



AKADEMIA GÓRNICZO-HUTNICZA
IM. STANISŁAWA STASZICA W KRAKOWIE

WYDZIAŁ ELEKTROTECHNIKI AUTOMATYKI INFORMATYKI I INŻYNIERII
BIOMEDYCZNEJ

INFORMATYKA

Wykorzystanie SAT Solverów w tworzeniu grafików

Izabela Kielbasa
Natalia Janecka

1 czerwca 2017

1 Użycie SAT Solvera w projekcie

Do rozwiązywania problemów SAT w projekcie została wykorzystana biblioteka dla Javy: **Sat4j**. Umożliwia ona rozwiązywanie problemów satysfakcji zdań logicznych oraz optymalizacji. Pozwala na rozwiązanie problemów SAT, MAXSAT, Pseudo-Boolean oraz Minimally Unsatisfiable Subset (MUS).

Biblioteka ta jest przyjazna dla użytkownika i zaprojektowana elastycznie używając wzorców projektowych takich jak dekorator i strategia. Jej zaletą jest również to, że jest open source.

SAT przyjmuje zdanie logiczne w formie **CNF** – koniunkcyjna postać normalna. Formuła zapisana w koniunkcji klauzul np.:

$$(x1 \vee x2 \vee x3) \wedge (x4 \vee x2)$$

Każda formuła CNF składa się ze **zmiennych**, które mogą przyjmować tylko wartości **prawda** lub **fałsz** (dla powyższego przykładu jest ich 4) oraz **klauzul** na zmiennych np.:

$$C1 \wedge C2 \wedge \dots \wedge Cn$$

(dla powyższego przykładu mamy 2 klauzule).

SAT znajduje rozwiązanie dla podanego zdania tak aby cała formuła była prawdziwa (wszystkie klauzule muszą być prawdziwe). Jeśli rozwiązanie nie istnieje to SAT zwraca odpowiednią informację.

MAXSAT znajduje jak największą ilość zmiennych tak aby zdanie było prawdziwe. Natomiast MUS znajduje jak najmniejszą ilość zmiennych dla której zdanie jest niespełnione.

Biblioteka sat4j przyjmuje jako wejście plik o rozszerzeniu **.cnf**. Plik ten składa się z 2 części: *preambuły* oraz *klauzul*.

1. Preambuła zawiera informacje na temat przykładu. Każda linia zaczyna się od litery która ją charakteryzuje.

- Linia komentarza zaczyna się od litery "c" a po niej następuje spacja i komentarz
- Problem line – jedna linia dla pliku, definiuje zawartość pliku. Dla pliku CNF wygląda ona następująco:

```
p cnf iloscZmiennych iloscKlauzul
```

2. Druga część pliku to część klauzul – pojawia się ona bezpośrednio po linii problemu. Zmienne są ponumerowane od 1 do n (gdzie n to ilość zmiennych). Każda klauzula jest reprezentowana jako sekwencja liczb, oddzielonych od siebie spacją, tabem lub znakiem nowej linii. Zmienna prawdziwa jest zapisywana jako *i*, a zmienna fałszywa jako $\neg i$. Każda klauzula jest zakończona wartością 0. Dla powyższego przykładu plik .cnf ma postać

```
p cnf 4 2
1 2 3 0
4 2 0
```

Jako rozwiązanie SATa będzie tu przedstawiona linia

```
1 2 3 4 0
```

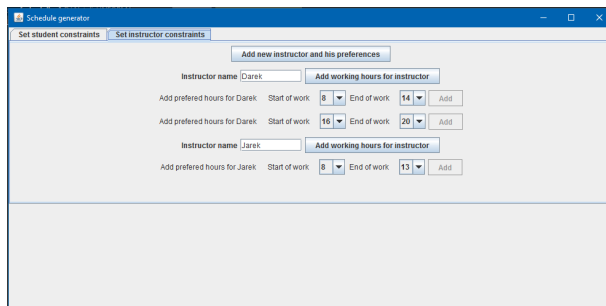
co oznacza, że aby dana formuła była prawdziwa wszystkie jej zmienne muszą być prawdziwe. Jeśli przed którąś ze zmiennych byłby znak \neg to zmienna ta powinna być fałszywa aby formuła była spełniona.

2 Słownik pojęć

- Instruktor - osoba pracująca, która prowadzi jazdy, ma określone godziny pracy (patrz niżej), jednocześnie może prowadzić zajęcia tylko z jednym kursantem
- kursant - osoba uczestnicząca w zajęciach, może wybrać jednego instruktora z listy, z którym chce odbywać jazdy. Ma określone preferowane godziny zajęć (patrz niżej). Osteczna godzina zajęć zostanie wybrana z listy preferowanych przez kursanta jeśli wybrany przez niego instruktor jest wtedy dostępny
- Godziny pracy instruktora - para liczb: godzina rozpoczęcia pracy i godzina zakończenia pracy. Pierwsza liczba zawsze musi być mniejsza od drugiej. Może być kilka takich par - oznacza to, że instruktor pracuje z przerwami w godzinach, które nie zawierają się pomiędzy liczbami w parach.
- Preferowane godziny zajęć - pary liczb - godzina rozpoczęcia i zakończenia jazdy. W tym czasie kursant może odbywać jazdy. Może być ich kilka

3 Tworzenie grafiku

W aplikacji możemy dodać instruktorów i wybrać dla nich godziny pracy.



W drugiej zakładce można dodać kursantów. Wybieramy dla każdego jednego instruktora oraz godziny, w których chciałby mieć zajęcia.

Schedule generator

Set student constraints | Set instructor constraints

Add new Student and his preferences

Student name: **Add preferences for student**

Select instructor:

Add preferred hours for Mia: Start of lesson: End of lesson: **Add**

Add preferred hours for Mia: Start of lesson: End of lesson: **Add**

Student name: **Add preferences for student**

Select instructor:

Add preferred hours for Joey: Start of lesson: End of lesson: **Add**

Student name: **Add preferences for student**

Select instructor:

Add preferred hours for Emma: Start of lesson: End of lesson: **Add**

Add preferred hours for Emma: Start of lesson: End of lesson: **Add**

Generate schedule

Kiedy dodaliśmy już wszystkie preferencje klikamy przycisk *Generate*. W nowym okienku pokaże nam się wygenerowany przez SAT grafik, który spełnia wszystkie ograniczenia. Wynikowa tabelka jest posortowana według instruktorów, a następnie godzin rozpoczęcia jazdy.

Schedule

Students name	Start of the lesson	End of the lesson	Instructor
Mia	8	11	Darek
Emma	16	17	Darek
Joey	9	10	Jarek

Schedule generator

Add new Student and his preferences

Student name: **Add preferences for student**

Select instructor:

Add preferred hours for Mia: Start of lesson: End of lesson: **Add**

Add preferred hours for Mia: Start of lesson: End of lesson: **Add**

Student name: **Add preferences for student**

Select instructor:

Add preferred hours for Joey: Start of lesson: End of lesson: **Add**

Student name: **Add preferences for student**

Select instructor:

Add preferred hours for Emma: Start of lesson: End of lesson: **Add**

Add preferred hours for Emma: Start of lesson: End of lesson: **Add**

Generate schedule