

Definicja:

Alfabet języka logiki zdań składa się z nieskończonego (najczęściej zakładamy: przeliczalnego) zbioru P , o którym myślimy jak o zbiorze **zmiennych zdaniowych** i skończonego zbioru symboli, o których myślimy jak o **spójnikach zdaniowych**:

- ▶ $\neg$ – jednoargumentowy, negacja;
- ▶ $\wedge$ – dwuargumentowy, koniunkcja;
- ▶ $\vee$ – dwuargumentowy, alternatywa;
- ▶ $\rightarrow$ – dwuargumentowy, implikacja.

Dodatkowo “dorzucamy” też do naszego alfabetu nawiasy (i).

Język logiki zdań L definiujemy teraz jako najmniejszy podzbiór monoidu L^* spełniający warunki:

1. $P \subset L$;
2. jeśli $A, B \in L$, to do L należą również następujące formuły:

$$\neg A, \quad A \wedge B, \quad A \vee B, \quad A \rightarrow B.$$

Niech $A, B, C, \dots$ oznaczają formuły. Przyjmujemy następujące aksjomaty:

1. $A \rightarrow (B \rightarrow A)$ (prawo poprzedzania),
2. $(A \rightarrow (B \rightarrow C)) \rightarrow ((A \rightarrow B) \rightarrow (A \rightarrow C))$ (sylogizm Fregego),
3. $A \wedge B \rightarrow A$,
4. $A \wedge B \rightarrow B$,
5. $(A \rightarrow B) \rightarrow ((A \rightarrow C) \rightarrow (A \rightarrow (B \wedge C)))$,
6. $A \rightarrow A \vee B$,
7. $B \rightarrow A \vee B$,
8. $(B \rightarrow A) \rightarrow ((C \rightarrow A) \rightarrow (B \vee C \rightarrow A))$,
9. $\neg A \rightarrow (A \rightarrow B)$ (prawo Dunsza Szkota),
10. $(A \rightarrow \neg A) \rightarrow \neg A$ (prawo redukcji do absurdu),
11. $\neg\neg A \rightarrow A$ (mocne prawo podwójnego przeczenia);

oraz następującą regułę, zwaną **regułą odrywania**:

$$\frac{A \rightarrow B, A}{B} r_0.$$

Matrycą (Łukasiewicz, Tarski, 1930) dla języka logiki zdaniowej nazywamy parę

$$\mathcal{M} = (\mathbf{M}, M^*),$$

gdzie $\mathbf{M}$ jest algebrą podobną do algebry języka logiki zdań, zaś zbiór $\emptyset \neq M^* \subset M$ nazywamy zbiorem elementów wyróżnionych matrycy (zakładamy często, że M^* jest jednoelementowy).

Piszemy też

$$\mathcal{M} = (M, M^*, F_{\neg}, F_{\wedge}, F_{\vee}, F_{\rightarrow}).$$

Niech $\mathcal{M}$ będzie pewną matrycą dla języka logiki zdań.

Wartościowaniem w $\mathcal{M}$ nazywamy dowolne odwzorowanie $\varphi : P \rightarrow M$. Każde takie odwzorowanie przedłuża się jednoznacznie do homomorfizmu algebry **L** w algebrę **M**. Homomorfizm ten będziemy również oznaczać przez φ .

Niech A będzie dowolną formułą języka zdań i niech $\mathcal{M}$ będzie matrycą. Ustalmy wartościowanie $\varphi : P \rightarrow M$. Mówimy, że φ **spełnia** A w $\mathcal{M}$, gdy $\varphi(A) \in M^*$. Formułę A nazwiemy **tautologią matrycy** $\mathcal{M}$, gdy A jest spełniona w $\mathcal{M}$ przy każdym wartościowaniu (mówimy wtedy też, że A jest **prawdziwe** w $\mathcal{M}$). Zbiór wszystkich tautologii matrycy $\mathcal{M}$ nazywamy **zawartością** tej matrycy i oznaczamy przez $E(\mathcal{M})$. Mówimy, że A **wynika logicznie** z B w matrycy $\mathcal{M}$, jeżeli dla każdego wartościowania $\varphi : P \rightarrow M$

$$\text{jeżeli } \varphi(A) \in M^*, \text{ to } \varphi(B) \in M^*.$$

Mówimy, że A i B są **logicznie równoważne** w matrycy $\mathcal{M}$, gdy dla każdego wartościowania $\varphi : P \rightarrow M$

$$\varphi(A) = \varphi(B).$$

Twierdzenie (o zgodności)

Każda teza KRZ jest tautologią matrycy $\mathcal{M}_2$.

Twierdzenie (o pełności)

Każda tautologia $\mathcal{M}_2$ jest tezą KRZ.

Wniosek

KRZ jest rozstrzygalny w intuicyjnym sensie, tzn. istnieje efektywna procedura dająca odpowiedź, czy $\vdash A$ czy $\nvdash A$.

Język pierwszego rzędu składa się z:

- ▶ symboli relacyjnych P_i , $i \in I$, gdzie $\#(P_i)$ oznaczać będzie ilość argumentów symbolu P_i ,
- ▶ symboli funkcyjnych f_j , $j \in J$, gdzie $\#(f_j)$ oznaczać będzie ilość argumentów f_j ,
- ▶ stałych c_k , $k \in K$.

Formalnie język będziemy definiowali jako zbiór

$L = \{(P_i)_{i \in I}, (f_j)_{j \in J}, (c_k)_{k \in K}\}$. P_i, f_j, c_k nazywamy symbolami pozalogicznymi.

Symbole logiczne, niezależne od L , są następujące:

- ▶ $\neg, \rightarrow$;
- ▶ zbiór zmiennych indywiduowych $V = \{v_0, v_1, v_2, \dots\}$;
- ▶ kwantyfikator $\forall$;
- ▶ binarny symbol relacyjny $\equiv$ odpowiadający identyczności;
- ▶ nawiasy.

Niech $M \neq \emptyset$. **Interpretacją** języka L w zbiorze M nazywamy odwzorowanie I takie, że

- ▶ $I(P_i) \subset M^{\sharp(P_i)}$,
- ▶ $I(f_j) : M^{\sharp(f_j)} \rightarrow M$,
- ▶ $I(c_k) \in M$.

Modelem (albo **strukturą**) dla L nazywamy parę $\mathfrak{M} = (M, I)$. Dla wygody zamiast $I(P_i)$, $I(f_j)$, $I(c_k)$ piszemy $P_i^{\mathfrak{M}}$, $f_j^{\mathfrak{M}}$ i $c_k^{\mathfrak{M}}$ oraz, gdy L zawiera skończenie wiele symboli pozalogicznych:

$$\mathfrak{M} = (M, P_1^{\mathfrak{M}}, \dots, P_i^{\mathfrak{M}}, f_1^{\mathfrak{M}}, \dots, f_j^{\mathfrak{M}}, c_1^{\mathfrak{M}}, \dots, c_k^{\mathfrak{M}}).$$

Moc modelu określamy jako moc zbioru M .

Rozważmy model $\mathfrak{M}$ języka L i oznaczmy przez V zbiór wszystkich zmiennych indywiduowych. Odwzorowanie $\alpha : V \rightarrow M$ nazywamy **wartościowaniem** w M .

Wartość termu $t^{\mathfrak{M}}[\alpha] \in M$ definiujemy następująco:

- ▶ $(v_i)^{\mathfrak{M}}[\alpha] = \alpha(v_i),$
- ▶ $(c_k)^{\mathfrak{M}}[\alpha] = c_k^{\mathfrak{M}},$
- ▶ $(f_j t_1 \dots t_n)^{\mathfrak{M}}[\alpha] = f_j^{\mathfrak{M}}(t_1^{\mathfrak{M}}[\alpha], \dots, t_n^{\mathfrak{M}}[\alpha]).$

Niech A będzie formułą języka L , $\mathfrak{M}$ modelem dla języka L , a α wartościowaniem w M . Definiujemy **spełnialność** formuły A w modelu $\mathfrak{M}$ z wartościowaniem α , $\mathfrak{M} \models A[\alpha]$:

- ▶ $\mathfrak{M} \models (t \equiv s)[\alpha]$ wtedy i tylko wtedy, gdy $t^{\mathfrak{M}}[\alpha] = s^{\mathfrak{M}}[\alpha]$,
- ▶ $\mathfrak{M} \models Pt_1 \dots t_n[\alpha]$ wtedy i tylko wtedy, gdy $P^{\mathfrak{M}}(t_1^{\mathfrak{M}}[\alpha], \dots, t_n^{\mathfrak{M}}[\alpha])$,
- ▶ $\mathfrak{M} \models \neg A[\alpha]$ wtedy i tylko wtedy, gdy nieprawda, że $\mathfrak{M} \models A[\alpha]$,
- ▶ $\mathfrak{M} \models (A \vee B)[\alpha]$ wtedy i tylko wtedy, gdy $\mathfrak{M} \models A[\alpha]$ lub $\mathfrak{M} \models B[\alpha]$,
- ▶ $\mathfrak{M} \models (A \wedge B)[\alpha]$ wtedy i tylko wtedy, gdy $\mathfrak{M} \models A[\alpha]$ i $\mathfrak{M} \models B[\alpha]$,
- ▶ $\mathfrak{M} \models (A \rightarrow B)[\alpha]$ wtedy i tylko wtedy, gdy nieprawda, że $\mathfrak{M} \models A[\alpha]$ lub $\mathfrak{M} \models B[\alpha]$,
- ▶ $\mathfrak{M} \models \forall x A[\alpha]$ wtedy i tylko wtedy, gdy $\mathfrak{M} \models A[\alpha \frac{x}{a}]$ dla każdego $a \in M$,
- ▶ $\mathfrak{M} \models \exists x A[\alpha]$ wtedy i tylko wtedy, gdy $\mathfrak{M} \models A[\alpha \frac{x}{a}]$ dla pewnego $a \in M$,

przy czym

$$x \quad \left(\alpha(y) \text{ dla } y \neq x. \right.$$

Formuła A jest **prawdziwa** w $\mathfrak{M}$, $\mathfrak{M} \models A$, gdy $\mathfrak{M} \models A[\alpha]$ dla każdego wartościowania $\alpha : V \rightarrow M$.

Formuła A jest **tautologią** (lub jest **ogólnie prawdziwa**), $\models A$,
gdy $\mathfrak{M} \models A$ dla każdego modelu $\mathfrak{M}$.

Ponieważ spełnialność formuły zależy wyłącznie od wartościowania jej zmiennych, dla formuły $A(x_1, \dots, x_n)$ i wartościowania α takiego, że $\alpha(x_i) = a_i$, dla $1 \leq i \leq n$, będziemy pisać

$$\mathfrak{M} \models A(x_1, \dots, x_n)[a_1, \dots, a_n].$$

Ustalmy język L . **Aksjomatami rachunku predykatów** (lub **klasycznego rachunku kwantyfikatorów** lub **KRK**) są domknięcia względem wszystkich układów zmiennych formuł danych za pomocą następujących schematów:

- (Q0) domknięcia schematów aksjomatów KRZ w języku $\{\neg, \rightarrow\}$, np. $A \rightarrow (B \rightarrow A)$ itd.
- (Q1) $\forall x A \rightarrow A(t/x)$ dla $x \in Ff(t, A)$,
- (Q2) $\forall x (A \rightarrow B) \rightarrow (\forall x A \rightarrow \forall x B)$,
- (Q3) $A \rightarrow \forall x A$, gdzie $x \notin vf(A)$,
- (Q4) aksjomaty identyczności: dla dowolnych termów $t_1, t_2, t_3, t'_1, t'_2, t'_3$ i dowolnych symboli funkcyjnych f i relacyjnych P :
 - ▶ $t_1 \equiv t_1$,
 - ▶ $t_1 \equiv t_2 \rightarrow t_2 \equiv t_1$,
 - ▶ $t_1 \equiv t_2 \wedge t_2 \equiv t_3 \rightarrow t_1 \equiv t_3$,
 - ▶ $(t_1 \equiv t'_1 \wedge \dots \wedge t_n \equiv t'_n) \rightarrow (ft_1 \dots t_n \equiv ft'_1 \dots t'_n)$,
 - ▶ $(t_1 \equiv t'_1 \wedge \dots \wedge t_n \equiv t'_n) \rightarrow (Pt_1 \dots t_n \rightarrow Pt'_1 \dots t'_n)$,

Ponadto definiujemy **regułę odrywania**:

$$A \quad A \rightarrow B$$

Mówimy, że zbiór formuł Γ jest **spełnialny**, gdy istnieje model $\mathfrak{M}$ i wartościowanie $\alpha : V \rightarrow M$ takie, że $\mathfrak{M} \models A[\alpha]$ dla każdej formuły $A \in \Gamma$.

Mówimy, że zbiór zdań Γ jest spełnialny, gdy dla pewnego modelu $\mathfrak{M}$ zachodzi $\mathfrak{M} \models A$ dla $A \in \Gamma$.

Twierdzenie (o istnieniu modelu)

Każdy niesprzeczny zbiór formuł jest spełnialny. W szczególności, każdy niesprzeczny zbiór zdań ma model.

Dowód tego twierdzenia pomijamy.

Wniosek (twierdzenie o pełności)

Dla dowolnego zbioru zdań Σ i dowolnego zdania A :

$$\Sigma \vdash A \Leftrightarrow \Sigma \models A.$$

Nierozstrzygalność rachunku predykatów

Nierozstrzygalny jest następujący problem decyzyjny:

Czy dana formuła logiki pierwszego rzędu jest tautologią?

Aby wykazać, że tak jest, posłużymy się nierozstrzygalnością problemu stopu dla **maszyn Turinga**.

- ▶ Przypomnijmy, że maszyna Turinga jest abstrakcyjnym modelem komputera, składającym się z nieskończenie długiej taśmy podzielonej na pola.
- ▶ Każde pole może znajdować się w jednym z N stanów.
- ▶ Maszyna zawsze jest ustawiona nad jednym z pól i znajduje się w jednym z M stanów.
- ▶ Zależnie od kombinacji stanu maszyny i pola maszyna zapisuje nową wartość w polu, zmienia stan, a następnie może przesunąć się o jedno pole w prawo lub w lewo.
- ▶ Każdą z takich operacji nazywamy **rozkazem**.
- ▶ Maszyna Turinga jest sterowana listą zawierającą dowolną liczbę takich rozkazów.
- ▶ N i M są dowolnymi liczbami skończonymi.
- ▶ Lista rozkazów dla maszyny Turinga może być traktowana jako jej program.

Formalnie, maszynę Turinga nad alfabetem $\mathcal{A}$ definiujemy jako uporządkowaną piątkę $\mathcal{M} = (\Delta, Q, \delta, q_0, q_f)$, gdzie:

- ▶ Δ jest skończonym alfabetem, zawierającym $\mathcal{A}$ oraz symbol $B \notin \mathcal{A}$,
- ▶ Q jest skończonym zbiorem stanów,
- ▶ $q_0 \in Q$ jest stanem początkowym,
- ▶ $q_f \in Q$ jest stanem końcowym,
- ▶ $\delta : (Q \setminus \{q_f\}) \times \Delta \rightarrow \Delta \times Q \times \{-1, 0, +1\}$ jest funkcją przejścia.

Zakładając, że zbiory Δ i Q są rozłączne, można zdefiniować **konfigurację** maszyny jako słowo postaci wqv , gdzie $q \in Q$ oraz $w, v \in \Delta$, przy czym utożsamiamy konfiguracje wqv i $wqvB$.

Interpretacja tej definicji jest następująca:

Taśma maszyny jest nieskończona w prawo.

Na początku taśmy zapisane jest słowo wv , dalej w prawo są same blanki, a głowica maszyny "patrzy" na pierwszy znak na prawo od w .

Konfigurację postaci $C_w = q_0w$, gdzie $w \in \mathcal{A}^*$, nazywamy **początkową**, a konfigurację postaci $wq_f v$ nazywamy **końcową** lub **akceptującą**.

Relację $\rightarrow_{\mathcal{M}}$ na konfiguracjach definiujemy następująco:

- ▶ Jeśli $\delta(q, a) = (b, p, +1)$ to $wqav \rightarrow_{\mathcal{M}} wbpv$;
- ▶ Jeśli $\delta(q, a) = (b, p, 0)$ to $wqav \rightarrow_{\mathcal{M}} wpbv$;
- ▶ Jeśli $\delta(q, a) = (b, p, -1)$ to $wcqav \rightarrow_{\mathcal{M}} wpcbv$ oraz $qav \rightarrow_{\mathcal{M}} pbv$ (gdy ruch w lewo jest niemożliwy.)

Symbolem $\twoheadrightarrow_{\mathcal{M}}$ oznaczamy przechodnio-zwrotne domknięcie relacji $\rightarrow_{\mathcal{M}}$.

Jeżeli $C_w \twoheadrightarrow_{\mathcal{M}} C'$, gdzie C' jest konfiguracją akceptującą to mówimy, że maszyna akceptuje słowo w .

Problem stopu można równoważnie sformułować następująco: czy dana maszyna Turinga akceptuje słowo puste?

Lemat

Niech ζ oznacza koniunkcję następujących formuł

- ▶ $\forall y \neg P(y, c),$
- ▶ $\forall x \exists y P(x, y),$
- ▶ $\forall x \forall y (P(x, y) \rightarrow R(x, y)),$
- ▶ $\forall x \forall y \forall z (R(x, y) \rightarrow (R(y, z) \rightarrow R(x, z))),$
- ▶ $\forall x \neg R(x, x).$

Formuła ζ jest zdaniem. Zdanie to jest spełnialne, a każdy jego model $\mathfrak{A}$ zawiera nieskończony ciąg różnych elementów $c^{\mathfrak{A}} = a_0, a_1, a_2, \dots$ takich że $(a_i, a_{i+1}) \in P^{\mathfrak{A}}$ dla każdego i .

Twierdzenie

Nie istnieje algorytm rozstrzygający, czy dana formuła logiki predykatów jest tautologią.

Dowód:

Naszym celem jest efektywna konstrukcja, która dla dowolnej maszyny Turinga $\mathcal{M}$ utworzy formułę $\phi_{\mathcal{M}}$ o następującej własności:

$\mathcal{M}$ akceptuje słowo puste wtedy i tylko wtedy, gdy $\phi_{\mathcal{M}}$ jest tautologią.

Stąd natychmiast wynika teza twierdzenia.

W przeciwnym razie taka konstrukcja pozwalałaby bowiem na rozstrzygnięcie problemu stopu.

W istocie, wygodniej będzie skonstruować taką formułę $\phi_{\mathcal{M}}$, że $\mathcal{M}$ zapętla się na słowie pustym wtedy i tylko wtedy, gdy $\phi_{\mathcal{M}}$ jest spełnialna,

i na koniec przyjąć $\phi_{\mathcal{M}} = \neg\phi_{\mathcal{M}}$.

Język, w którym zapiszemy naszą formułę, będzie zależała od maszyny $\mathcal{M}$.

Składa się on z:

- ▶ jednoargumentowych symboli relacyjnych S_q , dla wszystkich stanów $q \in Q$;
- ▶ dwuargumentowych symboli relacyjnych C_a , dla wszystkich symboli $a \in \Delta$;
- ▶ dwuargumentowego symbolu relacyjnego G ;
- ▶ stałej c i symboli P i Q występujących w formule ζ z Lematu.

Formuła $\phi_{\mathcal{M}}$ jest koniunkcją złożoną z wielu składowych, które teraz opiszemy.

Pierwszą składową jest formuła ζ z Lematu.

Każdy model tej formuły zawiera różnowartościowy ciąg $a_0, a_1, a_2, \dots$, który posłuży nam jako substytut ciągu liczb naturalnych (o elemencie a_i myślimy jak o liczbie i).

Intuicyjny sens formuł atomowych naszej sygnatury jest taki:

- ▶ Formułę $S_q(x)$ czytamy: po x krokach obliczenia maszyna jest w stanie q .
- ▶ Formułę $G(x, y)$ czytamy: po x krokach głowica maszyny znajduje się w pozycji y .
- ▶ Formułę $C_a(x, y)$ czytamy: po x krokach na pozycji y znajduje się symbol a .

Dalsze składowe naszej formuły $\phi_{\mathcal{M}}$ są tak dobrane, aby każdy jej model reprezentował nieskończone obliczenie maszyny zaczynające się od słowa pustego:

1. $S_{q_0}(c) \wedge G(c, c) \wedge \forall x C_B(c, x)$;
2. $\forall x (\bigvee_{q \in Q} S_q(x))$;
3. $\forall x (S_q(x) \rightarrow \neg S_p(x))$, dla $q, p \in Q$, $q \neq p$;
4. $\forall x \forall y (\bigvee_{a \in \Delta} C_a(x, y))$;
5. $\forall x \forall y (C_a(x, y) \rightarrow \neg C_b(x, y))$, dla $a, b \in \Delta$, $a \neq b$;
6. $\forall x \exists y G(x, y)$;
7. $\forall x \forall y \forall z (G(x, y) \wedge G(x, z) \rightarrow y = z)$;
8. $\forall x \forall y \forall z (S_q(x) \wedge G(x, y) \wedge C_a(x, y) \wedge P(x, z) \rightarrow S_p(z) \wedge C_b(z, y))$, gdy $\delta(q, a) = (p, b, i)$;
9. $\forall x \forall y \forall z (\neg G(x, y) \wedge C_a(x, y) \wedge P(x, z) \rightarrow C_a(z, y))$;
10. $\forall x \forall y \forall z \forall v (S_q(x) \wedge G(x, y) \wedge P(x, z) \wedge P(y, v) \rightarrow G(z, v))$, gdy $\delta(q, a) = (p, b, +1)$;

11. $\forall x \forall y \forall z (S_q(x) \wedge G(x, y) \wedge P(x, z) \rightarrow G(z, y))$, gdy $\delta(q, a) = (p, b, 0)$;
12. $\forall x \forall y \forall z \forall v (S_q(x) \wedge G(x, y) \wedge P(x, z) \wedge P(v, y) \rightarrow G(z, v))$, gdy $\delta(q, a) = (p, b, -1)$;
13. $\forall x \forall y \forall z \forall v (S_q(x) \wedge G(x, c) \wedge P(x, z) \rightarrow G(z, c))$, gdy $\delta(q, a) = (p, b, -1)$;
14. $\forall x \neg S_{q_f}(x)$.

Przypuśćmy teraz, że maszyna $\mathcal{M}$ ma nieskończone obliczenie dla pustego słowa wejściowego.

Zbudujemy model $\mathfrak{A}$ dla formuły $\phi_{\mathcal{M}}$.

Dziedziną tego modelu jest zbiór $\mathbb{N}$ liczb naturalnych.

Stałą c interpretujemy jako zero, relacja $P^{\mathfrak{A}}$ to relacja następnika, a $R^{\mathfrak{A}}$ to (ostra) relacja mniejszości.

Relacje $S_q^{\mathfrak{A}}$, $C_a^{\mathfrak{A}}$ i $G^{\mathfrak{A}}$ określamy zgodnie z ich intuicyjną interpretacją na podstawie przebiegu nieskończonego obliczenia maszyny.

Nietrudno się przekonać, że wszystkie człony koniunkcji są prawdziwe w $\mathfrak{A}$, czyli zdanie $\phi_{\mathcal{M}}$ jest spełnialne.

Przystąpmy więc do trudniejszej części dowodu.

Założmy mianowicie, że $\mathfrak{A} \models \phi_{\mathcal{M}}$ dla pewnego modelu $\mathfrak{A}$.

Wtedy także $\mathfrak{A} \models \zeta$, niech więc a_i będą elementami $\mathfrak{A}$, o których mowa w Lemacie.

Model $\mathfrak{A}$ spełnia też wszystkie pozostałe składowe formuły $\phi_{\mathcal{M}}$.

Składowe (2) i (3) gwarantują, że każdy z elementów a_i należy do dokładnie jednej z relacji S_q .

Podobnie, każda para (a_i, a_j) należy do dokładnie jednej z relacji C_a , oraz każde a_i jest w relacji G z dokładnie jednym elementem modelu $\mathfrak{A}$ – tu używamy składowych (4)–(7).

Rozpatrzmy obliczenie maszyny $\mathcal{M}$ dla słowa pustego.

Pokażemy, że jest to obliczenie nieskończone.

Jeśli obliczenie maszyny $\mathcal{M}$ składa się z co najmniej n kroków, to przez q_n oznaczmy stan, w którym znajduje się maszyna $\mathcal{M}$ po wykonaniu tych n kroków, a przez k_n pozycję głowicy w tym momencie.

Zawartością m -tej komórki taśmy po n -tym kroku obliczenia niech zaś będzie symbol b_{nm} .

Przez indukcję ze względu na n dowodzimy, że dla dowolnego $n \in \mathbb{N}$ obliczenie składa się z co najmniej n kroków, oraz:

$$a_n \in S_{q_n}^{\mathfrak{A}}, (a_n, a_m) \in C_{b_{nm}}^{\mathfrak{A}}, (a_n, a_{k_n}) \in G^{\mathfrak{A}}.$$

Inaczej mówiąc, model $\mathfrak{A}$ prawidłowo reprezentuje kolejne konfiguracje maszyny.

Dla $n = 0$ powyższe związki wynikają wprost z prawdziwości członu (1) naszej koniunkcji.

W kroku indukcyjnym skorzystamy przede wszystkim z członu (14), który gwarantuje, że stan q_n nie jest końcowy.

Określona jest więc wartość funkcji przejścia $\delta(q_n, b_{mk_n})$.

Możemy więc odwołać się do składowych (9–12), które zapewniają poprawną zmianę stanu i symbolu pod głowicą (8), zachowanie bez zmian reszty taśmy (9) i przesunięcie głowicy (10–12).