Abstract Specification Theory

Andrzej Tarlecki

Institute of Informatics, Warsaw University
and
Institute of Computer Science, Polish Academy of Sciences
Warsaw, Poland

Marktoberdorf Summer School 2002
"Models, Algebras, and Logic of Engineering Software"

Long-term goal

A *formal basis*

for *systematic development*

of *correct programs*

from *requirements specifications*

by *verified refinement steps*.

Overall plan

- Underlying logical framework:
  - basic algebraic framework; institutions
- Specifications:
  - basic and structured specifications; proof systems for specifications
- Program development:
  - simple refinements; constructor refinements; local constructions in global refinement steps; architectural specifications
- Observational approach:
  - observational interpretation of specifications, refinements, and architectural specifications
- Conclusions

Literature

A rudimentary knowledge of basic concepts of universal algebra and category theory is expected. Apart from a number of standard textbooks, these can be learnt from:

- D. Sannella, A. Tarlecki. Algebraic preliminaries. In: *Algebraic Foundations of System Specification*, E. Astesiano, H.-J. Kreowski, B. Krieg-Brückner, eds., 13–30, Springer Verlag 1999.
- D. Sannella, A. Tarlecki. Category theory. In: *Foundations of Algebraic Specifications and Formal Program Development*, under preparation.

See `http://www.mimuw.edu.pl/~tarlecki/marktoberdorf/`.

The lectures will loosely follow the ideas presented in the following papers, which therefore should make a good background reading:

- D. Sannella, A. Tarlecki. Essential concepts of algebraic specification and program development. *Formal Aspects of Computing* 9(1997) 229–269.
- M. Bidoit, D. Sannella, A. Tarlecki. Architectural specifications in CASL. *Formal Aspects of Computing*, to appear (2002).
- M. Bidoit, D. Sannella, A. Tarlecki. Global development via local observational construction steps. *Proc. MFCS'02*, Springer LNCS, to appear (2002).

See also `http://www.mimuw.edu.pl/~tarlecki/marktoberdorf/`.

CoFI pages, `http://www.brics.dk/Projects/CoFI/`, provide all the material on CASL; for instance:

- CoFI Language Design Task Group. CASL — The Common Algebraic Specification Language — Summary. See `http://www.brics.dk/Projects/CoFI/Documents/CASL/Summary/`.

Institutions have been introduced in:

- J. Goguen, R. Burstall. Institutions: abstract model theory for specification and programming. *Journal of the Assoc. for Computing Machinery* 39(1):95–146, 1992.

The following book chapter provides an introduction to, motivation for, and some concepts and results built around this notion:

- A. Tarlecki. Institutions: an abstract framework for formal specifications. In: *Algebraic Foundations of System Specification*, E. Astesiano, H.-J. Kreowski, B. Krieg-Brückner, eds., 105–130, Springer-Verlag 1999.
  See also `http://www.mimuw.edu.pl/~tarlecki/marktoberdorf/`.

Logical framework

- Basic algebraic framework
  - signatures, algebras, equations
  - signature morphisms and what they entail
- Institutions
  - signatures, models, sentences
  - signature morphisms and what they entail
  - some extra structure and properties

Software systems (modules, programs, databases . . .):

sets of data with operations on them

- Disregarding: code, efficiency, robustness, reliability, . . .
- Focusing on: CORRECTNESS

Rough analogy

module interface	$\rightsquigarrow$	signature
module	$\rightsquigarrow$	algebra
module specification	$\rightsquigarrow$	class of algebras

Signatures & algebras

- *Algebraic signature*:

$$\boxed{\Sigma = (S, \Omega)}$$

  – *sort names*: S

  – *operation names, with arity and result sort*: $\Omega = \langle \Omega_{w,s} \rangle_{w \in S^*, s \in S}$

- Σ*-algebra*:

$$\boxed{A = (|A|, \langle f_A \rangle_{f \in \Omega})}$$

  – *carrier sets*: $|A| = \langle |A|_s \rangle_{s \in S}$

  – *operations*: $f_A \colon |A|_{s_1} \times \ldots \times |A|_{s_n} \to |A|_s$, for $f \in \Omega_{s_1 \ldots s_n, s}$

- $f \colon s_1 \times \ldots \times s_n \to s$ stands for $s_1, \ldots, s_n, s \in S$ and $f \in \Omega_{s_1 \ldots s_n, s}$

Fix a signature $\Sigma = (S, \Omega)$ for a while.

Few further notions

- the class of all Σ-algebras: $Alg(\Sigma)$
- *subalgebra* $A_{sub} \subseteq A$: given by subset $|A_{sub}| \subseteq |A|$ closed under the operations
- *homomorphism* $h \colon A \to B$: map $h \colon |A| \to |B|$ that preserves the operations
- *congruence* ρ on A: equivalence $\rho \subseteq |A| \times |A|$ that is closed under the operations
- *quotient algebra* A/ρ: built in the natural way on the equivalence classes of ρ

Terms

Consider an S-sorted set X of variables, Σ-algebra A and valuation $v \colon X \to |A|$.

- *term* $t \in |T_\Sigma(X)|$: built using variables X, constants and operations from Ω in the usual way
- *term algebra* $T_\Sigma(X)$: with the set of terms as the carrier, and operations defined "syntactically"
- *term evaluation* $v^\sharp \colon T_\Sigma(X) \to A$: the unique homomorphism from $T_\Sigma(X)$ to A that extends v
- *term value* $t_A[v] = v^\sharp(t)$: may also be determined inductively