Wprowadzenie do nulli

Jakub Wilk

Wydział Matematyki, Informatyki i Mechaniki Uniwersytetu Warszawskiego

7 grudnia 2006 r.

Null ($\perp$) — specjalna „wartość”, która będzie tu znaczyć *wartość jest, ale nieznana*.

Definicje

- Każda krotka jest **częściowa** (*partial*).
- Krotka t jest **określona** (*definite*) na atrybucie A:
  $t(A)\downarrow \overset{\text{def}}{\iff} t(A) \neq \perp$.
- Krotka t jest **określona** (*definite*) na atrybutach z X:
  $t(X)\downarrow \overset{\text{def}}{\iff} \forall_{A \in X}\ t(A)\downarrow$.
- Krotka t jest **całkowita** (*total*):
  $t\downarrow \overset{\text{def}}{\iff} \forall_A\ t(A)\downarrow$.
- Krotka t **ukonkretnia** (*subsumes*) krotkę u:
  $t \geqslant u \overset{\text{def}}{\iff} \forall_A\ (u(A)\downarrow \implies t(A) = u(A))$.

Definicje

- Każda relacja jest **częściowa** (*partial*).
- Zbiór wszystkich relacji częściowych o schemacie R oznaczamy przez $\mathrm{Rel}{\uparrow}(R)$.
- Relacja r jest **całkowita** (*total*):
  $r{\downarrow} \stackrel{\text{def}}{\iff} \forall_{t \in r}\, t{\downarrow}$.
- Zbiór wszystkich relacji całkowitych o schemacie R:
  $\mathrm{Rel}(R) \stackrel{\text{def}}{=} \{r \in \mathrm{Rel}{\uparrow}(R) \colon r{\downarrow}\}$.

Definicje

- Relacja r **ukonkretnia** (*subsumes*) relację s:
  $r \geqslant s \overset{\text{def}}{\iff} \forall_{t_s \in s} \exists_{t_r \in r}\, t_r \geqslant t_s.$
- Relacja r jest **rozszerzeniem** (*extension*) relacji s:
  $r \geqslant\!\downarrow s \overset{\text{def}}{\iff} (r \geqslant s) \wedge r\!\downarrow.$
- Relacja r **wzmacnia** (*augments*) relację s:
  $r \succcurlyeq s \overset{\text{def}}{\iff} \exists_f\ (r = \{f(t_s) \colon t_s \in s\} \wedge \forall_{t_s \in s} f(t_s) \geqslant f(s))$
- Relacja r jest **uzupełnieniem** (*completion*) relacji s:
  $r \succcurlyeq\!\downarrow s \overset{\text{def}}{\iff} (r \succcurlyeq s) \wedge r\!\downarrow.$

Przykład

r	YEAR	AUTHOR	NAME
	1957	$\perp$	FORTRAN
	1960	John McCarthy	LISP
	1975	Niklaus Wirth	$\perp$
	$\perp$	$\perp$	Scheme

Możemy spojrzeć na taką relację częściową jak na zbiór następujących aksjomatów:

$$\begin{aligned}
\exists_{x_1}\ & \mathrm{r}(1957, x_1, \mathsf{FORTRAN}) \\
& \mathrm{r}(1960, \mathsf{John\ McCarthy}, \mathsf{LISP}) \\
\exists_{x_2}\ & \mathrm{r}(1975, \mathsf{Niklaus\ Wirth}, x_2) \\
\exists_{x_3}\exists_{x_4}\ & \mathrm{r}(x_3, x_4, \mathsf{Scheme})
\end{aligned}$$

Przykład

r' (	YEAR	AUTHOR	NAME	)
	1957	John Backus	FORTRAN	
	1960	John McCarthy	LISP	
	1975	Niklaus Wirth	$\perp$	
	1975	$\perp$	Scheme	

$r' \succcurlyeq r \iff$ aksjomaty dla r' powstały poprzez ukonkretnienie aksjomatów dla r:

$$\begin{aligned} &\mathrm{r}'(1957, \text{John Backus}, \text{FORTRAN}) \\ &\mathrm{r}'(1960, \text{John McCarthy}, \text{LISP}) \\ \exists_{x_2}\ &\mathrm{r}'(1975, \text{Niklaus Wirth}, x_2) \\ \exists_{x_4}\ &\mathrm{r}'(1975, x_4, \text{Scheme}) \end{aligned}$$

Przykład

r	YEAR	AUTHOR	NAME
	1957	$\perp$	FORTRAN
	1960	John McCarthy	LISP
	1975	Niklaus Wirth	$\perp$
	$\perp$	$\perp$	Scheme

Przykład

r''	(	YEAR	AUTHOR	NAME	)
		1957	John Backus	FORTRAN	
		1960	John McCarthy	LISP	
		1975	Niklaus Wirth	Pascal	
		1975	Guy L. Steele	Scheme	
		$\perp$	G. J. Sussman	Scheme	

$r'' \geqslant r \iff$ aksjomaty dla r'' logicznie implikują aksjomaty dla r.
$r'' \geqslant_{\downarrow} r \iff r''$ jest skończonym modelem aksjomatów dla r.

Definicje

φ — formuła rachunku krotek nad relacją $r(R)$, bez zmiennych wolnych.
$I_s(\varphi)$ ($s \in \mathrm{Rel}(R)$) — zwyczajna interpretacja formuły nad $s{\downarrow}$.
Interpretacją formuły φ jest:

$$\hat{I}(\varphi) = \begin{cases} \text{true} & \text{jeśli } \forall_{s \succeq {\downarrow} r}\ I_s(\varphi) = \text{true} \\ \text{false} & \text{jeśli } \forall_{s \succeq {\downarrow} r}\ I_s(\varphi) = \text{false} \\ \text{unknown} & \text{wpp.} \end{cases}$$

Nie da się zdefiniować $\hat{I}(\varphi)$ w terminach wartości $\hat{I}$ dla podformuł φ.

Definicje

Relacja r jest **dopuszczalna** (*permissible*) przez zależności funkcyjne F $\overset{\text{def}}{\iff}$ $\exists_{s \succcurlyeq\downarrow r}$ (s spełnia F). O relacji s mówimy, że jest **dopuszczalnym uzupełnieniem** względem F.

Przykład

r (

YEAR	AUTHOR	NAME
1957	John Backus	$\perp$
$\perp$	John McCarthy	LISP

)

r'(

YEAR	AUTHOR	NAME
1957	John Backus	FORTRAN
1960	John McCarthy	LISP

)

r''(

YEAR	AUTHOR	NAME
1957	John Backus	LISP
1960	John McCarthy	LISP

)

Jeżeli $F = \{\text{NAME} \to \text{YEAR}\}$ to r' jest dopuszczalnym a r'' niedopuszczalnym uzupełnieniem r.

Przykład

- $$r\left(\begin{array}{cccc} \mathsf{A} & \mathsf{B} & \mathsf{C} & \mathsf{D} \\ \hline \mathsf{a} & \mathsf{b} & \perp & \perp \\ \mathsf{a} & \perp & \perp & \mathsf{d} \end{array}\right)$$
- $F = \{\mathsf{A} \to \mathsf{C}, \mathsf{C} \to \mathsf{D}\}$.
- $$r'\left(\begin{array}{cccc} \mathsf{A} & \mathsf{B} & \mathsf{C} & \mathsf{D} \\ \hline \mathsf{a} & \mathsf{b} & \perp & \mathsf{d} \\ \mathsf{a} & \perp & \perp & \mathsf{d} \end{array}\right)$$

Przykład

- r (

A	B	C	D
a	b	$\perp$	$\perp$
a	$\perp$	$\perp$	d

)

- $F = \{\mathsf{A} \to \mathsf{C}, \mathsf{C} \to \mathsf{D}\}$.

- r'(

A	B	C	D
a	b	$\perp_1$	d
a	$\perp_2$	$\perp_1$	d

)

Wprowadzamy **nulle znakowane** (*marked nulls*).

- $\mathrm{nchase}_F(r)$:
  1. Zamień każde wystąpienie nieznakowanego nulla na unikatowy znakowany null $\perp_i$.
  2. Dopóki $\exists_{r \ni t \neq t' \in r} \exists_{X \to A \in F}\ t(X) = t'(X) \wedge t(A) \neq t'(A)$:
     - 2.1 Jeśli $t(A){\downarrow} \wedge t'(A){\downarrow}$ to:
       zakończ z porażką.
     - 2.2 Jeśli $t(A){\downarrow} \wedge t'(A) = \perp_i$ to:
       $r := r[\perp_i / t(X)]$.
     - 2.3 Jeśli $t'(A){\downarrow} \wedge t(A) = \perp_i$ to:
       $r := r[\perp_i / t'(X)]$.
     - 2.4 Jeśli $t(A) = \perp_i \wedge t'(A) = \perp_j$ to:
       $r := r[\perp_{\max(i,j)} / \perp_{\min(i,j)}]$.
  3. Zakończ z sukcesem.
- Algorytm zawsze się kończy i daje poprawne wyniki (o ile dziedziny atrybutów są *nieskończone*).

Definicje

Niech $q \geqslant\downarrow r$.

- q jest **bliskim rozszerzeniem** (*close extension*) $r \overset{\text{def}}{\iff} \forall_{t_q \in q} \exists_{t_s \in s}\, t_q \geqslant t_r$.
- q jest **minimalnym rozszerzeniem** (*minimal extension*) $r \overset{\text{def}}{\iff} \neg\exists_{q' \subsetneq q}\, q' \geqslant r$.

Przykład

$r($

A	B	C
1	2	$\perp$
$\perp$	2	3
1	$\perp$	3

$)$ $q_{\text{close}}($

A	B	C
1	2	3
4	2	3
1	5	3

$)$ $q_{\text{min}}($

A	B	C
1	2	6
7	2	3
1	8	3

$)$

Definicje

- **Funkcja możliwości** (*possiblity function*) to funkcja $\mathrm{POSS}\colon \mathrm{Rel}{\uparrow}(R) \to 2^{\mathrm{Rel}(R)}$.
- POSS jest **rozsądna** (*reasonable*) $\overset{\text{def}}{\iff}$:
  1. $\forall_{q \in \mathrm{POSS}(r)}\ q \gtrsim{\downarrow}\ r$;
  2. q jest minimalnym rozszerzeniem $r \implies q \in \mathrm{POSS}(r)$;
  3. $s \in \mathrm{POSS}(r) \iff \mathrm{POSS}(s) \subseteq \mathrm{POSS}(r) \wedge s{\downarrow}$.
- POSS jest **domknięta** (*closed*) $\overset{\text{def}}{\iff} q \in \mathrm{POSS}(r) \implies q$ jest bliskim rozszerzeniem r.

Przykład

- $\mathrm{POSS}_{\mathsf{C}}(r) = \{s\colon s \succcurlyeq{\downarrow}\ r\}$ jest rozsądna i domknięta.
- $\mathrm{POSS}_{\mathsf{O}}(r) = \{s\colon s \gtrsim{\downarrow}\ r\}$ jest rozsądna, ale nie domknięta.

Definicje

Niech γ będzie n-argumentowym operatorem na relacjach całkowitych, a γ' – na relacjach częściowych.

- Operator γ' jest **wierny** (*faithful*) $\gamma \overset{\text{def}}{\iff} \forall_{r_1\downarrow,\ldots,r_n\downarrow}\, \gamma(r_1,\ldots,r_n) = \gamma'(r_1,\ldots,r_n)$
- Operator γ' jest **precyzyjnym** (*precise*) uogólnieniem $\gamma \overset{\text{def}}{\iff} \forall_{r_1,\ldots,r_n}\, \mathrm{POSS}(\gamma'(r_1,\ldots,r_n)) = \{\gamma(s_1,\ldots,s_n) : s_i \in \mathrm{POSS}(r_i)\}$

Warunek precyzyjności jest często *zbyt mocny*.

Definicje

Niech γ będzie n-argumentowym operatorem na relacjach całkowitch, a γ' – na relacjach częściowych.

- Operator γ' jest **wystarczający** (*adequate*) dla γ $\overset{\text{def}}{\Longleftrightarrow}$ $\forall_{r_1,\ldots,r_n}\ \mathrm{POSS}(\gamma'(r_1,\ldots,r_n)) \supseteq \{\gamma\,(s_1,\ldots,s_n) : s_i \in \mathrm{POSS}\,(r_i)\}$
- Operator γ' jest **ograniczony** (*restricted*) do γ $\overset{\text{def}}{\Longleftrightarrow}$ $\forall_{r_1,\ldots,r_n} \neg\exists_q\ \mathrm{POSS}(\gamma'(r_1,\ldots,r_n)) \supsetneq \mathrm{POSS}(q) \supseteq \{\gamma\,(s_1,\ldots,s_n) : s_i \in \mathrm{POSS}\,(r_i)\}$

Zadowolamy się uogólnieniami wystarczającymi i ograniczonymi. Czasem będziemy wymagać też innych właściwości pierwotnego operatora (np. łączności).

Przykład

Niech $r(R), s(S)$ będą relacjami częściowymi, $A \in R, C \in S, R \cap S = \varnothing$. Rozszerzamy equijoin $[A = C]$ do $[A = C]'$ w następujący sposób:

$$\begin{aligned} r[A = C]'s = \{t(RS)\colon & \\ & t(R) \in r, t(S) \in S, \\ & t(A) = t(C) \vee t(A) = \bot \vee t(C) = \bot \\ \}. & \end{aligned}$$

Problem: operator nie jest wystarczający dla żadnej rozsądnej POSS.

Przykład

$\bowtie^Z$: łączymy krotki, gdy dla każdego wspólnego atrybutu: albo się zgadzają albo dokładnie jeden jest $\perp$. Atrybuty z $\perp$ są wypełniane wartościami z drugiej krotki. Krotki nie pasujące do niczego są wypychane nullami i dodawane do wyniku.

Problem: operator nie jest łączny dla żadnej rozsądnej POSS.

$\mathrm{POSS}_{\mathsf{O}}(r) = \{s \colon s \geqslant\!\downarrow r\}$:

- Operator $\pi^{\mathsf{O}} = \pi$ jest precyzyjny.
- Operator $\cup^{\mathsf{O}} = \cup$ jest precyzyjny.
- Nie istnieją precyzyjne uogólnienia σ, ale istnieją wystarczające i ograniczone:
  - $\sigma^{\mathsf{O}}_{A=a}(r) = \{t \in r \colon t(A) = a\}$;
  - $\sigma^{\mathsf{O}}_{A=B}(r) = \{t \in r \colon t(A)\downarrow, t(B)\downarrow, t(A) = t(B)\}$;
- Nie istnieje precyzyjne uogólnienie $\bowtie$.
- $r \bowtie^{\mathsf{O}} s = \{t(RS) \colon \exists_{t_r \in r, t_s \in s}\, t_r(X)\downarrow, t_s(X)\downarrow, t(R) = t_r, t(S) = t_s\}$, gdzie $X = R \cap S$; operator $\bowtie^{\mathsf{O}}$ jest wystarczający i ograniczony.

$\mathrm{POSS}_{\mathsf{CE}}(r) = \{s\colon s \text{ jest bliskim rozserzeniem } r\}$:

- Operator $\pi^{\mathsf{CE}} = \pi$ jest wystarczający i ograniczony.
- Operator $\cup^{\mathsf{CE}} = \cup$ jest wystarczający i ograniczony.
- Nie istnieją wystarczające uogólnienia σ.
- Nie istnieje wystarczające uogólnienie $\bowtie$.

$\mathrm{POSS}_{\mathsf{CE}}(r) = \{s\colon s$ jest bliskim rozserzeniem $r\}$:

- Operator $\pi^{\mathsf{CE}} = \pi$ jest wystarczający i ograniczony.
- Operator $\cup^{\mathsf{CE}} = \cup$ jest wystarczający i ograniczony.
- Nie istnieją wystarczające uogólnienia σ.
- Nie istnieje wystarczające uogólnienie $\bowtie$.

$\mathrm{POSS}_{\mathsf{C}}(r) = \{s\colon s \succcurlyeq_{\downarrow} r\}$:

- Nie istnieją wystarczające uogólnienia σ.

$\mathrm{POSS}_{\mathsf{ME}}(r) = \{s\colon s$ jest minimalnym rozserzeniem $r\}$:

- Nie istnieją wystarczające uogólnienia σ.

Definicje

- Dzielimy krotki relacji na **pewne** ($\mathrm{SURE}(r)$) i **niepewne** ($\mathrm{MAYBE}(r)$). Taką relację nazywamy **podzieloną** (*paritioned*).
- s **przybliża** (*approximate*) r: $s \rhd r \stackrel{\text{def}}{\Longleftrightarrow} r \supseteq s \supseteq \mathrm{SURE}(r)$.
- $\mathrm{POSS}_{\mathsf{B}}(r) = \{q \colon q$ jest bliskim rozszerzeniem pewnego $s \rhd r\}$.

- $\pi^{\mathsf{B}}_X(r) = s$, gdzie:

$$\mathrm{SURE}(s) = \{t(X) \colon t \in \mathrm{SURE}(r)\}\,;$$
$$\mathrm{MAYBE}(s) = \{t(X) \colon t \in \mathrm{MAYBE}(r)\}\,.$$

- Operator π^{B} jest wystarczający i ograniczony.

- $r \cup^{\mathsf{B}} s = q$, gdzie:

$$\mathrm{SURE}(q) = \mathrm{SURE}(r) \cup \mathrm{SURE}(s);$$
$$\mathrm{MAYBE}(q) = \mathrm{MAYBE}(r) \cup \mathrm{MAYBE}(s).$$

- Operator $\cup^{\mathsf{B}}$ jest precyzyjny.

- $\sigma^{\mathsf{B}}_{A=a}(r(R)) = s$, gdzie:

$$\begin{aligned}
\mathrm{SURE}(s) &= \{t\colon t \in \mathrm{SURE}(r), t(A) = a\}\,; \\
\mathrm{MAYBE}(s) &= \{t\colon t \in \mathrm{MAYBE}(r), t(A) = a\} \cup \\
&\quad \{t'\colon \exists_{t \in r}\ t(R_A) = t'(R_A), t(A) = \bot, t'(A) = a\}\,; \\
R_A &= R \setminus A.
\end{aligned}$$

- Operator $\sigma^{\mathsf{B}}_{A=a}$ jest precyzyjny.

- $\sigma^{\mathsf{B}}_{A=B}(r(R)) = s$, gdzie:

$$\begin{aligned}
\mathrm{SURE}(s) &= \{t \colon t \in \mathrm{SURE}(r), t(A) = t(B)\downarrow\}\,; \\
\mathrm{MAYBE}(s) &= \Big\{t' \colon \exists_{t \in r} \\
&\qquad t'(R \setminus AB) = t(R \setminus AB), \\
&\qquad t'(A) = t'(B) = \begin{cases} t(A) & \text{jeśli } t(A)\downarrow, \\ t(B) & \text{wpp.} \end{cases} \\
&\quad \Big\} \setminus \mathrm{SURE}(s).
\end{aligned}$$

- Operator $\sigma^{\mathsf{B}}_{A=B}$ jest precyzyjny.

- $r(R) \bowtie^{\mathsf{B}} s(S) = q(RS)$, gdzie:

$$
\begin{aligned}
\mathrm{SURE}(q) &= \{t\colon \exists_{t_r \in \mathrm{SURE}(r), t_s \in \mathrm{SURE}(s)} \\
&\qquad t_r(X)\downarrow, t_s(X)\downarrow, t(R) = t_r, t(S) = t_s\}; \\
\mathrm{MAYBE}(q) &= \Big\{t\colon \exists_{t_r \in r, t_s \in s} \\
&\qquad t_r \text{ i } t_s \text{ są zgodne na } X, \\
&\qquad t(R_X) = t_r(R_X), t(S_X) = t_s(S_X), \\
&\qquad \forall_{A \in X} t(A) = \begin{cases} t_r(A) & \text{jeśli } t_r(A)\downarrow, \\ t_s(A) & \text{wpp.} \end{cases} \\
&\quad \Big\} \setminus \mathrm{SURE}(q); \\
X &= R \cap S; \\
R_X &= R \setminus X.
\end{aligned}
$$

- Operator $\bowtie^{\mathsf{B}}$ jest wystarczający i ograniczony.

Definicje

- **Null value (null)**: A special value, or mark, that is used to indicate the absence of any data value.
- **Distinct**: Two values are said to be not distinct if either: both are the null value, or they compare equal according to the *comparison predicate*. Otherwise they are distinct. Two rows (or partial rows) are distinct if at least one of their pairs of respective values is distinct. Otherwise they are not distinct. The result of evaluating whether or not two values or two rows are distinct is never unknown.

- x `IS NULL`;
  x `IS NOT NULL`
- `COALESCE(`x_1`,` x_2`,` …`)`

Przykład

```
SELECT
COALESCE(
  NULL < NULL, NULL = NULL, NULL > NULL,
  NULL * 0, NULL || 'foo'
) FROM dual;
```

null

- ```
  SELECT x
  FROM
    (SELECT NULL FROM dual AS x UNION
     SELECT 0 FROM dual AS x)
  ORDER BY x;
  ```
  - MS SQL, MySQL:
    null
    0
  - IBM DB2, Oracle, PostreSQL, SQLite:
    0
    null
  - SQL: jedno lub drugie rozwiązanie.
- … `ORDER BY` c `IS NOT NULL,` c`, ...`
  ```

- SQL zezwala (jako *optional feature*) na nulle w kolumnach powiązanych przez `UNIQUE`.
- $\neg$ `NULL = NULL`!
- Zaimplementowe w: MS SQL, MySQL, Oracle, PostreSQL, SQLite.
- Nie w: IBM DB2.
- Ale:
  - MS SQL: `NULL = NULL`!?
  - Oracle: $\langle \mathtt{NULL}, c_2 \rangle = \langle c_1, c_2 \rangle$!?

Oracle utożsamia `NULL` i pusty napis.

Przykład

- `SELECT 'a' || NULL FROM dual;`
  - Oracle: 'a'
  - reszta świata: *null*
- Oracle:
  `SELECT LENGTH('') FROM dual;`
  null
- reszta świata:
  `SELECT CHARACTER_LENGTH('') FROM dual;`
  `0`

- David Maier, *The Theory of Relational Databases*: Null values, partial information and database semantics
  ⟨http://web.cecs.pdx.edu/~maier/TheoryBook/MAIER/C12.pdf⟩
- Serge Abiteboul, Richard Hull, Victor Vianu, *Foundation of Databases*: Incomplete Information
- Hans-Joachim Klein, *How to Modify SQL Queries in Order to Guarantee Sure Answers*
  ⟨http://acm.org/sigmod/record/issues/9409/sql.ps⟩
- Troels Arvin, *Comparison of different SQL implementations*
  ⟨http://troels.arvin.dk/db/rdbms/⟩
- *Database Server Feature Comparisons*
  ⟨http://dev.mysql.com/tech-resources/crash-me.php⟩
- *NULL Handling in SQLite Versus Other Database Engines*
  ⟨http://www.sqlite.org/nulls.html⟩