METODY DOWODZENIA TWIERDZEŃ I AUTOMATYZACJA ROZUMOWAŃ

WYKŁAD 6A: REZOLUCJA

III rok kognitywistyki UAM, 2016–2017

1 Rezolucja w KRZ

Dowody rezolucyjne w KRZ są równie proste, jak dowody tablicowe.

Metoda rezolucji także jest metodą nie wprost: dla wykazania, że φ jest tezą, staramy się wyprowadzić *zamkniętą derywację rezolucyjną* dla formuły $\neg\varphi$.

Dowody rezolucyjne wykorzystują koniunkcyjne postacie normalne: pracujemy na uogólnionych koniunkcjach uogólnionych alternatyw.

W propozycji Fittinga dowody rezolucyjne zapisujemy w sposób liniowy: krokami dowodowymi są poszczególne uogólnione alternatywy.

1.1 Reguły

Reguły *tworzenia derywacji rezolucyjnej* zapisane w notacji Smullyana (dla języka KRZ) są następujące:

$$\frac{\neg\neg\psi}{\psi} \qquad \frac{\neg\top}{\bot} \qquad \frac{\neg\bot}{\top}$$

$$\begin{array}{c} \beta \\ | \\ \beta_1 \\ | \\ \beta_2 \end{array} \qquad\qquad \begin{array}{c} \alpha \\ \diagup\quad\diagdown \\ \alpha_1 \quad \alpha_2 \end{array}$$

Reguła dla β-formuł działa wewnątrz alternatywy, reguła dla α-formuł tworzy dwie alternatywy, zapisywane jedna pod drugą.

Jeśli D_1 oraz D_2 są dwiema uogólnionymi alternatywami oraz φ występuje jako jeden z członów alternatywy w D_1, natomiast $\neg\varphi$ występuje jako jeden z członów alternatywy w D_2, to niech D będzie wynikiem wykonania następujących operacji:

1. usunięcia wszystkich wystąpień φ z D_1
2. usunięcia wszystkich wystąpień $\neg\varphi$ z D_2
3. utworzenia jednej alternatywy D z alternatyw otrzymanych w wyniku wykonania dwóch powyższych kroków.

Mówimy w takim przypadku, że D jest *rezolwentą* D_1 oraz D_2 (względem pary formuł: φ, $\neg\varphi$; czasami krócej: względem formuły φ).

Jeśli alternatywa F zawiera wystąpienia literału $\bot$, to alternatywę powstałą z niej poprzez usunięcie wszystkich wystąpień $\bot$ nazywamy *trywialną* rezolwentą F.

Reguła rezolucji.

D *wynika rezolucyjnie* z D_1 oraz D_2, gdy D jest rezolwentą D_1 oraz D_2 względem pewnej formuły φ. Podobnie, D *trywialnie wynika rezolucyjnie* z D_1, gdy D jest trywialną rezolwentą D_1.

Oczywiście szczególnym przypadkiem tak rozumianej reguły rezolucji jest rezolucja względem literału (i tak rozumie się regułę rezolucji w większości podręczników):

Jeśli alternatywa D_1 zawiera literał L, natomiast alternatywa D_2 zawiera literał do niego komplementarny $\overline{L}$, to *rezolwentą* D_1 oraz D_2 względem L jest alternatywa D powstająca poprzez usunięcie wszystkich wystąpień L z D_1, usunięcie wszystkich wystąpień $\overline{L}$ z D_2 i utworzenie jednej alternatywy z tak otrzymanych wyników.

Jeśli alternatywy potraktujemy jak zbiory formuł, to tak rozumiana reguła rezolucji przyjmuje następującą postać formalną:

$$\frac{D_1 \qquad D_2}{(D_1 - \{L\}) \cup (D_2 - \{\overline{L}\})}.$$

Derywację rezolucyjną definiujemy w sposób następujący:

1. Niech $\{ \varphi_1, \varphi_2, \ldots, \varphi_n \}$ będzie skończonym zbiorem formuł języka KRZ. *Derywacją rezolucyjną* zbioru $\{ \varphi_1, \varphi_2, \ldots, \varphi_n \}$ nazywamy ciąg:

$$\begin{array}{c} [\varphi_1] \\ | \\ [\varphi_2] \\ | \\ \vdots \\ | \\ [\varphi_n] \end{array}$$

2. Jeśli S jest derywacją rezolucyjną dla $\{ \varphi_1, \varphi_2, \ldots, \varphi_n \}$, zaś D powstaje z pewnych wierszy S poprzez zastosowanie jakiejś reguły tworzenia derywacji rezolucyjnej lub reguły rezolucji, to ciąg S przedłużony o element D także jest derywacją rezolucyjną dla $\{ \varphi_1, \varphi_2, \ldots, \varphi_n \}$.

Derywację rezolucyjną zawierającą alternatywę pustą [ ] nazywamy *zamkniętą*. Pamiętamy, że wartość logiczna alternatywy pustej równa jest 0 dla każdego wartościowania zmiennych zdaniowych. Alternatywę pustą oznacza się w wielu podręcznikach przez $\Box$.

Dowodem rezolucyjnym formuły φ nazywamy zamkniętą derywację rezolucyjną dla $\{\neg\varphi\}$. Mówimy, że φ jest *tezą* systemu rezolucyjnego dla KRZ, co zapisujemy $\vdash_{pr} \varphi$, gdy φ ma dowód rezolucyjny.

1.2 Przykład

Oto przykład dowodu rezolucyjnego formuły

$$((p \wedge q) \vee (r \to s)) \to ((p \vee (r \to s)) \wedge (q \vee (r \to s)))$$

1.	$[\neg(((p \wedge q) \vee (r \to s)) \to ((p \vee (r \to s)) \wedge (q \vee (r \to s))))]$	
2.	$[(p \wedge q) \vee (r \to s)]$	α, 1
3.	$[\neg((p \vee (r \to s)) \wedge (q \vee (r \to s)))]$	α, 1
4.	$[p \wedge q, r \to s]$	β, 2
5.	$[p, r \to s]$	α, 4
6.	$[q, r \to s]$	α, 4
7.	$[\neg(p \vee (r \to s)), \neg(q \vee (r \to s))]$	β, 3
8.	$[\neg p, \neg(q \vee (r \to s))]$	α, 7
9.	$[\neg(r \to s), \neg(q \vee (r \to s))]$	α, 7
10.	$[\neg p, \neg q]$	α, 8
11.	$[\neg p, \neg(r \to s)]$	α, 8
12.	$[\neg(r \to s), \neg q]$	α, 9
13.	$[\neg(r \to s), \neg(r \to s)]$	α, 9
14.	$[p, \neg q]$	RR: 7,12, $r \to s$
15.	$[\neg q]$	RR: 10,14, p
16.	$[r \to s]$	RR: 6,15, q
17.	$[\,]$	RR: 13,16, $r \to s$

Nie jest to jedyny dowód rezolucyjny rozważanej formuły (spróbuj znaleźć inne).

1.3 Trafność i pełność rezolucji w KRZ

Powiemy, że derywacja rezolucyjna S jest *spełnialna*, jeśli istnieje wartościowanie, przy którym każdy wiersz w S ma wartość 1.

Fakt. Dowolne zastosowanie reguły konstrukcji derywacji rezolucyjnej lub reguły rezolucji do spełnialnej derywacji rezolucyjnej daje spełnialną derywację rezolucyjną.

Fakt. Jeśli istnieje zamknięta derywacja rezolucyjna dla zbioru formuł S, to S nie jest spełnialny.

Twierdzenie o Trafności Metody Rezolucyjnej w KRZ. Jeśli φ ma dowód rezolucyjny, to φ jest tautologią KRZ.

Niech S będzie zbiorem alternatyw. *Derywacją rezolucyjną ze zbioru* S (*wyprowadzeniem rezolucyjnym ze zbioru* S) jest ciąg alternatyw, z których każda:

1. jest elementem S *lub*
2. powstaje z wcześniejszego elementu tego ciągu poprzez zastosowanie którejś z reguł tworzenia derywacji rezolucyjnej *lub*
3. powstaje z wcześniejszych elementów tego ciągu jako ich rezolwenta.

Mówimy, że alternatywa D jest *rezolucyjnie wyprowadzalna* z S, gdy D jest ostatnim elementem jakiejś derywacji rezolucyjnej z S.

Niech φ będzie dowolną formułą języka KRZ. Przez *φ-powiększenie* alternatywy $[\varphi_1, \varphi_2, \ldots, \varphi_n]$ rozumiemy zarówno $[\varphi, \varphi_1, \varphi_2, \ldots, \varphi_n]$ jak i $[\varphi_1, \varphi_2, \ldots, \varphi_n]$. Jeśli S jest zbiorem alternatyw, zaś S^* jest wynikiem zastąpienia każdego elementu S przez jego φ-powiększenie, to mówimy, że S^* jest *φ-powiększeniem* S.

Fakt o powiększeniach. Załóżmy, że S_1 i S_2 są zbiorami alternatyw oraz S_2 jest φ-powiększeniem S_1. Jeśli alternatywa D_1 jest rezolucyjnie wyprowadzalna z S_1, to istnieje φ-powiększenie D_2 alternatywy D_1 takie, że D_2 jest rezolucyjnie wyprowadzalna z S_2.

Faktu tego dowodzimy przez indukcję po długości wyprowadzenia rezolucyjnego z S_1 (zob. Fitting 1990, 59).

Skończony zbiór S formuł języka KRZ nazywamy *rezolucyjnie niesprzecznym*, gdy nie istnieje zamknięte wyprowadzenie rezolucyjne z S.

Na mocy tej definicji $\{ \varphi_1, \varphi_2, \ldots, \varphi_n \}$ jest rezolucyjnie niesprzeczny, gdy nie istnieje wyprowadzenie rezolucyjne z $\{ [\varphi_1], [\varphi_2], \ldots, [\varphi_n] \}$ pustej alternatywy [ ].

Zbiory formuł, które nie są rezolucyjnie niesprzeczne, nazwiemy *rezolucyjnie sprzecznymi*. Tak więc, S jest rezolucyjnie sprzeczny, gdy istnieje wyprowadzenie rezolucyjne z $\{ [\varphi_1], [\varphi_2], \ldots, [\varphi_n] \}$ pustej alternatywy [ ].

Lemat. Rodzina wszystkich zbiorów rezolucyjnie niesprzecznych jest zdaniową własnością niesprzeczności.

Dowód. Trzeba sprawdzić, że zachodzą wszystkie warunki definiujące zdaniową własność niesprzeczności.

Warunki dla zmiennych zdaniowych, $\perp$ oraz $\neg\top$ są oczywiste. Pozostałe przypadki wymagają odrobinę żmudnego, aczkolwiek nietrudnego sprawdzenia. Jak czyniliśmy to już poprzednio, w każdym z przypadków prowadzimy dowód nie wprost:

1. Załóżmy, że $\neg\neg\psi \in S$ i przypuśćmy, że $S \cup \{\psi\}$ nie jest rezolucyjnie niesprzeczny. Pokażemy, że S nie jest rezolucyjnie niesprzeczny.

   Niech zatem $S = \{\ \varphi_1, \varphi_2, \ldots, \varphi_n, \neg\neg\psi\ \}$. Ponieważ $S = \{\ \varphi_1, \varphi_2, \ldots, \varphi_n, \psi\ \}$ nie jest rezolucyjnie niesprzeczny, więc istnieje wyprowadzenie, powiedzmy $\mathbb{D}$, alternatywy pustej [ ] ze zbioru $\{\ [\varphi_1], [\varphi_2], \ldots, [\varphi_n], [\psi]\ \}$. Zbudujmy wyprowadzenie $\mathbb{D}^*$ z $\{\ [\varphi_1], [\varphi_2], \ldots, [\varphi_n], [\neg\neg\psi]\ \}$ w sposób następujący: najpierw stosujemy regułę dotyczącą podwójnej negacji, otrzymując $\{\ [\varphi_1], [\varphi_2], \ldots, [\varphi_n], [\neg\neg\psi], [\psi]\ \}$, a następnie wykonujemy wszystkie kroki wyprowadzenia $\mathbb{D}$. Wtedy oczywiście $\mathbb{D}^*$ jest wyprowadzeniem alternatywy pustej [ ] z $\{\ [\varphi_1], [\varphi_2], \ldots, [\varphi_n], [\psi]\ \}$, czyli S nie jest rezolucyjnie niesprzeczny.

2. Załóżmy, że $\alpha \in S$ i przypuśćmy, że $S \cup \{\alpha_1, \alpha_2\}$ nie jest rezolucyjnie niesprzeczny. Pokażemy, że S nie jest rezolucyjnie niesprzeczny.

   Niech zatem $S = \{\ \varphi_1, \varphi_2, \ldots, \varphi_n, \alpha\ \}$. Ponieważ $S \cup \{\alpha_1, \alpha_2\}$ nie jest rezolucyjnie niesprzeczny, więc istnieje wyprowadzenie, powiedzmy $\mathbb{D}$, alternatywy pustej [ ] ze zbioru $\{\ [\varphi_1], [\varphi_2], \ldots, [\varphi_n], [\alpha], [\alpha_1], [\alpha_2]\ \}$. Zbudujmy wyprowadzenie $\mathbb{D}^*$ z $\{\ [\varphi_1], [\varphi_2], \ldots, [\varphi_n], [\alpha]\ \}$ w sposób następujący: najpierw stosujemy α-regułę otrzymując $\{\ [\varphi_1], [\varphi_2], \ldots, [\varphi_n], [\alpha], [\alpha_1], [\alpha_2]\ \}$, a następnie wykonujemy wszystkie kroki wyprowadzenia $\mathbb{D}$. Wtedy oczywiście $\mathbb{D}^*$ jest wyprowadzeniem alternatywy pustej [ ] z $\{\ [\varphi_1], [\varphi_2], \ldots, [\varphi_n], [\alpha]\ \}$, czyli S nie jest rezolucyjnie niesprzeczny.

3. Załóżmy, że $\beta \in S$ i przypuśćmy, że ani $S \cup \{\beta_1\}$ ani $S \cup \{\beta_2\}$ nie jest rezolucyjnie niesprzeczny. Pokażemy, że S nie jest rezolucyjnie niesprzeczny.

   Niech zatem $S = \{\ \varphi_1, \varphi_2, \ldots, \varphi_n, \beta\ \}$. Zauważmy, że zastosowanie rezolucyjnej β-reguły do wyprowadzenia rozpoczynającego się od $[\varphi_1], [\varphi_2], \ldots, [\varphi_n], [\beta]$ pozwala nam dodać doń $[\beta_1, \beta_2]$. Tak więc, aby pokazać, że S nie jest rezolucyjnie niesprzeczny, wystarczy pokazać, że istnieje rezolucyjne wyprowadzenie alternatywy pustej [ ] z $[\varphi_1], [\varphi_2], \ldots, [\varphi_n], [\beta], [\beta_1], [\beta_2]$.

Ponieważ $S \cup \{\beta_1\}$ nie jest rezolucyjnie niesprzeczny, więc istnieje rezolucyjne wyprowadzenie alternatywy pustej [ ] z $\{ [\varphi_1], [\varphi_2], \ldots, [\varphi_n], [\beta], [\beta_1] \}$. Wtedy na mocy faktu o powiększeniach istnieje rezolucyjne wyprowadzenie z $[\varphi_1], [\varphi_2], \ldots, [\varphi_n], [\beta], [\beta_1], [\beta_2]$: albo alternatywy pustej [ ], albo $[\beta_2]$. W pierwszym przypadku dowód nie wprost jest zakończony. W przypadku drugim, ponieważ $S \cup \{\beta_2\}$ nie jest rezolucyjnie niesprzeczny, więc istnieje rezolucyjne wyprowadzenie, powiedzmy $\mathbb{D}$, alternatywy pustej [ ] z:

$$\{ [\varphi_1], [\varphi_2], \ldots, [\varphi_n], [\beta], [\beta_2] \}$$

Wszystkie alternatywy tworzące to wyprowadzenie występują w skonstruowanym wcześniej wyprowadzeniu z $[\varphi_1], [\varphi_2], \ldots, [\varphi_n], [\beta], [\beta_1], [\beta_2]$. Dołączając do nich wyprowadzenie $\mathbb{D}$ otrzymujemy wyprowadzenie alternatywy pustej [ ] z $\{ [\varphi_1], [\varphi_2], \ldots, [\varphi_n], [\beta], [\beta_2] \}$, co kończy dowód nie wprost.

Twierdzenie o Pełności Metody Rezolucyjnej w KRZ. Jeśli φ jest tautologią KRZ, to φ ma dowód rezolucyjny.
Dowód. Przeprowadzimy dowód nie wprost. Jeśli φ nie ma dowodu rezolucyjnego, to nie istnieje zamknięta derywacja rezolucyjna dla $\{\neg\varphi\}$. Wtedy $\{\neg\varphi\}$ jest rezolucyjnie niesprzeczny, a zatem jest spełnialny, na mocy udowodnionego przed chwilą lematu oraz Twierdzenia o Istnieniu Modelu. A zatem φ nie jest tautologią KRZ.

Podobnie, jak w przypadku tablic analitycznych możemy też rozważać rezolucyjne wyprowadzenia ze zbioru przesłanek. Jeśli S jest zbiorem formuł języka KRZ, to dla dowolnej $\varphi \in S$ w derywacji rezolucyjnej dodajemy jako osobny wiersz $[\varphi]$. Piszemy $S \vdash_{pr} \varphi$, gdy istnieje zamknięta derywacja rezolucyjna dla $\neg\varphi$, wykorzystująca przesłanki z S.

2 Zalecenia praktyczne

Powyżej omówiono metodę rezolucji w KRZ wedle propozycji w Fitting 1990. Nieco inaczej omawia się to zagadnienie w podręczniku Nerode, Shore 1997. Zachęcamy słuchaczy do stosowania metody rezolucji w KRZ z wykorzystaniem reguły rezolucji dla literałów, a nie (jak w powyższym przykładzie Fittinga) dla dowolnych formuł. Jest to bodaj bardziej rozpowszechniona wersja metody rezolucji.

W tym punkcie podamy kilka przykładów tak właśnie rozumianych derywacji rezolucyjnych. Omówimy dwa typy zagadnień:

1. ustalanie czy dana formuła jest tezą systemu rezolucyjnego w KRZ

2. ustalanie, czy dany zbiór formuł jest rezolucyjnie sprzeczny.

W każdym z tych przypadków będziemy wykorzystywać notację Smullyana oraz algorytm podany przez Fittinga. Z notacją Smullyana słuchacze są już chyba dobrze oswojeni, algorytm Fittinga był omawiany na wykładzie dotyczącym postaci normalnych.

2.1 Ustalanie czy formuła jest tezą

Przypomnijmy:

1. Derywację rezolucyjną zawierającą alternatywę pustą [ ] nazywamy *zamkniętą*. Pamiętamy, że wartość logiczna alternatywy pustej równa jest 0 dla każdego wartościowania zmiennych zdaniowych. Alternatywę pustą oznacza się w wielu podręcznikach przez $\Box$.

2. *Dowodem rezolucyjnym* formuły φ nazywamy zamkniętą derywację rezolucyjną dla $\{\neg\varphi\}$. Mówimy, że φ jest *tezą* systemu rezolucyjnego dla KRZ, co zapisujemy $\vdash_{pr} \varphi$, gdy φ ma dowód rezolucyjny.

Dla ustalenia czy formuła φ jest tezą systemu rezolucyjnego badamy zatem czy z $[\neg\varphi]$ możemy wyprowadzić klauzulę pustą, stosując algorytm Fittinga oraz regułę rezolucji. Innymi słowy:

1. Rozpoczynamy od $[\neg\varphi]$.

2. Stosujemy algorytm Fittinga tak długo, aż otrzymamy alternatywy elementarne, zapisując każdy krok w postaci osobnego wiersza (lub dwóch wierszy). Przy tym:

   (a) jeśli działamy na β-formule, to wiersz $[\ldots, \beta, \ldots]$ przekształcamy na wiersz $[\ldots, \beta_1, \beta_2, \ldots]$,

   (b) jeśli działamy na α-formule, to wiersz $[\ldots, \alpha, \ldots]$ przekształcamy na dwa wiersze $[\ldots, \alpha_1, \ldots]$ oraz $[\ldots, \alpha_2, \ldots]$,

   (c) jeśli działamy na formule podwójnie zanegowanej, to wiersz $[\ldots, \neg\neg\varphi, \ldots]$ przekształcamy na wiersz $[\ldots, \varphi, \ldots]$,

3. Stosujemy – tam, gdzie to możliwe – regułę rezolucji.

4. Sprawdzamy, czy otrzymaliśmy klauzulę pustą:

(a) jeśli otrzymaliśmy klauzulę pustą, to formuła φ jest tezą systemu rezolucyjnego

(b) jeśli nie jest możliwe otrzymanie klauzuli pustej, to formuła φ nie jest tezą systemu rezolucyjnego.

PRZYKŁAD. Czy formuła $((p \to q) \wedge \neg q) \to \neg p$ ma dowód rezolucyjny?

1.	$[\neg(((p \to q) \wedge \neg q) \to \neg p)]$	
2.	$[(p \to q) \wedge \neg q]$	α, 1
3.	$[\neg\neg p]$	α, 1
4.	$[p]$	$\neg\neg$, 2
5.	$[p \to q]$	α, 2
6.	$[\neg q]$	α, 2
7.	$[\neg p, q]$	β, 5
8.	$[q]$	RR: 4,7
9.	$[\,]$	RR: 6,8

Otrzymaliśmy klauzulę pustą. A zatem formuła $((p \to q) \wedge \neg q) \to \neg p$ ma dowód rezolucyjny.

PRZYKŁAD. Czy formuła $(p \to q) \to (q \to p)$ ma dowód rezolucyjny?

1.	$[\neg((p \to q) \to (q \to p))]$	
2.	$[p \to q]$	α, 1
3.	$[\neg(q \to p)]$	α, 1
4.	$[q]$	α, 3
5.	$[\neg p]$	α, 3
6.	$[\neg p, q]$	β, 2

Dotarliśmy do alternatyw elementarnych. Nie jest możliwe stosowanie reguły rezolucji. Nie otrzymaliśmy klauzuli pustej. Badana formuła nie ma dowodu rezolucyjnego.

2.2 Ustalanie czy zbiór formuł jest niespełnialny

Zbiór $\{ \varphi_1, \varphi_2, \ldots, \varphi_n \}$ formuł języka KRZ jest *rezolucyjnie sprzeczny*, jeśli istnieje zamknięta derywacja rezolucyjna tego zbioru, czyli taka, która zawiera klauzulę pustą.

Dla ustalenia, czy zbiór formuł $\{ \varphi_1, \varphi_2, \ldots, \varphi_n \}$ jest rezolucyjnie sprzeczny wykonujemy następujące czynności:

1. Tworzymy listę $[\varphi_1], [\varphi_2], \ldots, [\varphi_n]$. Przyjętym zwyczajem jest zapisywanie poszczególnych elementów listy w osobnych wierszach.

2. Stosujemy algorytm Fittinga do elementów tej listy tak długo, aż otrzymamy alternatywy elementarne. Przy tym:

   (a) jeśli działamy na β-formule, to wiersz $[\ldots, \beta, \ldots]$ przekształcamy na wiersz $[\ldots, \beta_1, \beta_2, \ldots]$,

   (b) jeśli działamy na α-formule, to wiersz $[\ldots, \alpha, \ldots]$ przekształcamy na dwa wiersze $[\ldots, \alpha_1, \ldots]$ oraz $[\ldots, \alpha_2, \ldots]$,

   (c) jeśli działamy na formule podwójnie zanegowanej, to wiersz $[\ldots, \neg\neg\varphi, \ldots]$ przekształcamy na wiersz $[\ldots, \varphi, \ldots]$,

3. Stosujemy – tam, gdzie to możliwe – regułę rezolucji.

4. Sprawdzamy, czy otrzymaliśmy klauzulę pustą:

   (a) jeśli otrzymaliśmy klauzulę pustą, to wyjściowy zbiór formuł jest rezolucyjnie sprzeczny;

   (b) jeśli nie jest możliwe otrzymanie klauzuli pustej, to wyjściowy zbiór formuł nie jest rezolucyjnie sprzeczny (w konsekwencji, jest spełnialny).

PRZYKŁAD. Ustal czy jest rezolucyjnie sprzeczny zbiór formuł:

$$\{\ q \to p,\ \neg r \lor q,\ s \to r,\ \neg(s \to p)\ \}$$

1.	$[q \to p]$	
2.	$[\neg r \lor q]$	
3.	$[s \to r]$	
4.	$[\neg(s \to p)]$	
5.	$[\neg q, p]$	β,1
6.	$[\neg r, q]$	β,2
7.	$[\neg s, r]$	β,3
8.	$[s]$	α,4
9.	$[\neg p]$	α,4
10.	$[r]$	RR:7,8
11.	$[\neg q]$	RR:5,9
12.	$[\neg r]$	RR:6,11
13.	$[\,]$	RR:10,12

Otrzymaliśmy klauzulę pustą, a więc badany zbiór jest rezolucyjnie sprzeczny.

PRZYKŁAD. Ustal czy jest rezolucyjnie sprzeczny zbiór formuł:

$$\{\ \neg q \lor p,\ s \land \neg p,\ r \to q,\ s \to r\ \}$$

1.	$[\neg q \vee p]$	
2.	$[s \wedge \neg p]$	
3.	$[r \to q]$	
4.	$[s \to r]$	
5.	$[\neg q, p]$	β,1
6.	$[s]$	α,2
7.	$[\neg p]$	α,2
8.	$[\neg r, q]$	β,3
9.	$[\neg s, r]$	β,4
10.	$[r]$	RR:6,9
11.	$[q]$	RR:8,10
12.	$[p]$	RR:5,11
13.	[ ]	RR:7,12

Otrzymaliśmy klauzulę pustą, a więc badany zbiór jest rezolucyjnie sprzeczny.

PRZYKŁAD. Ustal czy jest rezolucyjnie sprzeczny zbiór formuł:

$$\{\ s \wedge \neg p,\ \neg q \vee p,\ \neg r \vee q,\ \neg s \vee r\ \}$$

1.	$[s \wedge \neg p]$	
2.	$[\neg q \vee p]$	
3.	$[\neg r \vee q]$	
4.	$[\neg s \vee r]$	
5.	$[s]$	α,1
6.	$[\neg p]$	α,1
7.	$[\neg q, p]$	β,2
8.	$[\neg r, q]$	β,3
9.	$[\neg s, r]$	β,4
10.	$[\neg q]$	RR:6,7
11.	$[\neg r]$	RR:8,10
12.	$[\neg s]$	RR:9,11
13.	[ ]	RR:5,12

Otrzymaliśmy klauzulę pustą, a więc badany zbiór jest rezolucyjnie sprzeczny.

PRZYKŁAD. Ustal czy jest rezolucyjnie sprzeczny zbiór formuł:

$$\{\ s \to r,\ q \to p,\ \neg(s \to p),\ q \vee \neg r\ \}$$

1.	$[s \to r]$	
2.	$[q \to p]$	
3.	$[\neg(s \to p)]$	
4.	$[q \vee r]$	
5.	$[\neg s, r]$	β,1
6.	$[\neg q, p]$	β,2
7.	$[s]$	α,3
8.	$[\neg p]$	α,3
9.	$[q, \neg r]$	β,4
10.	$[r]$	RR:5,7
11.	$[q]$	RR:9,10
12.	$[p]$	RR:6,11
13.	$[\,]$	RR:8,12

Otrzymaliśmy klauzulę pustą, a więc badany zbiór jest rezolucyjnie sprzeczny.
PRZYKŁAD. Ustal czy jest rezolucyjnie sprzeczny zbiór formuł:

$$\{\ s \to r,\ \neg s \vee p,\ q \vee \neg r\ \}$$

1.	$[s \to r]$	
2.	$[\neg s \vee p]$	
3.	$[q \vee \neg r]$	
4.	$[\neg s, r]$	β, 1
5.	$[\neg s, p]$	β, 2
6.	$[q, \neg r]$	β, 3
7.	$[\neg s, q]$	RR: 4,6

Dotarliśmy do alternatyw elementarnych. Możliwe było zastosowanie reguły rezolucji, jednak nie otrzymaliśmy klauzuli pustej. Badany zbiór nie jest zatem rezolucyjnie sprzeczny.
PRZYKŁAD. Ustal czy jest rezolucyjnie sprzeczny zbiór formuł:

$$\{\ p \to q, q \wedge \neg r, r \to \neg p\ \}$$

1.	$[p \to q]$	
2.	$[q \wedge \neg r]$	
3.	$[r \to \neg p]$	
4.	$[q]$	α, 2
5.	$[\neg r]$	α, 2
6.	$[\neg p, q]$	β, 1
7.	$[\neg r, \neg p]$	β, 3

Dotarliśmy do alternatyw elementarnych. Nie ma możliwości stosowania reguły rezolucji. Nie otrzymaliśmy klauzuli pustej. Badany zbiór nie jest rezolucyjnie sprzeczny.
UWAGA. Należy pamiętać, że regułę rezolucji można stosować do danych wierszy jedynie raz.

3 Rezolucja w KRP

W tym wykładzie ograniczymy się do bardzo ogólnych uwag, dotyczących metody rezolucji w językach pierwszego rzędu. Planujemy osobny wykład, w którym omówiona zostanie procedura *unifikacji* (*uzgadniania*) i jej znaczenie dla bardziej subtelnych analiz z wykorzystaniem tablic analitycznych (ze zmiennymi wolnymi) oraz rezolucji w językach pierwszego rzędu.

3.1 Reguły

W przypadku języków pierwszego rzędu, dodajemy następujące reguły rozszerzania rezolucyjnego (w notacji Smullyana):

$$\begin{array}{c} \gamma \\ | \\ \gamma(t) \end{array} \qquad \text{(dla dowolnego termu domkniętego języka } L^{\mathbf{par}}\text{)}$$

$$\begin{array}{c} \delta \\ | \\ \delta(a) \end{array} \qquad \text{(dla dowolnego nowego parametru } a\text{)}$$

Wprowadzone uprzednio pojęcia (derywacja rezolucyjna, dowód rezolucyjny, itp.) przenoszą się, z oczywistymi modyfikacjami składniowymi, na przypadek języków pierwszego rzędu.

Piszemy $S \vdash_{fr} \varphi$, gdy istnieje zamknięta derywacja rezolucyjna dla $\neg\varphi$, wykorzystująca przesłanki z S.

3.2 Przykład

Oto dowód rezolucyjny zdania $\forall x(P(x) \vee Q(x)) \rightarrow (\exists x P(x) \vee \forall x Q(x))$:

1.	$[\neg(\forall x(P(x) \vee Q(x)) \rightarrow (\exists x P(x) \vee \forall x Q(x)))]$	
2.	$[\forall x(P(x) \vee Q(x))]$	α, 1
3.	$[\neg(\exists x P(x) \vee \forall x Q(x))]$	α, 1
4.	$[\neg\exists x P(x)]$	α, 3
5.	$[\neg\forall x Q(x)]$	α, 3
6.	$[\neg Q(a)]$	δ, 5, a
7.	$[\neg P(a)]$	γ, 4, a
8.	$[P(a) \vee Q(a)]$	γ, 2, a
9.	$[P(a), Q(a)]$	β, 8
10.	$[Q(a)]$	RR: 7,9, $P(a)$
11.	$[\,]$	RR: 10,6, $Q(a)$

3.3 Trafność i pełność rezolucji w KRP

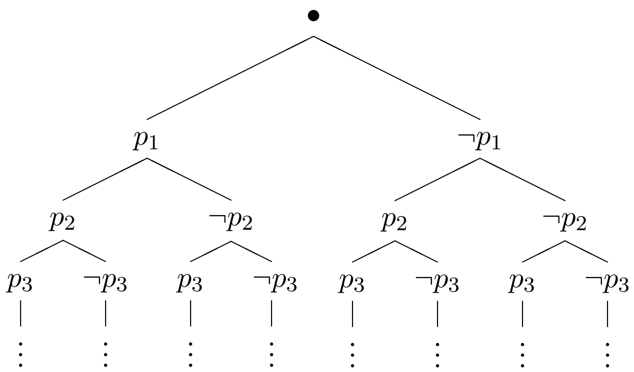
Derywacja rezolucyjna $\mathbb{D}$ jest *spełnialna*, jeśli istnieje model, w którym prawdziwa jest każda alternatywa należąca do $\mathbb{D}$.
Fakt. Zastosowanie dowolnej reguły konstrukcji derywacji rezolucyjnej oraz reguły rezolucji do spełnialnej derywacji rezolucyjnej daje w wyniku spełnialną derywację rezolucyjną.
Trafność rezolucji w KRP. Jeśli φ ma dowód rezolucyjny, to φ jest tautologią KRP.

Skończony zbiór S języka $L^{\mathbf{par}}$ jest *rezolucyjnie niesprzeczny*, gdy nie istnieje zamknięta derywacja rezolucyjna dla S.
Fakt. Rodzina wszystkich zbiorów rezolucyjnie niesprzecznych jest własnością niesprzeczności pierwszego rzędu.
Pełność rezolucji w KRP. Jeśli φ jest tautologią KRP, to φ ma dowód rezolucyjny.

4 Dodatek: drzewo semantyczne

Przeprowadzimy dowód twierdzenia głoszącego, że nasycony rezolucyjnie niespełnialny zbiór klauzul musi zawierać klauzulę pustą (Proposition 3.8.4 w Fitting 1990, 63–63). Dowód korzysta z pojęcia *drzewa semantycznego*.

Ustalmy porządek zmiennych zdaniowych: niech będą one wyliczone jako p_1, p_2, p_3, p_4,... Przez *drzewo semantyczne* rozumiemy pełne drzewo dwójkowe, którego wierzchołki znakowane są literałami, wedle następującej zasady:



Każda gałąź θ w tym drzewie jednoznacznie określa wartościowanie literałów na niej występujących (a więc także wartościowanie wszystkich zmiennych zdaniowych).

Jeśli $\mathbf{C}$ jest zbiorem klauzul, to powiemy, że $\mathbf{C}$ jest *rezolucyjnie nasycony*, gdy zastosowanie reguły rezolucji do elementów $\mathbf{C}$ daje w wyniku również element $\mathbf{C}$.

Udowodnimy teraz następujące twierdzenie:

TWIERDZENIE. Jeśli $\mathbf{C}$ jest rezolucyjnie nasycony oraz niespełnialny, to w $\mathbf{C}$ znajduje się klauzula pusta.

DOWÓD. Przyjmujemy skrótowe oznaczenia dotyczące literałów komplementarnych: $\overline{p} = \neg p$, $\overline{\neg p} = p$. Drzewo semantyczne oznaczmy przez T.

Niech θ będzie ścieżką w T. Powiemy, że θ jest *sprzeczna* z klauzulą C, jeśli dla każdego literału $L \in C$, na θ występuje $\overline{L}$. Ścieżkę nazywamy $\mathbf{C}$-*zamkniętą*, jeśli jest ona sprzeczna z jakąś klauzulą w nasyconym rezolucyjnie zbiorze $\mathbf{C}$. Wierzchołek N drzewa T nazywamy *wierzchołkiem porażki*, gdy ścieżka od korzenia drzewa T do N jest $\mathbf{C}$-zamknięta.

Chcemy pokazać, że korzeń drzewa jest wierzchołkiem porażki. Będzie to oznaczało, że w $\mathbf{C}$ znajduje się klauzula pusta, a więc otrzymamy tezę twierdzenia.

Każda gałąź w T musi być $\mathbf{C}$-zamknięta, ponieważ w przeciwnym przypadku istniałaby gałąź θ w T, która nie byłaby sprzeczna z żadną klauzulą w $\mathbf{C}$, a w takim przypadku wartościowanie zmiennych zdaniowych v_θ odpowiadające θ spełniałoby $\mathbf{C}$.

Niech θ będzie gałęzią w T. Ponieważ θ jest $\mathbf{C}$-zamknięta, więc istnieje klauzula $C \in \mathbf{C}$, z którą θ jest sprzeczna. Ponieważ C jest skończona, pewien skończony fragment początkowy gałęzi θ jest sprzeczny z C. Inaczej mówiąc, każda gałąź w T ma skończony fragment początkowy, który jest $\mathbf{C}$-zamknięty, czyli każda gałąź w T zawiera wierzchołek porażki.

Niech T^* będzie poddrzewem drzewa T, w którym opuszczamy wszystkich potomków wierzchołków porażki. Chcemy pokazać, że T^* składa się jedynie z korzenia drzewa T.

Przeprowadzamy dowód nie wprost. Przypuśćmy zatem, że T^* jest różne od korzenia drzewa T.

Drzewo T^* jest skończonym drzewem dwójkowym. Na mocy lematu Königa, musi zatem być skończone. Ma więc skończoną liczbę gałęzi, a zatem istnieje w nim gałąź M o maksymalnej długości. Z przypuszczenia dowodu nie wprost wynika, że gałąź M zakończona jest wierzchołkiem, który jest potomkiem jakiegoś wierzchołka. Rozważmy przypadek, gdy ostatni wierzchołek M jest lewym potomkiem, a więc gdy M ma postać θ, L, gdzie θ jest ścieżką od korzenia drzewa T do przodka wierzchołka L (przypadek prawego potomka rozpatruje się podobnie).

Zauważmy, że na mocy konstrukcji drzewa T^*: L jest wierzchołkiem porażki, ścieżka θ, L jest **C**-zamknięta, a ścieżka θ nie jest **C**-zamknięta.

Niech R będzie prawym wierzchołkiem, z którym L ma wspólnego bezpośredniego przodka. Wtedy R także jest wierzchołkiem porażki, gdyż w przeciwnym razie najkrótsza **C**-zamknięta ścieżka zaczynająca się od θ, R byłaby dłuższa od θ, L, co przeczyłoby temu, że θ, L jest ścieżką o maksymalnej długości w T^*. Ponieważ θ nie jest **C**-zamknięta, więc θ, R musi być gałęzią w T^*.

Niech literałem na wierzchołku L będzie zmienna p. Wtedy literałem na wierzchołku R jest $\neg p$. Obie gałęzie: θ, L oraz θ, R są **C**-zamknięte, a więc są sprzeczne z pewną klauzulą w **C**. Niech np. θ, L będzie sprzeczna z C_L, zaś θ, R będzie sprzeczna z C_R. Wtedy $\neg p$ musi występować w C_L, bo gdyby tak nie było, to już ścieżka θ byłaby sprzeczna z C_L, pamiętamy jednak, że θ nie jest **C**-zamknięta. Podobnie, p musi występować w C_R. Niech teraz C będzie rezolwentą klauzul C_L oraz C_R. Wtedy θ jest sprzeczna z C. Z założenia, że **C** jest rezolucyjnie nasycony otrzymujemy: $C \in \mathbf{C}$, a to oznacza, że θ jest **C**-zamknięta. Przypuszczenie dowodu nie wprost musimy zatem odrzucić, a w konsekwencji, drzewo T^* jest identyczne z korzeniem drzewa T. To z kolei oznacza, że korzeń drzewa T jest wierzchołkiem porażki, czyli **C** zawiera klauzulę pustą.

JERZY POGONOWSKI
Zakład Logiki i Kognitywistyki UAM
www.kognitywistyka.amu.edu.pl
http://logic.amu.edu.pl/index.php/Dydaktyka
pogon@amu.edu.pl