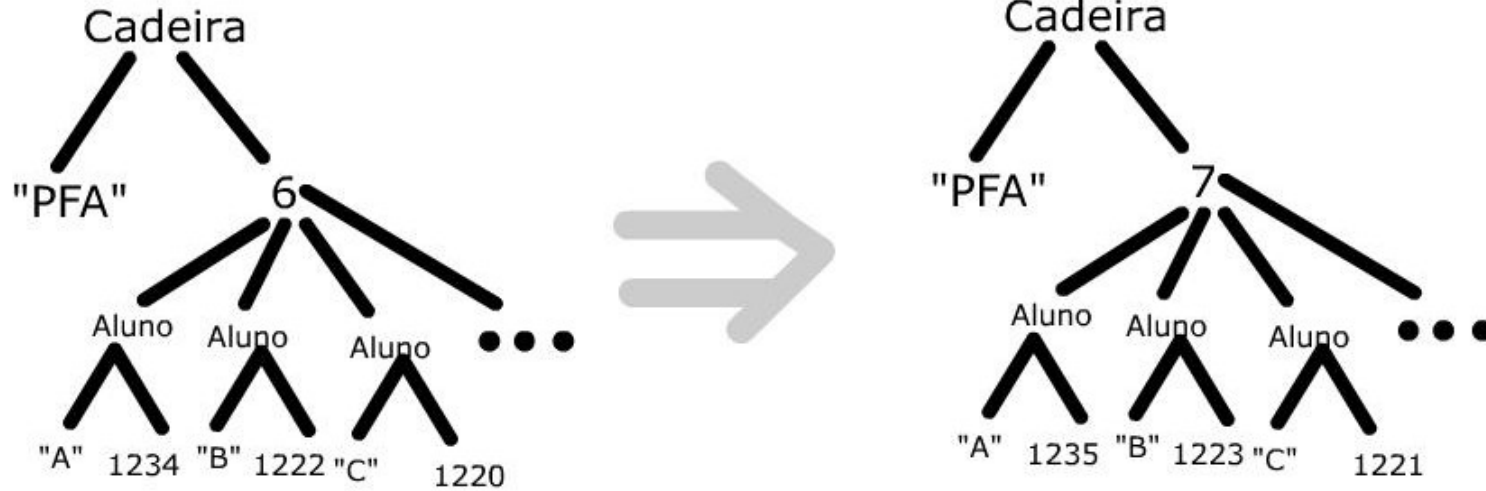
once_td: processa até obter sucesso pela primeira vez.



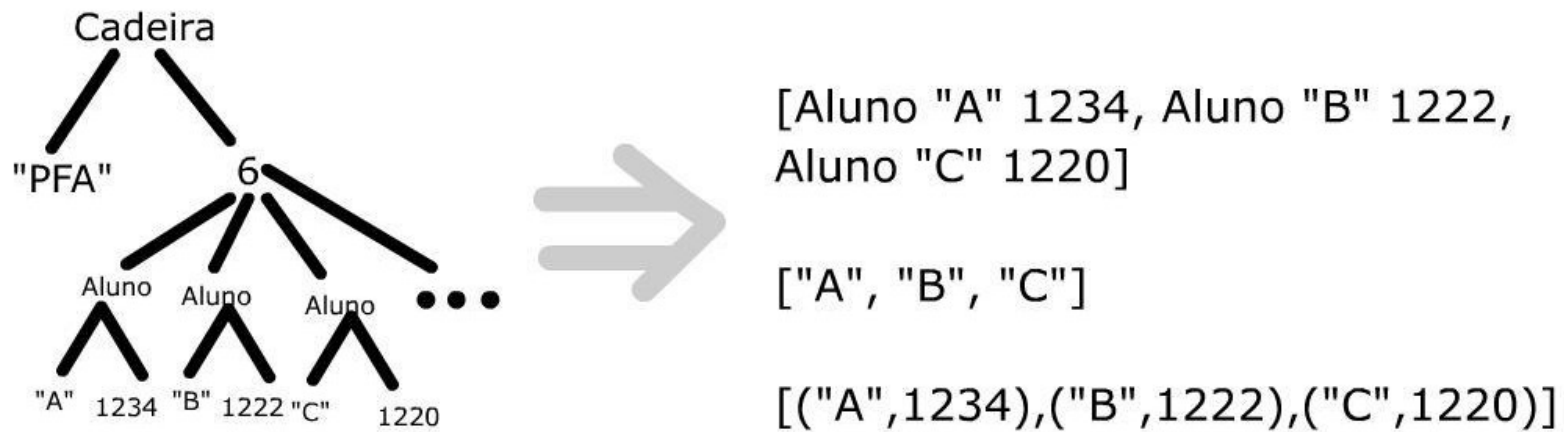
stop_td: processa todas as sub-árvores apenas até obter sucesso.

Tipos de estratégias

Type Preserving (TP)

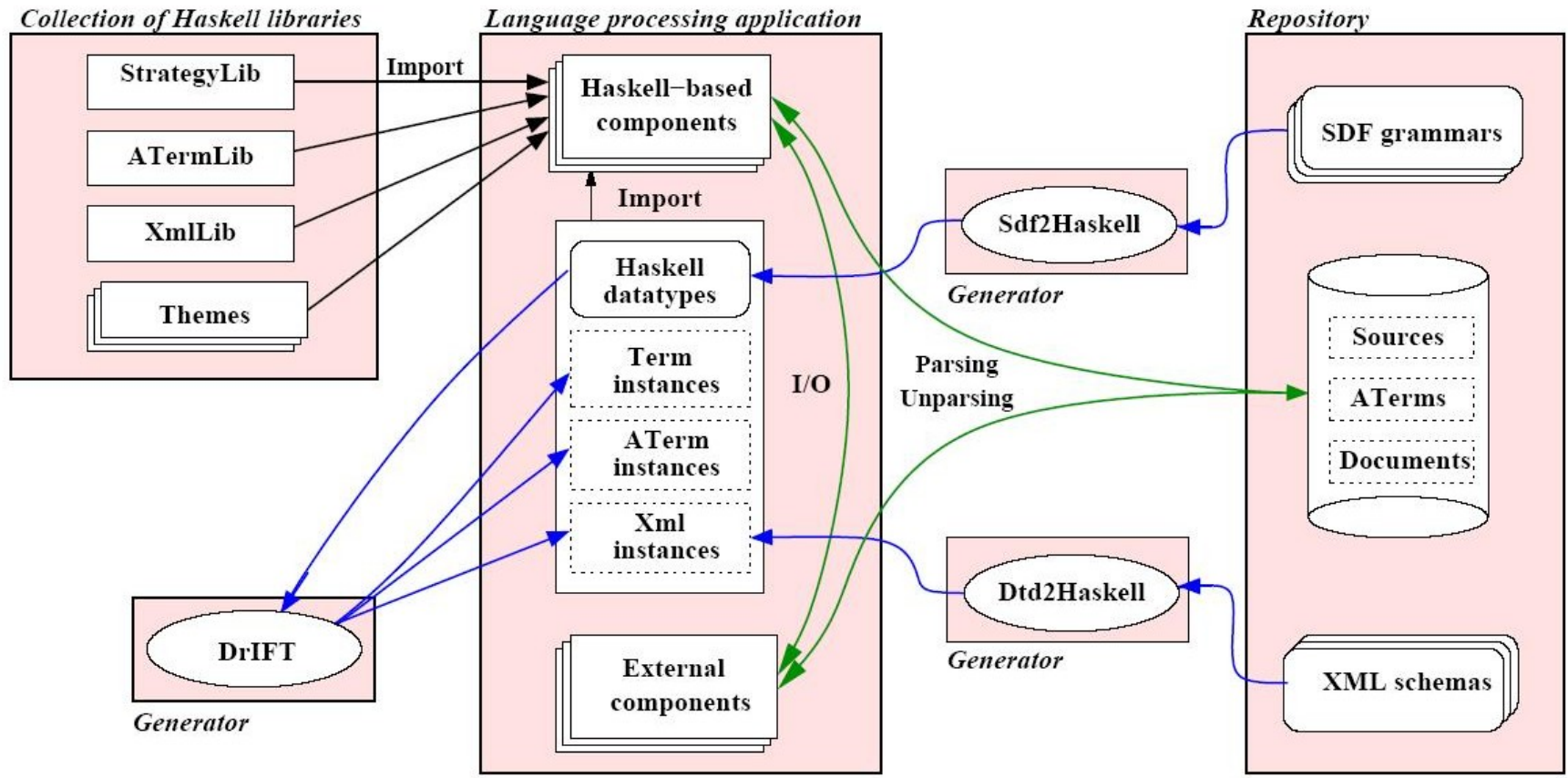


Type Unifying (TU)



Componentes externos

- Formatos de intercâmbio
 - XML, suportado pela biblioteca *XmlLib* do HaXml – intercâmbio de dados.
 - ATerm, suportado pela biblioteca *ATermLib* – *intercâmbio de tipos de dados*.
- Integração de Parsers
 - Através do suporte a gramáticas SDF.



Aplicações práticas (1)

- COBOL reverse engineering
 - Cálculo do grafo de PERFORMs:
 - Objectivo: encontrar todos os PERFORM e construir grafo
 - Gramática do COBOL definido em SDF
 - Derivação em Haskell dos tipos algébricos através do *SDFtoHaskell*.
 - findPerform definido com a estratégia fulltd_TU.
 - Visualização através da integração de componentes externos: dot.

Aplicações práticas(2)

- Java code metrics
 - Objectivo: Contagem de *statements*, *números de condições e condições aninhadas* e *exportação dos resultados em XML*.
 - *Gramática do java definida em SDF*.
 - Vários combinadores de estratégias para calcular as várias métricas.
 - Métricas Java definidas por DTD, importadas para o Haskell com o *Dtd2Haskell*.
 - Estratégias Type Unifying, devolvem tipos de dados: Métricas Java.
 - Exportação para XML através do HaXml.

Aplicações práticas(3)

- Haskell re-engineering
 - Eliminação de declarações em módulos.
 - Problema de transformação (e não de análise, como os anteriores).
 - Por exemplo, eliminações de cláusulas *where* inutilizadas.