

## Specifica – parte IIC

Leggere Sez. 5.6.3 Ghezzi et al.



università di ferrara  
DA SEICENTO ANNI GUARDIAMO AVANTI.

## Specifiche algebriche

- Tradizionalmente, un'*algebra* è composta da un *insieme* e *operazioni* sull'insieme.
- Esempio: insieme N con addizione, sottrazione, moltiplicazione, divisione.
- Le algebre definite su un solo insieme si dicono *omogenee*.

Specifica 3C



università di ferrara  
DA SEICENTO ANNI GUARDIAMO AVANTI.

2

## Specifiche algebriche

- Definiscono un sistema come un'*algebra eterogenea*, cioè come collezione di insiemi diversi su cui sono definite operazioni.
- Adatte alla definizione di tipi di dati astratti.

Specifica 3C



università di ferrara  
DA SEICENTO ANNI GUARDIAMO AVANTI.

3

## Algebra per definizione di stringhe

- Vogliamo definire le operazioni:
  - **new**: creazione di nuove stringhe
  - **append**: concatenazione di stringhe
  - **add**: aggiunta di un carattere in coda a una stringa
  - **length**: calcolo della lunghezza di una stringa
  - **isEmpty**: controllo se una stringa è vuota
  - **equal**: controllo se due stringhe sono uguali

Specifica 3C



università di ferrara  
DA SEICENTO ANNI GUARDIAMO AVANTI.

4

## Algebra per definizione di stringhe

- Insiemi che formano l'algebra:
  - String
  - Nat
  - Bool
  - Char
- Ogni insieme si dice *sort*.
- La collezione di insiemi si dice *segnatura*.

## Sintassi

- La *sintassi* di un'algebra è definita dalla sua segnatura e dall'elenco delle operazioni, con i rispettivi domini e codomini.
- Larch: famiglia di linguaggi per specifiche formali, formati da:
  - parte comune (Larch Shared Language)
  - parte di interfaccia con linguaggi di programmazione (es. Larch/Pascal)

## Esempio

**algebra** StringSpec;  
**introduce**  
**sorts** String, Char, Nat, Bool;  
**operations**  
new: () → String;  
append: String, String → String;  
add: String, Char → String;  
length: String → Nat;  
isEmpty: String → Bool;  
equal: String, String → Bool

## Semantica

- Il significato delle operazioni deve essere specificato formalmente.
- La *semantica* di un'algebra è definita da un insieme di *equazioni*
- Rappresentano le *proprietà* essenziali delle operazioni.
- Sono dette anche *assiomi*, in quanto usate per dedurre proprietà delle algebre.

## Esempio

```
constrains new, append, add, length, isEmpty, equal so that
String generated by [new, add]
for all [s, s1, s2: String; c: Char]
  isEmpty (new ()) = true;
  isEmpty (add (s, c)) = false;
  length (new ()) = 0;
  length (add (s, c)) = length (s) + 1;
  append (s, new ()) = s; (2)
  append (s1, add (s2,c)) = add (append (s1,s2),c); (1)
  equal (new (),new ()) = true;
  equal (new (), add (s, c)) = false;
  equal (add (s, c), new ()) = false;
  equal (add (s1, c), add (s2, c)) = equal (s1,s2);
end StringSpec.
```



## Deduzione di proprietà

- $\text{append}(\text{new}(), \text{add}(\text{new}(),c)) = \text{add}(\text{new}(), c)$
- Dall'assioma (1) con  $s_1=s_2=\text{new}()$ ,  
 $\text{append}(\text{new}(), \text{add}(\text{new}(),c)) = \text{add}(\text{append}(\text{new}(),\text{new}()),c)$ ; (3)
  - Dall'assioma (2), con  $s = \text{new}()$ ,  
 $\text{append}(\text{new}(), \text{new}()) = \text{new}()$
  - E sostituendo in (3) la tesi.



## Deduzione di proprietà

- $\text{append}(\text{new}(),s) = s$  (4)
- Per induzione sulla lunghezza di s:
  - se  $s = \text{new}()$ : (4) è vera per l'assioma (2)
  - se (4) è vera per ogni  $s_0$  di lunghezza L:  
ogni  $s_1$  di lunghezza L+1 sarà  $\text{add}(s_0, c)$  per qualche  $s_0$  e c, e  
 $\text{append}(\text{new}(),s_1) =$   
 $\text{append}(\text{new}(),\text{add}(s_0,c)) =$   
 $\text{add}(\text{append}(\text{new}(),s_0),c) = \text{add}(s_0,c) = s_1$



## Possibili problemi

- Incompletezza
- Sovraspecifica
- Incoerenza
- Ridondanza



## Incompletezza

- Si può dimostrare che  $\text{equal}(\text{add}(s, 'a'), \text{add}(s, 'b')) = \text{false}$ ?
- No. Occorre sostituire l'ultimo assioma con  
 $\text{equal}(\text{add}(s_1, c_1), \text{add}(s_2, c_2)) = \text{equal}(s_1, s_2) \text{ and } \text{equalC}(c_1, c_2)$ ;

## Sovraspecifica

- Ad es., caso in cui l'ultimo assioma sia  $\text{equal}(\text{add}(s_1, c_1), \text{add}(s_2, c_2)) = \text{equal}(s_1, s_2) \text{ and } \text{equalC}(c_1, c_2) \text{ and } \text{not } \text{equal}(c_1, 'a')$ ;

## Incoerenza

- Si ha quando si riesce a dimostrare  $\text{true} = \text{false}$
- Ad es. se includessimo l'assioma  $\text{equal}(\text{add}(s, c), s) = \text{true}$

## Ridondanza

- Presenza di assiomi non necessari in quanto derivabili
- Ad es. l'assioma  $\text{append}(\text{new}(), s) = s$

## Esempio: editor di testi

### Operazioni:

- NewF: crea un nuovo file vuoto
- isEmptyF: controlla se un file è vuoto
- addF: aggiunge una stringa alla fine del file
- insertF: inserisce una stringa nella posizione specificata
- appendF: concatena due file

## Sintassi

algebra TextEditor;

introduces

sorts Text, String, Char, Bool, Nat;

operations

```
newF: () → Text;
isEmptyF: Text → Bool;
addF: Text, String → Text;
insertF: Text, Nat, String → Text;
appendF: Text, Text → Text;
deleteF: Text → Text;
lengthF: Text → Nat;
equalF: Text, Text → Bool;
addFC: Text, Char → Text;
```

## Semantica

**constrains** newF, isEmptyF, addF, appendF, insertF, deleteF

**so that** TextEditor generated by [newF, addFC]

for all [f, f1, f2: Text; s: String; c: Char; cursor: Nat]

isEmptyF (newF ()) = true;

isEmptyF (addFC (f, c)) = false;

addF (f, newS ()) = f;

addF (f, addS (s, c)) = addFC (addF (f, s), c);

lengthF (newF ()) = 0;

lengthF (addFC (f, c)) = lengthF (f) + 1;

appendF (f, newF ()) = f;

## Semantica

appendF (f1, addFC (f2, c)) =

addFC (appendF (f1, f2), c);

equalF (newF (), newF ()) = true;

equalF (newF (), addFC (f, c)) = false;

equalF (addFC (f, c), new ()) = false;

equalF (addFC (f1, c1), addFC (f2, c2)) =

equalF (f1, f2) and equalC (c1, c2);

insertF (f, cursor, newS ()) = f;

((equalF (f, appendF (f1, f2)) and (lengthF (f1) = cursor - 1))

**implies**

equalF (insertF (f, cursor, s), appendF (addF (f1, s), f2))) = true;

**end** TextEditor.

## Notazione alternativa

- L'assioma
  - A implies Bsi può scrivere
  - if A then B
- La coppia di assiomi
  - A implies B
  - not A implies Csi può scrivere
  - if A then B
  - else C