Automatyczne dowodzenie twierdzeń metodą rezolucji

Wojciech Buszkowski

16 kwietnia 2010

Rezolucja zdaniowa

Formuły rachunku zdań: zbudowane ze zmiennych zdaniowych za pomocą spójników logicznych $\neg$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$ i nawiasów

Wartości logiczne: 1 (prawda), 0 (fałsz)

V - przeliczalny zbiór zmiennych zdaniowych

Wartościowanie: dowolna funkcja $w : V \mapsto \{0, 1\}$

$w(A)$ - wartość logiczna formuły A dla wartościowania w

$w(\neg A) = 1 - w(A)$,

$w(A \wedge B) = \min(w(A), w(B))$, $w(A \vee B) = \max(w(A), w(B))$,

$w(A \rightarrow B) = 1 \Leftrightarrow w(A) \leq w(B)$, $w(A \leftrightarrow B) = 1 \Leftrightarrow w(A) = w(B)$

Dowolne zmienne zdaniowe oznaczamy literami $p.q, r, s$ (z indeksami). Litery A, B, C, D oznaczają dowolne formuły. Litera S oznacza dowolny zbiór formuł.

Definicja 1

Wartościowanie w *spełnia* formułę A, jeżeli $w(A) = 1$; spełnia zbiór formuł, jeżeli spełnia każdą formułę tego zbioru. *Tautologia* jest to formuła spełniona przez każde wartościowanie.

Definicja 2

Formułę nazywamy *spełnialną*, jeżeli istnieje wartościowanie, które ją spełnia. Zbiór formuł nazywamy *spełnialnym*, jeżeli istnieje wartościowanie, które spełnia ten zbiór.

Definicja 3

Formuła A *logicznie wynika* ze zbioru formuł S, jeżeli każde wartościowanie spełniające zbiór S spełnia formułę A.

Fakt 1

Formuła A jest tautologią wtw, gdy formuła $\neg A$ nie jest spełnialna.

Fakt 2

Formuła A logicznie wynika ze zbioru S wtw, gdy zbiór $S \cup \{\neg A\}$ nie jest spełnialny.

Literał: dowolna formuła postaci p lub $\neg p$

Klauzula: alternatywa skończenie wielu literałów, np. $p \vee \neg q \vee r$

Fakt 3

Wartościowanie w spełnia klauzulę A wtw, gdy w spełnia przynajmniej jeden literał klauzuli A.

Klauzula pusta: 0 (nie jest spełniona przez żadne wartościowanie)

Definicja 4

Formuła A jest *logicznie równoważna* formule B, jeżeli $w(A) = w(B)$ dla każdego wartościowania w (tzn. $A \leftrightarrow B$ jest tautologią).

Fakt 4

Każda formuła jest logicznie równoważna koniunkcji skończenie wielu (niepustych) klauzul (tzn. formule w koniunkcyjnej postaci nornmalnej, kpn).

$(p \to q) \leftrightarrow (\neg p \vee q)$ jest tautologią, więc $p \to q$ i $\neg p \vee q$ są logicznie równoważne.

Algorytmy sprowadzania formuł do kpn działają w czasie wykładniczym.

Wielomianowa procedura przekształcania dowolnej formuły A w formułę B w kpn taką, że:

A jest spełnialna wtw, gdy B jest spełnialna.

Jeżeli A jest zmienną, to $B = A$. Zakładamy, że A nie jest zmienną.

Każdej podformule C formuły A przypisujemy zmienną p_C, przy czym $p_q = q$ dla zmiennych q występujących w A.

$C_1, \ldots, C_k$ - wszystkie złożone podformuły formuły A, przy czym $C_1 = A$.

$B = p_A \wedge F(C_1) \wedge \cdots \wedge F(C_k)$, gdzie:

$F(C_i)$ jest kpn formuły $p_{C_i} \leftrightarrow \neg p_D$, jeżeli $C_i = \neg D$, czyli $F(C_i) = (\neg p_{C_i} \vee \neg p_D) \wedge (p_D \vee p_{C_i})$,

$F(C_i)$ jest kpn formuły $p_{C_i} \leftrightarrow (p_D \wedge p_E)$, jeżeli $C_i = D \wedge E$;

podobnie dla $C_i = D \vee E$, $C_i = D \rightarrow E$, $C_i = D \leftrightarrow E$.

Fakt 5

Formuła w kpn jest spełnialna wtw, gdy zbiór klauzul tej formuły jest spełnialny.

Metoda dowodzenia tautologii:

(1) wyznaczamy formułę B, która jest spełnialna wtw, gdy formuła $\neg A$ jest spełnialna,
(2) wykazujemy, że zbiór klauzul formuły B jest niespełnialny.

Reguła rezolucji zdaniowej (RRZ); A. Blake 1937, J Robinson 1965.

$$\frac{A \vee p\,;\;\; B \vee \neg p}{A \vee B}$$

Na przykład:

$$\frac{p \vee q \vee r\,;\;\; p \vee \neg s \vee \neg r}{p \vee q \vee \neg s}$$

Kolejność i powtórzenia literałów klauzuli są nieistotne.

Wniosek reguły (RRZ) nazywamy *rezolwentą* przesłanek.

Fakt 6

Wniosek reguły (RRZ) logicznie wynika ze zbioru przesłanek.

Definicja 5

Dowodem rezolucyjnym klauzuli A na podstawie zbioru klauzul S nazywamy skończony ciąg klauzul $(A_1, \ldots, A_n)$ taki, że $A_n = A$ i każda klauzula A_i jest elementem zbioru S lub rezolwentą pewnych klauzul A_j, A_k dla $j, k < i$.

Przykład. $S = \{p \vee \neg q,\ \neg p \vee r,\ q\}$.

1. $p \vee \neg q$ (z S)
2. q (z S)
3. p (z 1,2)
4. $\neg p \vee r$ (z S)
5. r (z 3,4)

Definicja 6

$S \vdash_{RZ} A$ (A jest wyprowadzalne z S) wtw, gdy istnieje dowód rezolucyjny klauzuli A na podstawie zbioru klauzul S.

Fakt 7

Jeżeli $S \vdash_{RZ} A$, to A logicznie wynika z S.

Twierdzenie 1 (pełności rezolucji zdaniowej)

Zbiór klauzul S jest niespełnialny wtw, gdy $S \vdash_{RZ} 0$.

DOWÓD. Implikacja ($\Leftarrow$) wynika z Faktu 7.
Implikację ($\Rightarrow$) udowodnimy najpierw dla skończonych zbiorów S przez indukcję po liczbie zmiennych występujących w klauzulach zbioru S; oznaczmy tę liczbę przez $v(S)$.
$v(S) = 0$. Zakładamy, że zbiór S jest niespełnialny. Ponieważ pusty zbiór formuł jest spełnialny, więc $S \neq \emptyset$. Zatem $S = \{0\}$, a stąd $S \vdash_{RZ} 0$.

$v(S) = n > 0$. Zakładamy, że $(\Rightarrow)$ zachodzi dla wszelkich zbiorów S' takich, że $v(S') < n$. Niech $p_1, \ldots, p_n$ będą wszystkimi zmiennymi, występującymi w klauzulach zbioru S.

Określamy zbiory S_1 i S_2. S_1 powstaje z S przez usunięcie wszystkich klauzul, zawierających p_n, i usunięcie $\neg p_n$ z pozostałych klauzul. S_2 określamy analogicznie, zamieniając role p_n i $\neg p_n$.

Zakładamy, że zbiór S jest niespełnialny. Wtedy zbiory S_1 i S_2 są niespełnialne. Ponieważ $v(S_1) < n$ i $v(S_2) < n$, więc $S_1 \vdash_{RZ} 0$ i $S_2 \vdash_{RZ} 0$.

Jeżeli istnieje dowód rezolucyjny klauzuli 0 z S_1 lub z S_2, będący dowodem z S, to $S \vdash_{RZ} 0$.

W przeciwnym przypadku, dowód klauzuli 0 z S_1 zawiera istotnie przynajmniej jedną klauzulę, powstającą z klauzuli z S przez usunięcie $\neg p_n$. Wtedy $S \vdash_{RZ} \neg p_n$. Podobnie $S \vdash_{RZ} p_n$, ponieważ dowód klauzuli 0 z S_2 zawiera istotnie przynajmniej jedną klauzulę powstającą z klauzuli z S przez usunięcie p_n. Zatem $S \vdash_{RZ} 0$.

Dla nieskończonych zbiorów S korzystamy z **twierdzenia o zwartości**:

Zbiór S jest spełnialny wtw,gdy każdy skończony podzbiór zbioru S jest spełnialny.

Zakładamy, że zbiór S jest niespełnialny. Wtedy istnieje skończony zbiór $S' \subseteq S$, który nie jest spełnialny. Mamy $S' \vdash_{RZ} 0$, a więc $S \vdash_{RZ} 0$. Q.E.D.

DYGRESJA. Twierdzenie o zwartości w rachunku zdań wynika ze zwartości przestrzeni topologicznej $\{0,1\}^N$ (nieskończonych ciągów binarnych, czyli wartościowań) z topologią produktową wyznaczoną przez topologię dyskretną na $\{0,1\}$. Dla każdego A zbiór $W(A) = \{w : w(A) = 1\}$ jest domknięty. Jeżeli każdy skończony podzbiór zbioru S jest spełnialny, to rodzina $\{W(A) : A \in S\}$ jest scentrowaną rodziną zbiorów domkniętych; stąd $\bigcap_{A \in S} W(A) \neq \emptyset$, a więc zbiór S jest spełnialny.

Termy: zbudowane ze zmiennych indywiduowych i stałych indywiduowych za pomocą symboli funkcyjnych.

V - przeliczalny zbiór zmiennych indywiduowych

Zmienne oznaczamy x, y, z, stałe indywiduowe a, b, c, symbole funkcyjne f, g. Literą t oznaczamy dowolny term.
Przykłady: $x, y, f(x, y), a, b, f(a, f(x, b)), g(x), g(f(x, y))$

Formuły atomowe (atomy): $P(t_1, \ldots, t_n)$, gdzie P jest n–argumentowym symbolem relacyjnym, a $t_1, \ldots, t_n$ są termami.
Symbole relacyjne oznaczamy P, Q, R.

Przykłady: $P(x, y), P(a, b), P(f(x, y), f(a, x))$

Język: jest określony przez skończony zbiór symboli relacyjnych, skończony zbiór symboli funkcyjnych i skończony zbiór stałych indywiduowych. Symbole relacyjne i funkcyjne mają jednoznacznie określoną liczbę argumentów.

Formuły: zbudowane z atomów za pomocą spójników logicznych i kwantyfikatorów $\forall x$,$\exists x$.

Zdania: formuły bez zmiennych wolnych

Przykład: $\forall x(P(x) \to \exists y Q(x, y))$ - zdanie

Termy i atomy *ustalone*: termy i atomy nie zawierające zmiennych (ang. ground).

Przykłady: $a, b, f(a, b), g(a), f(g(a), b), P(f(a, b), g(a))$

Modele

Struktura dla języka L: struktura relacyjna M z niepustym uniwersum U_M, w której każdy symbol relacyjny jest interpretowany jako pewna relacja na U_M, każdy symbol funkcyjny jako pewne działanie na U_M, a każda stała indywiduowa jako pewien element zbioru U_M.

Określa się relację $M \models A$ (zdanie A jest prawdziwe w M) w naturalny sposób (A. Tarski 1933).

Strukturę M nazywamy *modelem* zbioru zdań S, jeżeli $M \models A$ dla każdego $A \in S$ (piszemy $M \models S$).

Zdanie A nazywamy *tautologią rachunku predykatów*, jeżeli $M \models A$ dla każdej struktury M (dla danego języka).

Zdanie A nazywamy *spełnialnym*, jeżeli istnieje M takie, że $M \models A$. Zbiór zdań S nazywamy *spełnialnym*, jeżeli istnieje M takie, że $M \models S$.

Mówimy, że zdanie A *logicznie wynika* ze zbioru zdań S, jeżeli A jest prawdziwe w każdym modelu zbioru S (piszemy $S \models_{RP} A$).

Prawdziwe są Fakt 1 i Fakt 2.

Zdanie uniwersalne: zdanie $\forall x_1 \ldots \forall x_n A$ takie, że A jest formułą bez kwantyfikatorów (*otwartą*).

Niech $B = \forall x_1 \ldots \forall x_n A$ będzie zdaniem uniwersalnym. Określamy $gr(B)$ jako zbiór wszystkich *ustalonych instancji* formuły A, tzn. wszystkich zdań postaci $A[x_1/t_1, \ldots, x_n/t_n]$ takich, że $t_1, \ldots, t_n$ są termami ustalonymi. Zakładamy $gr(B) \neq \emptyset$.

Niech S będzie zbiorem zdań uniwersalnych. Określamy:

$$gr(S) = \bigcup_{B \in S} gr(B)$$

Twierdzenie 2 (twierdzenie Herbranda)

Zbiór zdań uniwersalnych S jest spełnialny w rachunku predykatów wtw, gdy zbiór $gr(S)$ jest spełnialny w rachunku zdań (różne atomy traktujemy jako różne zmienne zdaniowe).

Wniosek 1

Zbiór zdań uniwersalnych S jest niespełnialny w rachunku predykatów wtw, gdy $gr(S) \vdash_{RZ} 0$.

Dla każdego zdania A możemy skonstruować zdanie uniwersalne A_u, spełniające warunek: A jest spełnialne wtw, gdy A_u jest spełnialne.

(1) Zdanie A przekształcamy w logicznie równoważne zdanie A' w *preneksowej postaci normalnej* $Q_1 x_1 \ldots Q_n x_n C$, gdzie C jest formułą otwartą, a $Q_1, \ldots, Q_n \in \{\forall, \exists\}$. A jest logicznie równoważne A', jeżeli:

$$\text{dla każdego } M,\ M \models A \Leftrightarrow M \models A'.$$

(2) Eliminujemy kwantyfikatory istnienia z A'.

Zdanie $\exists x D$ zastępujemy zdaniem $D[x/c]$, gdzie c jest nową stałą.

Zdanie $\forall x_1 \ldots \forall x_n \exists x D$ zastępujemy zdaniem $\forall x_1 \ldots \forall x_n D[x/f(x_1, \ldots, x_n)]$, gdzie f jest nowym symbolem funkcyjnym.

Po skończonej liczbie takich przekształceń otrzymamy zdanie A_u.

Podobnie dla każdego skończonego zbioru zdań S możemy skonstruować skończony zbiór zdań uniwersalnych S_u, spełniający warunek: zbiór S jest spełnialny wtw, gdy zbiór S_u jest spełnialny.

Fakt 8

Dla każdego zdania A następujące warunki są równoważne:
(a) zdanie A jest tautologią rachunku predykatów,
(b) zdanie $\neg A$ jest niespełnialne,
(c) zdanie $(\neg A)_u$ jest niespełnialne,
(d) $gr((\neg A)_u) \vdash_{RZ} 0$.

Fakt 9

Niech S będzie skończonym zbiorem zdań, a A zdaniem. Następujące warunki są równoważne:
(a) $S \models_{RP} A$,
(b) zbiór $S \cup \{\neg A\}$ *jest niespełnialny,*
(c) zbiór $(S \cup \{\neg A\})_u$ *jest niespełnialny,*
(d) $gr((S \cup \{\neg A\})_u) \vdash_{RZ} 0$.

Fakt 9 można uogólnić na nieskończone zbiory S.

Przykład

Stałe a, b, c, d, e. Relacja $R(x, y)$; sens: x jest rodzicem y.
Relacja $F(x)$; sens: x jest kobietą.
$S = \{R(a, b), R(b, c), R(c, d), R(d, e), F(a), F(c), F(e)\}$ (baza danych)
$A = \exists x(F(a) \wedge R(a, x) \wedge R(x, c))$, sens: a jest babcią c.
Wykażemy $S \models_{RP} A$.

$\neg A$ jest równoważne zdaniu $\forall x(\neg F(a) \vee \neg R(a, x) \vee \neg R(x, c))$,
$(\neg A)_u = \neg A$.
$gr((\neg A)_u)$ zawiera klauzulę: $\neg F(a) \vee \neg R(a, b) \vee \neg R(b, c)$.

1. $\neg F(a) \vee \neg R(a, b) \vee \neg R(b, c)$
2. $F(a)$ (z S)
3. $\neg R(a, b) \vee \neg R(b, c)$ (z 1,2)
4. $R(a, b)$ (z S)
5. $\neg R(b, c)$ (z 3,4)
6. $R(b, c)$ (z S)
7. 0 (z 5,6)

$B = \exists x \forall y P(x,y) \rightarrow \forall y \exists x P(x,y)$

$\neg B$ jest równoważne zdaniu $\exists x \forall y P(x,y) \wedge \exists y \forall x \neg P(x,y)$.

Sprowadzamy $\neg B$ do preneksowej postaci normalnej.

$\exists x \forall y P(x,y) \wedge \exists u \forall z \neg P(z,u)$

$\exists x \exists u \forall y \forall z (P(x,y) \wedge \neg P(z,u))$

$(\neg B)_u = \forall y \forall z (P(a,y) \wedge \neg P(z,b))$

$gr((\neg B)_u)$ zawiera zdanie: $P(a,b) \wedge \neg P(a,b)$.

Zatem $gr((\neg B)_u) \vdash_{RZ} 0$.

Jeżeli w języku początkowym lub w wyniku skolemizacji pojawiają się symbole funkcyjne, to zbiór ustalonych instancji jest nieskończony; np. za x trzeba podstawiać $a, f(a), f(f(a))$ itd. Opisana procedura daje tylko *algorytm pozytywny*.

Problem spełnialności zdania z dowolnymi kwantyfikatorami (uniwersalnego z symbolami funkcyjnymi) jest nierozstrzygalny.

Złożoność obliczeniowa

Dane reprezentujemy jako *łańcuchy* symboli.

Σ - skończony zbiór symboli (alfabet)

Σ^* - zbiór wszystkich łańcuchów nad Σ

$\Sigma = \{0, 1\}$, $\Sigma^* = \{\epsilon, 0, 1, 00, 01, 10, 11, 000, \ldots\}$

Język nad Σ: dowolny zbiór $L \subseteq \Sigma^*$

$|x|$ - długość łańcucha x

Maszyna Turinga (MT): automat akceptujący pewne łańcuchy nad Σ; MT *akceptuje* x, jeżeli obliczenie dla wejścia x prowadzi do stanu akceptującego (wtedy obliczenie kończy się).

$L(M)$ - zbiór wszystkich łańcuchów akceptowanych przez maszynę M (język akceptowany przez M)

Detrministyczna MT ma najwyżej jeden ruch w każdej sytuacji; dla danego wejścia x generuje dokładnie jedno obliczenie.

Język L nazywamy *rekurencyjnym*, jeżeli $L = L(M)$ dla pewnej deterministycznej MT M, nie generującej nieskończonych obliczeń.

$t_M(x)$ - czas obliczenia maszyny M dla łańcucha wejściowego x. Dla deterministycznej MT jest to liczba kroków, czyli długość, jedynego obliczenia.

$L \in \mathrm{P}$ wtw, gdy istnieją deterministyczna MT M bez nieskończonych obliczeń oraz wielomian $p(n)$ takie, że $L = L(M)$ i $t_M(x) \leq p(|x|)$ dla każdego łańcucha wejściowego $x \in \Sigma^*$.

Systemem dowodzenia dla języka $L \subseteq \Sigma^*$ nazywamy predykat $P \subseteq \Sigma^* \times \Gamma^*$ taki, że $P \in \mathrm{P}$ oraz dla każdego $x \in \Sigma^*$ zachodzi równoważność:

$$x \in L \Leftrightarrow (\exists \pi \in \Gamma^*) P(x, \pi) .$$

Mówimy, że system dowodzenia P dla L jest *wielomianowo ograniczony*, jeżeli istnieje wielomian $p(n)$ taki, że dla każdego $x \in \Sigma^*$:

$$x \in L \Leftrightarrow (\exists \pi \in \Gamma^*)(|\pi| \leq p(|x|) \wedge P(x, \pi)).$$

$L \in$ NP wtw, gdy istnieje wielomianowo ograniczony system dowodzenia dla L.

Fakt 10

P $\subseteq$ NP.

Dowód. Niech $L = L(M)$, przy czym $t_M(x) \leq p(|x|)$ dla każdego $x \in \Sigma^*$. Określamy predykat: $P(x, \pi)$ wtw, gdy π jest obliczeniem akceptującym maszyny M dla wejścia x. Jeżeli $P(x, \pi)$, to $|\pi| = O((p(|x|))^2)$. Q.E.D.

Problem.

P $=$ NP?

Zdanie P $=$ NP to słynna hipoteza, w której prawdziwość nikt nie wierzy, lecz nie udowodniono jej fałszywości.

Dla $L \subseteq \Sigma^*$ oznaczamy $\overline{L} = \{x \in \Sigma^* : x \notin L\}$ (dopełnienie języka L).

$L \in$ co-K wtw, gdy $\overline{L} \in$ K

Ponieważ klasa P jest zamknięta ze względu na dopełnienie języków, więc P $=$ co-P.

Problem.

NP $=$ co-NP?

Fakt 11

Jeżeli P $=$ NP*, to* NP $=$ co-NP *(jeżeli* NP $\neq$ co-NP*, to* P $\neq$ NP*).*

$L \subseteq \Sigma^*$, $L' \subseteq \Gamma^*$. Określamy relację: $L' \leq_P L$ wtw, gdy istnieje funkcja $f : \Gamma^* \mapsto \Sigma^*$, obliczalna przez deterministyczną MT z wielomianowym ograniczeniem czasu i spełniająca warunek:

$$(\forall x \in \Gamma^*)(x \in L' \Leftrightarrow f(x) \in L).$$

Język L nazywamy NP-*trudnym*, jeżeli $L' \leq_P L$ dla każdego języka $L' \in$ NP; NP-*zupełnym*, jeżeli ponadto $L \in$ NP.

SAT - język wszystkich spełnialnych formuł rachunku zdań

CSAT - język wszystkich spełnialnych formuł rachunku zdań w kpn

Twierdzenie 3 (Cook 1971)

SAT i CSAT są NP*-zupełne.*

Fakt 12

P $=$ NP *wtw, gdy istnieje język* NP*-zupełny, należący do* P.

TAUTOLOGY - zbiór wszystkich tautologii rachunku zdań

System dowodzenia dla języka TAUTOLOGY nazywamy *systemem dowodzenia dla rachunku zdań*.

TAUTOLOGY należy do co-NP. Jest to język co-NP-zupełny

Twierdzenie 4

NP $=$ co-NP *wtw, gdy TAUTOLOGY należy do* NP*, czyli, gdy istnieje wielomianowo ograniczony system dowodzenia dla rachunku zdań (Cook, Reckhow 1979).*

Złożoność dowodów rezolucyjnych

π - dowód rezolucyjny klauzuli 0 ze zbioru klauzul S

Miary złożoności:

$s(S)$ (size) - liczba wystąpień literałów w zbiorze klauzul S; podobnie dla klauzuli A

$l(\pi)$ (length) - liczba klauzul występujących w π

$s(\pi)$ (size) - suma wszystkich $s(A)$ dla A występujących w π

$w(\pi)$ (width) - maksymalna wartość $s(A)$ dla A występujących w π

$sp(\pi)$ (space) - maksymalna liczba klauzul, które należy przechowywać w pamięci w trakcie wykonywania π

$L(S)$ - najmniejsza wartość $l(\pi)$ dla dowodów $\pi : S \vdash 0$

$W(S)$, $\mathrm{Sp}(S)$ - określamy podobnie

Haken 1985: rezolucja nie jest wielomianowo ograniczonym systemem dowodzenia.

Silniejsze wyniki: Urquhart 1987, Beame, Karp, Pitassi, Saks 2002 i inni.

$|\pi| = \mathrm{O}(s(\pi))$

$s(\pi) \leq l(\pi) \cdot w(\pi)$

$w(\pi) \leq s(S)$

Przyjmujemy, ze klauzule nie zawierają dwóch przeciwnych literałów $p, \neg p$.

$W(S) = \mathrm{O}(\sqrt{n \cdot \log\, L(S)})$; n - liczba zmiennych w S

(Ben-Sasson, Wigderson 2001)

S jest $k-$CNF. Wtedy $W(S) \leq \mathrm{Sp}(S) + c$, gdzie c jest stałą zależną od k (Atserias, Dalmau 2003)

[1] M. Ben-Ari, Logika matematyczna w informatyce, WNT.

[2] K. Doets, From Logic to Logic Programming, The MIT Press.

[3] J. E. Hopcroft i J.D. Ullman, Wprowadzenie do teorii automatów, języków i obliczeń, Wydawnictwo Naukowe PWN.

[4] P. Beame i T. Pitassi, Propositional proof complexity: Past, present, and future, 1998 (link z Wikipedii).

[5] S.A. Cook and R. Reckhow, The relative efficiency of propositional proof systems, Journal of Symbolic Logic 44.1, 1979.

[6] A. Haken, The intractability of resolution, Theoretical Computer Science 39.2-3, 1985.

[7] P. Clote i E. Kranakis, Boolean Functions and Computation Models, Springer, 2002.