# Declarative Heuristics for ASP

Dipl.-Ing. Paul Ogris, AAU Klagenfurt

2018-09-11

Answer Set Programming (*ASP*) has been used successfully to model and solve combinatorial search problems in industrial domains

No silver bullet, complex instances can still be prohibitively slow

No silver bullet, complex instances can still be prohibitively slow

**Use Heuristics!** (Gebser et al. 2013)

Modern ASP solvers use some variation of the CDCL algorithm

```
if (unitPropagation( , ) == conflict):
    return UNSAT
while not all variables assigned:
    (x, v) ← decide( , )
    dl ← dl + 1
      ←    {(x,v)}
    if (unitPropagation( , ) == conflict):
          ← conflictAnalysis( , )
        if (  < 0):
            return UNSAT
        else:
            backtrack( , , )
            dl ←
return SAT
```

`decide` can be anything that returns a literal!

`decide` can be anything that returns a literal!

General purpose heuristics are available

- VSIDS

`decide` can be anything that returns a literal!

General purpose heuristics are available

- VSIDS
- VMTF

`decide` can be anything that returns a literal!

General purpose heuristics are available

- VSIDS
- VMTF
- BerkMin

`decide` can be anything that returns a literal!

General purpose heuristics are available

- VSIDS
- VMTF
- BerkMin
- ...

General purpose heuristics do not always perform well

→ Use *domain specific heuristics* instead

General purpose heuristics do not always perform well

→ Use *domain specific heuristics* instead

The ASP solver needs to support the use of special heuristics

# Related Work

## Clingo Domain Heuristics

quux

bar

### HWasp

- foo

bar

## HWasp

- foo
- bar

# Procedural Heuristics with Clingo

# Declarative Heuristics

Gebser, Martin, Benjamin Kaufmann, Javier Romero, Ramón Otero, Torsten Schaub, and Philipp Wanko. 2013. "Domain-Specific Heuristics in Answer Set Programming." In *AAAI*. Citeseer. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.447.8848&rep=rep1&type=pdf.