Coloring Map Problem with *Answer Set Programming – Clingo* An Encoding

Claudio Cesar de Sá[1]

Independent Researcher

Roteiro

Roteiro

Attention: some background in logic and declarative language is recommended!

Answer Set Programming

in a Nutshell (this page was borrowed from official repository)

Answer Set Programming

in a Nutshell (this page was borrowed from official repository)

- ASP is an approach to declarative problem solving, combining
  - a rich yet simple modeling language
  - with high-performance solving capacities

Answer Set Programming

in a Nutshell (this page was borrowed from official repository)

- ASP is an approach to declarative problem solving, combining
  - a rich yet simple modeling language
  - with high-performance solving capacities
- ASP has its roots in
  - (deductive) databases
  - logic programming (with negation)
  - (logic-based) knowledge representation and (nonmonotonic) reasoning
  - constraint solving (in particular, SATisfiability testing)

Answer Set Programming

in a Nutshell (this page was borrowed from official repository)

- ASP is an approach to declarative problem solving, combining
  - a rich yet simple modeling language
  - with high-performance solving capacities
- ASP has its roots in
  - (deductive) databases
  - logic programming (with negation)
  - (logic-based) knowledge representation and (nonmonotonic) reasoning
  - constraint solving (in particular, SATisfiability testing)
- ASP allows for solving all search problems in NP (and NP^{NP}) in a uniform way

Answer Set Programming

in a Nutshell (this page was borrowed from official repository)

- ASP is an approach to declarative problem solving, combining
  - a rich yet simple modeling language
  - with high-performance solving capacities
- ASP has its roots in
  - (deductive) databases
  - logic programming (with negation)
  - (logic-based) knowledge representation and (nonmonotonic) reasoning
  - constraint solving (in particular, SATisfiability testing)
- ASP allows for solving all search problems in NP (and NP^{NP}) in a uniform way
- ASP fundaments in: propositional, first-order, auto-epistemic, default. Many aspects in theory and programs.

Answer Set Programming

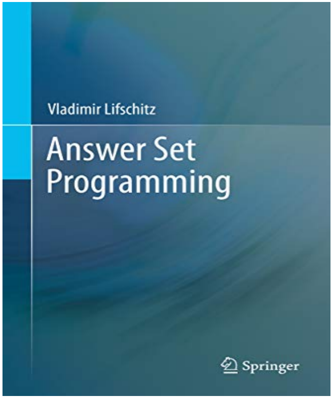
in a Nutshell (this page was borrowed from official repository)

- ASP is an approach to declarative problem solving, combining
  - a rich yet simple modeling language
  - with high-performance solving capacities
- ASP has its roots in
  - (deductive) databases
  - logic programming (with negation)
  - (logic-based) knowledge representation and (nonmonotonic) reasoning
  - constraint solving (in particular, SATisfiability testing)
- ASP allows for solving all search problems in NP (and NP^{NP}) in a uniform way
- ASP fundaments in: propositional, first-order, auto-epistemic, default. Many aspects in theory and programs.
- ASP embraces many emerging application areas

Hystorical and references

- This programming language has its root at the Universität Potsdam – 1999
- **Potassco**, the **Po**tsdam **A**nswer **S**et **S**olving **Co**llection – `https://potassco.org/`
- Official repository with a full-courese: `https://github.com/potassco-asp-course/`
- Support to start: an active forum and a course covered by videos in the Youtube
- This presentation and its code: `https://github.com/claudiosa/CCS/tree/master/asp_Answer_Set_Programming`
- Books:

Some References:



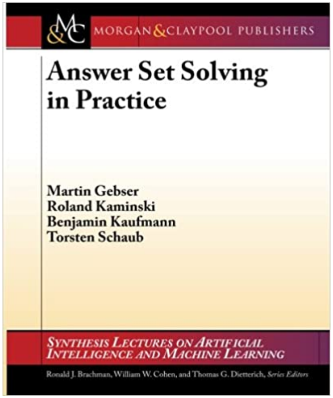

MORGAN & CLAYPOOL PUBLISHERS
Answer Set Solving in Practice
Martin Gebser
Roland Kaminski
Benjamin Kaufmann
Torsten Schaub
SYNTHESIS LECTURES ON ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING
Ronald J. Brachman, William W. Cohen, and Thomas G. Dietterich, Series Editors

Figure: Estou usando o do Vladmir

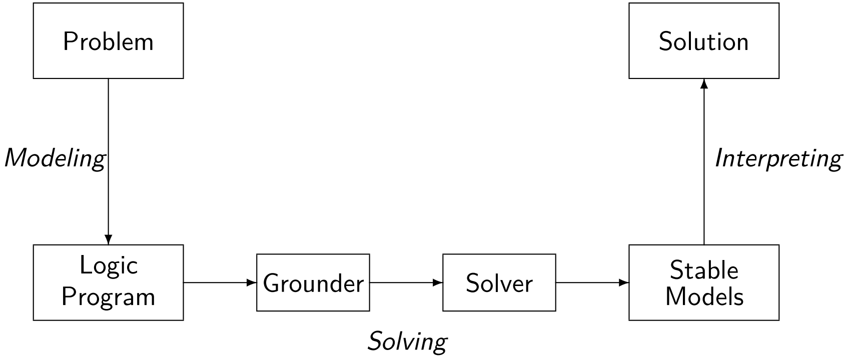
Características

- Mais declarativa que Prolog e seus predecessores (**apenas**, *e quase nada mais, tem uma sintaxe que lembra Prolog*)
- Raízes em várias lógicas, incluindo as que tratam de informações incompletas: LP, LPO, *default*, circunscrição (suposição do mundo-fechado) e negação como falha, (auto-epistêmica)
- Usa o conceito de **modelo estável**: *semântica bem-fundamentada* e *ramificação*
- Uso: problemas combinatoriais baseados em conhecimento declarativo – faremos um exemplo
- Consistem de **decisões** e **restrições**
- Tudo isto na ordem de milhões!
- Na indústria: desde gerenciador de pacotes do Debian a sistemas da NASA

Nesta apresentação

- A linguagem ASP com o sistema *clingo*
- *clingo* = *gringo* + *clasp*
- Há outras ramificações: *clingocon*, *aspcud* e *asprin*
- Alguns elementos da linguagem e um exemplo

Modelagem, aterramento, e resolução



Fonte: `https://github.com/potassco-asp-course/`

Graph coloring – a set of problems related

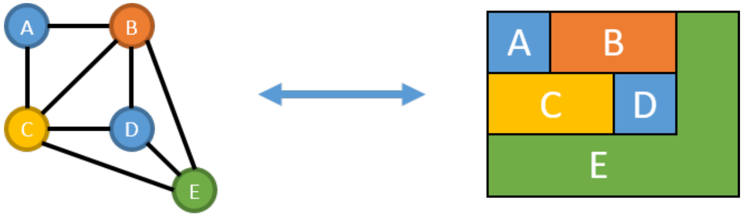


Figure: Let's consider a planar map for readgibility

- Input: a graph (map) G with n vertices (countries) and integer k (colors)
- Output: does G admit a proper vertex coloring with k colors?
- Complexity: NP-Complete
- Optimization: NP-Hard (lesser chromatic number – k)
- More details: `https://en.wikipedia.org/wiki/Graph_coloring`

The map to be colorized!

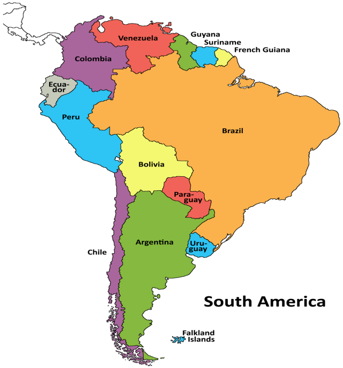


Figure: South America map, the author had the input data – ☺

The map to be colorized!

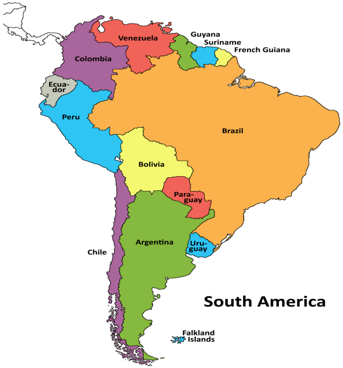


Figure: South America map, the author had the input data – ☺

So, let's find a minimal color number to paint this map!

Some comments:

- The modelling is immediate with an old Prolog code
- Many approaches for this problem can be taken: Simulated Annealing, Ant Colony, Depth-First Search, ..., meta-heuristics in general presents a good efficiency
- The full code disussed here is found in: `https://github.com/claudiosa/CCS/tree/master/asp_Answer_Set_Programming/map_coloring.lp`
- We are commenting it in parts

Colors availble (k) and countries (n):

```
color(red).
color(blue).
color(green).
color(yellow).

country(antilles).       country(argentina).
country(bolivia).        country(brazil).
country(columbia).       country(chile).
country(ecuador).        country(french_guiana).
country(guyana).         country(paraguay).
country(peru).           country(surinam).
country(uruguay).        country(venezuela).
```

Ground terms, exactly written like Prolog syntax.

The map, countries and their relations with neighbours:

```
neighbour(antilles,venezuela).
neighbour(argentina,brazil).
neighbour(argentina,paraguay).
neighbour(bolivia,brazil).
neighbour(bolivia,paraguay).
neighbour(brazil,columbia).
neighbour(brazil,guyana).
neighbour(brazil,peru).
neighbour(brazil,uruguay).
neighbour(chile,peru).
```

```
neighbour(argentina,bolivia).
neighbour(argentina,chile).
neighbour(argentina,uruguay).
neighbour(bolivia,chile).
neighbour(bolivia,peru).
neighbour(brazil,french_guiana)
neighbour(brazil,paraguay).
neighbour(brazil,surinam).
neighbour(brazil,venezuela).
neighbour(columbia,ecuador).
```

Another representation is possible, but until now, evertything was reused!

Modelling the problem under its requisites:

```
%%  Country X Colors - Assign any color for each country
1 { country_color(P, C) : color(C) } 1 :- country(P).

%% Brazil must be green
:- not country_color(brazil,green).
%% OR
%% country_color(brazil,green).

%% Finally: none adjacents countries receive at the same color
% C != C1 :- country_color(P, C), country_color(P1, C1),
             neighbour(P,P1).
%% OR -- by Susana - ASP Community
:- country_color(P, C), country_color(P1, C), neighbour(P,P1).
```

Basically, that's all!

Preparing for a optimization

```
%% number of colors used
n_colors(N) :-   N = #count{C : country_color(P,C)}.

%% A minimizations on this value
#minimize{ N : n_colors(N) }.

%% OUTPUT
#show country_color/2.
#show n_colors/1.
```

That's all!

An output

```
clingo ../map_coloring.lp 0 --out-ifs='\n' --out-atomf=%s.
clingo version 5.3.0
Reading from ../map_coloring.lp
Solving...
Answer: 1
country_color(argentina,red).
country_color(columbia,red).
country_color(surinam,red).
country_color(guyana,blue).
country_color(paraguay,blue).
...........................
country_color(french_guiana,yellow).
country_color(venezuela,yellow).
n_colors(4).
Optimization: 4
OPTIMUM FOUND
Models       : 1
  Optimum    : yes
Optimization : 4
Calls        : 1
```

Conclusions

-
-
-
-

Contact and Comments:

- `https://claudiocesar.wordpress.com/`
- `https://github.com/claudiosa`
- Neste git, repostiório CCS $\Rightarrow$ asp...
- Email: `ccs1664@gmail.com`
- *Thank you so much*!