Wyniki prof. Rasiowej w informatyce I

G. Mirkowska & A. Salwicki

Instytut Informatyki UKSW
salwicki@mimuw.edu.pl

przyczynek -

22 czerwca 2017 konferencja PTKwM Rzeszów

Plan

1. Aksjomat instrukcji przypisania, H. Rasiowa w r. 1963
2. H. Rasiowa i automatyczne dowodzenie twierdzeń
3. Lemat Rasiowej i Sikorskiego
4. Lemat R-S w logice algorytmicznej
5. Zasługi H. Rasiowej dla środowiska informatycznego

Główna teza tej prezentacji

H. Rasiowa & R. Sikorski

Mathematics of metamathematics

1963, PWN, Warszawa

519 stron

Wpływ tej książki na badania w podstawach matematyki, informatyki i logiki jest olbrzymi. Jesteśmy przekonani, że następne pokolenia będą ją odkrywać na nowo.

rozdział 1

W r. 1963 prof. Rasiowa sformułowała twierdzenie, które kilka lat później zostało nazwane aksjomatem instrukcji przypisania.

Aksjomat instrukcji przypisania

W książce *Mathematics of metamathematics* na str. 232 znajdujemy takie twierdzenie

Twierdzenie 1

Dla każdego podstawienia s, dla każdej formuły otwartej α i dla każdego wartościowania v zmiennych zachodzi równość

$$\alpha_R(s_R(v)) = \overline{s\,\alpha}_R(v).$$

Co to znaczy?

Objaśniamy

s - jest funkcją ze zbioru zmiennych w zbiór termów,
α - jest formułą bez kwantyfikatorów,
$\overline{s\alpha}$ - wyrażenie powstające z formuły α przez równoczesne zastąpienie wszystkich wystąpień zmiennych odpowiednimi termami,

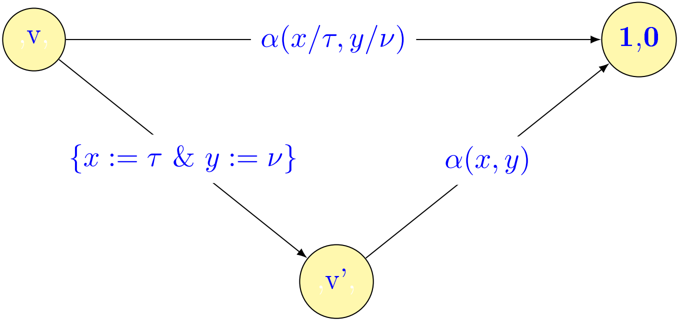
Przykład 1

$\alpha : xy + 3x - 66y - 198 > z$
$s : \{x := 70 \ \& \ y := -3\}$
$\overline{s\alpha} : 70 * (-3) + 3 * 70 - 66 * (-3) - 198 > z$

v - jest odwzorowaniem ze zbioru zmiennych w zbiór wartości, np. liczb całkowitych,
R - określa w jaki sposób rozumiemy funktory np. $+$ i *, oraz predykaty np. $=$ i $<$.

Twierdzenie powyższe stwierdza przemienność diagramu.

diagram



Rysunek: $\alpha_R(s_R(v)) = \overline{s\,\alpha}_R(v)$

Mówiąc po ludzku, warunek $\alpha(x, y)$ zachodzi po wykonaniu instrukcji {x:=τ & y:=ν } tj. w stanie pamięci v' zmienionym przez wykonanie tej instrukcji, wtedy i tylko wtedy gdy w początkowym stanie pamięci v zachodzi formuła $\alpha(x/\tau, y/\nu)$. Formułę tę otrzymuje się przez równoczesne zastąpienie wszystkich wolnych wystąpień zmiennych x i y w α przez termy τ i ν, odpowiednio.

Jak to odczytać?

Dla każdej realizacji języka programowania R i dla każdego stanu pamięci v,
po wykonaniu instrukcji przypisania $x := \tau \,\&\, y := \nu$ zachodzi warunek $\alpha(x, y)$ wtedy i tylko wtedy gdy przed wykonaniem tej instrukcji zachodzi warunek $\alpha(x/\tau, y/\nu)$.
Zapiszemy to w rachunku AL (od 1969) w ten sposób:

$$\{x := \tau \,\&\, y := \nu\}\alpha(x, y) \Leftrightarrow \alpha(x/\tau, y/\nu)$$

W rachunku Hoare'a (od 1969) zapisuje się to tak

$$\{\alpha(x/\tau, y/\nu)\}[x := \tau \,\&\, y := \nu]\{\alpha(x, y)\}$$

U Dijkstry (od 1974)

$$wp(\alpha(x, y), [x := \tau \,\&\, y := \nu]) = \alpha(x/\tau, y/\nu)$$

itd.

Co z tego wynika?

Co z tego wynika?

1. Aksjomat ten wystarczy do badania własności programów liniowych, tzn. ciągów instrukcji przypisania.

Co z tego wynika?

1. Aksjomat ten wystarczy do badania własności programów liniowych, tzn. ciągów instrukcji przypisania.
2. Umożliwia dowód poprawności kompilatora programów liniowych.

Co z tego wynika?

1. Aksjomat ten wystarczy do badania własności programów liniowych, tzn. ciągów instrukcji przypisania.
2. Umożliwia dowód poprawności kompilatora programów liniowych.
3. **Tw**. Instrukcje przypisania tworzą półgrupę.

Co z tego wynika?

1. Aksjomat ten wystarczy do badania własności programów liniowych, tzn. ciągów instrukcji przypisania.
2. Umożliwia dowód poprawności kompilatora programów liniowych.
3. **Tw**. Instrukcje przypisania tworzą półgrupę.
4. Własności tej półgrupy są wykorzystywane przez optymalizator – część kompilatora.

Co z tego wynika?

1. Aksjomat ten wystarczy do badania własności programów liniowych, tzn. ciągów instrukcji przypisania.
2. Umożliwia dowód poprawności kompilatora programów liniowych.
3. **Tw**. Instrukcje przypisania tworzą półgrupę.
4. Własności tej półgrupy są wykorzystywane przez optymalizator – część kompilatora.
5. **Tw**. Każdy komputer (lub maszyna wirtualna), który gwarantuje prawdziwość aksjomatu przypisania musi działać tak samo: *oblicz wartość termu τ, przypisz ją zmiennej x*.

wypowiedź prof. Ewy Orłowskiej

Prof. Rasiowa interesowała się automatycznym dowodzeniem twierdzeń.

H. Rasiowa i automatyczne dowodzenie twierdzeń

- W r. 1960 prof. Rasiowa w pracy wspólnej z R. Sikorskim zawarli nowy dowód twierdzenia Gentzena o eliminacji reguły cięcia z dowodów. Metoda zaproponowana przez nich jest metodą dualną do metody tablic semantycznych W. Betha. Praca ta zainspirowała wiele badań, których przegląd można znaleźć w monografii

  E. Orłowska and J. Golińska-Pilarek, Dual tableau: Foundations, Methodology, Case Studies. Trends in Logic 33, Springer, 2011

H. Rasiowa i automatyczne dowodzenie twierdzeń

- W r. 1960 prof. Rasiowa w pracy wspólnej z R. Sikorskim zawarli nowy dowód twierdzenia Gentzena o eliminacji reguły cięcia z dowodów. Metoda zaproponowana przez nich jest metodą dualną do metody tablic semantycznych W. Betha. Praca ta zainspirowała wiele badań, których przegląd można znaleźć w monografii
  E. Orłowska and J. Golińska-Pilarek, Dual tableau: Foundations, Methodology, Case Studies. Trends in Logic 33, Springer, 2011
- Podczas seminariów prowadzonych przez prof. Rasiową i prof. Pawlaka studiowano metodę rezolucji zaproponowaną w 1965 przez Alana Robinsona.

H. Rasiowa i automatyczne dowodzenie twierdzeń

- W r. 1960 prof. Rasiowa w pracy wspólnej z R. Sikorskim zawarli nowy dowód twierdzenia Gentzena o eliminacji reguły cięcia z dowodów. Metoda zaproponowana przez nich jest metodą dualną do metody tablic semantycznych W. Betha. Praca ta zainspirowała wiele badań, których przegląd można znaleźć w monografii
  E. Orłowska and J. Golińska-Pilarek, Dual tableau: Foundations, Methodology, Case Studies. Trends in Logic 33, Springer, 2011
- Podczas seminariów prowadzonych przez prof. Rasiową i prof. Pawlaka studiowano metodę rezolucji zaproponowaną w 1965 przez Alana Robinsona.
- W 1967 roku Helena Rasiowa zaproponowała mi podjęcie studiów doktoranckich na temat systemów automatycznego wnioskowania. Okres pracy w Jej Katedrze, a później Zakładzie Logiki (1967-1980) wspominam z ogromną wdzięcznością. Pod Jej kierunkiem zrobiłam doktorat (1971) i habilitację (1978). Przez cały ten czas poświęcała mi wiele godzin na indywidualne dyskusje, nie tylko w katedrze, ale często także u Niej w domu na Wiejskiej.

Lemat Rasiowej i Sikorskiego zob. tw. II.9.3 str. 87

Twierdzenie 2

Niech $\boldsymbol{A}$ *będzie algebrą Boole'a. Niech* (Q) *będzie przeliczalnym zbiorem nieskończonych działań*

$$(Q) \quad \begin{array}{l} \{a_t = \bigcup_{s \in S} a_{t\ s}\}_{t \in T} \\ \{b_u = \bigcap_{r \in R} a_{t\ r}\}_{u \in U} \end{array}$$

gdzie T *i* U *są pewnymi zbiorami* ***przeliczalnymi****.*

Dla każdego niezerowego elementu $a \neq 0 \in \boldsymbol{A}$ *istnieje* Q*-filtr zawierający ten element.*

Ale, co to jest Q-filtr?

Ale, co to jest Q-filtr?

- *Filtrem* w algebrze Boole'a $\mathbf{A}$ jest każdy podzbiór $\nabla \subset \mathbf{A}$, który spełnia następujące warunki
  - jeśli $a, b \in \nabla$, to $a \cap b \in \nabla$,
  - jeśli $a \in \nabla$ i $a \leq b$, to $b \in \nabla$.

Ale, co to jest Q-filtr?

- *Filtrem* w algebrze Boole'a $\mathbf{A}$ jest każdy podzbiór $\nabla \subset \mathbf{A}$, który spełnia następujące warunki
  - jeśli $a, b \in \nabla$, to $a \cap b \in \nabla$,
  - jeśli $a \in \nabla$ i $a \leq b$, to $b \in \nabla$.
- Filtr ∇ jest filtrem *pierwszym*, jeśli nie zawiera elementu $\mathbf{0}$ i ponadto spełniony jest następujący warunek
  jeśli $a \cup b \in \nabla$, to albo $a \in \nabla$ albo $b \in \nabla$.

Ale, co to jest Q-filtr?

- *Filtrem* w algebrze Boole'a $\mathbf{A}$ jest każdy podzbiór $\nabla \subset \mathbf{A}$, który spełnia następujące warunki
  - jeśli $a, b \in \nabla$, to $a \cap b \in \nabla$,
  - jeśli $a \in \nabla$ i $a \leq b$, to $b \in \nabla$.
- Filtr ∇ jest filtrem *pierwszym*, jeśli nie zawiera elementu $\mathbf{0}$ i ponadto spełniony jest następujący warunek
  jeśli $a \cup b \in \nabla$, to albo $a \in \nabla$ albo $b \in \nabla$.
- Filtr ∇ jest *Q-filtrem* jeśli jest filtrem pierwszym i ponadto zachowuje działania nieskończone (Q), tj.
  dla każdego $t \in T$ oraz dla każdego $u \in U$ zachodzą implikacje
  - jeśli $a_t \in \nabla$, to istnieje $s \in S$ takie, że $a_{ts} \in \nabla$,
  - jeśli dla każdego $r \in R$, $b_{ur} \in \nabla$, to $b_u \in \nabla$.

Dlaczego ten lemat(to twierdzenie) jest ważny?

Ponieważ otwiera drogę do algebraicznego dowodu twierdzenia o pełności rachunku predykatów.
Analiza metamatematycznych własności rachunku zdań może przebiegać w ten sposób:

1. W zbiorze formuł wprowadzamy relację równoważności $\alpha \approx \beta$ wttw obie formuły $\alpha \implies \beta$ i $\beta \implies \alpha$ są twierdzeniami.
2. Klasy równoważności tworza algebrę Boole'a (niekoniecznie dwuelementową), nazywamy ją *algebrą Lindenbauma*.
3. W każdej algebrze Boole'a, dla każdego elementu niezerowego a istnieje filtr pierwszy ∇ zawierający element a.
4. Wynika stąd istnienie homomorfizmu z algebry Lindenbauma w dwuelementową algebrę Boole'a takiego, że wartość a jest równa $\mathbf{1}$.
5. **Wniosek**: formuła nie będąca twierdzeniem może zostać sfalsyfikowana, czyli formuła prawdziwa posiada dowód tzn. jest twierdzeniem. $\square$

Dlaczego ten lemat(to twierdzenie) jest ważny? II

W rachunku predykatów napotykamy trudności:

- w algebrze Lindenbauma dowolnej teorii, kwantyfikatory są działaniami nieskończonymi,
- na ogół nie istnieją filtry pierwsze zachowujące działania nieskończone,
- Halmos, Tarski i in. podejmowali próby wprowadzając nowe algebry (algebry polyadyczne - Halmos, algebry cylindryczne - Tarski),
- Rasiowa i Sikorski (w r. 1950) zauważyli, że działań nieskończonych kresów górnych i kresów dolnych w algebrze Lindenbauma jest przeliczalnie wiele
- udowodnili lemat (metodami topologicznymi),
- udowodnili (w sposób algebraiczny) twierdzenie o pełności klasycznego rachunku zdań.

Tw. Dla każdego niezerowego elementu $a \neq 0 \in \mathbf{A}$ istnieje Q-filtr zawierający ten element.

Fakt 1. Załóżmy, że $c \neq 0$ oraz $a_i = \bigcup_{a \in A_i} a$. Wykażemy, że istnieje $a_i^* \in A_i$, takie że

$$c \cap (a_i \Rightarrow a_i^*) \neq \mathbf{0}.$$

Dowód. Przypuśćmy przeciwnie, że dla dowolnego $a \in A_i$, zachodzi $c \cap (a_i \Rightarrow a) = \mathbf{0}$. Wtedy {pamiętaj, że $(b \Rightarrow d = -b \cup d)$}

(1) $c \cap -a_i = \mathbf{0}$

(2) $c \cap a = \mathbf{0}$ dla wszystkich $a \in A_i$.

Z (2) mamy $\bigcup_{a \in A_i} (c \cap a) = \mathbf{0}$ i w konsekwencji

(3) $c \cap a_i = \mathbf{0}$.

Z (1) i (3) wynika, że $c = \mathbf{0}$, co jest sprzeczne z założeniem. □

Konstrukcja Q-filtru

Niech (Q) będzie przeliczalnym zbiorem działań nieskończonych

$$(Q) \quad a_1 = \bigcup_{a \in A_1} a, \quad a_2 = \bigcup_{a \in A_2} a, \quad \ldots \quad a_i = \bigcup_{a \in A_i} a \quad \ldots$$

Konstrukcja Q-filtru

Niech (Q) będzie przeliczalnym zbiorem działań nieskończonych

$$(Q) \quad a_1 = \bigcup_{a \in A_1} a, \quad a_2 = \bigcup_{a \in A_2} a, \quad \ldots \quad a_i = \bigcup_{a \in A_i} a \quad \ldots$$

Wykorzystując Fakt 1, konstruujemy rosnący ciąg $\{C_i\}$ skończonych podzbiorów zbioru A, taki że $C_0 = \{a_0\}$ i dla każdej liczby naturalnej i, $C_i = C_{i-1} \cup \{ \bigcup_{a \in A_i} a \Rightarrow a_i^*\}$ i taki, że przecięcie wszystkich elementów zbioru C_i jest różne od zera $\bigcap C_i \neq 0$.

Konstrukcja Q-filtru

Niech (Q) będzie przeliczalnym zbiorem działań nieskończonych

$$(Q) \quad a_1 = \bigcup_{a \in A_1} a, \quad a_2 = \bigcup_{a \in A_2} a, \quad \dots \quad a_i = \bigcup_{a \in A_i} a \quad \dots$$

Wykorzystując Fakt 1, konstruujemy rosnący ciąg $\{C_i\}$ skończonych podzbiorów zbioru A, taki że $C_0 = \{a_0\}$ i dla każdej liczby naturalnej i, $C_i = C_{i-1} \cup \{ \bigcup_{a \in A_i} a \Rightarrow a_i^* \}$ i taki, że przecięcie wszystkich elementów zbioru C_i jest różne od zera $\bigcap C_i \neq 0$.

Fakt 2. Zbiór $C = \bigcup_{i \in N} C_i$ ma własność skończonych przecięć.

Konstrukcja Q-filtru

Niech (Q) będzie przeliczalnym zbiorem działań nieskończonych

$$(Q) \quad a_1 = \bigcup_{a \in A_1} a, \quad a_2 = \bigcup_{a \in A_2} a, \quad \dots \quad a_i = \bigcup_{a \in A_i} a \quad \dots$$

Wykorzystując Fakt 1, konstruujemy rosnący ciąg $\{C_i\}$ skończonych podzbiorów zbioru A, taki że $C_0 = \{a_0\}$ i dla każdej liczby naturalnej i, $C_i = C_{i-1} \cup \{ \bigcup_{a \in A_i} a \Rightarrow a_i^* \}$ i taki, że przecięcie wszystkich elementów zbioru C_i jest różne od zera $\bigcap C_i \neq 0$.

Fakt 2. Zbiór $C = \bigcup_{i \in N} C_i$ ma własność skończonych przecięć.

Fakt 3. Istnieje filtr maksymalny ∇ będący rozszerzeniem zbioru C.

Konstrukcja Q-filtru

Niech (Q) będzie przeliczalnym zbiorem działań nieskończonych

$$(Q) \quad a_1 = \bigcup_{a \in A_1} a, \quad a_2 = \bigcup_{a \in A_2} a, \quad \dots \quad a_i = \bigcup_{a \in A_i} a \quad \dots$$

Wykorzystując Fakt 1, konstruujemy rosnący ciąg $\{C_i\}$ skończonych podzbiorów zbioru A, taki że $C_0 = \{a_0\}$ i dla każdej liczby naturalnej i, $C_i = C_{i-1} \cup \{ \bigcup_{a \in A_i} a \Rightarrow a_i^* \}$ i taki, że przecięcie wszystkich elementów zbioru C_i jest różne od zera $\bigcap C_i \neq 0$.

Fakt 2. Zbiór $C = \bigcup_{i \in N} C_i$ ma własność skończonych przecięć.

Fakt 3. Istnieje filtr maksymalny ∇ będący rozszerzeniem zbioru C.
Fakt 4. Filtr ∇ jest $Q-$filtrem.

Konstrukcja Q-filtru

Niech (Q) będzie przeliczalnym zbiorem działań nieskończonych

$$(Q) \quad a_1 = \bigcup_{a \in A_1} a, \quad a_2 = \bigcup_{a \in A_2} a, \quad \ldots \quad a_i = \bigcup_{a \in A_i} a \quad \ldots$$

Wykorzystując Fakt 1, konstruujemy rosnący ciąg $\{C_i\}$ skończonych podzbiorów zbioru A, taki że $C_0 = \{a_0\}$ i dla każdej liczby naturalnej i, $C_i = C_{i-1} \cup \{\bigcup_{a \in A_i} a \Rightarrow a_i^*\}$ i taki, że przecięcie wszystkich elementów zbioru C_i jest różne od zera $\bigcap C_i \neq 0$.

Fakt 2. Zbiór $C = \bigcup_{i \in N} C_i$ ma własność skończonych przecięć.

Fakt 3. Istnieje filtr maksymalny ∇ będący rozszerzeniem zbioru C.
Fakt 4. Filtr ∇ jest $Q-$filtrem.

Rzeczywiście, weźmy $a_i = \bigcup_{a \in A_i} a \in \nabla$, wtedy $(\bigcup_{a \in A_i} a \Rightarrow a_i^*) \in C$, a więc należy też do ∇. Ale $a_i \cap (a_i \Rightarrow a_i^*) \leq a_i^*$, zatem $a_i^* \in \nabla$. $\square$

Lemat R-S w logice algorytmicznej

Tylko metoda algebraiczna zaproponowana przez Rasiową i Sikorskiego może zostać wykorzystana w dowodzie twierdzenia o pełności rachunku programów, ponieważ w rachunku programów są działania nieskończone, tzw. kwantyfikatory iteracji,

$$\bigcap M\alpha \text{ – dla każdej iteracji programu } M \text{ zachodzi formuła } \alpha$$

$$\bigcup M\alpha \text{ – istnieje iteracja programu } M \text{ taka, że zachodzi formuła } \alpha,$$

które nie sprowadzają się do kwantyfikatorów klasycznych.

język AL =/= język pierwszego rzędu

W języku logiki pierwszego rzędu (i teorii elementarnych) mamy *termy* i *formuły*.

W języku logiki algorytmicznej mamy termy, formuły i *programy*. Co więcej formuły mogą być budowane przy pomocy programów. Np. formuła

$$n > m > 0 \implies \left\{ \begin{array}{l} t := m; \\ \textbf{while } t \leq n \textbf{ do } t := t + m \textbf{ od} \end{array} \right\} (t > n)$$

wyraża prawo Archimedesa - własność niewyrażalną w języku logiki pierwszego rzędu.

język AL =/= język pierwszego rzędu

W języku logiki pierwszego rzędu (i teorii elementarnych) mamy *termy* i *formuły*.

W języku logiki algorytmicznej mamy termy, formuły i *programy*. Co więcej formuły mogą być budowane przy pomocy programów. Np. formuła

$$\underbrace{n > m > 0}_{formula} \implies \underbrace{\left\{ \begin{array}{l} t := m; \\ \textbf{while } t \leq n \textbf{ do } t := t + m \textbf{ od} \end{array} \right\}}_{program} \underbrace{(t > n)}_{formula}$$

wyraża prawo Archimedesa - własność niewyrażalną w języku logiki pierwszego rzędu.

W języku logiki pierwszego rzędu (i teorii elementarnych) mamy *termy* i *formuły*.

W języku logiki algorytmicznej mamy termy, formuły i *programy*. Co więcej formuły mogą być budowane przy pomocy programów. Np. formuła

$$\underbrace{n > m > 0}_{formula} \implies \underbrace{\left\{ \begin{array}{l} t := m; \\ \textbf{while } t \le n \textbf{ do } t := t + m \textbf{ od} \end{array} \right\}}_{program} \underbrace{(t > n)}_{formula}$$

wyraża prawo Archimedesa - własność niewyrażalną w języku logiki pierwszego rzędu.
To samo prawo zapisane inaczej

$$n > m > 0 \implies t := m; \bigcup \textbf{if } t \le n \textbf{ then } t := t + m \textbf{ fi} \, (t > n)\}$$

przy pomocy kwantyfikatora iteracji.

wzajemne usytuowanie 4 rachunków logicznych

program calculus AL

$$\mathcal{WFF}_{AL} = \{\mathcal{T}_{AL} \cup \mathcal{F}_{AL} \cup \mathcal{P}_{AL}\}$$

$$\mathcal{F}_{FOL} \subsetneq \mathcal{F}_{AL}$$

predicate calculus FOL

$$\mathcal{WFF}_{FOL} = \{\mathcal{T}_{FOL} \cup \mathcal{F}_{FOL}\}$$

calculus of program schemes PAL

$$\mathcal{WFF}_{PAL} = \{\mathcal{F}_{PAL} \cup \mathcal{P}_{PAL}\}$$

propositional calculus PL

$$\mathcal{WFF}_{PL} = \{\mathcal{F}_{PL}\}$$

Prace H. Rasiowej dot. logiki algorytmicznej

Prace H. Rasiowej dot. logiki algorytmicznej

- Algorithmic Logic - wykłady w Simon Fraser University, 1975, 206 stron

Prace H. Rasiowej dot. logiki algorytmicznej

- Algorithmic Logic - wykłady w Simon Fraser University, 1975, 206 stron
- wspólna z doktorantami publikacja w Banach Center Publications vol. 2, 1977, pp.7-99

Prace H. Rasiowej dot. logiki algorytmicznej

- Algorithmic Logic - wykłady w Simon Fraser University, 1975, 206 stron
- wspólna z doktorantami publikacja w Banach Center Publications vol. 2, 1977, pp.7-99
- prace nt. wielowartościowych logik algorytmicznych (w algebrach Posta)

Prace H. Rasiowej dot. logiki algorytmicznej

- Algorithmic Logic - wykłady w Simon Fraser University, 1975, 206 stron
- wspólna z doktorantami publikacja w Banach Center Publications vol. 2, 1977, pp.7-99
- prace nt. wielowartościowych logik algorytmicznych (w algebrach Posta)
- prace nt. współprogramów w AL

Prace H. Rasiowej dot. logiki algorytmicznej

- Algorithmic Logic - wykłady w Simon Fraser University, 1975, 206 stron
- wspólna z doktorantami publikacja w Banach Center Publications vol. 2, 1977, pp.7-99
- prace nt. wielowartościowych logik algorytmicznych (w algebrach Posta)
- prace nt. współprogramów w AL
- w sumie 16 prac poświęconych różnym zagadnieniom logiki algorytmicznej

Zasługi Prof. Rasiowej dla społeczności informatycznej

Zasługi Prof. Rasiowej dla społeczności informatycznej

- Poddała pomysł konferencji MFCS'72 i wielokrotnie uczestniczyła w kolejnych konferencjach Mathematical Foundations of Computer Science.

Zasługi Prof. Rasiowej dla społeczności informatycznej

- Poddała pomysł konferencji MFCS'72 i wielokrotnie uczestniczyła w kolejnych konferencjach Mathematical Foundations of Computer Science.
- Stworzyła, nie szczędząc pracy i zachodów, jedyne polskie naukowe czasopismo informatyczne *Fundamenta Informaticae* i była jego naczelną przez wiele lat.

Zasługi Prof. Rasiowej dla społeczności informatycznej

- Poddała pomysł konferencji MFCS'72 i wielokrotnie uczestniczyła w kolejnych konferencjach Mathematical Foundations of Computer Science.
- Stworzyła, nie szczędząc pracy i zachodów, jedyne polskie naukowe czasopismo informatyczne *Fundamenta Informaticae* i była jego naczelną przez wiele lat.
- Była wieloletnią przewodniczącą rady Naukowej IPI PAN.

Zasługi Prof. Rasiowej dla społeczności informatycznej

- Poddała pomysł konferencji MFCS'72 i wielokrotnie uczestniczyła w kolejnych konferencjach Mathematical Foundations of Computer Science.
- Stworzyła, nie szczędząc pracy i zachodów, jedyne polskie naukowe czasopismo informatyczne *Fundamenta Informaticae* i była jego naczelną przez wiele lat.
- Była wieloletnią przewodniczącą rady Naukowej IPI PAN.
- Zainspirowała spotkania informatyków z Warszawy i Berlina, które pod nazwą sympozja CS&P trwają do dziś.

Zasługi Prof. Rasiowej dla społeczności informatycznej

- Poddała pomysł konferencji MFCS'72 i wielokrotnie uczestniczyła w kolejnych konferencjach Mathematical Foundations of Computer Science.
- Stworzyła, nie szczędząc pracy i zachodów, jedyne polskie naukowe czasopismo informatyczne *Fundamenta Informaticae* i była jego naczelną przez wiele lat.
- Była wieloletnią przewodniczącą rady Naukowej IPI PAN.
- Zainspirowała spotkania informatyków z Warszawy i Berlina, które pod nazwą sympozja CS&P trwają do dziś.
- Dwukrotnie organizowała semestr informatyczny w Centrum Banacha.

Koniec

Dziękujemy za uwagę.