Capítulo 3

Álgebras e Especificação Equacional

3.1 Conceitos Básicos

Para tentar descrever um qualquer objecto é necessário poder descrever a sua forma, assim como as suas propriedades intrínsecas. Além disso se esse objecto cumpre um dado fim é necessário ser capaz de descrever a sua operacionalidade.

Em termos matemáticos vamos enveredar por uma aproximação algébrica, descrevendo "a forma dos objecto", a sua componente sintáctica, através de *assinaturas*, conjuntos de símbolos descrevendo as espécies, as constantes e as operações. A "operacionalidade dos objectos", a sua semântica, vai ser descrita através de álgebras.

Sem ainda definir formalmente estes conceitos, temos que:

- *Assinaturas*, conjunto de espécies, e conjuntos de símbolos de constantes e símbolos de operações.

- *Álgebras*, (para uma dada assinatura), um conjunto de elementos para cada espécie, um elemento (constante) para cada símbolo de constante, uma operação (função) para cada símbolo de operação.

Os símbolos de operação e de constante podem ser combinados de forma a dar origem a *termos*.

Mas se já temos as ferramentas que nos permitem descrever a "forma dos objectos", ou seja as assinaturas, falta-nos ainda uma forma de descrever as propriedades intrínsecas dos objectos. Tal descrição vai tomar a forma de um conjunto de equações, obtêm-se assim uma *especificação algébrica equacional*.

As álgebras que satisfazem as equações são as álgebras que verificam a especificação.

Antes de começar a definir formalmente todos estes conceitos vejamos alguns exemplos.

3.2 Exemplos

Um semi-grupo $(A, *_A)$, consiste num conjunto da base A, e uma operação binária $*_A : A \times A \rightarrow A$ que é associativa, isto é verifica-se que $(a_1 *_A a_2) *_A a_3 = a_1 *_A (a_2 *_A a_3)$, para quaisquer a_1, a_2, e a_3.

Como exemplos de semi-grupos temos:

$$(\mathbb{N}, +_{\mathbb{N}}), \quad (\mathbb{N}, *_{\mathbb{N}}), \quad (\mathbb{Z}, +_{\mathbb{Z}}), \quad (\mathbb{Z}, *_{\mathbb{Z}})$$

outro exemplo é dado pelo semi-grupo livre

$$(A^+, \cdot_{A^+})$$

de todas as sequências não vazias de caracteres de um dado alfabeto A com a operação de concatenação.

Temos então duas entidades a definir/compreender.

- representação semântica de um semi-grupo;
- representação (finita) sintáctica de um semi-grupo.

Comecemos por analisar em primeiro lugar a questão da representação sintáctica.

em vez do conjunto A, da operação binária $*_A : A \times A \rightarrow A$, da propriedade associativa $\forall_{a_1, a_2, a_3 \in A}(a_1 *_A a_2) *_A a_3 = a_1 *_A (a_2 *_A a_3)$.

vamos falar da espécie s, do símbolo de operação $* : s\ s \rightarrow s$, da equação $(m_1 * m_2) * m_3 = m_1 * (m_2 * m_3)$ para todos os símbolos m_1, m_2, m_3 de espécie s.

Obtemos assim uma possível representação sintáctica (especificação) de um semi-grupo.

```
module SEMI-GRUPO {
    [S]
    op _*_ : S S -> S
    vars M1 M2 M3 : S
    eq (M1 * M2) * M3 = M1 * (M2 * M3) .
}
```

Em termos de regras de re-escrita, com vista a uma implementação em *CafeOBJ* (Goguen *et al.*, 1992; Nakagawa *et al.*, 1999), a equação referente à associatividade não é válida. Em *CafeOBJ* teríamos então a seguinte representação:

```
module SEMI-GRUPO {
    [S]
    op _*_ : S S -> S  {assoc}
}
```

Da especificação de semi-grupo obtemos a de monóide por adição de:

- um símbolo de constante n;
- duas equações, $n * e = e \quad e * n = e$.

```
module MONOIDE {
  [S]
  op _*_ : S S -> S
  op n : -> S
  vars M1 M2 M3 : S
  eq n * M1 = M1 .
  eq M1 * n = M1 .
  eq (M1 * M2) * M3 = M1 * (M2 * M3) .
}
```

ou, usando atributivos para os operadores binários.

```
module MONOIDE {
  [S]
  op n : -> S
  op _*_ : S S -> S  {assoc id: n}
}
```

Não é possível construir a especificação do Monóide por extensão da especificação de um Semi-Grupo, em *CafeOBJ*, dado que não é possível modificar os atributos de um dado operador.

Vejamos então a definição formal de Assinatura.

Definição 3.1 (Assinatura) *Uma assinatura* $\Sigma = (E, OP)$ *consiste:*

- *num conjunto* E*, o conjunto das espécies de* Σ*.*
- *num conjunto* OP*, o conjunto dos símbolos das constantes e dos símbolos de operações.*

O conjunto OP *é a união dos sub-conjuntos, disjuntos dois a dois:*

- C_e*, o conjunto dos símbolos de constantes de espécie* $e \in E$*;*
- $OP_{d,c}$*, o conjunto dos símbolos de operação com domínio nas espécies* $d \in E^+$*, e co-domínio na espécie* $c \in E$*, para todo o* $d \in E^+$*, e* $c \in E$*.*

Observações:

- $C = \bigcup_{e \in E} C_e$, e $OP = C \bigcup_{\substack{d \in E^+ \\ c \in E}} OP_{d,c}$
- Se $N \in C_e$, e $M \in OP_{d,c}$, com $d = e_1 \ldots e_n$ então escreve-se:

  $N : \to e$
  $M : d \to c \quad \text{ou} \quad M : e_1 \ldots e_n \to c$
- As constantes são também designadas por operadores de aridade 0.
- O facto de os conjuntos serem disjuntos dois a dois permite a sobrecarga dos identificadores.

Vejamos agora o conceito de Σ-álgebra, isto é, uma álgebra que se controí tendo como ponto de partida uma dada assinatura Σ.

Definição 3.2 (Σ-álgebra) *Uma álgebra $A = (E_A, OP_A)$, de uma assinatura $\Sigma = (E, OP)$ também designado por Σ-álgebra, é determinada por duas famílias $E_A = (A_e)_{e \in E}$ e $OP_A = (N_A)_{N \in OP}$ onde:*

- *A_e são conjuntos para todo o $e \in E$, são designados por conjuntos portadores de espécie e.*
- *$N_A \in A_e$, são elementos para todo o símbolo de constante $N \in C_e$ e $e \in E$. N_A é designada por uma constante de A.*
- *$N_A : A_{e_1} \times A_{e_2} \times \ldots \times A_{e_n} \to A_e$ são funções, para todo o símbolo de operação $N \in OP_{d,c}$, com $d \in E^+, c \in E$. N_A é designada por operação em A, o símbolo "$\times$" designa o produto cartesiano de conjuntos.*

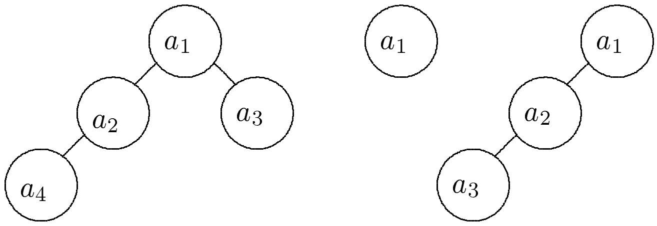
Exemplos Para cada uma das especificações apresentadas é possível obter Σ-álgebras apropriadas, para tal basta estabelecer uma correspondência apropriada entre espécies e conjuntos, e entre símbolos de operação e funções.

Árvores Binárias Uma possível especificação equacional para árvores binárias é dada pelo seguinte *objecto* em *OBJ3*.

```
module ARVBIN-BASE {
 [Alfabeto < Arvbin]
 ops a1 a2 a3 a4 a5 : -> Alfabeto
 op folha : Alfabeto -> Arvbin
 op esq : Arvbin Alfabeto -> Arvbin
 op dir : Alfabeto Arvbin -> Arvbin
 op esqdir : Arvbin Alfabeto Arvbin -> Arvbin
}
```

Pretende-se descrever árvores binárias, não vazias, nas quais os nós guardam informação do tipo alfabeto.

Ou seja, árvores dos seguintes tipos:



Como é que poderemos descrever, algebricamente, estes objectos?

Seja B o conjunto de tais árvores.

Seja A o conjunto dos elementos do alfabeto $A = \{a_1, a_2, a_3, a_4, a_5\}$.

Podemos então construir a seguinte Σ-álgebra, no caso presente uma `ARVBIN-BASE`-álgebra.

$$(\{A, B\}, \{a_1, a_2, a_3, a_4, a_5, folha, esq, dir, esqdir\})$$

com:

$$\begin{array}{rcll}
a_i & : & 1 \longrightarrow A & \\
 & & * \longmapsto a_i, \quad \text{para } i = 1, \ldots, 5 & \\
folha & : & A \longrightarrow B & \\
 & & a \longmapsto folha(a) & \\
esq & : & B \times A \longrightarrow B & \\
 & & (b, a) \longmapsto esq(b, a), & \text{adiciona uma sub-} \\
 & & & \text{árvore esquerda.} \\
dir & : & A \times B \longrightarrow B & \\
 & & (a, b) \longmapsto dir(a, b), & \text{adiciona uma sub-} \\
 & & & \text{árvore direita.} \\
esqdir & : & B \times A \times B \longrightarrow B & \\
 & & (b_1, a, b_2) \longmapsto esqdir(b_1, a, b_2), & \text{adiciona duas sub-} \\
 & & & \text{árvores.}
\end{array}$$

Antes de se poder relacionar os dois níveis, o da especificação equacional (sintáctico), e o da Σ-álgebra (semântico), é necessário introduzir outros conceitos.

É necessário definir formalmente a forma de construir palavras (termos) a partir dos símbolos de operadores e de constantes para uma dada assinatura.

Definição 3.3 (Termos e Variáveis) *Seja $\Sigma = (E, OP)$ uma assinatura e X_e, para cada $e \in E$, um conjunto, designado por conjunto das variáveis* *de espécie e. Assuma-se que estes conjuntos X_e são disjuntos dois a dois e que são disjuntos de OP. Temos então que:*

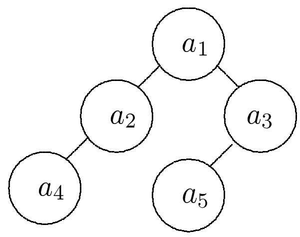
- *A união $X = \bigcup_{e \in E} X_e$, é designado por conjunto das variáveis de Σ.*
- *O conjunto $T_{OP,e}(X)$ de termos de espécie e é definido indutivamente por:*
  1. *$X_e \bigcup C_e \subseteq T_{OP,e}(X)$, isto é as variáveis e as constantes são termos, designados por termos básicos.*
  2. *$N(t_1, \ldots, t_n) \in T_{OP,e}(X)$, são termos designados por termos compostos para todo o símbolo de operação $N \in OP$ com $N : e_1 \ldots e_n \rightarrow e$ e verifica-se $t_1 \in T_{OP,e_1}(X), \ldots, t_n \in T_{OP,e_n}(X)$.*
  3. *Não existem mais nenhuns termos de espécie e em $T_{OP,e}(X)$.*

O conjunto $T_{OP,e}$ de termos sem variáveis de espécie e ($X = \emptyset$) é designado por conjunto dos termos de base ("ground terms") de espécie e.

Os conjuntos de termos da assinatura são definidos por:

$$\begin{aligned} T_{OP}(X) &= \bigcup_{e \in E} T_{OP,e}(X) \\ T_{OP} &= \bigcup_{e \in E} T_{OP,e} \end{aligned}$$

Exemplo, dado a árvore



Temos o termo:

```
esqdir(esq(folha(a3),a2),esq(folha(a5),a4))
```

As propriedades nos termos podem ser demonstradas por indução no comprimento dos mesmos, no entanto isso não reflecte a estrutura dos mesmos, como tal vai-se enunciar uma variante do método de indução, designada por indução estrutural.

Teorema 3.1 (Indução Estrutural) *Seja p um predicado definido nos termos $t \in T_{OP}(X)$ de uma assinatura $\Sigma = (E, OP)$, com um conjunto de variáveis X, ou seja, para cada $t \in T_{OP}(X)$ a asserção $p(t)$ tem um determinado valor lógico. Temos então:*

A asserção $p(t)$ é verdadeira para todo o $t \in T_{OP}(X)$ se as seguintes condições são satisfeitas:

1. *(Caso de Base) $p(t)$ é verdadeira para todo o símbolo de constante $t \in C$, e toda a variável $t \in X$.*

2. *(Passo Indutivo) para todo o termo $N(t_1, \ldots .t_n) \in T_{OP}(X)$ tem-se que se $p(t_1)$ e ... e $p(t_n)$ são verdadeiras então $p(N(t_1, \ldots, t_n))$ é verdadeira.*

Diz-se então que se tem uma demonstração por indução estrutural.

Demonstração.

Vai-se demonstrar a indução estrutural recorrendo à indução em $\mathbb{N}$ (indução natural).

Seja q o seguinte predicado em $\mathbb{N}$: $q(n)$ é verdadeiro se $p(t)$ é verdadeiro para todos os termos $t \in T_{OP}(X)$ com, no máximo, $n + 1$ símbolos de operação.

Pretende-se demonstrar que $p(t)$ é verdadeira para todo o $t \in T_{OP}(X)$ assumindo que as condições 1 e 2 são verificadas.

Caso de Base: $q(0)$ é verdadeira dado que os termos com um só símbolo são as constantes e as variáveis e, por 1, $p(t)$ é verdadeira nesses casos.

Caso Indutivo: Seja $q(n)$ verdadeira, isto é, $p(t)$ é verdadeira para todos os termos $t \in T_{OP}(X)$ com $n + 1$ símbolos no máximo. Pretende-se demonstrar que nesse caso, $q(n + 1)$ é verdadeira.

Seja t um termo com no máximo $n+2$ símbolos. Então existe um $N \in T_{OP}(X)$ tal que $t = N(t_1, t_2, \ldots, t_m)$ com $t_1 \in T_{OP,e_1}(X), \ldots, t_m \in T_{OP,e_m}(X)$ são termos com, no máximo, $n + 1$ símbolos de operação. Então $p(t_1)$, ..., $p(t_m)$ são verdadeiras, mas então pelo passo indutivo da indução estrutural $p(N(t_1, \ldots, t_m)) = p(t)$ é verdadeira, ou seja, $q(n + 1)$ é verdadeira.

Então por indução natural $q(n)$ é verdadeira para todo o $n \in \mathbb{N}$ ou seja $p(t)$ é verdadeira para todo o $t \in T_{OP}(X)$.

c.q.d.

Muitas das definições que se seguem vão ser definições indutivas na estrutura dos termos, como tal sempre que for necessário demonstrar alguma propriedade relacionada com a estrutura dos termos vai-se usar a indução estrutural.

Definição 3.4 (Avaliação de termos) *1. Seja T_{OP} o conjunto dos termos de base de uma assinatura $\Sigma = (E, OP)$, e A uma Σ-álgebra. A* valoração $\mathrm{val} : T_{OP} \to A$ *é definida recursivamente por:*

(a) $\mathrm{val}(c) = c_A$, *para todo o símbolo de constante* $c \in C$

(b) $\mathrm{val}(N(t_1, \ldots, t_n)) = N_A(\mathrm{val}(t_1), \ldots, \mathrm{val}(t_n))$, *para todo o* $N(t_1, \ldots, t_n) \in T_{OP}$.

2. Dado um conjunto de variáveis X, para $\Sigma = (E, OP)$, e uma atribuição atr $: X \to A$, *com* $\mathrm{atr}(x) \in A_e$, *para todo o* $x \in X_e$ *e* $e \in E$, *então a atribuição extendida, ou simplesmente extensão,* $\overline{\mathrm{atr}} : T_{OP}(X) \to A$ *da atribuição* atr, *é definida recursivamente por:*

(a) $\overline{\mathrm{atr}}(x) = \mathrm{atr}(x)$, *para todo o símbolo de variável* $x \in X$.

(b) $\overline{\mathrm{atr}}(c) = c_A$, *para todo o símbolo de constante* $c \in C$.

(c) $\overline{\mathrm{atr}}(N(t_1, \ldots, t_n)) = N_A(\overline{\mathrm{atr}}(t_1), \ldots, \overline{\mathrm{atr}}(t_n))$, *para todo o* $N(t_1, \ldots, t_n) \in T_{OP}(X)$.

Notas: para $X = \emptyset$ existe uma e uma só atribuição, a atribuição nula, e tem-se $\overline{\mathrm{atr}} = \mathrm{val}$.

Exemplo: Seja `NATURAIS` o seguinte módulo em *CafeOBJ*:

```
module NATURAIS {
 [ Nat ]
 op 0 : -> Nat
 op succ : Nat -> Nat
 op soma : Nat Nat -> Nat
 vars N M : Nat
 eq soma(N,0) = N .
 eq soma(N,succ(M))=succ(soma(N,M)) .
}
```

Definindo-se a valoração val desta teoria na álgebra dos Naturais do seguinte modo:

$$\begin{array}{rcl} \texttt{Nat} & \longmapsto & \mathbb{N} \\ \texttt{0} & \longmapsto & 0 \\ \texttt{succ} & \longmapsto & +_1 \\ \texttt{soma} & \longmapsto & + \end{array}$$

Temos então:

val`(soma(succ(0),succ(0))=` val`(succ(0))+` val`(succ(0)) =`
`(` val`(0)+1)+(` val`(0)+1)=(0+1)+(0+1)=2`

Proposição 3.1 (Compatibilidade de Valorações) *Dado uma assinatura $\Sigma = (E, OP)$, dois conjuntos de variáveis X e Y, uma Σ-álgebra A, e atribuições* $\mathrm{atrx} : X \to A$, $\mathrm{atry} : Y \to A$ *e* $\mathrm{atr} : X \to T_{OP}(Y)$. *O diagrama seguinte*

$$\begin{tikzcd} T_{OP}(X) \arrow[rr, "\overline{\mathrm{atr}}"] \arrow[dr, "\overline{\mathrm{atrx}}"'] & & T_{OP}(Y) \arrow[dl, "\overline{\mathrm{atry}}"] \\ & A & \end{tikzcd} \tag{3.1}$$

comuta, desde que o diagrama seguinte comute

$$\begin{tikzcd} X \arrow[rr, "\mathrm{atr}"] \arrow[dr, "\mathrm{atrx}"'] & & T_{OP}(Y) \arrow[dl, "\overline{\mathrm{atry}}"] \\ & A & \end{tikzcd} \tag{3.2}$$

Nota: A possibilidade de definir $\mathrm{atr} : X \to T_{OP}(Y)$ advêm do facto de que $T_{OP}(Y)$ ser uma Σ-álgebra, ver-se-á isso mais à frente.
Demonstração.
Demonstração por indução estrutural.

Caso de Base: para todo o $x \in X$ e $c \in C$ tem-se

$$\begin{array}{l} \overline{\mathrm{atry}} \circ \overline{\mathrm{atr}}(x) = \overline{\mathrm{atry}}(\mathrm{atr}(x)) \overset{(3.2)}{=} \mathrm{atrx}(x) = \overline{\mathrm{atrx}}(x) \\ \overline{\mathrm{atry}} \circ \overline{\mathrm{atr}}(c) = \overline{\mathrm{atry}}(c) = c_A = \overline{\mathrm{atrx}}(c) \end{array}$$

Passo indutivo: para todo o $N(t_1, \ldots, t_n) \in T_{OP}(X)$ tem-se:

$$\begin{aligned} \overline{\mathrm{atry}} \circ \overline{\mathrm{atr}}(N(t_1, \ldots, t_n)) &= \overline{\mathrm{atry}}(N_A(\overline{\mathrm{atr}}(t_1), \ldots, \overline{\mathrm{atr}}(t_n))) = \\ = \quad & N_A(\overline{\mathrm{atry}} \circ \overline{\mathrm{atr}}(t_1), \ldots, \overline{\mathrm{atry}} \circ \overline{\mathrm{atr}}(t_n)) \underset{\text{hipótese de indução}}{\overset{(3.1)}{=}} \\ = \quad & N_A(\overline{\mathrm{atrx}}(t_1), \ldots, \overline{\mathrm{atrx}}(t_n)) = \overline{\mathrm{atrx}}(N(t_1, \ldots, t_n)) \end{aligned}$$

c.q.d.

Definição 3.5 (Equações e Validade) *Dado uma assinatura $\Sigma = (E, OP)$ e variáveis X para a assinatura Σ, temos que:*

1. *Um triplo $eq = (X, LE, LD)$ com $E, D \in T_{OP,e}(X)$, para algum $e \in E$, é designado por uma* equação de espécie e *na assinatura Σ.*

2. *A equação* $eq = (X, LE, LD)$ *é válida na* Σ*-álgebra* A *se para toda a atribuição* atr $: X \to A$ *tem-se:*

$$\overline{\mathrm{atr}}(LE) = \overline{\mathrm{atr}}(LD)$$

Se eq *é válida em* A *diz-se também que* A *satisfaz* eq*.*

3. *Equações fechadas ("ground equations") são equações* $eq = (X, LE, LD)$ *em que* $X = \emptyset$*. Neste caso* E *e* D *são termos fechados.*

Uma equação $LE = LD$ tem implicitamente todas as suas variáveis quantificadas universalmente, ou seja a equação referida representa a seguinte fórmula na lógica de predicados

$$\forall_{x_1 \in e_1} \forall_{x_2 \in e_2} \ldots \forall_{x_n \in e_n} LE = LD$$

Definição 3.6 (Especificação Equacional) *Uma especificação equacional* $Esp = (E, OP, Eq)$ *consiste numa assinatura* $\Sigma = (E, OP)$*, e de um conjunto de equações* Eq *com símbolos da assinatura.*

Definição 3.7 (Esp**-álgebra)** *Uma álgebra da especificação* Esp*, designada por* Esp*-álgebra, é uma álgebra* A *com assinatura* Σ*, e que satisfaz todas as equações em* Eq*.*

Vejamos alguns exemplos.

```
module NATURAIS {
 [Nat]
 op zero : -> Nat
 op succ : Nat -> Nat
 op soma : Nat Nat -> Nat
 vars N M : Nat
 eq soma(N,zero) = N .
 eq soma(N,succ(M)) = succ(soma(N,M)) .
}
```

Para que $(\mathbb{N}, 0, +_1, +)$ seja uma `NATURAIS`-álgebra é necessário verificar que para toda a atribuição atr $: X \to \mathbb{N}$, com $X = \{N, M\}$ as equações da especificação equacional são satisfeitas. Temos então pela definição da atribuição extendida:

$$\overline{\mathrm{atr}}(\mathtt{soma}(\mathtt{N},\mathtt{zero})) = \overline{\mathrm{atr}}(\mathtt{N}) + \overline{\mathrm{atr}}(\mathtt{zero}) = \overline{\mathrm{atr}}(\mathtt{N}) + 0 = \overline{\mathrm{atr}}(\mathtt{N})$$

$$\begin{aligned}
&\overline{\mathrm{atr}}(\mathtt{soma}(\mathtt{N},\mathtt{succ}(\mathtt{M}))) = \overline{\mathrm{atr}}(\mathtt{N}) + \overline{\mathrm{atr}}(\mathtt{succ}(\mathtt{M})) = \\
&= \overline{\mathrm{atr}}(\mathtt{N}) + \overline{\mathrm{atr}}(\mathtt{M}) + 1 = \overline{\mathrm{atr}}(\mathtt{soma}(\mathtt{N},\mathtt{M})) \\
&= \overline{\mathrm{atr}}(\mathtt{succ}(\mathtt{soma}(\mathtt{N},\mathtt{M})))
\end{aligned}$$

Definição 3.8 (Sub-termo) *Seja "$\leq$" a seguinte relação binária em $T_{OP}(X)$:*

1. $t \leq t$, *para todo o* $t \in T_{OP}(X)$
2. $t \leq N(t_1, \ldots, t_n)$, *se* $t = t_i$ *para um dado* $1 \leq i \leq n$ *e* $N(t_1, \ldots, t_n) \in T_{OP}(X)$
3. $t_1 \leq t_2$ *e* $t_2 \leq t_3$ *implica* $t_1 \leq t_3$, *para todos os* $t_1, t_2, t_3 \in T_{OP}(X)$

Temos então que t_1 *é um sub-termo de* t_2 *sempre que* $t_1 \leq t_2$.

Se se tem que $t \leq t'$, então existe pelo menos um *contexto esquerdo*, *ce*, e um *contexto direito*, *cd*, de t em t' tal que:

$$t' = ce\ t\ cd$$

ce e *cd* não são em geral termos, mas simplesmente sequências de símbolos de Σ.

Definição 3.9 (Substituição de Termos) *Dado um conjunto de equações Eq, para uma dada assinatura* $\Sigma = (E, OP)$, *com um conjunto fixo de variáveis* $X = X_{eq}$ *para cada equação eq. Então a equação* $(X, LE, LD) \in Eq$ *define duas regras de substituição:*

1. $LE \Longrightarrow LD$ *(regra esq-dir)*
2. $LD \Longrightarrow LE$ *(regra dir-esq)*

Uma regra $t_1 \Rightarrow t_2$ é aplicável a um termo $t \in T_{OP}(X)$ se existe uma atribuição $\mathrm{atr} : X \to T_{OP}(X)$ com extensão $\overline{\mathrm{atr}} : T_{OP}(X) \to T_{OP}(X)$ tal que para $\overline{t_1} = \overline{\mathrm{atr}}(t_1)$ e $\overline{t_2} = \overline{\mathrm{atr}}(t_2)$, tem-se que: $\overline{t_1}$ é um sub-termo de t.

A substituição de $\overline{t_1}$ em t por $\overline{t_2}$ dá origem a um termo t', sendo que se denota essa substituição por $t' = t(\overline{t_1}/\overline{t_2})$.

Se houver mais do que uma ocorrência de t_1 em t a regra de substituição deve especificar como proceder.

Definição 3.10 (Derivação de Termos) *Dado um conjunto Eq de equações para uma dada assinatura* $\Sigma = (E, OP)$ *com um conjunto fixo de variáveis* X, *temos que:*

1. *Se a regra* $t_1 \Rightarrow t_2$ *é aplicável ao termo* $t \in T_{OP}(X)$ *então diz-se que* $t \Rightarrow t'$ *é uma derivação num passo de* t *para* t' *em Eq, usando a regra* $t_1 \Rightarrow t_2$ *e a atribuição* atr.
2. *Uma sequência de* $n \geq 0$ *derivações num só passo* $t_0 \Rightarrow t_1 \Rightarrow \cdots \Rightarrow t_n$ *com* $t = t_0$ *e* $t' = t_n$, *denotada por* $t \overset{*}{\Rightarrow} t'$, *é designada por derivação de* t *para* t' *em Eq, e* $eq' = (X, t, t')$ *é designada por equação derivada de Eq com* X *fixo.*

3. *A derivação $t \stackrel{*}{\Rightarrow} t'$ é dita correcta com respeito à Esp-álgebra A se se tem que para toda a atribuição* $\mathrm{atr} : X \to A$ *se verifica* $\overline{\mathrm{atr}}(t) = \overline{\mathrm{atr}}(t')$.

Tendo estabelecido como se procede à derivação de termos, e o que se entende por uma derivação correcta para uma dada álgebra A, falta-nos definir o que se entende por derivações correctas.

Teorema 3.2 (Correcção das Derivações) *Dado uma especificação $Esp = (E, OP, Eq)$, onde Eq é um conjunto de equações com respeito a um conjunto fixo de variáveis X, cada derivação $t \stackrel{*}{\Rightarrow} t'$ de termos $t, t' \in T_{OP}(X)$ em Eq é correcta com respeito a todas as Esp-álgebras A.*

Demonstração.

Demonstração por indução estrutural.

Pela transitividade da igualdade na definição de derivação correcta, isto é $\overline{\mathrm{atr}}(t) = \overline{\mathrm{atr}}(t')$ basta provar a correcção com respeito à Esp-álgebra A para uma derivação num passo. Assume-se que se usou a regra $t_1 \Rightarrow t_2$ com $(X, t_1, t_2) \in Eq$ e uma atribuição $\mathrm{atr} : X \to T_{OP}(X)$ de forma a obter $t' = t(\overline{t_1}/\overline{t_2})$ com $\overline{t_1} = \overline{\mathrm{atr}}(t_1)$ e $\overline{t_2} = \overline{\mathrm{atr}}(t_2)$.

De molde a demonstrar a correcção da derivação com respeito a A é necessário mostrar que para cada atribuição $\mathrm{atr1} : X \to A$ se tem que $\overline{\mathrm{atr1}}(t) = \overline{\mathrm{atr1}}(t')$.

Usando o facto de que $t' = t(\overline{t_1}/\overline{t_2})$ basta mostrar que $\overline{\mathrm{atr1}}(\overline{t_1}) = \overline{\mathrm{atr1}}(\overline{t_2})$.

Dado que (X, t_1, t_2) é válida em A (A é uma Esp-álgebra) sabe-se que para a atribuição $\mathrm{atr2} : X \to A$ definida por $\mathrm{atr2} = \overline{\mathrm{atr1}} \circ \mathrm{atr}$

$$\begin{tikzcd}
X \arrow[r, "\mathrm{atr}"] \arrow[dr, "\mathrm{atr2}"'] & T_{OP}(X) \arrow[d, "\overline{\mathrm{atr1}}"] \\
 & A
\end{tikzcd}$$

tem-se que

$$\overline{\mathrm{atr2}}(t_1) = \overline{\mathrm{atr2}}(t_2)$$

pela definição de atr2 e pela compatibilidade de atribuições, temos então que:

$$\overline{\mathrm{atr2}} = \overline{\mathrm{atr1}} \circ \overline{\mathrm{atr}}$$

$$\begin{tikzcd} X \arrow[r, "\mathrm{atr}"] \arrow[dr, "\mathrm{atr2}"'] & T_{OP}(X) \arrow[d, "\overline{\mathrm{atr1}}"] & T_{OP}(X) \arrow[l, "\overline{\mathrm{atr}}"'] \arrow[dl, "\overline{\mathrm{atr2}}"] \\ & A & \end{tikzcd}$$

O que por $\overline{\mathrm{atr2}}(t_1) = \overline{\mathrm{atr2}}(t_2)$ e por $\overline{t_1} = \overline{\mathrm{atr}}(t_1)$ e $\overline{t_2} = \overline{\mathrm{atr}}(t_2)$ implica que $\overline{\mathrm{atr1}}(\overline{t_1}) = \overline{\mathrm{atr1}}(\overline{t_2})$.

c.q.d.