

IMPLEMENTACJA RACHUNKU SEKWENTOWEGO

Waldemar Wietrzykowski

Digital Intelligence Laboratory, czerwiec 2018

Streszczenie *niniejsza praca przedstawia implementację rachunku sekwentowego.*

Abstract *this work presents the implementation of sequent calculus.*

Implementacją rachunku tautologicznego jest rachunek sekwentowy. Okazuje się, że implementacja ta ma bardziej ogólne i uniwersalne znaczenie niż jego pierwowzór, jakim jest rachunek tautologiczny. Umożliwia nawet postawienie na wstępie niniejszej pracy hipotezy, że mechanizm funkcjonowania mózgu w zakresie świadomego działania jest o wiele prostszy niż do tej pory się sądzi, a powodem nie wyjścia jeszcze z etapu stawiania hipotez w tej dziedzinie jest nie znalezienie spójnego i szczegółowego wyjaśnienia tego zjawiska. Sądzę, że dobrym kandydatem na podstawę tego wyjaśnienia jest rachunek sekwentowy, będący implementacją rachunku tautologicznego.

produkcje

Podstawą rachunku sekwentowego jest implikacja¹, której poprzednik stanowią alternatywnie połączone predykaty (lub jeden predykat dla prostej implikacji), natomiast następnik stanowią alternatywnie połączone elementy (lub jeden element) posiadające różne nazwy, jak: sygnał zespolony² (składający się z sygnału wejściowego i uczącego), obiekt zespolony (składający się z obiektu i własności), produkcja³ lub po prostu alternatywa⁴. Element ten można też nazwać regułą. W pracy⁵ Ariel Zylberberga i innych autorów element ten nazwano produkcją, którą zdefiniowano jako regułę składającą się z warunku i działania. W tej pracy skłoniłem się na to, aby element ten nazwać także produkcją.

Rachunek segmentowy traktuje produkcję całościowo, bez wnikania z szczegóły wewnętrzne produkcji ani w jej naturę. Produkcję na przykład można traktować jako regułę składającą się z warunku i działania, ale może to też być wystąpienie obiektu z którąś z jej własności lub następujące pary elementów: <symbol, desygnat symbolu>, <identyfikacja obiektu, ciało obiektu>, <identyfikacja zjawiska, przebieg zjawiska>, <fakt, dowód zaistnienia faktu>, <postulat, uzasadnienie prawdziwości postulat (jego dowód)>, <identyfikacja obiektu, cecha rzeczywista obiektu>, <identyfikacja akcji lub działania lub zdarzenia, dowód wystąpienia akcji lub działania lub zdarzenia>, <nieprecyzyjna identyfikacja czegoś w języku naturalnym, dowód jego wystąpienia>, <element świadomości, podświadomość związana z postrzeganiem>, {nazwa pojęcia, definicja pojęcia}, itd. Produkcja może też być

¹ Waldemar Wietrzykowski, *Wystarczalność rachunku sekwentowego. Rachunek tautologiczny*, DIL 2018

² Waldemar Wietrzykowski, *Samoucząca się maszyna motoryczna*, DIL 2017

³ Waldemar Wietrzykowski, *Teoria świadomej maszyny*, DIL 2018

⁴ Waldemar Wietrzykowski, *Teoria wiedzy praktycznej*, DIL 2018

⁵ Ariel Zylberberg, Stanislas Dehaene, Pieter R. Roelfsema, Mariano Sigman, *The human Turing machine: a neural framework for mental programs*, Trends in Cognitive Sciences, July 2011

punktem połączenia teorii z praktyką, gdzie pierwszym elementem jest element teorii, a drugim elementem odpowiadająca teorii praktyka. Można to być też złożenie elementu mentalnego z elementem rzeczywistym, którym może być rzeczywista cecha, własność, akcja, działanie, itd., a także połączenie podświadomości ze świadomością, jako propozycja rozwiązania trudnego problemu związanego z qualia.

Jedyną operacją wykonywaną na produkcjach w rachunku sekwentowym jest operacja równości dwóch produkcji. Operacja ta umożliwia sprawdzenie obecności określonej produkcji bez wnikania w szczegóły złożoności samej produkcji.

Prawo zamiany koniunkcji w implikacje

Wyobraźmy sobie tradycyjną maszynę Turinga, na której taśmie w kolejnych komórkach są umieszczone produkcje. Maszyna produkcje te traktuje w sposób całościowy jako symbole, które następnie porównywane są z częścią warunkową reguł tworzących program maszyny. W zależności od treści tego programu aktywowane mogą być różne akcje maszyny. W odróżnieniu od tradycyjnej maszyny, samoucząca się maszyna Turinga w stanie początkowym może nie mieć żadnego programu (czyli żadnej reguły). W tym stanie tradycyjna maszyna nie mogłaby funkcjonować, dlatego wyposażono ją w pewien uniwersalny dla wszystkich samouczących się maszyn program ich obsługi oparty na prawach rachunku sekwentowego.

Prawa rachunku sekwentowego wywodzą się z praw rachunku tautologicznego⁶. Pierwsze prawo dotyczące zamiany koniunkcji w implikacje⁷, umożliwia zamianę ciągu produkcji na taśmie maszyny na ciąg implikacji. Treść tego prawa jest następująca: *Prawo zamiany koniunkcji w implikacje polega na zastąpieniu znaków koniunkcji “^” znakami implikacji “=>” a następnie rozdzieleniu pojedynczych implikacji znakami koniunkcji “^”.*

Przykład:

$$[(p \wedge q \wedge r \wedge s)] \Rightarrow [(p \Rightarrow q) \wedge (q \Rightarrow r) \wedge (r \Rightarrow s)]$$

Powyższe prawo rachunku tautologicznego zamiany koniunkcji w implikacje można uogólnić do pierwszego prawa rachunku sekwentowego zamiany dowolnego ciągu elementów w ciąg par w ten sposób, że:

Pierwsze prawo rachunku sekwentowego – zamiany elementów w pary

do każdego elementu ciągu, oprócz pierwszego i ostatniego, dołączamy ten sam element i uzyskujemy ciąg kolejnych par elementów.

Przykład:

$$p, q, r, s \Rightarrow p q, q r, r s$$

⁶ Waldemar Wietrzykowski, *Wystarczalność rachunku sekwentowego. Rachunek tautologiczny*, DIL 2018

⁷ Waldemar Wietrzykowski, *Wystarczalność rachunku sekwentowego. Rachunek tautologiczny*, DIL 2018, str. 5

Jakie ma znaczenie pierwsze prawo rachunku sekwentowego? Mówi ona, że każda produkcja ma reprezentację w maszynie w postaci pary tych samych produkcji..

Przykład:

$$q \rightarrow q, q$$

Każda z produkcji tej pary spełnia inną rolę w maszynie. Pierwsza produkcja jest produkcją zapamiętaną w maszynie, a druga produkcja pełni rolę adresu pojemnika w maszynie na produkcje występujące po tej produkcji. Mechanizm zapełniania się tego pojemnika opisuje piąte prawo rachunku tautologicznego – prawo dodawania następników. Treść tego prawa jest następująca: *Prawo dodawania następników polega na przekształceniu implikacji posiadających ten sam poprzednik w jedną implikację, której poprzednikiem jest powtarzający się poprzednik tych implikacji a następnikiem alternatywa następników tych implikacji.*⁸

Przykład:

$$[(p \Rightarrow q) \wedge (p \Rightarrow r)] \Rightarrow [p \Rightarrow (q \vee r)]$$

Piąte prawo rachunku tautologicznego dodawania następników można uogólnić do piątego prawa rachunku sekwentowego **konsolidacji** wszystkich par elementów posiadających ten sam poprzednik w ten sposób, że:

Piąte prawo rachunku sekwentowego - dodawania następników

Jeżeli w ciągu par elementów występują pary o tym samym poprzedniku to ulegają one konsolidacji w jedną parę, w której następnikiem jest zbiór nie powtarzających się następników tych par.

Piąte prawo można również określić następująco:

Jeżeli pewien predykat wskazuje na kilka produkcji, to produkcje te leżą w tej samym stanie co ta produkcja.

Przykład:

$$\boxed{p \ q, \ p \ r \Rightarrow p \ (q, \ r)}$$

Piąte prawo rachunku sekwentowego ze względu na zawarte w nim prawo przemienności uniezależnia każdą parę od zajmowanej przez nią pozycji w tym ciągu, gdyż niezależnie od tej pozycji wynik konsolidacji jest ten sam.

Okazuje się, że piąte prawo odnosi się również do poprzednika pary w postaci czwartego prawa rachunku tautologicznego. Treść tego prawa jest następująca: *Prawo dodawania poprzedników polega na*

⁸ Waldemar Wietrzykowski, *Wystarczalność rachunku sekwentowego. Rachunek tautologiczny*, DIL 2018, str. 6

*przekształceniu implikacji posiadających ten sam następnik w jedną implikację, której poprzednikiem jest alternatywa poprzedników tych implikacji, a następnikiem powtarzający się następnik*⁹.

Przykład:

$$[(p \Rightarrow r) \wedge (q \Rightarrow r)] \Rightarrow [(p \vee q) \Rightarrow r]$$

Czwarte prawo rachunku tautologicznego dodawania poprzedników można uogólnić do czwartego prawa rachunku sekwentowego **konsolidacji** wszystkich par elementów posiadających ten sam następnik w ten sposób, że:

Czwarte prawo rachunku sekwentowego - dodawania poprzedników

Jeżeli w ciągu par elementów występują pary o tym samym następniku to ulegają one konsolidacji w jedną parę, w której poprzednikiem jest zbiór nie powtarzających się poprzedników tych par.

Piąte prawo można również określić następująco:

Jeżeli na pewną produkcję wskazuje kilka predykatów to leżą one w tym samym stanie co ta produkcja.

Przykład:

$$p \ r, q \ r \Rightarrow (p, q) \ r$$

Czwarte prawo rachunku sekwentowego ze względu na zawarte w nim prawo przemienności uniezależnia każdą parę od zajmowanej przez nią pozycji w tym ciągu, gdyż niezależnie od tej pozycji wynik konsolidacji jest ten sam.

Czwarte i piąte prawo rachunku tautologicznego zostały połączone w postaci szóstego prawa dodawania poprzedników i następników o następującej treści: *Prawo dodawania poprzedników i następników polega na przekształceniu implikacji posiadających ten sam poprzednik lub ten sam następnik w jedną implikację, której poprzednikiem jest alternatywa poprzedników, a następnikiem jest alternatywa następników tych implikacji*¹⁰.

Przykład:

$$[(p \Rightarrow r) \wedge (q \Rightarrow r) \wedge (p \Rightarrow q) \wedge (p \Rightarrow r)] \Rightarrow [(p \vee q) \Rightarrow (q \vee r)]$$

Szóste prawo rachunku tautologicznego dodawania poprzedników i następników można uogólnić do szóstego prawa rachunku sekwentowego **konsolidacji** wszystkich par elementów posiadających ten sam poprzednik lub ten sam następnik w ten sposób, że:

⁹ Waldemar Wietrzykowski, *Wystarczalność rachunku sekwentowego. Rachunek tautologiczny*, DIL 2018, str. 6

¹⁰ Waldemar Wietrzykowski, *Wystarczalność rachunku sekwentowego. Rachunek tautologiczny*, DIL 2018, str. 6

Szóste prawo rachunku sekwentowego - dodawania poprzedników i następników

Jeżeli w ciągu par elementów występują pary o tym samym poprzedniku lub tym samym następniku to ulegają one konsolidacji w jedną parę, w której poprzednikiem jest zbiór nie powtarzających się poprzedników tych par a następnikiem zbiór nie powtarzających się następników tych par.

Piąte prawo można również określić następująco:

Jeżeli pewien predykat wskazuje na kilka produkcji, a na jedną z nich wskazuje kilka innych predykatów, to te wszystkie predykaty i produkcje leżą w tym samym stanie.

Przykład:

$$\boxed{p \ r, q \ r, p \ q, p \ r \Rightarrow (p, q) (q, r)}$$

Uwieńczeniem praw dotyczących konsolidacji implikacji uwzględniającym również implikacje złożone w postaci struktur jest dziewiąte prawo rachunku tautologicznego - dodawania konstrukcji. Ponieważ konstrukcja jest uogólnioną implikacją, prawo to jest uogólnieniem praw dodawania poprzedników i następników uwzględniającym konstrukcje. Treść tego prawa jest następująca: *Prawo dodawania konstrukcji polega na tym, że jeżeli w ciągu konstrukcji istnieją konstrukcje, których zbiory alternatyw poprzedników lub zbiory alternatyw następników są nierozłączne, to konstrukcje te można połączyć w jedną konstrukcję, której poprzednikiem jest suma zbiorów alternatyw poprzedników a następnikiem jest suma zbiorów alternatyw następników tych konstrukcji*¹¹.

Przykład:

$$\{[(t \vee z) \Rightarrow (q \vee r)] \wedge [(p \vee q) \Rightarrow (q \vee s)]\} \Rightarrow [(t \vee z \vee p \vee q) \Rightarrow (q \vee r \vee s)]$$

Dziewiąte prawo rachunku tautologicznego dodawania konstrukcji można uogólnić do dziewiątego prawa rachunku sekwentowego **konsolidacji** wszystkich par elementów posiadających ten sam poprzednik lub ten sam następnik w ten sposób, że:

Dziewiąte prawo rachunku sekwentowego - dodawania konstrukcji

Jeżeli w ciągu par elementów istnieją pary, których zbiory poprzedników lub zbiory następników są nierozłączne, to można je skonsolidować w jedną parę, w której poprzednikiem jest suma zbiorów poprzedników, a następnikiem jest suma zbiorów następników tych par.

Dziewiąte prawo można również określić następująco:

Zbiory predykatów wskazujące na łączne zbiory produkcji leżą w jednym stanie, jak również łączne zbiory predykatów wskazujące na zbiory produkcji leżą w jednym stanie.

¹¹ Waldemar Wietrzykowski, *Wystarczalność rachunku sekwentowego. Rachunek tautologiczny*, DIL 2018, str. 8

Przykład:

$$(t, z) (q, r), (p, q) (q, s) \Rightarrow (t, z, p, q) (q, r, s)$$

Uniwersalne prawo konsolidacji implikacji i konstrukcji

Jeżeli implikację potraktujemy jako pewną odmianę konstrukcji z jednoelementowym zbiorem poprzedników i jednoelementowym zbiorem następników, to takich implikacji będzie miało również zastosowanie dziewiąte prawo rachunku tautologicznego - dodawania konstrukcji. Będzie się to również przenosiło na rachunek sekwentowy. Oznacza to, że dziewiąte prawo rachunku tautologicznego i sekwentowego będzie łączyło wszystkie prawa dotyczące konsolidacji implikacji i konstrukcji.

Definicja samouczącej się maszyny Turinga

W rachunku tautologicznym i sekwentowym jeden stan samouczącej się maszyny Turinga reprezentowany jest w postaci konstrukcji. Taka definicja pozwoli nam na sprecyzowanie znaczenia zbioru poprzedników i zbioru następników tej konstrukcji.

W celu przeprowadzenia dalszych rozważań weźmy następującą przemianę:

$$p, q, r, s \Rightarrow p q, q r, r s$$

Z prawa tej przemiany wynika, że pierwszy i ostatni element ciągu może wystąpić w ciągu co najmniej raz (pierwszy jest zawsze predykatem, a ostatni zawsze produkcją), pozostałe elementy mogą wystąpić co najmniej dwa razy.

Z pracy „Teoria świadomej maszyny” predykatem nazywano produkcję, której zadaniem jest *“wskazywanie (identyfikacja) stanu poprzez swoją obecność w tym stanie”*¹² oraz *“predykaty umożliwiają tworzenie procesów w ten sposób, że ostatnio uaktywniona produkcja wyznacza predykat o tej samej wartości co ta produkcja i umieszczony jest w tym stanie, w którym oczekiwane jest następne uaktywnienie produkcji”*¹³.

Neuronalna analogia

Produkcje reprezentują obszary swoiste, natomiast predykaty obszary nieswoiste kory mózgowej. Podobnie jak w korze mózgowej do swoistego pobudzenia punktu kory mózgowej potrzebne jest uprzednie nieswoiste jej pobudzenie, tak samo do pobudzenia produkcji nie wystarcza treść swoistego pobudzenia, lecz wskazanie tej produkcji poprzez nieswoiste pobudzenie predykatu wskazującego tą produkcję.

Pobudzenie swoiste produkcji wynika bezpośrednio z zewnętrznego pobudzenia, natomiast pobudzenie nieswoiste wynika ze stanów wewnętrznych wynikających zarówno pośrednio z pobudzenia swoistego, jak i z woli wewnętrznej jednostki..

¹² Waldemar Wietrzykowski, *Teoria świadomej maszyny*, DIL 2018

¹³ Waldemar Wietrzykowski, *Teoria świadomej maszyny*, DIL 2018

Implementacja komputerowa 1 i 9 prawa sekwentowego

Pierwsza implementacja dotyczy pierwszego i dziewiątego prawa rachunku sekwentowego. Na wstępie zostaje zrealizowane pierwsze prawo sekwentowe w wyniku czego z ciągu produkcji zostaje utworzony ciąg struktur jednostkowych (z jednym poprzednikiem i jednym następnikiem). Pierwsze prawo sekwentowe określa sposób postępowania: *do każdego elementu ciągu, oprócz pierwszego i ostatniego, dołączamy ten sam element i uzyskujemy ciąg kolejnych par elementów*. Para elementów jest strukturą jednostkową. Oznacza to, że

następnik struktury ciągu będzie powtórzony w poprzedniku kolejnej struktury

Pierwszą czynnością będzie zdefiniowanie produkcji, niech nią będzie liczba całkowita, oraz struktury (implikacji), którą będzie para wskaźników do ciągów produkcji, przedstawiających poprzednik i następnik tej struktury. Ciągiem produkcji będzie też wejściowy ciąg, który będzie przetwarzany na ciąg struktur.

Po zdefiniowaniu produkcji i struktury kolejną czynnością będzie implementacja pierwszego prawa sekwentowego określonego następująco: *do każdego elementu ciągu, oprócz pierwszego i ostatniego, dołączamy ten sam element i uzyskujemy ciąg kolejnych par elementów*. Jest to proste do zrealizowania. Kolejną czynnością będzie implementacja dziewiątego prawa sekwentowego określonego następująco: *Jeżeli w ciągu par elementów istnieją pary, których zbiory poprzedników lub zbiorów następników są nierozłączne, to można je skonsolidować w jedną parę, w której poprzednikiem jest suma zbiorów poprzedników, a następnikiem jest suma zbiorów następników tych par*. Oznacza to, że ciąg struktur (implikacji) należy porównywać niepowtarzalnymi parami, co oznacza, że mamy sprawdzić zbiór kombinacji bez powtórzeń pary implikacji ze zbioru struktur w ciągu. Ilość takich par jest równa $n!/2!(n-2)!$. Implementację takiej kombinacji robimy w ten sposób, że najpierw pierwszy element zbioru porównujemy z następnymi elementami, potem drugi element porównujemy z następnymi elementami, itd., aż przedostatni element porównujemy z ostatnim elementem tego ciągu. Wynikiem porównania pary jest sprawdzenie, czy para implikacji jest łączna (nierozłączna) po stronie poprzednika lub następnika. Jeżeli para implikacji jest łączna sumujemy elementy implikacji i sumę umieszczamy w drugiej implikacji. Po zakończeniu porównywania pierwszej implikacji z kolejnymi, pierwszą implikację kasujemy.

Implementację łączności pary implikacji ze względu na poprzedniki lub następniki robimy w ten sposób, że każdy element ciągu np. poprzedników pierwszej implikacji porównujemy ze wszystkimi elementami ciągu poprzedników drugiej implikacji. Taki sam element dla pierwszej i drugiej implikacji oznacza że implikacje są łączne. Różne elementy pierwszej implikacji zapamiętywane są w implikacji tymczasowej. Po zakończeniu porównywania elementów, do implikacji tymczasowej dodawane są elementy drugiej implikacji, a następnie zawartość implikacji tymczasowej zapamiętywana jest w drugiej implikacji. Pierwsza implikacja jest kasowana. Jeżeli implikacje pary są rozłączne zawartości pierwszej i drugiej implikacji pozostają bez zmian.

Testowanie implementacji 1 i 9 prawa sekwencyjnego

Implementację pierwszego i dziewiątego prawa sekwentowego zrealizowano w aplikacji "consolid" (wersja 7) w języku C++..

Działanie aplikacji sprawdzono na przykładzie 1 z pracy “Teoria świadomej maszyny”¹⁴ z tematu “Reguły scalania predykatów i produkcji”, gdzie z ciągu liczb:

3,5,0,6,4,2,1,0,3,6,4,7,1,0,6,2,1,0,6,1,0,6,7,2,1,5,3,3,6,1,0 uzyskano konsolidację: (4,7,2,6) (7,2,4,1), (0,3,5,1) (3,6,5,0).

W przykładzie 2, z ciągu liczb: 3,5,0,6,4,2,1,0,3,6,4,7,1,0,6,2,1,0,6,1,0 uzyskano konsolidację (3,0) (5,3,6), (4,7,2,6) (7,2,4,1), (5,1) 0.

W przykładzie 3, strukturę: (3,0) (5,3,6), (4,7,2,6) (7,2,4,1), (5,1) 0 połączono z ciągiem liczb: 6,7,2,1,5,3,3,6,1,0 i uzyskano konsolidację (4,7,2,6) (7,2,4,1), (3,0,5,1) (3,6,5,0)

W przykładzie 4, strukturę: (5,3,6), (4,7,2,6) (7,2,4,1), (5,1) 0 połączono z ciągiem 0 5 6 3 i otrzymano konsolidację: (1,5,4,7,2,3,0,6) (0,7,2,4,1,5,6,3)

W przykładzie 5, ciąg liczb 3,5,0,6,4,2,1,0,3,6,4,7,1,0,6,2,1,0,6,1,0,6,7,2,1,5,3,3,6,1,0 zakłócono ciągiem liczb 11,13,8,14,12,10,9,8,11,14,12,15,9,8,14,10,9,8,14,9,8. Otrzymano konsolidację: (0,3,5,1) (3,6,5,0), (4,7,2,6) (7,2,4,1) oraz dodatkowe struktury zakłócające: (11,8) (13,11,14), (12,15,10,14) (15,10,12,9), (13,9) 8.

Podsumowując wyniki obliczeń, były zgodne z wynikami zamieszczonymi w przykładach, z wyjątkiem przykładu 4, gdzie program dokonał dodatkowej konsolidacji wyników przedstawionych w przykładzie.

Testowanie implementacji w operacjach binarnych

Na podstawie moich notatek z 2015 roku dotyczących rozważań dotyczących możliwości opracowywania maszyny samouczącej się, można przetestować program konsolid na zamieszczonych tam przykładach, dotyczących zadań i wyników obliczeń jakie musiałaby realizować, hipotetyczna wtedy maszyna..

Rozpocznijmy od ostatniego przykładu z 2016 roku dotyczącego konsolidacji ciągu liczb.

Maszynie podano ciąg złożony z czterech ciągów o postaci:

6,5,0,3,1,2,4,0
6,3,1,7,4,0
3,2,4,0,3,4,0,0,0
3,7,2,4,5,6,6,3,4,0,0

i otrzymano konsolidację:

(1,7,2,3) (7,2,1,4)
(6,4,5,0) (5,6,3,0)

natomiast zadanie rozwiązano na konsolidację:

(011, 101, 000, 001) (000, 101, 110, 011)

¹⁴ Waldemar Wietrzykowski, *Teoria świadomej maszyny*, DIL 2018

(110, 100, 010, 111) (001, 010, 100, 111)

czyli

(6,5,0,4) (0,5,3,6)

(3,1,2,7) (4,2,1,7)

przy liczbie binarnej czytanej od lewej do prawej.

Kolejność elementów w poprzedniku i następniku nie ma znaczenia, jak również kolejność wystąpienia implikacji złożonych. Oznacza to, że wyniki są te same. **Zaznaczyć należy, że o wyborze implikacji złożonej decydują jego elementy poprzednika..**

W kolejnym przykładzie maszynie podano ciąg liczb: 3,2,4,0,3,4,0,0,0 i otrzymano konsolidację:

(2,3) (2,4)

(4,0) (3,0)

gdzie 4 i 3 są krawędziami. Oznacza to bardzo wysoką skuteczność i poprawność zastosowanego algorytmu..

Dla ciągu: otrzymujemy:

Dla ciągu: otrzymujemy:

Dla ciągu: otrzymujemy:

Pozostałe wyniki w tabeli:

Ciąg wejściowy	Konsolidacja
6,0,0,3,1,2,4,6,6,3,4,6,5,0	1 2,(2,3) (1,4),(4,0,6,5) (5,6,3,0)
6,0,0,3,1,2,4,6,6,3,4,6,5,0, 6,3,1,7,4	1 (2,7), (3,2,7) (1,4), (5,0,6,4) (5,6,3,0)
3,7,2,4,5,6,6,3,4,0,0	7 2,(5,6) (6,3),(2,3) (7,4),(4,0) (5,0)
3,4,6,6,5,0,6,6,0,0	3 4,(6,4,5,0) (5,6,0)
0,3,2,2,1,4,6,6,0,0	(3,2) (2,1),1 4,(4,6,0) (6,3,0)
6,5,0,3,1,2,4,0	6 5,0 3,3 1,1 2,2 4,(5,4) 0
3,2,4,0,3,4,0,0,0	(2,3) (2,4),(4,0) (3,0)
5,6,0,5,5,3,2,7,4,5,6,0	3 2,2 7,7 4,(4,0,5) (3,5,6),6 0
6,0,3,1,7,7,7,7,2,7,4,0	0 3,3 1,(2,1,7) (2,7,4),(6,4) 0
6,5,3,1,4,5,6,3,4,0	(1,3) (1,4),(5,6,4) (6,3,5,0)
6,5,0,6,3,2,1,7,2,7,7,4,0	0 6,6 (5,3),(2,1,3,7) (1,7,2,4),(5,4) 0
6,3,7,1,1,1,2,4,6,0,3,2,4,0,3,4,0	(7,1,2,3) (1,2,7,4),(0,6,4) (3,6,0)
6,3,7,1,2,4,0,3,2,4,3,2,4,0,3,4,0	7 1,(1,2,3) (2,7,4),(0,6,4) (3,0)
5,6,0,3,7,2,1,4,5,3,2,4,5,0	(7,3) (7,2),(1,2) (1,4),4 5,(0,6,5) (3,6,0)
6,5,5,3,2,7,1,2,7,2,4,0	(6,5) (5,3),(1,3,7) (1,2),2 (7,4),4 0
6,5,5,3,6,3,5,6,3,6	(5,6,3) (3,5,6)

Różnica między programami “mw” a “consolid”.

Pomimo tego, że aplikacje *mw* i *consolid* zmagają się do osiągnięcia jednego celu, a jest nim automatyczna budowa programu maszyny Turinga, to jednak sposób dojścia do tego dla poszczególnych aplikacji jest zupełnie inna. Algorytm programu *mw* opiera się na algorytmie funkcjonowania maszyny Turinga, natomiast punktem wyjścia algorytmu programu *consolid* jest definicja maszyny jednofunkcyjnej (która jest zgodna z maszyną Turinga) “*Dla pewnego ciągu produkcji maszyna jednofunkcyjna jest zbiorem tych produkcji, jakie wystąpiły lub mogą wystąpić w następnym kroku po określonej produkcji*”.

Warto porównać też struktury, na jakich opierają się algorytm *mw* i *consolid*. Dla algorytmu *mw* jest to:

```
typedef struct rule { // struktura reguły rozmiaru 1 bajta
    byte free:1;      // flaga obecności reguły
    byte bit_x1:1;    // bit ścieżki x1
    byte bit_x2:1;    // bit ścieżki x2; (bit_x1, bit_x2) - offset lub obiekt
    byte bit_y1:1;    // bit ścieżki y1; określa przynależność obiektu do segmentu.
    byte seg_next:4;  // następny wskazywany przez regułę segment ze zbiorem reguł
} trule, *tprule; // typ struktury i jego wskaźnika (do określ. położenia reguły lub segmentu)
```

Natomiast dla algorytmu *consolid* jest to:

```
typedef struct implikacja { // definicja implikacji
    bool k;                // true: do zwolnienia pamięci
    int cprior;             // liczba poprzedników
    int cnext;              // liczba następników
    tprod *prior;           // lista poprzedników
    tprod *next;            // lista następników
} timplik;
```

Jak z tego wynika algorytm programu *mw* jest dostosowany do określonej produkcji, i operuje na elementach tej produkcji, natomiast algorytm *consolid* tworzy kontener, w którym umieszczona jest dowolna produkcja, a operacją wykonywaną na produkcji jest jedynie tożsamość jej z inną produkcją, a więc ma wymiar bardziej uniwersalny i inteligentny. Jest też niezależny od maszyny na jakiej jest wykonywany oraz może być analizowany w oparciu jedynie na umiejętności porównywania i uogólniania umysłu człowieka, bez znajomości języka programowania i wiedzy o działaniu komputera.

Prawo zamiany implikacji w koniunkcje

Drugie prawo tautologiczne dotyczy zamiany implikacji w koniunkcję.¹⁵ Może być stosowane tylko wtedy, kiedy poprzednik każdej implikacji jest równy następnikowi implikacji poprzedzającej. Zamiana polega na zastąpieniu każdego podwójnego wystąpienia zmiennej w postaci $q) \wedge (q \text{ jedną zmienną } q$ a następnie zastąpieniu znaków implikacji $\Rightarrow$ znakami koniunkcji $\wedge$.

¹⁵ Waldemar Wietrzykowski, *Wystarczalność rachunku sekwentowego. Rachunek tautologiczny*, DIL 2018, str. 5

Przykład:

$$[(p \Rightarrow q) \wedge (q \Rightarrow r) \wedge (r \Rightarrow s)] \leq (p \wedge q \wedge r \wedge s)$$

Należy zauważyć, że z jednej implikacji da się wyprowadzić dwa elementy połączone koniunkcją:

Przykład:

$$(p \Rightarrow q) \leq (p \wedge q)$$

Drugie prawo rachunku tautologicznego zamiany implikacji w koniunkcję można uogólnić do drugiego prawa rachunku sekwentowego zamiany ciągu par w ciąg elementów w ten sposób, że:

Drugie prawo rachunku sekwentowego – zamiany par w elementy

w ciągu par, w którym następnik danej pary jest równy poprzednikowi następnej pary w postaci ..q, q.. zamieniany jest na jeden element q.

Przykład:

$$p, q, r, r, s \leq p, q, r, s$$

Należy zauważyć, że z jednej pary da się wyprowadzić dwa elementy:

Przykład:

$$p, q \leq p, q$$

Prawa rozłączania implikacji

Oprócz praw łączenia implikacji występują jeszcze prawa ich rozłączania, a także prawo zamiany implikacji w koniunkcję.

Prawa rozłączania złożonych implikacji w proste implikacje są etapem przygotowawczym do zastosowania prawa zamiany implikacji w koniunkcję, ponieważ niepotrzebne jest rozdzielanie w jakimś innym celu.

Mechanizm rozłączania implikacji opisuje dziesiąte prawo rachunku tautologicznego – prawo rozłączania poprzedników. Treść tego prawa jest następująca: *Prawo rozłączania poprzedników polega na uzyskaniu ciągu implikacji, którego elementy posiadają taki sam następnik co implikacja rozłączana a poprzednik jest jedną z alternatyw poprzednika tej implikacji.*¹⁶

¹⁶ Waldemar Wietrzykowski, *Wystarczalność rachunku sekwentowego. Rachunek tautologiczny*, DIL 2018, str. 9

Przykład:

1. $[(p \vee q) \Rightarrow r] \Leftarrow [(p \Rightarrow r) \wedge (q \Rightarrow r)]$
2. $[(p \vee q) \Rightarrow (q \vee r)] \Leftarrow \{[(p \Rightarrow (q \vee r))] \wedge [(q \Rightarrow (q \vee r))]\}$

Dziesiąte prawo rachunku tautologicznego rozłączania poprzedników można uogólnić do dziesiątego prawa rachunku sekwentowego rozłączania na pary elementów posiadających ten następnik w ten sposób, że:

Dziesiąte prawo rachunku sekwentowego – rozłączania poprzedników

Parę można rozłączyć na ciąg par o tym samym następniku co rozłączana para i poprzedniku ze zbioru poprzedników tej pary.

Przykład:

$$(p, q) r \Leftarrow p r, q r$$

$$(p, q) (q, r) \Leftarrow p (q, r), q (q, r)$$

Oprócz rozłączania poprzedników możemy też rozłączać następniki. Mechanizm takiego rozłączania implikacji opisuje jedenaste prawo rachunku tautologicznego – prawo rozłączania następników. Treść tego prawa jest następująca: *Prawo rozłączania następników polega na uzyskaniu ciągu implikacji, którego elementy posiadają taki sam poprzednik co implikacja rozłączana a następnik jest jedną z alternatyw następnika tej implikacji.*¹⁷

Jedenaste prawo rachunku tautologicznego rozłączania następników można uogólnić do jedenastego prawa rachunku sekwentowego rozłączania na pary elementów posiadających ten poprzednik w ten sposób, że:

Jedenaste prawo rachunku sekwentowego – rozłączania następników

Parę można rozłączyć na ciąg par o tym samym poprzedniku co rozłączana para i następniku ze zbioru następników tej pary.

Przykład:

$$p (q, r) \Leftarrow p q, p r$$

$$(p, q) (q, r) \Leftarrow (p, q) q, (p, q) r$$

Prawa dziesiąte i jedenaste można połączyć w dwunaste prawo rozłączania poprzedników i następników. Mechanizm takiego rozłączania implikacji opisuje dwunaste prawo rachunku tautologicznego – prawo rozłączania poprzedników i następników. Treść tego prawa jest następująca: *Prawo*

¹⁷ Waldemar Wietrzykowski, *Wystarczalność rachunku sekwentowego. Rachunek tautologiczny*, DIL 2018, str. 9

rozłączania poprzedników i następników polega na stosowaniu prawa rozłączania poprzedników a następnie prawa rozłączania następników lub odwrotnie..¹⁸

Przykład:

$[(p \vee q) \Rightarrow (q \vee r)]$	(tautologia)
$\Leftarrow \{[(p \Rightarrow (q \vee r)) \wedge [(q \Rightarrow (q \vee r))]\}$	rozdzielenie poprzedników
$\Leftarrow [(p \Rightarrow q) \wedge (p \Rightarrow r) \wedge (q \Rightarrow q) \wedge (q \Rightarrow r)]$	rozdzielenie następników

Jedenaste prawo rachunku tautologicznego rozłączania następników można uogólnić do jedenaściego prawa rachunku sekwentowego rozłączania na pary elementów posiadających ten poprzednik w ten sposób, że:

Dwunaste prawo rachunku sekwentowego – rozłączanie poprzedników i następników

Parę można rozłączyć stosując prawo rozłączania poprzedników a następnie prawo rozłączania następników lub odwrotnie.

Przykład:

$$(p, q) (q, r) \Leftarrow p (q, r), q (q, r) \Leftarrow p q, p r, q q, q r$$

Implementacja komputerowa 2 i 12 prawa sekwentowego

Na początku zostaje zrealizowane 12 prawo sekwentowe a następnie 2 prawo sekwentowe.

Implementacja komputerowa 1, 9 i 12 prawa sekwentowego

Aplikacja implementacyjna nosi nazwę “rachsekw” od rachunku sekwentowego. Na podstawie tej aplikacji przeprowadzono badanie, polegające na tym, że z operacji dodawania dwóch liczb całkowitych, stosując 1 prawo sekwentowe wygenerowano ciągu produkcji, który następnie przemieniono na ciąg implikacji jednostkowych. Następnie stosując 9 prawo sekwentowe ciąg implikacji został skonsolidowany są struktur. Z kolei stosując 12 prawo sekwentowe otrzymane struktury zostały rozłączone na ciąg implikacji jednostkowych, który w oparciu o 9 prawo sekwentowe został ponownie skonsolidowany. Celem badania było porównanie produktów skonsolidowania po pierwszej konsolidacji i po drugiej konsolidacji oraz ciągów implikacji jednostkowych przed pierwszą konsolidacją i po rozłączeniu uzyskanych konstrukcji.. Wynik badania przedstawiono poniżej:

$$3566 + 4972 = 8538$$

$$\Rightarrow 0, 5, 3, 7, 4, 3, 7, 1, 7, 2, 1, 1, 2, 4, 0$$

$$\Rightarrow 0, 5, 5, 3, 3, 7, 7, 4, 4, 3, 3, 7, 7, 1, 1, 7, 7, 2, 2, 1, 1, 1, 1, 2, 2, 4, 4, 0$$

$$\Rightarrow 0, 5, (3, 1, 7, 2) (7, 2, 1, 4), (5, 4) (3, 0)$$

$$\Leftarrow 0, 5, 3, 7, 3, 2, 3, 1, 3, 4, 1, 7, 1, 2, 1, 1, 1, 4, 7, 7, 7, 2, 7, 1, 7, 4, 2, 7, 2, 2, 2, 1, 2, 4, 5, 3, 5, 0, 4, 3, 4, 0$$

¹⁸ Waldemar Wietrzykowski, *Wystarczalność rachunku sekwentowego. Rachunek tautologiczny*, DIL 2018, str. 9

$\Rightarrow 0\ 5,(7,1,3,2)\ (1,2,7,4),(5,4)\ (3,0)$

$4632 + 965 = 5597$

$\Rightarrow 6, 0, 6, 5, 5, 0, 6, 6, 6, 3, 4, 0, 5, 0$

$\Rightarrow 6\ 0,0\ 6,6\ 5,5\ 5,5\ 0,0\ 6,6\ 6,6\ 6,6\ 3,3\ 4,4\ 0,0\ 5,5\ 0$

$\Rightarrow 3\ 4,(4,0,6,5)\ (3,6,5,0)$

$\leq 3\ 4,4\ 3,4\ 6,4\ 5,4\ 0,0\ 3,0\ 6,0\ 5,0\ 0,6\ 3,6\ 6,6\ 5,6\ 0,5\ 3,5\ 6,5\ 5,5\ 0$

$\leq 3\ 4,(6,0,4,5)\ (5,6,3,0)$

$5828 + 2722 = 8550$

$\Rightarrow 0, 6, 5, 0, 0, 6, 5, 3, 4, 3, 1, 2, 1, 4, 0$

$\Rightarrow 0\ 6,6\ 5,5\ 0,0\ 0,0\ 6,6\ 5,5\ 3,3\ 4,4\ 3,3\ 1,1\ 2,2\ 1,1\ 4,4\ 0$

$\Rightarrow 6\ 5,(2,3,1)\ (2,1,4),(0,5,4)\ (3,6,0)$

$\leq 6\ 5,2\ 2,2\ 1,2\ 4,3\ 2,3\ 1,3\ 4,1\ 2,1\ 1,1\ 4,0\ 3,0\ 6,0\ 0,5\ 3,5\ 6,5\ 0,4\ 3,4\ 6,4\ 0$

$\Rightarrow 6\ 5,(3,2,1)\ (1,2,4),(5,0,4)\ (6,3,0)$

$6732 + 7644 = 14376$

$\Rightarrow 0, 0, 3, 7, 2, 4, 3, 2, 2, 1, 2, 7, 7, 4, 0$

$\Rightarrow 0\ 0,0\ 3,3\ 7,7\ 2,2\ 4,4\ 3,3\ 2,2\ 2,2\ 1,1\ 2,2\ 7,7\ 7,7\ 4,4\ 0$

$\Rightarrow (1,3,2,7)\ (1,7,2,4),(0,4)\ (3,0)$

$\leq 1\ 1,1\ 7,1\ 2,1\ 4,3\ 1,3\ 7,3\ 2,3\ 4,2\ 1,2\ 7,2\ 2,2\ 4,7\ 1,7\ 7,7\ 2,7\ 4,0\ 3,0\ 0,4\ 3,4\ 0$

$\Rightarrow (2,3,1,7)\ (2,7,1,4),(0,4)\ (3,0)$

$(6453 + 4449 = 10902), (1055 + 1237 = 2292), (4084 + 1686 = 5770)$

$\Rightarrow (3, 4, 5, 0, 5, 3, 2, 4, 3, 4, 0, 5, 3, 4, 0), (3, 1, 7, 1, 7, 4, 6, 6, 0, 0, 3, 4, 0), (0, 6, 3, 4, 3, 1, 1, 7, 1, 7, 7, 1, 4, 0)$

$\Rightarrow (3\ 4,4\ 5,5\ 0,0\ 5,5\ 3,3\ 2,2\ 4,4\ 3,3\ 4,4\ 0,0\ 5,5\ 3,3\ 4,4\ 0), (3\ 1,1\ 7,7\ 1,1\ 7,7\ 4,4\ 6,6\ 6,6\ 0,0\ 0,0\ 3,3\ 4,4\ 0), (0\ 6,6\ 3,3\ 4,4\ 3,3\ 1,1\ 1,1\ 7,7\ 1,1\ 7,7\ 7,7\ 1,1\ 4,4\ 0)$

$\Rightarrow (7,2,3,1)\ (7,1,2,4),(6,0,5,4)\ (6,3,5,0)$

$\leq 7\ 7,7\ 1,7\ 2,7\ 4,2\ 7,2\ 1,2\ 2,2\ 4,3\ 7,3\ 1,3\ 2,3\ 4,1\ 7,1\ 1,1\ 2,1\ 4,6\ 6,6\ 3,6\ 5,6\ 0,0\ 6,0\ 3,0\ 5,0\ 0,5\ 6,5\ 3,5\ 5,5\ 0,4\ 6,4\ 3,4\ 5,4\ 0$

$\Rightarrow (3,2,7,1)\ (2,1,7,4),(5,0,6,4)\ (5,3,6,0)$

$(8353 + 1334 = 9687), (1046 + 9885 = 10931), (1122 + 2783 = 3905)$

$\Rightarrow (5, 6, 6, 0, 6, 3, 4, 5, 6, 0, 6, 0, 0, 5, 0), (6, 5, 3, 2, 7, 4, 0, 6, 0, 6, 3, 4, 0, 6, 0), (6, 3, 2, 2, 2, 1, 7, 2, 4, 6, 5, 6, 0)$

$\Rightarrow (5\ 6,6\ 6,6\ 0,0\ 6,6\ 3,3\ 4,4\ 5,5\ 6,6\ 0,0\ 6,6\ 0,0\ 0,0\ 5,5\ 0), (6\ 5,5\ 3,3\ 2,2\ 7,7\ 4,4\ 0,0\ 6,6\ 0,0\ 6,6\ 3,3\ 4,4\ 0,0\ 6,6\ 0), (6\ 3,3\ 2,2\ 2,2\ 2,2\ 1,1\ 7,7\ 2,2\ 4,4\ 6,6\ 5,5\ 6,6\ 0)$

$\Rightarrow (1,7,3,2)\ (1,7,2,4),(4,0,5,6)\ (5,3,6,0)$

$\leq 1\ 1,1\ 7,1\ 2,1\ 4,7\ 1,7\ 7,7\ 2,7\ 4,3\ 1,3\ 7,3\ 2,3\ 4,2\ 1,2\ 7,2\ 2,2\ 4,4\ 5,4\ 3,4\ 6,4\ 0,0\ 5,0\ 3,0\ 6,0\ 0,5\ 5,5\ 3,5\ 6,5\ 0,6\ 5,6\ 3,6\ 6,6\ 0$

$\Rightarrow (3,7,1,2)\ (2,7,1,4),(5,0,4,6)\ (6,3,5,0)$

$(603 + 4813 = 5416)$, $(7986 + 1494 = 9480)$, $(1443 + 858 = 2301)$
 $\Rightarrow (3, 1, 2, 7, 1, 4, 3, 2, 4, 3, 4, 0, 6, 0), (0, 3, 2, 4, 3, 1, 2, 2, 7, 1, 7, 1, 1, 4, 0), (5, 3, 4, 6, 6, 5, 6, 5, 3, 2, 1, 4, 0)$
 $\Rightarrow (3, 1, 1, 2, 2, 7, 7, 1, 1, 4, 4, 3, 3, 2, 2, 4, 4, 3, 3, 4, 4, 0, 0, 6, 6, 0), (0, 3, 3, 2, 2, 4, 4, 3, 3, 1, 1, 2, 2, 2, 2, 7, 7, 1, 1, 7, 7, 1, 1, 1, 4, 4, 0), (5, 3, 3, 4, 4, 6, 6, 6, 6, 5, 5, 6, 6, 5, 5, 3, 3, 2, 2, 1, 1, 4, 4, 0)$
 $\Rightarrow (2, 7, 3, 1), (7, 1, 2, 4), (5, 0, 6, 4), (5, 6, 3, 0)$
 $\leq (2, 7, 2, 1, 2, 2, 2, 4, 7, 7, 7, 1, 7, 2, 7, 4, 3, 7, 3, 1, 3, 2, 3, 4, 1, 7, 1, 1, 1, 2, 1, 4, 5, 5, 5, 6, 5, 3, 5, 0, 0, 5, 0, 6, 0, 3, 0, 0, 6, 5, 6, 6, 6, 3, 6, 0, 4, 5, 4, 6, 4, 3, 4, 0)$
 $\Rightarrow (3, 7, 2, 1), (2, 1, 7, 4), (6, 0, 5, 4), (3, 6, 5, 0)$

$(2506 + 2945 = 5451)$, $(2574 + 1437 = 4011)$, $(8795 + 5882 = 14677)$, $(9269 + 9510 = 18779)$, $(4324 + 1605 = 5929)$, $(8848 + 6359 = 15207)$, $(4622 + 8140 = 12762)$, $(8659 + 1167 = 9826)$, $(7775 + 1898 = 9673)$, $(4892 + 7960 = 12852)$
 $\Rightarrow (6, 5, 0, 5, 0, 0, 5, 3, 7, 2, 4, 3, 4, 0), (6, 5, 3, 7, 2, 4, 0, 6, 6, 5, 6, 5, 0), (5, 3, 4, 3, 7, 2, 7, 2, 4, 3, 2, 4, 6, 5, 0), (5, 6, 3, 4, 5, 3, 4, 0, 6, 0, 3, 4, 0, 3, 4, 0), (6, 0, 3, 4, 0, 5, 3, 1, 4, 6, 6, 0, 5, 0), (6, 6, 6, 0, 3, 4, 6, 3, 4, 5, 0, 6, 6, 5, 0), (0, 5, 3, 7, 4, 0, 6, 6, 6, 3, 2, 2, 7, 4, 0), (3, 7, 2, 2, 1, 4, 5, 3, 1, 4, 6, 0, 0, 5, 0), (5, 3, 1, 7, 1, 2, 7, 4, 6, 3, 7, 1, 1, 4, 0), (0, 0, 5, 3, 7, 4, 0, 0, 3, 7, 2, 2, 7, 4, 0)$
 $\Rightarrow (6, 5, 5, 0, 0, 5, 5, 0, 0, 0, 5, 5, 3, 3, 7, 7, 2, 2, 4, 4, 3, 3, 4, 4, 0), (6, 5, 5, 3, 3, 7, 7, 2, 2, 4, 4, 0, 0, 6, 6, 6, 5, 5, 6, 6, 5, 5, 0), (5, 3, 3, 4, 4, 3, 3, 7, 7, 2, 2, 7, 7, 2, 2, 4, 4, 3, 3, 2, 2, 4, 4, 6, 6, 5, 5, 0), (5, 6, 6, 3, 3, 4, 4, 5, 5, 3, 3, 4, 4, 0, 0, 6, 6, 0, 0, 3, 3, 4, 4, 0, 0, 3, 3, 4, 4, 0), (6, 0, 0, 3, 3, 4, 4, 0, 0, 5, 5, 3, 3, 1, 1, 4, 4, 6, 6, 6, 0, 0, 5, 5, 0), (6, 6, 6, 6, 6, 0, 0, 3, 3, 4, 4, 6, 6, 3, 3, 4, 4, 5, 5, 0, 0, 6, 6, 6, 5, 5, 0), (0, 5, 5, 3, 3, 7, 7, 4, 4, 0, 0, 6, 6, 6, 6, 6, 3, 3, 2, 2, 2, 2, 7, 7, 4, 4, 0), (3, 7, 7, 2, 2, 2, 1, 1, 4, 4, 5, 5, 3, 3, 1, 1, 4, 4, 6, 6, 0, 0, 0, 5, 5, 0), (5, 3, 3, 1, 1, 7, 7, 1, 1, 2, 2, 7, 7, 4, 4, 6, 6, 3, 3, 7, 7, 1, 1, 1, 1, 4, 4, 0), (0, 0, 0, 5, 5, 3, 3, 7, 7, 4, 4, 0, 0, 0, 3, 3, 7, 7, 2, 2, 2, 2, 7, 7, 4, 4, 0)$
 $\Rightarrow (1, 3, 2, 7), (1, 7, 2, 4), (6, 0, 5, 4), (6, 3, 5, 0)$
 $\leq (1, 1, 1, 7, 1, 2, 1, 4, 3, 1, 3, 7, 3, 2, 3, 4, 2, 1, 2, 7, 2, 2, 2, 4, 7, 1, 7, 7, 7, 2, 7, 4, 6, 6, 6, 3, 6, 5, 6, 0, 0, 6, 0, 3, 0, 5, 0, 0, 5, 6, 5, 3, 5, 5, 5, 0, 4, 6, 4, 3, 4, 5, 4, 0)$
 $\Rightarrow (2, 3, 1, 7), (2, 7, 1, 4), (5, 0, 6, 4), (5, 3, 6, 0)$

Wnioski

1. Wyniki rozłączania po ponownym skonsolidowaniu dają ten sam wynik niż przed ich rozłączeniem co potwierdza prawidłowość algorytmu rozłączania.
2. Ciąg implikacji uzyskany po skonsolidowaniu i rozłączeniu pierwotnego ciągu implikacji zawiera dodatkowe implikacje nadmiarowe, które nie znajdują się w ciągu pierwotnym. Oznacza to, że konsolidacja powoduje poszerzenie wiedzy o dodatkowe elementy, których nie było w elementach konsolidowanych.

Implementacja komputerowa 2 prawa sekwentowego

Zastosowanie 12 prawa sekwentowego powoduje rozłączenie struktur sekwentowych na ciąg implikacji, na którym ciężko jest zastosować 2 prawo sekwentowe. Można spróbować zastosować 2 prawo sekwentowe bezpośrednio na strukturach, wykorzystując zasadę, że produkcja aktywuje predykat o tej samej wartości. Pierwsze co należy zrobić, to zlokalizować predykat na liście implikacji (każdy predykat jest niepowtarzalny i pełni rolę indeksu implikacji).

Do znalezienia ciągu implikacji potrzebny jest predykat początkowy. Jest on równy pierwszemu elementowi ciągu predykatów, który mamy uzyskać po zastosowaniu 12 i 2 prawa sekwentowego. Po zastosowaniu funkcji $index(predykat)$ znajdujemy implikację, z której możemy odczytać kolejną pro-

dukcję do inicjowania. Inicjowana produkcja wywołuje kolejny predykat o tej samej wartości, itd. Ten sposób generacji produkcji ze struktur prowadzi do generacji, nie jednego ciągu, lecz alternatywnych ciągów produkcji.

Adresowanie predykatami

Adresowanie predykatami zapewnia nam funkcja *index(predykat)*. Znajduje ona index implikacji, na której liście poprzedników znajduje się podany predykat.

Zapamiętanie produkcji

Aby zapamiętać produkcję musimy mieć predykat, pod którym produkcja ma być umieszczona. W ten sposób produkcję można umieszczać do struktury o określonym predykanie. Funkcja zapamiętywania realizuje 1 i 9 prawo sekwentowe (prawo zamiany produkcji w implikacje i prawo łączenia implikacji).

Bibliografia

1. Waldemar Wietrzykowski, *Samoucząca się maszyna*, DIL 2016
2. Waldemar Wietrzykowski, *Jednofunkcyjne maszyny*, DIL 2016
3. Waldemar Wietrzykowski, *Interpretacje maszyn*, DIL 2016
4. Waldemar Wietrzykowski, *Znaczenie interpretacji maszyn*, DIL 2017
5. Waldemar Wietrzykowski, *Wiele maszyn*, DIL 2017
6. Waldemar Wietrzykowski, *Świadomość wielu maszyn*, DIL 2017
7. Waldemar Wietrzykowski, *Obraz rzeczywistości z poziomu maszyny*, DIL 2017
8. Waldemar Wietrzykowski, *Scalanie interpretacji maszyn*, DIL 2017
9. Waldemar Wietrzykowski, *Strumień świadomości*, DIL 2017
10. Waldemar Wietrzykowski, *Samoucząca się maszyna motoryczna*, DIL 2017
11. Waldemar Wietrzykowski, *Świadoma reakcja maszyn*, DIL 2017
12. Waldemar Wietrzykowski, *Świadoma maszyna*, DIL 2017
13. Waldemar Wietrzykowski, *Założenia wstępne do projektu świadomej maszyny*, DIL 2017
14. Waldemar Wietrzykowski, *Teoria świadomej maszyny*, DIL 2018
15. Waldemar Wietrzykowski, *Teoria wiedzy praktycznej*, DIL 2018
16. Waldemar Wietrzykowski, *Teoria wiedzy teoretycznej*, DIL 2018
17. Waldemar Wietrzykowski, *Teoria wiedzy*, DIL 2018
18. Waldemar Wietrzykowski, *Wystarczalność rachunku sekwentowego. Rachunek tautologiczny*, DIL 2018