Prace Naukowe Wyższej Szkoły Pedagogicznej w Częstochowie
Matematyka V, Częstochowa 1997

Operacja konsekwencji w trójsekwencyjnym rachunku predykatów Łukasiewicza

Piotr Borowik

Streszczenie

W niniejszej pracy określono pojęcie konsekwencji dla trójsekwencyjnego rachunku predykatów Łukasiewicza, oraz wykazano, że wprowadzona operacja ma własności ogólnej operacji konsekwencji w sensie Tarskiego.

1. Preliminaria

Niech S będzie standardowym językiem rachunku predykatów pierwszego rzędu, generowanym z formuł atomicznych przy pomocy spójników logicznych: negacji $\sim$, implikacji $\Rightarrow$, koniunkcji $\wedge$ i alternatywy $\vee$, oraz kwantyfikatorów wiążących zmienne indywiduowe: dużego $\forall$ i małego $\exists$.

W rachunku zdań wartości formuł są zdeterminowane przez wartości ich podformuł. Formuły kwantyfikatorowe mogą mieć jednak nieskończenie wiele podformuł. Związki między prawdziwością formuł kwantyfikatorowych a wartościami ich podformuł ustala się za pomocą pewnej dodatkowej funkcji zwanej niekiedy *funkcją dystrybucji.*

Niech $E_3 = \{0, 1, 2\}$ i niech

$$\mathbf{E}_3 = (E_3, \{\sim, \Rightarrow, \wedge, \vee\}, \{2\})$$

będzie trójwartościową matrycą Łukasiewicza, której operacje określone są następująco:

(a) $\sim x = 2 - x$,

(b) $x \Rightarrow y = min\{2, 2 - x + y\}$,

(c) $x \wedge y = min\{x, y\}$,

(d) $x \vee y = max\{x, y\}$.

Niech $D_3 = P(E_3) - \emptyset = \{\{0\}, \{1\}, \{2\}, \{0,1\}, \{0,2\}, \{1,2\}, \{0,1,2\}\}$.

Niech funkcje $q_\exists, q_\forall : D_3 \longrightarrow E_3$ będą funkcjami interpretującymi odpowiednie kwantyfikatory $\exists$ i $\forall$. Określa się je w następujący sposób:

$$q_\exists(U) = max\ U,$$
$$q_\forall(V) = min\ V$$

dla dowolnych $U, V \in D_3$.

Czwórkę

$$\mathcal{E}_3 = (E_3, \{2\}, \{\sim, \Rightarrow, \wedge, \vee\}, \{q_\exists, q_\forall\})$$

nazywać będziemy *prestrukturą* dla języka trójwartościowego rachunku predykatów Łukasiewicza.

Niech teraz U będzie dowolnym niepustym zbiorem, zwanym dalej uniwersum i niech $\mathfrak{R}$ oznacza następujący układ:

$$\mathfrak{R} = (U, \{p_i : i \in I\}, \{g_j : j \in J\}, \{a_k : k \in N\}, \mathcal{E}_3),$$

gdzie p_i oraz g_j są funkcjami takimi, że

$$p_i : U^{\arg(p_i)} \longrightarrow E_3 \text{ dla } i \in I, I \subseteq N$$

$$g_j : U^{\arg(g_j)} \longrightarrow U \text{ dla } j \in J \text{ oraz}$$

$$a_k \in U \text{ dla } k \in N,$$

tzn. a_k są pewnymi wyróżnionymi elementami ze zbioru U. (W powyższych wzorach $\arg(p_i)$ oznacza liczbę argumentów funkcji p_i, podobnie $\arg(g_j)$ — liczbę argumentów funkcji g_j).
Układ $\mathfrak{R}$ będziemy nazywać *strukturą języka rachunku predykatów Łukasiewicza* lub krótko *strukturą*.

Interpretacją języka S w strukturze $\mathfrak{R}$ nazywamy funkcję $\iota : S \longrightarrow \mathfrak{R}$ spełniającą następujące warunki:

(a) $\iota(x_j) \in U$ dla $j \in N$,

(b) $\iota(c_k) = a_k$ dla $k \in N$,

(c) $\iota(P_s) = p_s$ dla $s \in I, I \subseteq N$,

(d) $\iota(f_j) = g_j$ dla $j \in J$,

(e) $\iota(\sim) = \sim$,

(f) $\iota(\Rightarrow) = \Rightarrow$

(g) $\iota(\wedge) = \wedge$

(h) $\iota(\vee) = \vee$

(i) $\iota(\exists) = q_\exists$,

(j) $\iota(\forall) = q_\forall$.

Powiemy, że formuła α jest *spełnialna* w interpretacji ι wtedy i tylko wtedy, gdy istnieje ciąg $b \in U^N$ taki, że $v(\alpha, b) = 2$. Podobnie powiemy, że formuła α jest *spełnialna,* jeśli istnieje struktura $\mathfrak{R}$ i interpretacja ι, w której formuła α jest spełnialna.

Formuła $\alpha \in S$ jest *prawdziwa w interpretacji* ι wtedy i tylko wtedy, gdy dla każdego ciągu $b \in U^N$ $v(\alpha, b) = 2$, gdzie $v(\alpha, b)$ oznacza wartość formuły α na ciągu b.

Formuła $\alpha \in S$ jest *tautologią rachunku predykatów Łukasiewicza*, krótko *tautologią* wtedy i tylko wtedy, gdy jest ona prawdziwa w każdej interpretacji języka S w dowolnej strukturze $\mathfrak{R}$ tego języka S.

2. Trójsekwencyjny rachunek predykatów Łukasiewicza

Niech $\Gamma_0, \ldots \Gamma_{n-1}$ będą dowolnymi skończonymi zbiorami formuł. W szczególności niektóre ze zbiorów Γ_i mogą być puste. Uporządkowaną n-tkę zbiorów formuł $(\Gamma_0, \Gamma_1, \ldots, \Gamma_{n-1})$ *nazywamy* $n-$ *sekwentem*. Ten n-sekwent, złożony ze zbiorów formuł $\Gamma_0, \ldots \Gamma_{n-1}$, będziemy notować $\Gamma_0 \vdash \ldots \vdash \Gamma_{n-1}$. Jeżeli α jest formułą i $\Gamma \subseteq S$ to zapis Γ, α oznaczać będzie sumę zbiorów $\Gamma \cup \{\alpha\}$. Sekwenty będziemy oznaczać literą Σ, również z indeksami, jeśli zajdzie taka konieczność. W dalszym ciągu tej pracy, jeśli to nie będzie prowadzić do nieporozumień, n-sekwenty będziemy nazywać po prostu sekwentami.

Niech

$$\Sigma = \Gamma_0 \vdash \cdots \vdash \Gamma_{n-1}$$

i

$$\Pi = \Delta_0 \vdash \cdots \vdash \Delta_{n-1}$$

będą dwoma dowolnymi sekwentami. Sekwent Σ *zawiera się* w sekwencie Π wtedy i tylko wtedy, gdy dla dowolnego i, $0 \leq i \leq n-1$, $\Gamma_i \subseteq \Delta_i$. Fakt ten będziemy notować $\Sigma \subseteq \Pi$. Zapis $\Sigma * \Pi$ lub $\Sigma\Pi$ oznacza sekwent

$$\Lambda_0 \vdash \cdots \vdash \Lambda_{n-1}$$

gdzie

$$\Lambda_i = \Gamma_i \cup \Delta_i \quad \text{dla} \quad 0 \leq i \leq n-1,$$

nazywamy *złożeniem* sekwentów Σ oraz Π.

Niech $\Gamma \subseteq S$ będzie dowolnym zbiorem formuł. Przez zapis $\vdash_j \Gamma$ będziemy rozumieć sekwent

$$\Sigma = \Gamma_0 \vdash \cdots \vdash \Gamma_{n-1} \quad \text{taki, że}$$

$$\Gamma_i = \begin{cases} \Gamma & \text{jeśli} \quad i = j \\ \emptyset & \text{jeśli} \quad i \neq j. \end{cases}$$

W szczególności zapis $\alpha \vdash_j$ oznaczał będzie sekwent postaci $\vdash_j \{\alpha\}$. Z powyższych określeń łatwo wynika, że dowolny sekwent $\Sigma = \Gamma_0 \vdash \cdots \vdash \Gamma_{n-1}$ można przedstawić jako złożenie n sekwentów $\Gamma_j \vdash_j$ dla $0 \leq j \leq n-1$. Sekwent będziemy nazywać *atomicznym* jeśli wszystkie formuły w nim występujące są atomiczne.

Sekwent $\Sigma = \Gamma_0 \vdash \cdots \vdash \Gamma_{n-1}$ nazywać będziemy *sekwentem przepełnionym* wtedy i tylko wtedy gdy, istnieją r, s takie, że $0 \leq r \leq n-1$, $0 \leq s \leq n-1$ i $\Gamma_r \cap \Gamma_s \neq \emptyset$,
Sekwent Σ nazywać będziemy *sekwentem końcowym* wtedy i tylko wtedy, gdy jest sekwentem atomicznym lub przepełnionym. Sekwent atomiczny i nie przepełniony będziemy nazywać sekwentem *rozłożonym.* W dalszym ciągu nasze rozważania ograniczamy do $n = 3$.

Schematy reguł wprowadzania spójników i kwantyfikatorów do zbiorów formuł w sekwencie mają w systemie trójsekwencyjnym Łukasiewicza następującą postać:

(a) *reguły wprowadzające spójniki negacji*

$$(ln)_0 \quad \frac{\Gamma \vdash \Delta \vdash \Pi, \alpha}{\Gamma, \sim \alpha \vdash \Delta \vdash \Pi},$$

$$(ln)_1 \quad \frac{\Gamma \vdash \Delta, \alpha \vdash \Pi}{\Gamma \vdash \Delta, \sim \alpha \vdash \Pi},$$

$$(ln)_2 \quad \frac{\Gamma, \alpha \vdash \Delta \vdash \Pi}{\Gamma \vdash \Delta \vdash \Pi, \sim \alpha},$$

(b) *reguły wprowadzające spójnik implikacji*

$$(li)_0 \quad \frac{\Gamma, \beta \vdash \Delta \vdash \Pi, \alpha}{\Gamma, \alpha \Rightarrow \beta \vdash \Delta \vdash \Pi},$$

$$(li)_1 \quad \frac{\Gamma, \beta \vdash \Delta, \alpha \vdash \Pi; \ \Gamma \vdash \Delta, \beta \vdash \Pi, \alpha}{\Gamma \vdash \Delta, \alpha \Rightarrow \beta \vdash \Pi},$$

$$(li)_2 \quad \frac{\Gamma, \alpha \vdash \Delta \vdash \Pi; \ \Gamma \vdash \Delta, \alpha, \beta \vdash \Pi; \ \Gamma \vdash \Delta \vdash \Pi, \beta}{\Gamma \vdash \Delta \vdash \Pi, \alpha \Rightarrow \beta},$$

(c) *reguły wprowadzające koniunkcję*

$$(lc)_0 \quad \frac{\Gamma, \alpha \vdash \Delta \vdash \Pi; \ \Gamma, \beta \vdash \Delta \vdash \Pi}{\Gamma, \alpha \wedge \beta \vdash \Delta \vdash \Pi},$$

$(lc)_1$ $\dfrac{\Gamma \vdash \Delta, \alpha, \beta \vdash \Pi;\ \Gamma \vdash \Delta, \alpha \vdash \Pi, \beta;\ \Pi \vdash \Delta, \beta \vdash \Pi, \alpha}{\Gamma \vdash \Delta, \alpha \wedge \beta \vdash \Pi}$,

$(lc)_2$ $\dfrac{\Gamma \vdash \Delta \vdash \Pi, \alpha, \beta}{\Gamma \vdash \Delta \vdash \Pi, \alpha \wedge \beta}$,

(d) *reguły wprowadzające alternatywę*

$(la)_0$ $\dfrac{\Gamma, \alpha, \beta \vdash \Delta \vdash \Pi}{\Gamma, \alpha \vee \beta \vdash \Pi}$,

$(la)_1$ $\dfrac{\Gamma, \beta \vdash \Delta, \alpha \vdash \Pi;\ \Gamma \vdash \Delta, \alpha, \beta \vdash \Pi;\ \Gamma, \alpha \vdash \Delta, \beta \vdash \Pi}{\Gamma \vdash \Delta, \alpha \vee \beta \vdash \Pi}$,

$(la)_2$ $\dfrac{\Gamma \vdash \Delta \vdash \Pi, \alpha;\ \Gamma \vdash \Delta \vdash \Pi, \beta}{\Gamma \vdash \Delta \vdash \Pi, \alpha \vee \beta}$,

(e) *reguły wprowadzające kwantyfikatory: szczegółowy i ogólny*

$(l\exists)_0$ $\dfrac{\Gamma, \alpha(a) \vdash \Delta \vdash \Pi}{\Gamma, \exists x \alpha(x) \vdash \Delta \vdash \Pi}$,

$(l\exists)_1$ $\dfrac{\Gamma \vdash \Delta, \alpha(a), \vdash \Pi;\ \Gamma, \alpha(b) \vdash \Delta \vdash \Pi}{\Gamma \vdash \Delta, \exists x \alpha(x) \vdash \Pi}$,

$(l\exists)_2$ $\dfrac{\Gamma \vdash \Delta \vdash \Pi, \alpha(a)}{\Gamma \vdash \Delta \vdash \Pi, \exists x \alpha(x)}$,

$(l\forall)_0$ $\dfrac{\Gamma, \alpha(a) \vdash \Delta \vdash \Pi}{\Gamma, \forall x \alpha(x) \vdash \Delta \vdash \Pi}$,

$(l\forall)_1$ $\dfrac{\Gamma \vdash \Delta, \alpha(a), \vdash \Pi;\ \Gamma \vdash \Delta \vdash \Pi, \alpha(b)}{\Gamma \vdash \Delta}$,

$(l\forall)_2$ $\dfrac{\Gamma \vdash \Delta \vdash \Pi, \alpha(a)}{\Gamma \vdash \Delta \vdash \Pi, \forall x \alpha(x)}$,

gdzie stała a występująca w przesłankach reguł $(l\exists)_1, (l\exists)_2, (l\forall)_o$ i $(l\forall)_1$ nie występuje odpowiednio w ich wniosku.

3. Pojęcia dowodu

Drzewem dowodowym (krótko *drzewem*) trójsekwentu Σ nazywamy drzewo uporządkowane $\mathcal{D} = (D, p, r, \Sigma)$ takie, że

(a) D jest pewnym zbiorem trójsekwentów,

(b) $\Sigma \in D$,

(c) r jest relacją binarną określoną na zbiorze trójsekwentów D w następujący sposób: $(\Sigma_1, \Sigma_2) \in r$ wtedy i tylko wtedy, gdy istnieje reguła $(lx) \in R$ taka, że trójsekwent Σ_2 jest jej wnioskiem, a trójsekwent Σ_1 jedną z jej przesłanek.

Niech $\mathcal{D}_1$ i $\mathcal{D}_2$ będą uporządkowanymi drzewami. Powiemy, że drzewo $\mathcal{D}_2$ jest *rozszerzeniem prostym* drzewa $\mathcal{D}_1$ wtedy i tylko wtedy, gdy drzewo $\mathcal{D}_2$ powstaje z drzewa $\mathcal{D}_1$ przez zastosowanie reguł, których schematy należą do R, w ten sposób, że istnieje liść Σ drzewa $\mathcal{D}_\infty$ będący wnioskiem reguły r.

Wykorzystując pojęcie rozszerzenia prostego drzewa możemy podać inną definicję drzewa dowodowego sekwentu Σ. Powiemy więc, że drzewo $\mathcal{D}$ jest *drzewem dowodowym* sekwentu Σ wtedy i tylko wtedy, gdy istnieje skończony ciąg drzew $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_k$ taki, że $\mathcal{D}_k = \mathcal{D}$ i każdy liść drzewa $\mathcal{D}_k$ jest sekwentem przepełnionym lub atomicznym $\mathcal{D}_1 = \Sigma$ oraz dla każdego $i < k$ drzewo $\mathcal{D}_{i+1}$ jest prostym rozszerzeniem drzewa $\mathcal{D}_i$.

Drzewo będziemy nazywać *przepełnionym*, jeśli na każdej jego gałęzi wystąpi punkt będący sekwentem przepełnionym. Stąd w szczególności, drzewo $\mathcal{D}$ jest przepełnione, jeśli każdy jego liść jest sekwentem przepełnionym.

Podamy jeszcze procedurę konstrukcji *systematycznego drzewa dowodowego* dla sekwentu Σ. W dalszym ciągu, przez *drzewo* sekwentu Σ będziemy rozumieć systematyczne drzewo dowodowe utworzone zgodnie z niżej podaną procedurą. Ma ona następującą postać:

(a) Sekwent Σ przyjmujemy jako korzeń drzewa.

(b) Załóżmy, że zakończony został k-ty etap procedury konstrukcji drzewa. Rozważmy $k+1$ etap. Wtedy

(1) Jeżeli każdy liść skonstruowanego drzewa jest sekwentem przepełnionym lub atomicznym to kończymy procedurę konstrukcji drzewa.

(2) W przeciwnym przypadku, wybieramy na drzewie pierwszy od lewej sekwent nieatomiczny i nieprzepełniony Θ i uwzględniając pewien priorytet w stosowaniu reguł, rozszerzamy drzewo przez zastosowanie określonej reguły.

Mówiąc obrazowo konstruowane drzewo „rośnie” od korzenia przez rozgałęzienia do swoich liści.

4. Konfiguracje

Niech $Y \subseteq S$. Przyjmijmy następujące oznaczenia:

$\mathcal{F}(Y) = \{\vdash_j \beta : j = 0, 1 \;\&\; \beta \in Y\}$,

$\mathcal{T}(Y) = \{\vdash_2 \beta : \beta \in Y\}$,

$\mathcal{M}(Y) = \mathcal{F}(Y) \cup \mathcal{T}(Y).$

Niech $Z \subseteq M(Y)$. Konfiguracją $\mathcal{K}(Z)$ dla ciągu sekwentów Z (każdy zbiór sekwentów możemy ustawić w ciąg) nazywamy drzewo skonstruowane w następujący indukcyjny sposób:

(k_1) jako konfigurację dla ciągu jednowyrazowego Σ_0 bierzemy drzewo dowodowe,

(k_2) jeżeli już mamy utworzone drzewo dla sekwentów $\Sigma_0, \Sigma_1, \ldots, \Sigma_{k-1}$, to każdy nie przepełniony liść Θ tego drzewa, o ile taki istnieje, zastępujemy drzewem dowodowym sekwentu $\Theta\Sigma_k$,

(k_3) kończymy procedurę tworzenia drzewa, jeśli takich liści nie ma.

Łatwo zauważyć, że problem istnienia konfiguracji $\mathcal{K}(Z)$ dla całego zbioru sekwentów Z wiąże się z istnieniem drzew nie przepełnionych dla każdego sekwentu ze zbioru Z. Jeśli ciąg sekwentów Z jest nieskończony, to konfiguracja może być drzewem nieskończonym. Ze względu na to, że każdy punkt drzewa $\mathcal{K}(Z)$ ma skończoną ilość poprzedników, z lematu König'a wynika, że w takiej nieskończonej konfiguracji istnieje gałąź nieskończona.

Konfiguracja $\mathcal{K}(Z)$ jest przepełniona, jeśli na każdej jej gałęzi wystąpi sekwent przepełniony. Oczywiście, konfiguracja $\mathcal{K}(Z)$ jest przepełniona, jeśli każdy jej liść jest sekwentem przepełnionym.

Powiemy, że formuła $\alpha \in S$ jest *twierdzeniem* w sekwencyjnym systemie trójwartościowego rachunku predykatów Łukasiewicza wtedy i tylko wtedy, gdy istnieją przepełnione drzewa dowodowe sekwentów postaci $\vdash_j \alpha$ dla $j = 0,1$.

Lemat 1
Niech $Y \subseteq M(S)$. Jeżeli konfiguracja $\mathcal{K}(Y)$ jest przepełniona, to zawiera skończoną liczbę punktów.

Dowód.
Zgodnie z procedurą konstrukcji konfiguracji $\mathcal{K}(Y)$ ustawiamy elementy zbioru Y w ciąg $\mathbf{C} = (\Sigma_0, \Sigma_1, \ldots, \Sigma_s, \ldots)$ i realizujemy punkty $(k_1), (k_2)$ tej procedury. Załóżmy, że konfiguracja $\mathcal{K}(Y)$ jest przepełniona. Z przepełnienia wynika, że na każdej gałęzi $\mathcal{K}(Y)$ wystąpi sekwent przepełniony. Jest on liściem i w każdym przypadku ze względu na punkt (k_3) procedury, zatrzymuje konstrukcję konfiguracji na pewnym elemencie ciągu $\mathbf{C}$. ■

Lemat 2
Niech $Y \subseteq M(S)$. Jeżeli konfiguracja $\mathcal{K}(Y)$ jest przepełniona, to istnieje skończony podzbiór $Z \subseteq Y$ taki, że $\mathcal{K}(Z)$ jest przepełniona.

Dowód.

Jeśli zbiór Y jest skończony, to można przyjąć, że $Z = Y$. Jeśli Y nie jest zbiorem skończonym, to zgodnie z lematem 1 istnieje skończony ciąg sekwentów $\Sigma_0, \Sigma_1, \ldots, \Sigma_t$, który przepełnia konfigurację $\mathcal{K}(Y)$. Wtedy $Z = \{\Sigma_0, \Sigma_1, \ldots, \Sigma_t\}$. ∎

5. Operacja konsekwencji

Niech $X \subseteq S$ i niech $\alpha \in S$. Powiemy, że formuła α jest *konsekwencją sekwencyjną*, krótko *konsekwencją* zbioru formuł X, co będziemy notować, że $\alpha \in CsX$ wtedy i tylko wtedy, gdy istnieje konfiguracja przepełniona dla zbioru $\mathcal{T}(X)$ lub istnieją konfiguracje przepełnione dla zbiorów $\mathcal{T}(X) \cup \{\vdash_0 \alpha\}$ i $\mathcal{T}(X) \cup \{\vdash_1 \alpha\}$.

Lemat 3

Jeżeli konfiguracja zbioru $\mathcal{T}(X)$ jest przepełniona, to $CsX = S$.

Lemat ten wynika natychmiast z definicji funkcji Cs. ∎

Lemat 4

Jeżeli $X \subseteq S$, to $X \subseteq CsX$.

Dowód.

Niech $\alpha \in X$. Jeśli $\mathcal{K}(\mathcal{T}(X))$ jest przepełniona, to oczywiście $\alpha \in CsX$. Załóżmy że konfiguracja dla $\mathcal{T}(X)$ nie jest przepełniona. Konstruując konfigurację dla X ustawiamy elementy zbioru $\mathcal{T}(X)$ w ciąg. Sekwent $\vdash_2 \alpha$ wystąpi w tym ciągu na m-tym miejscu. W m-tym kroku procedury konstrukcji konfiguracji dla $\mathcal{T}(X)$ każdy nieprzepełniony liść Θ zastępujemy drzewem dla $\Theta * \vdash_2 \alpha$ i ewentualnie kontynuujemy dalej konstrukcję konfiguracji. Rozważmy konfiguracje dla zbiorów $\mathcal{T}(X) \cup \{\vdash_0 \alpha\}$ i $\mathcal{T}(X) \cup \{\vdash_1 \alpha\}$. Elementy tych zbiorów ustawiamy w dwa ciągi tak samo jak dla ciągu $\mathcal{T}(X)$, z tym, że między wyrazami m i $m+1$ wstawiamy odpowiednio $\vdash_0 \alpha$ w pierwszym i $\vdash_1 \alpha$ w drugim. Konfiguracje dla $\mathcal{T}(X) \cup \{\vdash_0 \alpha\}$ i $\mathcal{T}(X) \cup \{\vdash_1 \alpha\}$ tworzy się do $m-1$ kroku tak samo jak dla $\mathcal{T}(X)$. W m-tym kroku konstrukcji każdy liść nieprzepełniony Θ w obydwu konfiguracjach zastępujemy złożeniem $\Theta * \vdash_2 \alpha$ traktując $\vdash_2 \alpha$ jako drzewo jednopunktowe. Oczywiście, tak otrzymane konfiguracje nie są przepełnione. Następnie każdy liść nieprzepełniony $\Theta * \vdash_2 \alpha$ w obydwu konfiguracjach zastępujemy złożeniem $\Theta * \vdash_2 \alpha * \vdash_0 \alpha$ w pierwszej konfiguracji i odpowiednio złożeniem $\Theta * \vdash_2 \alpha * \vdash_1 \alpha$ w drugiej, również traktując sekwenty $\vdash_0 \alpha$ i $\vdash_1 \alpha$ jako drzewa jednopunktowe, które już przepełniają obydwie konfiguracje. ∎

Lemat 5

Niech $X \subseteq S$. Konfiguracja $\mathcal{K}(\mathcal{T}(CsX))$ nie jest przepełniona wtedy i tylko wtedy, gdy nie jest przepełniona konfiguracja $\mathcal{K}(\mathcal{T}(X))$. ∎

Lemat 6 Niech $X, Y \subseteq S$. Jeżeli $X \subseteq Y$, to $CsX \subseteq CsY$.

Dowód.

Załóżmy, że $X \subseteq Y$ i niech $\alpha \in CsX$. Jeżeli nie istnieje przepełniona konfiguracja dla $\mathcal{T}(X)$, to istnieją przepełnione konfiguracje dla $\mathcal{T}(X) \cup \{\vdash_0 \alpha\}$ i $\mathcal{T}(X) \cup \{\vdash_1 \alpha\}$. Jeżeli konfiguracja dla $\mathcal{T}(Y)$ jest przepełniona, to oczywiście $\alpha \in CsY$. Jeśli nie istnieje konfiguracja przepełniona dla $\mathcal{T}(Y)$, to łatwo zauważyć, że $\mathcal{T}(X) \subseteq \mathcal{T}(Y)$ i tym samym $\mathcal{T}(X) \cup \{\vdash_j \alpha\} \subseteq \mathcal{T}(Y) \cup \{\vdash_j \alpha\}$ dla $j = 0, 1$. Stąd konfiguracje dla $\mathcal{T}(Y) \cup \{\vdash_j \alpha\}$ $(j = 0, 1)$ są przepełnione. ▪

Lemat 7

Niech $X \subseteq S$ i niech $\alpha \in S$. Jeżeli $\alpha \in CsX$, to istnieje skończony zbiór $Y \subseteq X$ taki, że $\alpha \in CsY$.

Dowód.

Niech $\alpha \in CsX$. Jeśli nie istnieje konfiguracja przepełniona dla $\mathcal{T}(X)$, to istnieją przepełnione konfiguracje dla $\mathcal{T}(X) \cup \{\vdash_0 \alpha\}$ oraz $\mathcal{T}(X) \cup \{\vdash_1 \alpha\}$. Ponieważ $\mathcal{T}(X) \cup \{\vdash_j \alpha\}$ dla $j = 0, 1$ są przepełnione, więc są skończone. Zbiór Y składa się z tych formuł, które występują w korzeniu $\mathcal{K}(\mathcal{T}(X) \cup \{\vdash_j \alpha\})$ dla $j = 0, 1$ oraz w sekwentach występujących w realizacji punktu (k_2) definicji procedury konstrukcji tych konfiguracji. ▪

Lemat 8

Jeżeli $X \subseteq S$, to $CsCsX \subseteq CsX$.

Dowód.

Niech $X \subseteq S$ i niech $\alpha \in CsCsX$. Jeżeli konfiguracja $\mathcal{K}(\mathcal{T}(X))$ jest przepełniona, to $\alpha \in CsX$. Załóżmy więc, że $\mathcal{K}(\mathcal{T}(X))$ nie jest przepełniona. Na podstawie lematu 5 nie jest też przepełniona konfiguracja $\mathcal{K}(\mathcal{T}(CsX))$. Ponieważ $\alpha \in CsCsX$ więc konfiguracje $\mathcal{K}(\mathcal{T}(CsX) \cup \{\vdash_0 \alpha\}$ i $\mathcal{K}(\mathcal{T}(CsX) \cup \{\vdash_1 \alpha\})$ są przepełnione. Stąd i z definicji funkcji Cs przepełnione są konfiguracje $\mathcal{K}(\mathcal{T}(X) \cup \{\vdash_0 \alpha\})$ i $\mathcal{K}(\mathcal{T}(X) \cup \{\vdash_1 \alpha\})$. Wobec nieprzepełnienia konfiguracji $\mathcal{K}(\mathcal{T}(X))$, mamy $\alpha \in CsX$. ▪

Na podstawie lematów 4 i 7 otrzymujemy:

Wniosek 1

Dla każdego $X \subseteq S; CsCsX = CsX$. ▪

Bibliografia

1. Borowik, P. (1995) *Przykładowy algorytm provera automatycznego dowodzenia twierdzeń.* Streszczenie wykładów i Komunikatów, XXIV Ogólnopolska Konferencja Zastosowań Matematyki (An exemplary algorithm of an automatic theorem prover, Summary of lectures and

reports, 24th National Conference of Applications of Mathematics), Zakopane'95, 13–14.

2. Borowik, P. (1998) *Logiki trójwartościowe. Wersja sekwencyjna.* (W przygotowaniu)
3. Post, E. (1921) *Introduction to a general theory of elementary propositions.* American Journal of Mathematics 43, 163-185.
4. Rine, C. (ed) (1977) *Computer science and multiple-valued logic.* North Holland, Amsterdam.
5. Rosser, J.B., Turquette, A.R. (1952) *Many-valued logics.* North Holand, Amsterdam.
6. Wolf, R.G. (1975) *A critical survey of many-valued logic 1966-1974.* Proceedings of the 1875 International Symposium on Multiple-Valued logic, 468-474.