

Análise de Folhas de Cálculo em Haskell

Cupertino Miranda

Programação Funcional Avançada

Introdução

- Conversão de formatos
- Estrutura de dados
- Grafo de fluxo de dados
- Slicing e chopping
- Inferência de tipos

Conversão de Formatos

- Gnumeric (ssconvert)
- Vantagens
 - Open-Source
 - Exportação para XML (Schema conhecido)
- Desvantagens
 - Complexidade da estrutura XML
 - DTD para XML não conhecido
 - foi necessária conversão de Schema para DTD
 - correcções a alguns erros de geração

Estrutura de Dados

- Preocupações
 - Adaptativa
 - Totalidade
- Ferramentas Utilizadas
 - Dtd2Haskell
 - DrIFT
- Bibliotecas Utilizadas
 - HaXML

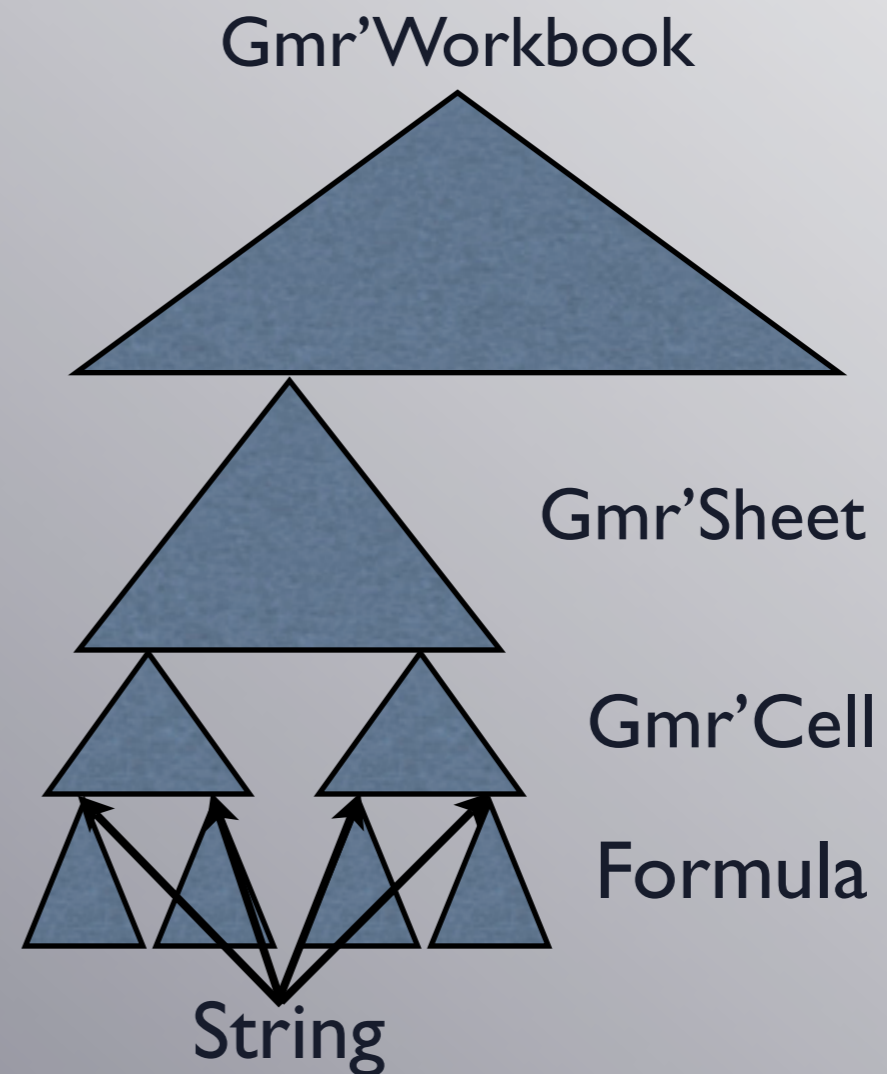
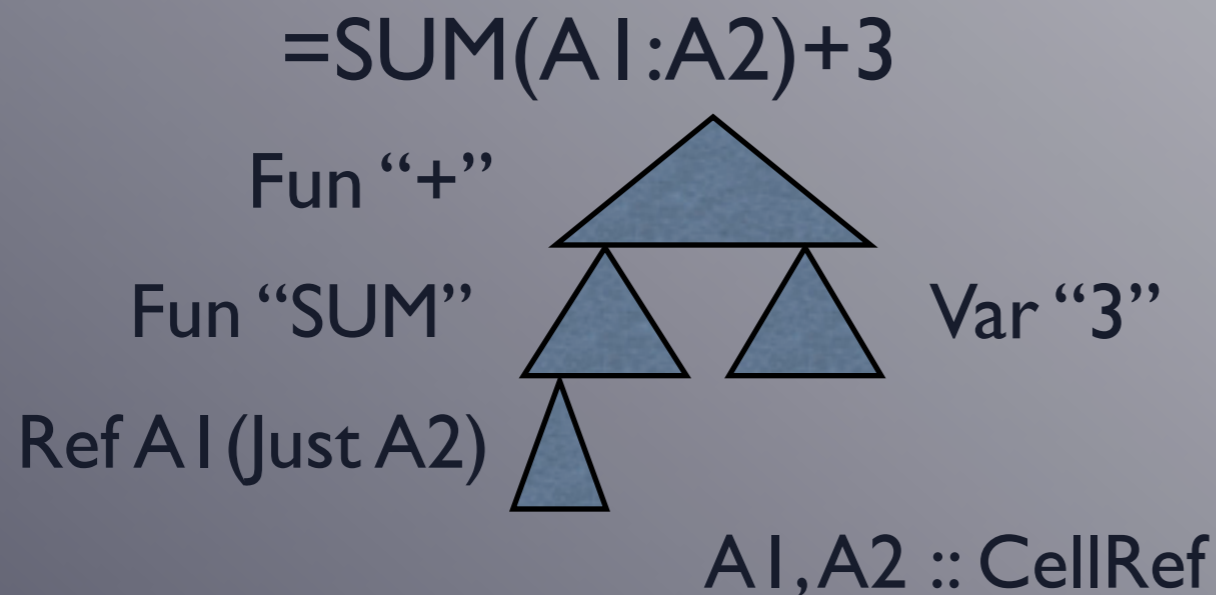
Estrutura de Dados

- `Syntax.hs`
 - Gerado pelo `Dtd2Haskell`
 - Necessário contornar alguns problemas
- `SyntaxTermInstances.hs`
 - Gerado pelo `DrIFT`
 - Cria instâncias para `Term` e `Typeable`

Estrutura de Dados

Exemplo Minimalista

- Conteúdo da Célula
- String não desejada
- Parsing para Formula



Estrutura de Dados

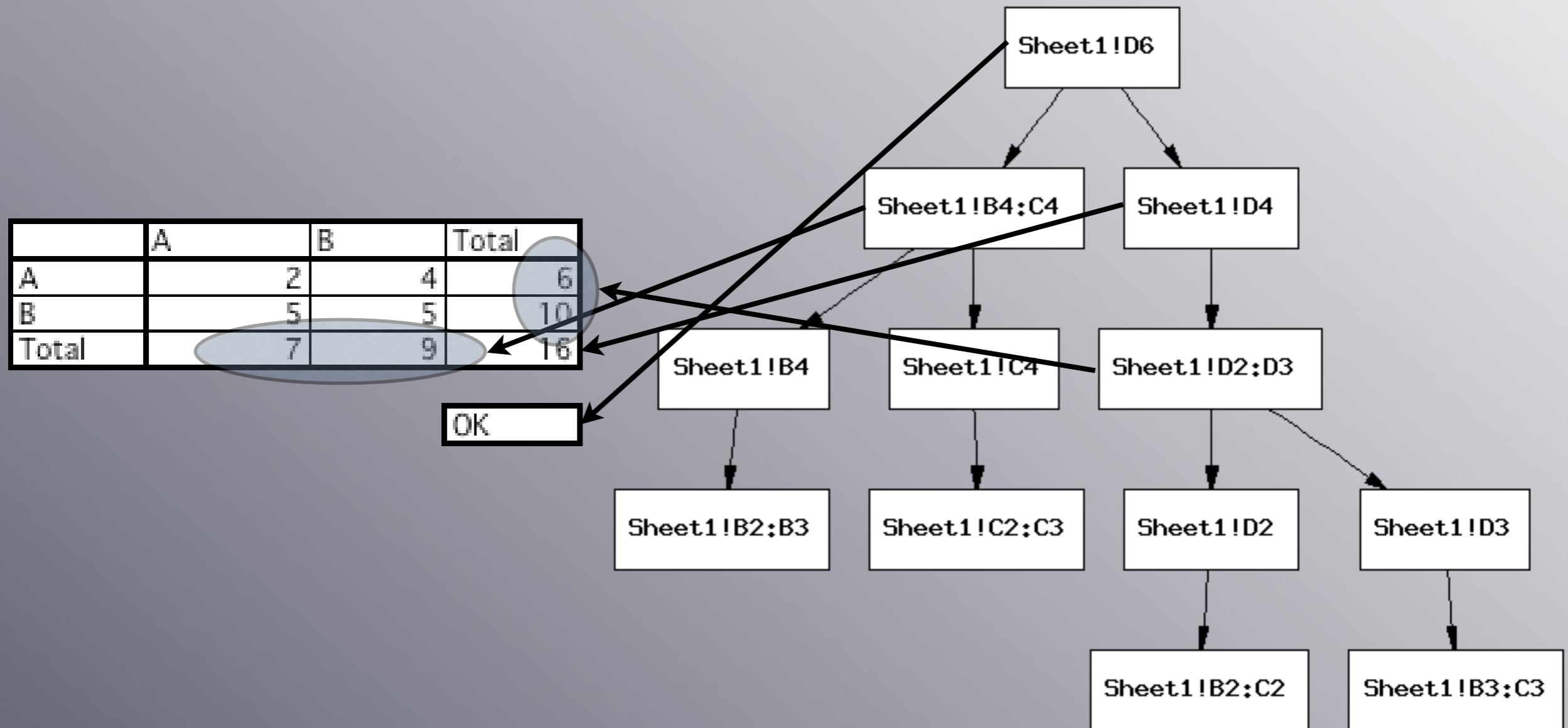
- Resultados Obtidos
 - Eficácia 100% a todos os ficheiros usados
 - Com certeza que existem erros
- Testes Realizados
 - `SyntaxHUnit.hs`
 - Testa todos os ficheiros de uma directoria

Grafo de Fluxo de Dados

- Validação de Spreadsheets
- Gerado a partir das formulas.
 - Cada referência gera um novo arco do grafo
 - Fornecido por função genérica
- Criou-se dois tipos de grafos
 - Normal
 - Expandido - utilizado para efectuar slicing
 - Conseguído através de funções 'collector'

Grafo de Fluxo de Dados

Exemplo de um Grafo



Slicing e Chopping

- Igualmente útil para validação
- Slicing
 - Forward - conjunto de células que produzem um resultado
 - Backward - conjunto de células com que contribui com o seu valor
- Chopping - intersecção entre Forward e Backward slice

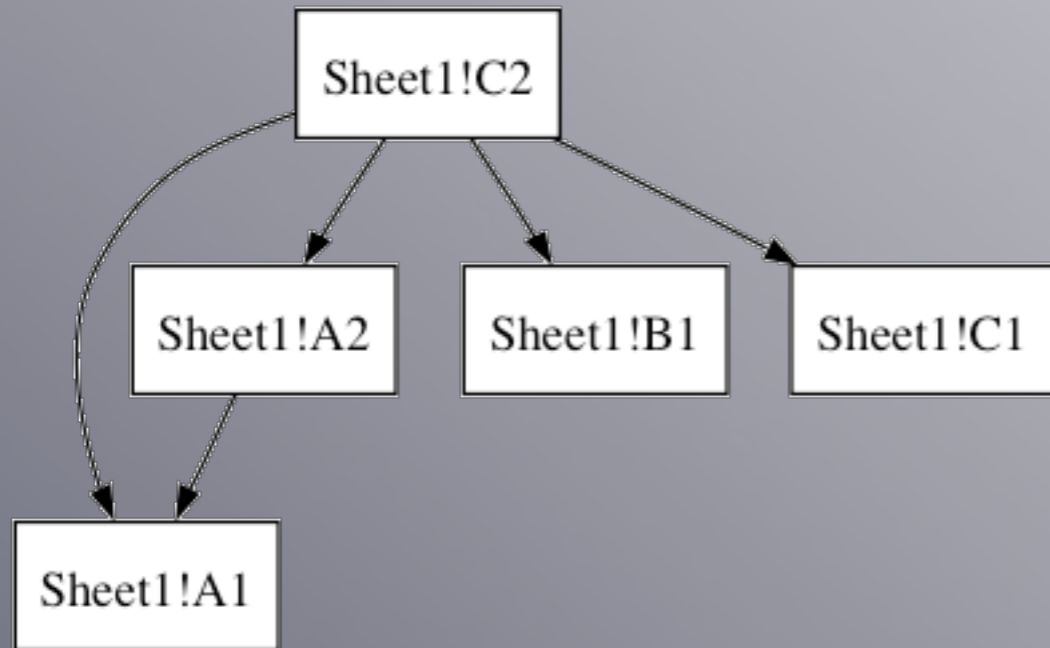
Slicing e Choping

- Filter.hs
 - capacidade de filtrar a estrutura (Gmr'Workbook)
 - utilizado para aplicar slices e chops
 - não adequada para Choping
- função sliceOrChopWith
 - permite criar conjunto de células envolvidas
 - necessita células sources/sinks e grafo expandido

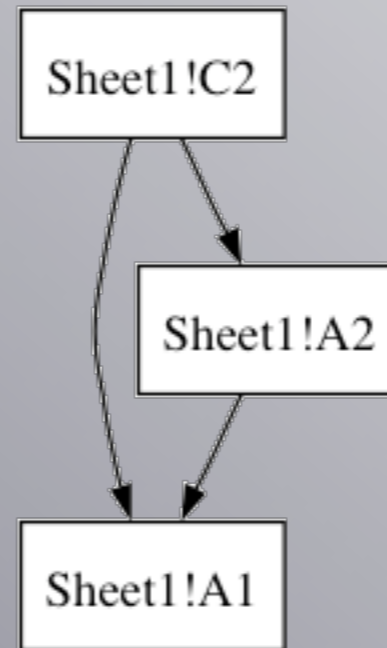
Slicing e Chopping

Exemplo

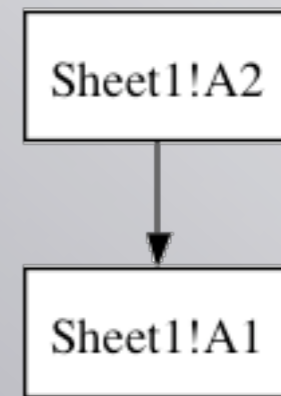
Dataflow



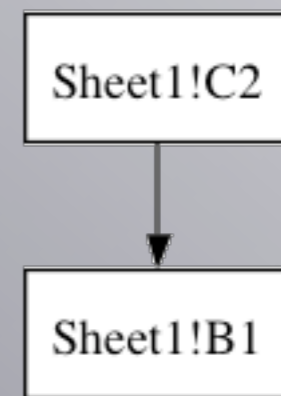
Chop C2-A1



Forward Slice A2



Backward Slice A2



Inferência de Tipos

- É construída com o modulo
 - `Data.Relation.TypeInference`
- É possível 3 tipos de relações:
 - Expressão
 - Array
 - Atribuição
- É possível verificação de erros através deste método

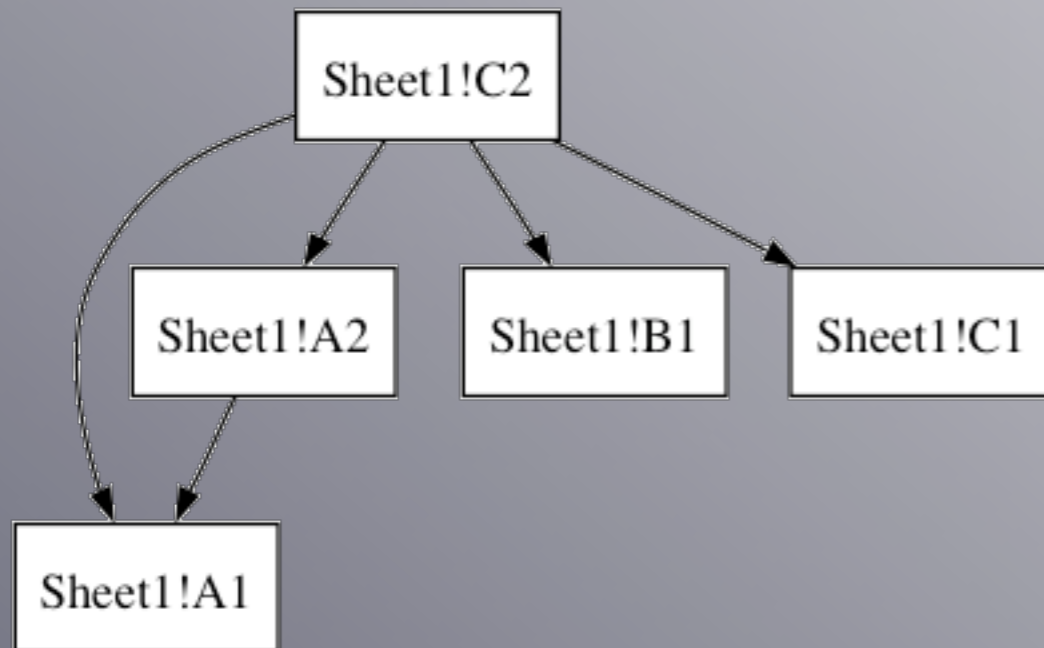
Inferência de Tipos

- Gerado a partir das formulas.
 - Cada função gera ou não uma nova relação
 - Fornecido por função genérica
 - O tipo de relação a calcular é definido por função de inferência
 - Ainda só definida função ‘inferExpression’
- Neste momento:
 - Permite verificar relações de igualdade entre tipos

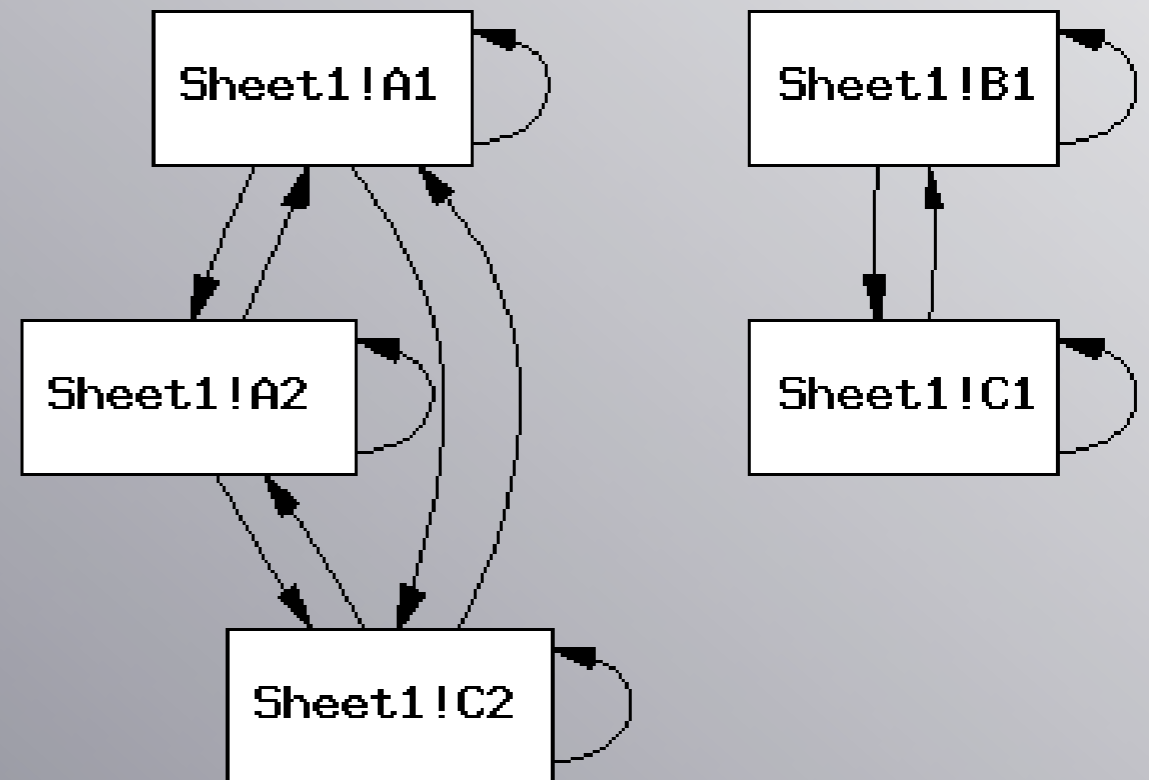
Inferência de Tipos

Exemplo

Dataflow



Type Inference Graph



$C2 = IF(B1 = SUM(C1), A1, A2)$
 $A2 = A1$

Demonstração

Trabalho Futuro

1. Corrigir falhas existentes
2. Iluminar chopping/slicing em vez de filtrar
3. Adicionar relações do tipo:
 - Array
 - Atribuição
4. Conseguir inferência de tipos mistos:
 - Com as funções (*, /, ... etc)
5. Tentar unir toda esta informação

Conclusão

- Incompleto mas de fácil extensão
- Filtragem com Backward Slicing e Chopping não produz bons resultados
- Grafos gerados produzem bons resultados para pequenos exemplos
- Simplifica correcção de erros