A relational model of programming

At a glance

- What is a
  - program?
  - problem?
- What does „specification” mean?
- How to
  - construct a program?
  - construct a correct program?

Origins

- ELTE, Budapest
- Ákos Fóthi et al, beginning of 1980s
- Teaching aid & research
- Describing most aspects of imperative programming

Influenced by

- H. D. Mills: Mathematical functions for structured programming (1972)
- C. A. R. Hoare: Proof of correctness of data representations. (1972)
- E. W. Dijkstra: A Discipline of Programming (1976)
- D. Gries: The Science of Programming (1981)

Publications

- Fóthi, Á.: A mathematical approach to programming. In: Annales Univ. Sci. Budapestin. De Rolando Eötvös Nom., Sect. Comp., Tomus IX. (1989), pp. 105-113.
- Fóthi, Á., Horváth, Z., Nyékyné Gaizler, J.: A Relational Model of Transformation in Programming. In: Proceedings of the 3rd International Conference on Applied Informatics (1997)

Scope

- Basics:
  - Problems & Programs
  - State spaces
- Structured constructs
  - Correctness of derivation rules

- Further:
  - Method of stepwise refinement
  - Theorems on common patterns
  - Program transformations
  - Type theory

Notation

- $f: A \to B$ (f *is total*)
- $f \in A \to B$ (f *is partial*)
- $f \in A \times B, H \subseteq A: f_{|H} = f \cap H \times B$
- $\mathcal{D}_f,\ \mathcal{R}_f$

- $\alpha: \mathbb{N} \to A, \mathcal{D}_\alpha \in \mathbb{N} \cup \{[1..n] | n \in \mathbb{N}\}$
  - $\mathcal{D}_\alpha = \mathbb{N}:\ \alpha \in A^\infty, |\alpha| := \infty$
  - $\mathcal{D}_\alpha = [1..n]: \alpha \in A^*, |\alpha| := n, \tau(\alpha) := \alpha_n$
  - $A^\infty \cup A^* = A^{**}$
  - $\chi(\alpha^1, \alpha^2, \ldots) := red(concat(\alpha^1, \alpha^2, \ldots))$

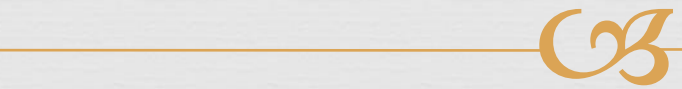
Notation

- $\times_{i \in I} A_i := \{x | x: I \to \cup_{i \in I} A_i \wedge \forall i \in I: x(i) \in A_i\}$
  $(generalized\ cross\ product)$

- $J \subseteq I, A = \times_{i \in I} A_i, B = \times_{j \in J} B_j$: B is a *subspace* of A
- $a \in A: pr_B(a) = a_{|J}$ $(projection)$

Programs & problems

State space



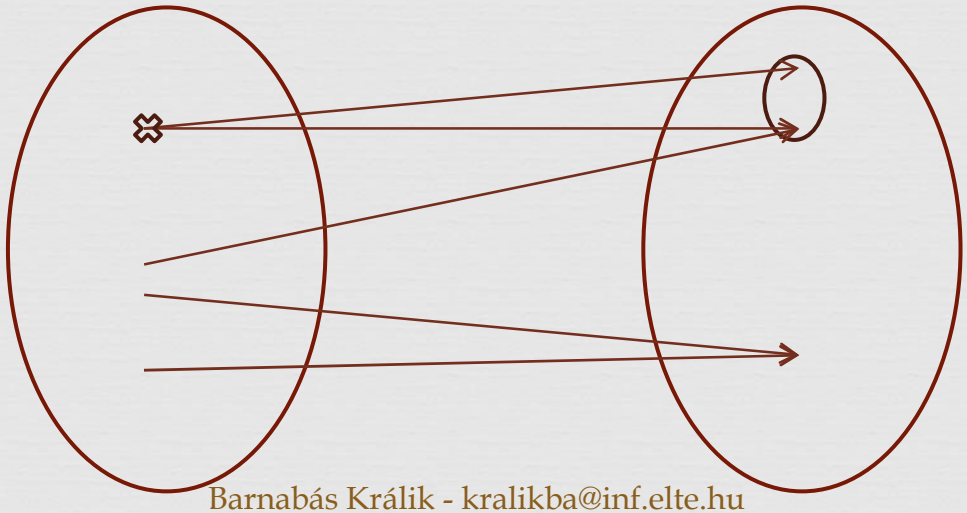
- Given finite set I and for all $i \in I, \mathbb{N} \geqslant A_i$

  $A = \times_{i \in I} A_i$ is a state space

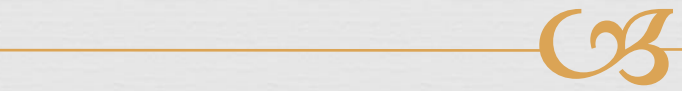
- $i \in I: v_i: A \to A_i,\ v_i \coloneqq pr_{\{i\}}$ is a *variable*

Problem

$$F \subseteq A \times A$$

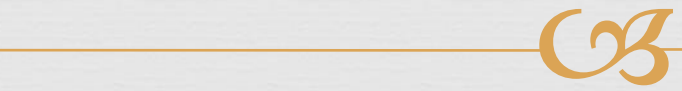


Program



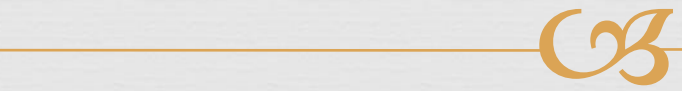
- The $S \subseteq A \times A^{**}$ relation is a program in A, iff
  1. $\mathcal{D}_\alpha = A$
  2. $\forall \alpha \in R_S : \alpha = red(\alpha)$
  3. $\forall a \in A : \forall \alpha \in S(a) : |\alpha| \neq 0 \wedge \alpha_1 = a$

Effect relation



- The $p(S) \subseteq A \times A$ relation is the effect relation of the program S, so that
  1. $\mathcal{D}_{p(S)} = \{a \in A | S(a) \subseteq A^*\}$
  2. $p(S)(a) = \{b \in A | \exists \alpha \in S(a) : \tau(\alpha) = b\}$

Solution



- An S program solves an F problem iff
  1. $\mathcal{D}_F \subseteq \mathcal{D}_{p(S)}$
  2. $\forall a \in D_F : p(S)(a) \subseteq F(a)$

Example

$A = \{1,2,3,4,5\}\ S \subseteq A \times A^{**}$

$$S = \left\{ \begin{array}{c} (1, \langle 1251 \rangle), (1, \langle 14352 \rangle), (1, \langle 132132 \ldots \rangle),\ (2, \langle 21 \rangle), (2, \langle 24 \rangle), \\ (3, \langle 333 \ldots \rangle),\ (4, \langle 41514 \rangle), (4, \langle 431251 \rangle), (4, \langle 41542 \rangle), \\ (5, \langle 524 \rangle), (5, \langle 5234 \rangle) \end{array} \right\}$$

$F = \{(2,1), (2,4), (4,1), (4,2), (4,5)\}$

$p(S) = \{(2,1), (2,4), (4,4), (4,1), (4,2), (5,4)\}$

Specification

and solution

Stating a complex problem

- Specifying complete programs using „raw relations” is cumbersome and error-prone
- Let us define notions of
  - pre- and postconditions, based on the
  - weakest precondition, yielding the concept of
  - specification.

Weakest precondition

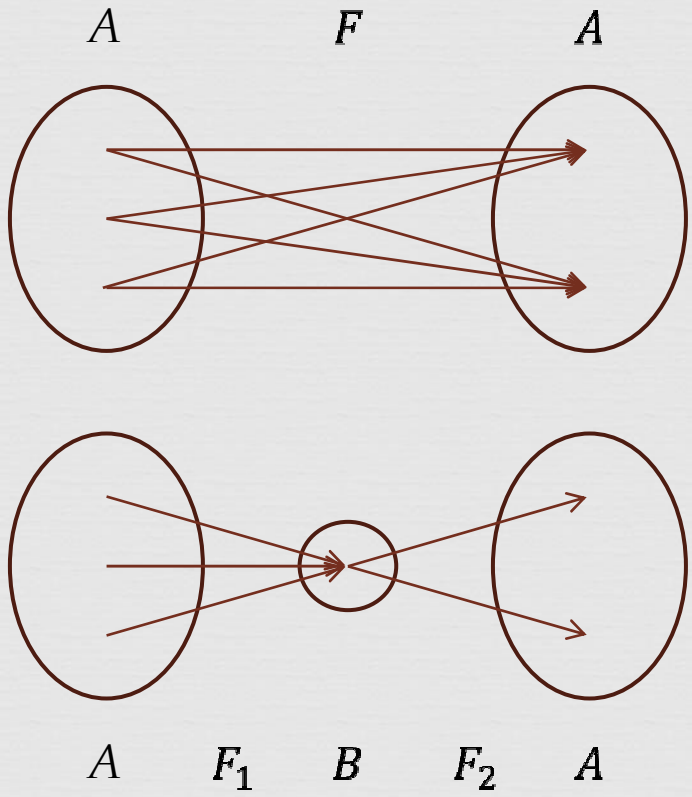
- Given $S \subseteq A \times A^{**}$ and $R: A \to \mathbb{L}$ predicates,
  - $wp(S,R): A \to \mathbb{L}$ is the weakest precondition of S for R, such that
  - $\lceil wp(S,R) \rceil = \{a \in A | a \in \mathcal{D}_{p(S)} \wedge p(s)(a) \subseteq \lceil R \rceil\}$

- Property: $\lceil wp(S,R) \rceil = \lceil R \circ p(S) \rceil$

Decomposing the problem

- Let us decompose $F \subseteq A \times A$ into relations F_1 and F_2, such that
  - $F_1 \subseteq A \times B$,
  - $F_2 \subseteq B \times A$,
  - $F = F_2 \circ F_1$.

The set B is then called the *parameter space*.



Theorem of specification

- Let $F \subseteq A \times A$ be a problem, B its problem space with $F_1 \subseteq A \times B, F_2 \subseteq B \times A, F = F_2 \circ F_1$. Let us define for all $b \in B$ the following predicates:
  - $[Q_b] = \{a \in A | (a,b) \in F_1\} = F_1^{(-1)}(b)$,
  - $[R_b] = \{a \in A | (b,a) \in F_2\} = F_2(b)$.
- If $\forall b \in B: Q_b \Rightarrow wp(S, R_b)$ holds, S solves F. ■

Example

$$A = \underset{a}{\mathbb{Z}} \times \underset{b}{\mathbb{N}}$$
$$B = \underset{a'}{\mathbb{Z}}$$
$$Q: a = a' \wedge a' \neq 0$$
$$R: b = |a| * 2 \wedge a' = 0$$

Construction

and properties thereof

Programming task

- Let $A = \times_{i \in I} A_i$ be the state space. We call the 3-tuple $(F, \mathbb{P}, \mathbb{K})$ a programming task, if
  - $F \subseteq A \times A$ is a task,
  - $\mathbb{P}$, the set of *primitive programs* is a finite set of programs in A
  - $\mathbb{K}$, the set of *allowed constructions* is a finite set of functions on the set of programs in A
- The S program is the solution of $(F, \mathbb{P}, \mathbb{K})$ iff S can be constructed from $\mathbb{P}$ using $\mathbb{K}$ and solves F.

SKIP and ABORT

- $SKIP$
  - $\forall a \in A: SKIP(a) = \langle a \rangle$
  - $wp(SKIP, R) = R$
- $ABORT$
  - $\forall a \in A: ABORT(a) = \langle a \ldots \rangle$
  - $wp(ABORT, R) = false$

Assignment

- $A = \times_{i \in I} A_i, F \subseteq A \times A$
  - S is called a *general assignment* iff $\forall a \in A$:

$$S(a) = \begin{cases} \{red(\langle a, b \rangle) | b \in F(a)\}, if\ a \in \mathcal{D}_F \\ \{\langle a, a, a, \dots \rangle\}, if\ a \notin \mathcal{D}_F \end{cases}$$

- Now, we can prove, that:
  - $F: A \to A$ $\quad wp(a := F(a), R) = R \circ F$
  - $\mathcal{D}_F = A$ $\quad \lceil wp(a :\in F(a), R) \rceil = \lceil R \circ F \rceil$
  - $F \in A \to A$

$$\forall b \in A: wp(a := F(a), R) = \begin{cases} R \circ F(b), if\ b \in \mathcal{D}_F \\ false, if\ b \notin \mathcal{D}_F \end{cases}$$

  - $\mathcal{D}_F \subset A$

$$\forall b \in A: wp(a := F(a), R) = \begin{cases} true, \quad if\ b \subseteq \lceil R \rceil \\ false, if\ b \notin \mathcal{D}_F \end{cases}$$

Sequence

- Given programs $S_1, S_2 \subseteq A \times A^{**}$, the following $S_1; S_2 \subseteq A \times A^{**}$ program is their sequence:
  - $\forall a \in A: S_1; S_2(a) = \{\alpha \in A^{\infty} | \alpha \in S_1(a)\} \cup \{\chi(\alpha, \beta) \in A^{**} | \alpha \in S_1(a) \cap A^{*} \wedge \beta \in S_2(\tau(\alpha))\}$
- effect relation:
  - $p(S) = p(S_2) \odot p(S_1)$
- Derivation rule:
  - If $Q \Rightarrow wp(S_1, Q') \wedge Q' \Rightarrow wp(S_2, R)$, then $Q \Rightarrow wp(S_1; S_2, R)$

Branch

Given $S_1, S_2, \ldots, S_n \subseteq A \times A^{**}$ programs and $\pi_1, \pi_2, \ldots, \pi_n : A \to \mathbb{L}$ predicates, we can construct $(\pi_1 : S_1, \ldots, \pi_n : S_n) = IF$ as:

$$\forall a \in A : IF(a) = w_0(a) \cup \bigcup_{i=1}^{n} w_i(a)$$

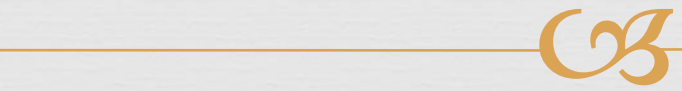
where $\forall i \in [1..n] : w_i(a) := \begin{cases} S_i(a), & if\ \pi_i(a) \\ \emptyset, & if \neg \pi_i(a) \end{cases}$

and $w_0(a) := \begin{cases} \langle a, a, a, a \ldots \rangle, if\ \forall_{i=1}^{n} \neg \pi_i(a) \\ \emptyset, \quad otherwise \end{cases}$

Branch

- effect relation:
  - $\mathcal{D}_{p(IF)} =$
    $\{a \in A | a \in \bigcup_{i=1}^{n} \lceil \pi_i \rceil \wedge \forall i \in [1..n]: \pi_i(a) \Rightarrow a \in \mathcal{D}_{p(S_i)}\}$
  - $\forall a \in \mathcal{D}_{p(IF)}: p(IF)(a) = \bigcup_{i=1}^{n} p(S_i)_{|\lceil \pi_i \rceil}(a)$
- Derivation rule: if
  - $Q \Rightarrow \bigvee_{i=1}^{n} \pi_i$ and
  - $\forall i \in [1..n]: Q \wedge \pi_i \Rightarrow wp(S_i, R)$ holds, then
    $Q \Rightarrow wp(IF, R)$

Loop



- Given $\pi: A \to \mathbb{L}$ condition and an $S_0 \subseteq A \times A^{**}$ program (loop body), $\mathrm{DO} = (\pi, S_0) \subseteq A \times A^{**}$ is called a loop and is defined as follows:
  - $\forall a \notin \lceil \pi \rceil: DO(a) = \{\langle a \rangle\}$
  - $\forall a \in \lceil \pi \rceil: DO(a) =$

$$\left\{ \alpha \in A^{**} \middle| \begin{array}{c} \exists \alpha^1, \alpha^2, \dots, \alpha^n \in A^{**}: \alpha = \chi(\alpha^1, \alpha^2, \dots, \alpha^n) \wedge \\ \alpha^1 \in S_0(a) \wedge \forall i \in [1..n-1]: \left(\alpha^i \in A^* \wedge \alpha^{i+1} \in S_0\left(\tau(\alpha^i)\right) \wedge \tau(\alpha^i) \in \lceil \pi \rceil\right) \wedge \\ (\alpha^n \in A^\infty \vee \alpha^n \in A^* \wedge \tau(\alpha^n) \notin \lceil \pi \rceil) \end{array} \right\}$$

$$\cup \left\{ \alpha \in A^\infty \middle| \begin{array}{c} \forall i \in \mathbb{N}: \exists \alpha^i \in A^*: \alpha = \chi(\alpha^1, \alpha^2, \dots) \wedge \\ \alpha_1 \in S_0(a) \wedge \forall i \in \mathbb{N}: \left(\alpha^i \in A^* \wedge \alpha^{i+1} \in S_0\left(\tau(\alpha^i)\right) \wedge \tau(\alpha^i) \in \lceil \pi \rceil\right) \end{array} \right\}$$

Loop

- effect relation:
  - $p(DO) = \overline{p(S_0)|\pi}$
- Derivation rule: if
  1. $Q \Rightarrow P$,
  2. $P \wedge \neg\pi \Rightarrow R$,
  3. $P \wedge \pi \Rightarrow t > 0$,
  4. $P \wedge \pi \Rightarrow wp(S_0, P)$ and
  5. $P \wedge \pi \wedge t = t_0 \Rightarrow wp(S_0, t < t_0)$ holds, then
     $Q \Rightarrow wp(S_0, R)$

Conclusions

Conclusions

- Model and course used with minimal modification
- 2-semester introductory course (BSc)
- Easier comprehension of later courses (BSc & MSc)

Questions?