

Repensando grades horárias no IME USP com lógica simbólica

Introdução

Construir boas grades horárias é um problema tão complexo quanto é importante. No IME USP, um único professor é encarregado por todo o escalonamento, resultando em um esforço monumental e resultados, por vezes, insatisfatórios.

Nesse sentido, o trabalho desenvolvido busca auxiliar na construção de boas grades horárias para o Departamento de Ciência da Computação por meio do desenvolvimento um sistema especialista de escalonamento usando Answer Set Programming.

O desafio

O especialista deve conseguir expressar o seu conhecimento acumulado no decorrer dos anos em regras simples. O sistema deve ser fácil de modificar para conseguir acompanhar a evolução dos requisitos do corpo acadêmico no decorrer dos anos.

Answer Set Programming (ASP)

Answer Set Programming é um paradigma de programação que se baseia na computação de modelos estáveis para solucionar problemas complexos de satisfação de restrições eficientemente (Lifschitz, 2019). Em particular, o paradigma é considerado uma ótima alternativa para modelar o problema do escalonamento universitário devido a sua ótima expressividade e desempenho de busca por soluções (Banbara et al., 2013).

Como funciona

A produção de uma grade horária é um processo em três etapas:

- ▶ **Gerar** um conjunto de grades horárias candidatas que respeitem a estrutura esperada no IME-USP.
- ▶ **Excluir** grades inválidas baseado em *hard constraints*.
- ▶ **Otimizar** as grades hipóteses restantes, segundo critérios definidos por *soft constraints*.

Hard e *soft constraints* podem ser habilitadas e desabilitadas por um arquivo de configuração. O usuário também pode modificar os pesos e as prioridades de cada *soft constraints* para alterar sua importância nas grades geradas.

O modelo base, que não pode ser modificado por configuração, é capaz de se adaptar a inúmeros cenários de uso devido a uma modelagem pouco restritiva e resistente a erros do usuário.

Respeitando o especialista

Como um sistema especialista, o escalonador tenta ao máximo replicar o conhecimento do professor responsável no DCC. No entanto, é comum que os trâmites burocráticos envolvidos nas decisões do escalonamento exijam que decisões não ótimas sejam tomadas sobre o escalonamento.

Para lidar com esses casos, o `scheduLy` permite que o especialista fixe horários de algumas disciplinas na grade horária, e então otimize apenas as soluções que respeitem a configuração proposta pelo professor.

Exemplos de codificação de regras em ASP

```
1 %*
2 HARD CONSTRAINT
3 -----
4 Duas disciplinas (CourseA e CourseB) lecionadas pelo mesmo
5 professor (Teacher) não podem conflitar.
6 %*
7 :- conflict(CourseA, GroupA, CourseB, GroupB),
8     lecturer(CourseA, GroupA, Teacher),
9     lecturer(CourseB, GroupB, Teacher).
10
11 %*
12 SOFT CONSTRAINT
13 -----
14 Penaliza grades com aulas de disciplinas de graduação do BCC na
15 sexta-feira a tarde.
16 %*
17 friday_afternoon_class(Course, Period) :-
18     class(Course, "bcc", friday, Period, false),
19     is_undergrad(Course),
20     part_of_day(Period, PartOfDay),
21     PartOfDay = "morning".
22
23 :- friday_afternoon_class(Course, Period).
24 [w_no_friday_afternoon@p_no_friday_afternoon, Course, Period]
```

Resultados e conclusão

Dadas as diversas considerações feitas durante o desenvolvimento do escalonador quanto à extensibilidade e a configurabilidade, a modelagem do escalonamento no IME USP foi um sucesso. Além disso, o escalonador parece ter boa receptividade do especialista, que pontou facilidade de compreensão das regras mesmo sem muitos conhecimentos em ASP.

Bibliografia

- ▶ Banbara, Mutsunori et al. (July 2013). "Answer set programming as a modeling language for course timetabling". *Theory and Practice of Logic Programming* 13.4, pp. 783–798. ISSN: 1471-0684, 1475-3081. DOI: 10.1017/S1471068413000495. URL: https://www.cambridge.org/core/product/identificator/S1471068413000495/type/journal_article (visited on 08/20/2023).
- ▶ Lifschitz, Vladimir (2019). "What Is Answer Set Programming?" In: *Twenty-Third AAIL Conference on Artificial Intelligence*. publisher = Springer Heidelberg.