**Wizualizacja formuł logicznych**

**Prowadzący: dr hab. inż. Radosław Klimek**

Damian Ciura,

Stanisław Mendel,

Wiktor Warmuz

**Spis treści**

1. Wstęp……………………………………..……………………3
2. Metody Embedding: DeepWalk…...………………………...6
3. Podobne Metody Embedding……….……………………….8
4. Prezentacja Danych DeepWalk………...……..…………….9
5. Wyniki i Interpretacja..………….…………………………...13
6. Aplikacja..…………………………………………………….16
7. Bibliografia..………………………………………………….19

**Wstęp**

Celem projektu była wizualizacja formuł logicznych, czyli reprezentacja graficzna struktury logicznych wyrażeń matematycznych lub symbolicznych, umożliwiająca łatwiejsze zrozumienie ich skomplikowanych relacji i interakcji.  
  
Naszymi formułami były pliki typu CNF opisane poniżej, które zawierały formuły logiczne w postaci tzw. Koniunkcyjnej Normalnej Formy (CNF), co umożliwiało efektywne reprezentowanie złożonych wyrażeń logicznych za pomocą koniunkcji i dysjunkcji klauzul.

**CNF** to katalog danych, który zawiera przykłady plików przechowywanych przy użyciu formatu pliku CNF DIMACS. Ten format służy do definiowania wyrażenia boolowskiego, napisane w spójnej formie normalnej, którą można wykorzystać jako przykład problemu satysfakcji.

Problem satysfakcji dotyczy przypadku, w którym N boolean zmienne są używane do utworzenia wyrażenia boolowskiego obejmującego negację (**NIE**), połączenie (**I**) i rozłączenie (**LUB**). Problem w tym w celu ustalenia, czy istnieje przypisanie wartości do wartości logicznej zmienne, które sprawiają, że formuła jest prawdziwa. To coś w rodzaju próby przełącz kilka przełączników, aby znaleźć ustawienie, które włącza żarówkę.

Formuły te należało „przeliczyć” jedną z metod embedding, tj. przekształcić je w reprezentację numeryczną przy użyciu algorytmu embedding, takiego jak DeepWalk. Algorytm ten konwertuje struktury grafów, w tym formuły logiczne w postaci CNF, na wektory numeryczne zwane embeddingami. W efekcie każda formuła logiczna jest reprezentowana w przestrzeni o

mniejszej wymiarowości, co ułatwia analizę i wizualizację złożonych struktur logicznych. Proces ten pozwala na bardziej efektywną pracę z formułami logicznymi w kontekście analizy danych za pomocą technik embeddingu.

Metody embedding są używane do reprezentacji obiektów, danych lub informacji w przestrzeni o mniejszej wymiarowości, zwanej przestrzenią embeddingu, w taki sposób, aby zachować pewne właściwości tych obiektów. Główne cele korzystania z metod embeddingu obejmują:

* Redukcję Wymiarów:
  + Embedding pozwala na reprezentację danych w przestrzeni o mniejszej liczbie wymiarów niż oryginalne dane. To ułatwia analizę, wizualizację i manipulację dużymi i złożonymi zbiorami danych.
* Utrzymanie Relacji:
  + W przypadku metod embeddingu grafu, takich jak DeepWalk, celem jest zachowanie struktury grafu i relacji między wierzchołkami w przestrzeni embeddingu. W rezultacie podobne obiekty w oryginalnej przestrzeni są blisko siebie w przestrzeni embeddingu.
* Efektywne Uczenie:
  + Embedding ułatwia uczenie maszynowe, ponieważ reprezentacja numeryczna obiektów może być bardziej efektywnie przetwarzana przez modele uczenia maszynowego. Umożliwia to lepsze wykorzystanie algorytmów uczenia maszynowego i głębokiego uczenia.
* Wizualizacja:
  + Przestrzeń embeddingu umożliwia wizualizację złożonych struktur danych. Możemy łatwo przedstawiać i analizować relacje między obiektami, co jest trudne do osiągnięcia w oryginalnej, wysokowymiarowej przestrzeni danych.
* Podobieństwo i Klasyfikacja:
  + Embedding pozwala na określenie podobieństwa między obiektami na podstawie odległości w przestrzeni embeddingu. Może to być wykorzystywane do zadań klasyfikacyjnych, gdzie obiekty o podobnej reprezentacji są często przypisywane do tych samych klas.
* Skalowalność:
  + Metody embedding są często skalowalne i mogą być stosowane do dużych zbiorów danych, co jest istotne w przypadku analizy danych na dużą skalę.

W przypadku formuł logicznych, embedding pozwala na przekształcenie złożonych struktur logicznych na bardziej zrozumiałe i efektywne reprezentacje numeryczne, co ułatwia ich analizę i manipulację w kontekście różnych zastosowań.

**Metody Embedding: DeepWalk**

**DeepWalk** to metoda używana głównie do reprezentowania węzłów w grafach za pomocą wektorów, co może być przydatne w różnych zadaniach analizy grafów. Jeśli masz plik CNF (Conjunctive Normal Form), co sugeruje, że mamy do czynienia z problemem związany z logiką boolowską, to zastosowanie DeepWalk może być interesujące.

Poniżej przedstawiamy kilka pomysłów, jak metoda DeepWalk mogłaby być użyta w kontekście pliku CNF:

* Reprezentacja zmiennych:
  + Każda zmienna w formule CNF może być traktowana jako węzeł w grafie. DeepWalk może pomóc w stworzeniu reprezentacji wektorowej dla każdej zmiennej, co może być użyte w dalszych analizach.
* Badanie relacji między zmiennymi:
  + Możemy użyć DeepWalk do odkrywania podobieństw między zmiennymi poprzez analizę ich reprezentacji wektorowej. To może pomóc zidentyfikować zależności lub wzorce w formule CNF.
* Rozpoznawanie wzorców klauzul:
  + Klauzule w formule CNF mogą być traktowane jako połączenia między zmiennymi. DeepWalk może pomóc w analizie tych połączeń, co może prowadzić do odkrywania istotnych wzorców w strukturze formuły.
* Predykcja wartości zmiennych:
  + Jeśli związane są zmiennymi wartości (np. prawda/fałsz), to DeepWalk może pomóc w predykcji tych wartości na podstawie ich reprezentacji wektorowej i relacji między nimi.
* Klastrowanie zmiennych:
  + Możemy użyć DeepWalk do klastrowania zmiennych w formule CNF na podstawie ich podobieństwa, co może pomóc w identyfikacji grup zmiennych o podobnym zachowaniu.

**Podobne Metody Embedding**

* Node2Vec Node2Vec jest rozszerzeniem DeepWalk, pozwalającym na elastyczną kontrolę nad eksploracją i eksploatacją grafu podczas generowania embeddingów.
* LINE (Large-scale Information Network Embedding) LINE koncentruje się na zachowaniu struktury grafu poprzez minimalizację funkcji straty reprezentacji pierwszego i drugiego rzędu.
* Grafowe Konwolucyjne Sieci Neuronowe (GCN) GCN to podejście oparte na warstwach konwolucyjnych, stosowane do analizy struktury grafów, co pozwala na bardziej zaawansowaną ekstrakcję cech.
* Doc2Vec Doc2Vec to technika embeddingu stosowana w analizie dokumentów tekstowych, w której każdy dokument jest reprezentowany jako wektor.
* Graph2Vec Graph2Vec stosuje podejście podobne do Doc2Vec, ale dla grafów, generując embeddingi grafów na podstawie ich struktury.

**Prezentacja Danych DeepWalk**

1. **Implementacja DeepWalk**

W implementacji algorytmu DeepWalk, zastosowano podejście do budowy sekwencji losowych spacerów po grafie, a następnie wykorzystano Word2Vec do nauki reprezentacji wierzchołków.

Word2Vec to algorytm używany do uczenia reprezentacji słów w przestrzeni o mniejszej wymiarowości. Pomimo że został pierwotnie opracowany do pracy z danymi tekstowymi, takimi jak zdania czy dokumenty, może być także zastosowany do innych rodzajów danych, w tym do reprezentacji wierzchołków w grafach, co jest często wykorzystywane w przypadku metod embeddingu grafu, takich jak DeepWalk.

W kontekście implementacji DeepWalk, wykorzystanie Word2Vec odnosi się do sposobu uczenia reprezentacji wierzchołków na podstawie sekwencji spacerów po grafie. Proces ten można opisać w kilku krokach:

* 1. Generowanie Sekwencji Spacerów:

Algorytm DeepWalk rozpoczyna od generowania sekwencji losowych spacerów po grafie. Spacer ten jest realizowany poprzez poruszanie się po wierzchołkach grafu zgodnie z pewnymi regułami, na przykład przy użyciu losowego błądzenia.

* 1. Przekształcanie Sekwencji na Konteksty i Słowa:

Następnie, każda sekwencja spacerów jest przekształcana na pary "kontekst - słowo", gdzie wierzchołki są traktowane jak słowa. Kontekstem dla danego słowa może być na przykład kilka wierzchołków, które występują w tej samej sekwencji spacerów.

* 1. Uczenie Word2Vec:

Otrzymane pary "kontekst - słowo" są używane do trenowania modelu Word2Vec. Model ten nauczy się reprezentacji numerycznych dla każdego wierzchołka w grafie na podstawie kontekstów, w jakich występuje.

* 1. Uzyskanie Embeddingów Wierzchołków:

Po zakończeniu procesu uczenia, otrzymujemy embeddingi wierzchołków, które są reprezentacjami numerycznymi wierzchołków grafu w przestrzeni o mniejszej wymiarowości.

Te embeddingi wierzchołków mogą być używane do różnych celów, takich jak analiza struktury grafu, rekomendacje, czy też klasyfikacja wierzchołków. Wyniki Word2Vec w przypadku DeepWalk pozwalają uzyskać semantyczne reprezentacje wierzchołków, co umożliwia lepsze zrozumienie ich roli i relacji w analizowanym grafie.

1. **Analiza Embeddingów za pomocą T-SNE**

Do wizualizacji uzyskanych embeddingów użyto algorytmu T-SNE, umożliwiającego przedstawienie wierzchołków grafu w przestrzeni o mniejszej wymiarowości, z zachowaniem ich wzajemnych odległości.

**T-SNE**, czyli t-distributed stochastic neighbor embedding, to algorytm do wizualizacji danych w przestrzeni o mniejszej wymiarowości. Jego głównym celem jest przeniesienie punktów z oryginalnej, wysokowymiarowej przestrzeni danych do przestrzeni o niższym wymiarze, zachowując przy tym istotne struktury odległości między punktami.

Oto kilka kluczowych cech algorytmu T-SNE:

1. Rozkład Studenta t-distribution:

T-SNE używa rozkładu Studenta (t-distribution) w celu modelowania odległości między punktami w oryginalnej przestrzeni i przestrzeni docelowej. Ten rozkład ma właściwość szerokiego ogona, co pozwala na skupienie się na utrzymaniu odległości między odległymi punktami w oryginalnych danych.

1. Dwuetapowy Proces:

Algorytm działa w dwóch etapach: konstrukcji macierzy podobieństwa dla oryginalnych danych i dla danych docelowych. W obu przypadkach stosuje się różnice między odległościami do konstrukcji macierzy podobieństwa.

1. Minimalizacja Funkcji Kosztu:

T-SNE minimalizuje funkcję kosztu, która mierzy różnice między macierzami podobieństwa w oryginalnej i docelowej przestrzeni. W rezultacie punkty, które były blisko siebie w oryginalnych danych, są bardziej prawdopodobne, aby pozostać blisko siebie po transformacji do przestrzeni o mniejszej wymiarowości.

1. Utrzymywanie Lokalnych Struktur:

T-SNE jest znane z utrzymania lokalnych struktur w danych, co oznacza, że sąsiadujące punkty w oryginalnej przestrzeni są bardziej skłonne do zachowania tej relacji w przestrzeni docelowej.

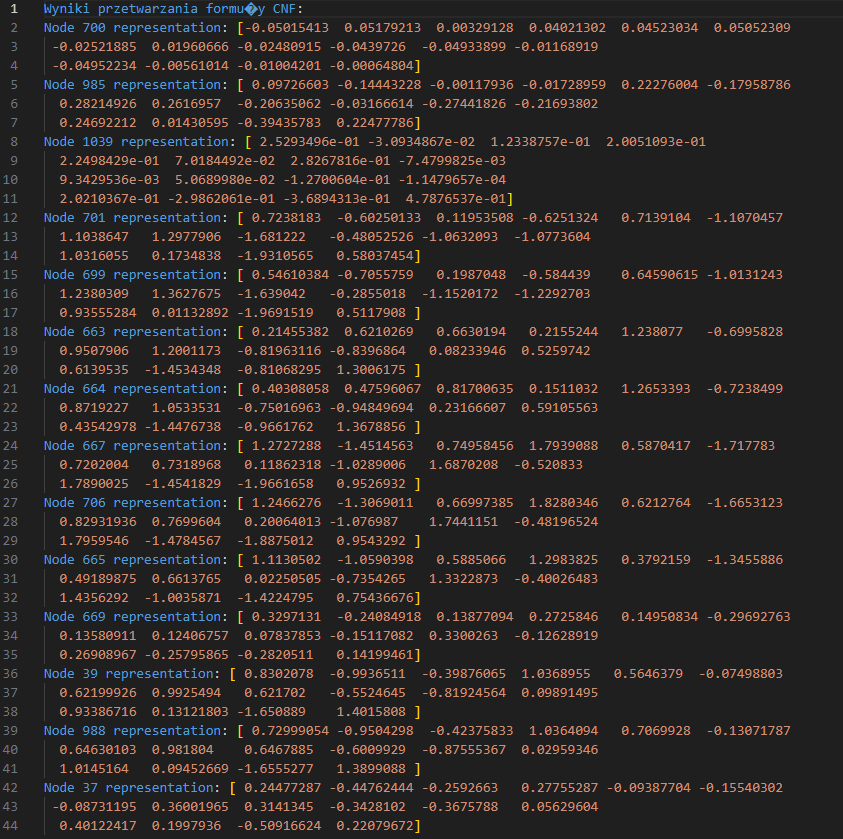
1. Wrażliwość na Parametr Perplexity:

Parametr perplexity w T-SNE wpływa na to, ile sąsiadów brane jest pod uwagę podczas konstrukcji macierzy podobieństwa. Różne wartości perplexity mogą prowadzić do różnych rezultatów w wizualizacji.

Algorytm T-SNE jest często używany do wizualizacji danych w obszarach takich jak analiza embeddingów wierzchołków grafów. W kontekście projektu, w którym analizowane są embeddingi wierzchołków uzyskane za pomocą DeepWalk, T-SNE pozwala na przedstawienie tych embeddingów w przestrzeni o niższym wymiarze, co ułatwia zrozumienie ich struktury i wzajemnych relacji.

**Wyniki i Interpretacja**

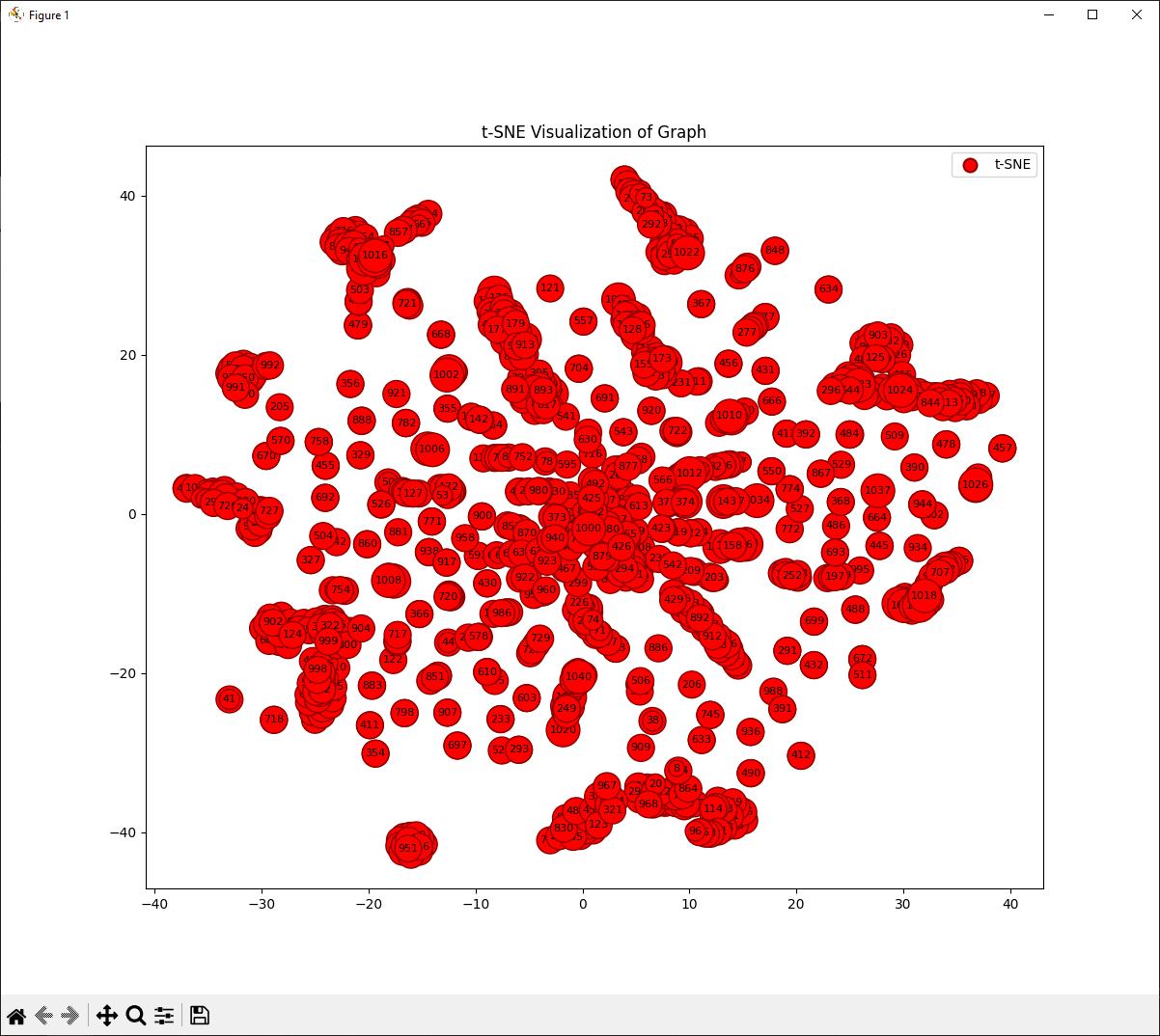
Wynikiem dla samego algorytmu DeepWalk dla plików CNF (Conjunctive Normal Form) powinny być numeryczne reprezentacje wierzchołków grafu logicznego, uzyskane poprzez proces embeddingu. Każdy wierzchołek grafu, reprezentujący formułę logiczną, zostanie przekształcony w wektor numeryczny, który zachowuje istotne relacje i strukturę logiczną.

Przykład:  
  


Wynikiem prezentacji danych za pomocą metody wizualizacji T-SNE są zbiorowiska wierzchołków, im bliżej siebie są wierzchołki na grafie, tym bardziej podobne są do siebie w oryginalnym, wysokowymiarowym zbiorze danych. Algorytm T-SNE jest zaprojektowany tak, aby zachować struktury podobieństw między wierzchołkami, co oznacza, że w przestrzeni o niższej wymiarowości zachowuje się odległości między wierzchołkami, które były blisko siebie w oryginalnej przestrzeni danych.

W rezultacie zbiorowiska wierzchołków, które obserwujemy po zastosowaniu T-SNE, odzwierciedlają ich podobieństwo w oryginalnym grafie. Wierzchołki reprezentujące podobne formuły logiczne są skupione razem, tworząc klastry lub grupy. To ułatwia analizę i zrozumienie struktury danych logicznych, ponieważ wierzchołki, które są ze sobą podobne, są reprezentowane jako bliskie sobie punkty w przestrzeni wizualizacji T-SNE.

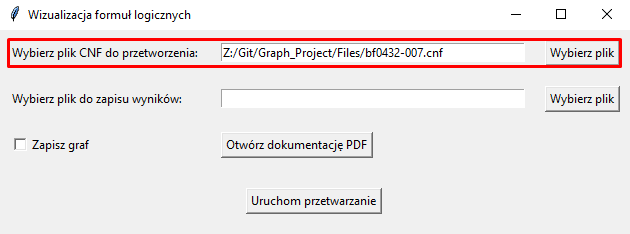
Warto jednak zauważyć, że w przypadku T-SNE, odległości między klastrami niekoniecznie odzwierciedlają rzeczywiste odległości między nimi w oryginalnych danych. T-SNE ma tendencję do skupiania się na utrzymaniu lokalnych struktur, więc odległości między klastrami mogą być zniekształcone. Dlatego interpretacja wyników T-SNE wymaga uwzględnienia ograniczeń tego algorytmu i zrozumienia, że odległości w przestrzeni T-SNE są bardziej związane z relacjami lokalnymi niż globalnymi w oryginalnych danych.



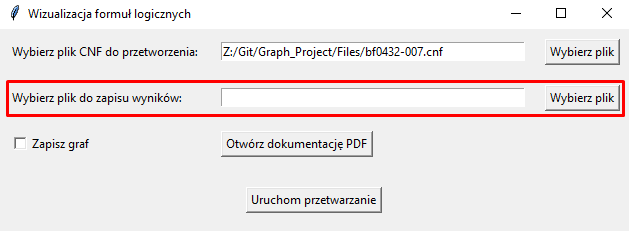
**Aplikacja**

Wymagania:

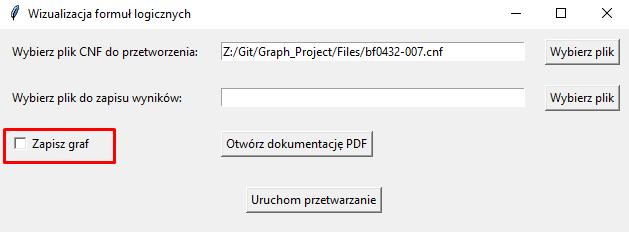
* Wymagania minimalne:
  + Wersja Pythona 3.x
* Zainstalowane biblioteki języka Python:
  + Tkinter
  + Networkx
  + Matplotlib
  + Webbrowser
  + Os
* Działanie aplikacji:

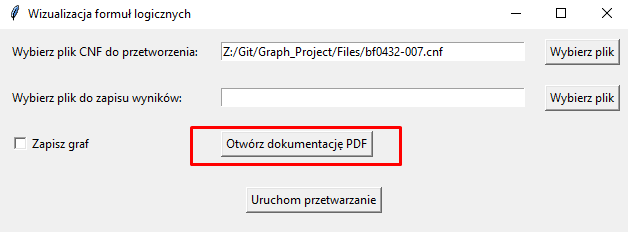


Wybieramy plik w formacie CNF, który zostanie przetworzony.

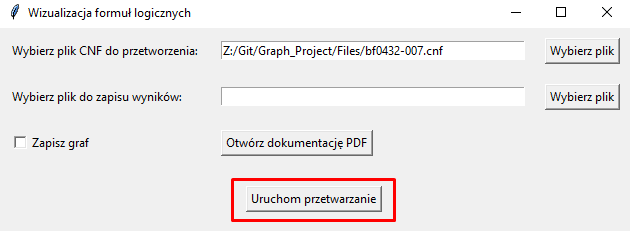


Jeżeli chcemy zapisać wyniki przeliczenia metodą DeepWalk możemy wprowadzić tutaj plik docelowy. Alternatywnie można ręcznie wpisać w tym oknie np. „wynik.txt”, co spowoduje zapisanie pliku w głównym folderze projektu.

  
Zaznaczenie opcji „Zapisz graf” daje możliwość zapisania w postaci pliku png wygenerowany graf.



Przycisk „Otwórz dokumentację PDF” otworzy aktualnie czytaną dokumentację w domyślnej przeglądarce.

  
Przycisk „Uruchom przetwarzanie” rozpoczyna właściwe działanie programu i rozpoczyna się przeliczenie, a następnie zostanie wyświetlony graf.

**Bibliografia**

<https://arxiv.org/abs/1403.6652>

<https://arxiv.org/abs/1607.00653>

<https://arxiv.org/abs/1503.03578>

<https://arxiv.org/abs/1609.02907>

<https://www.jmlr.org/papers/volume9/vandermaaten08a/vandermaaten08a.pdf>

<https://towardsdatascience.com/graph-embeddings-the-summary-cc6075aba007>

<http://forvis.agh.edu.pl/docs>

<https://people.sc.fsu.edu/~jburkardt/data/cnf/cnf.html>