THE EXTENSIONAL CONSTRAINT

Public defense

Hélène Verhaeghe

31 August 2021

ICTEAM, UCLouvain, Place Sainte Barbe 2, 1348 Louvain-la-Neuve, Belgium, *helene.verhaeghe@uclouvain.be*

Advisors:

- Pierre Schaus
- Christophe Lecoutre

Thesis jury:

- Yves Deville
- Claude-Guy Quimper
- Jean-Charles Régin
- Peter Van Roy

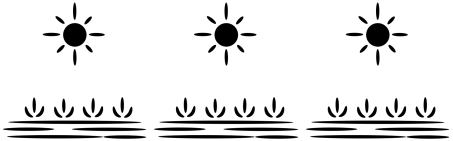
UCLouvain

The extensional constraint

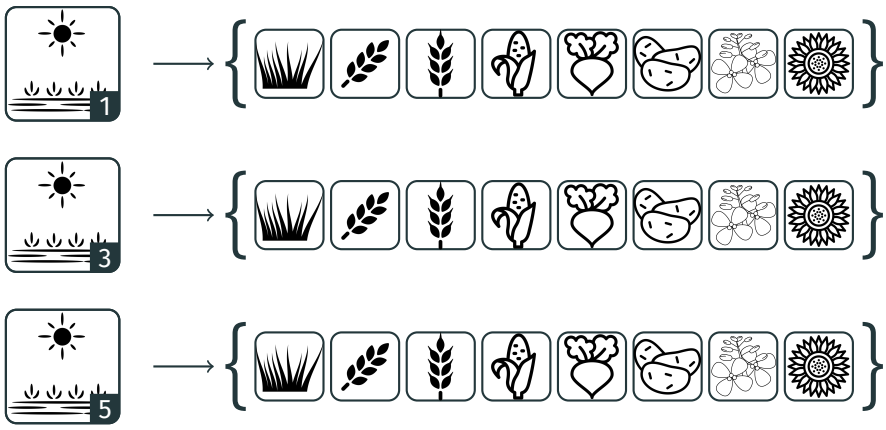
UCLouvain

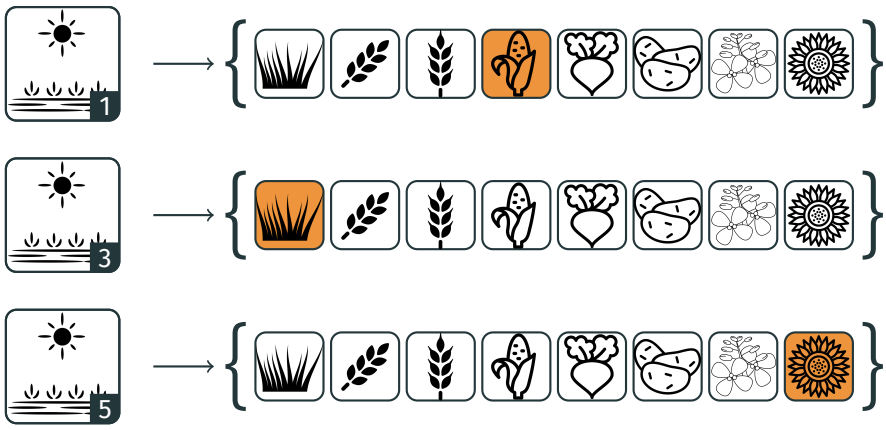
The extensional constraint

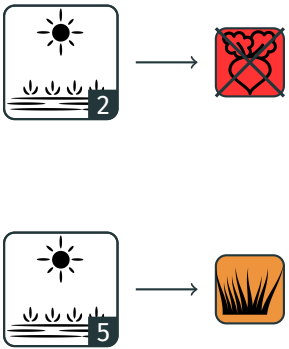
WHAT IS CONSTRAINT PROGRAMMING?



Variables


1
3
5
Variables
Domains


1
3
5
Variables
Domains



Unary constraints

$$\#i\left(\ \text{  }\ \longrightarrow\ \text{  }\ \right) \geq 10$$

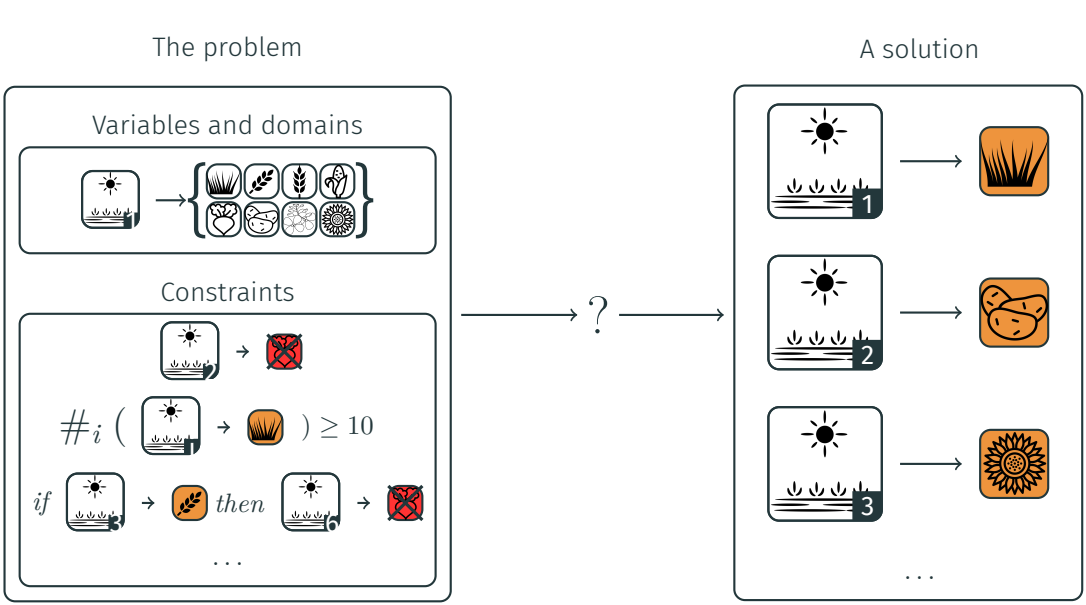
$$\#i\left(\ \text{  }\ \longrightarrow\ \text{  }\ \right) = 5$$

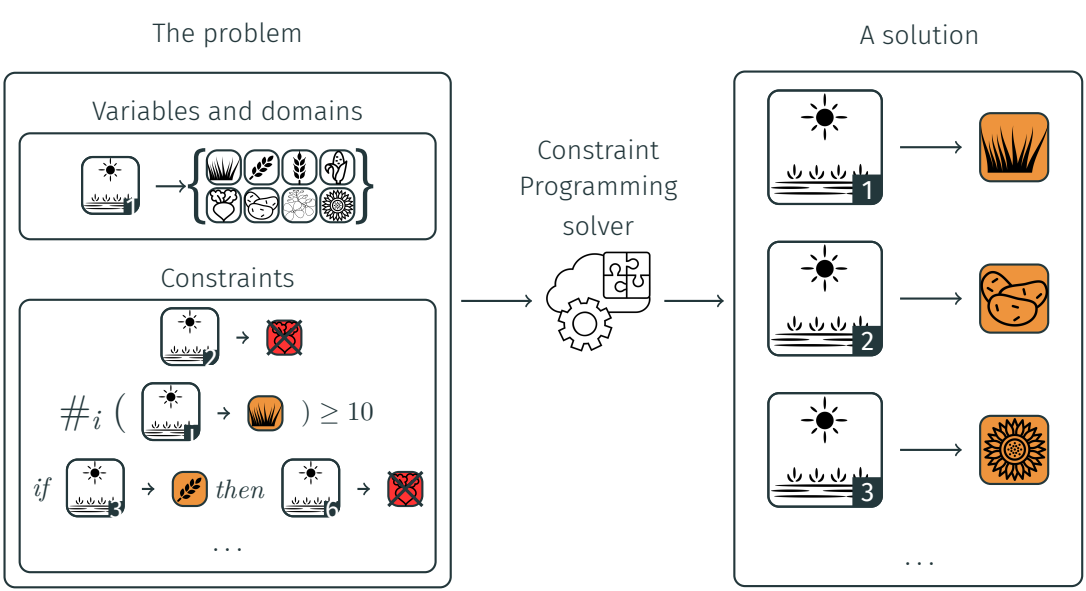
Arithmetic constraints

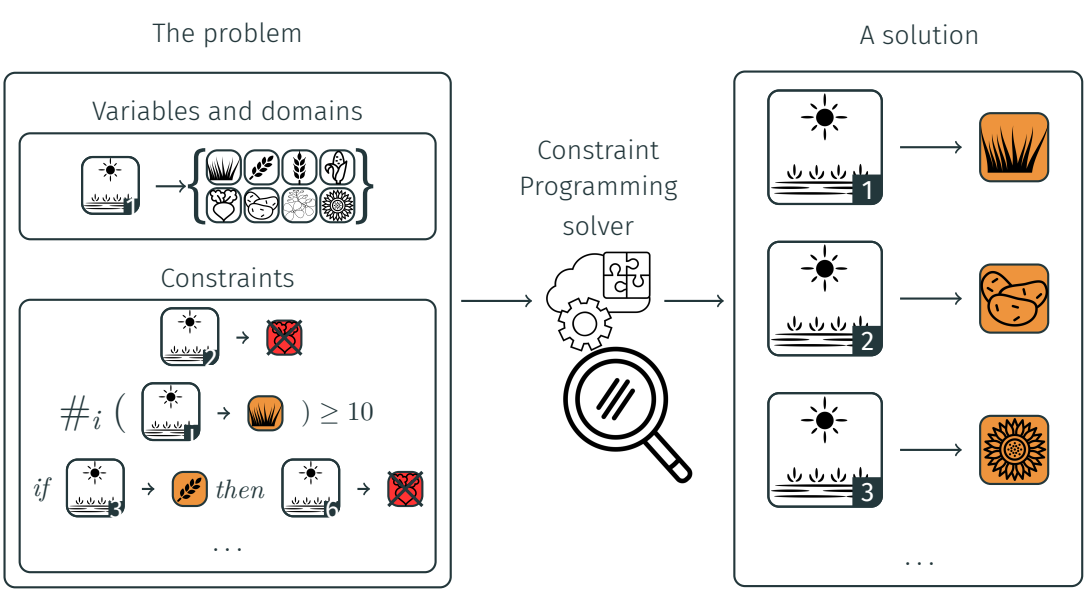
Logical constraints

Logical constraints

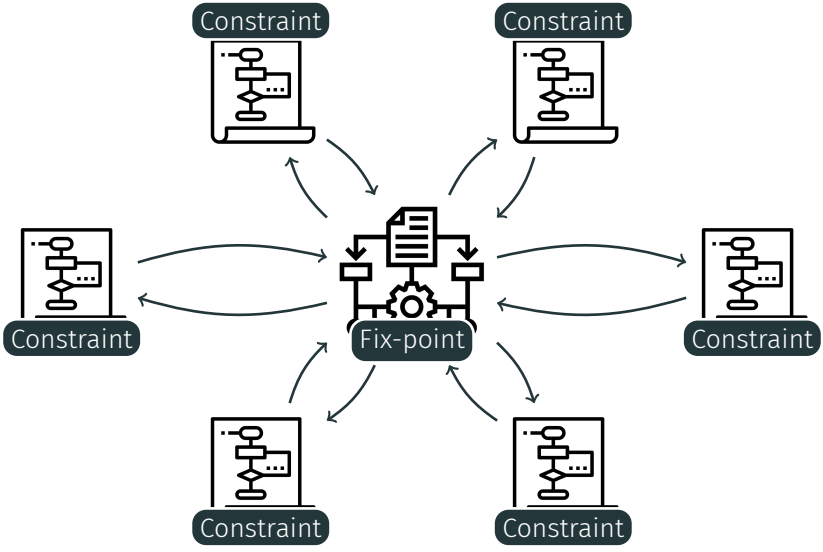
Combinatorial constraints


The problem
Variables and domains
Constraints
$\#_i$ ( ) ≥ 10
if then
...
?
A solution
1
2
3
...


The problem
Variables and domains
Constraints
if
then
...
Constraint Programming solver
A solution
1
2
3
...


The problem
Variables and domains
Constraints
$\#_i$ ( ) ≥ 10
if then
...
Constraint Programming solver
A solution
1
2
3
...


Fix-point


Constraint
Constraint
Constraint
Fix-point
Constraint
Constraint
Constraint

The extensional constraint

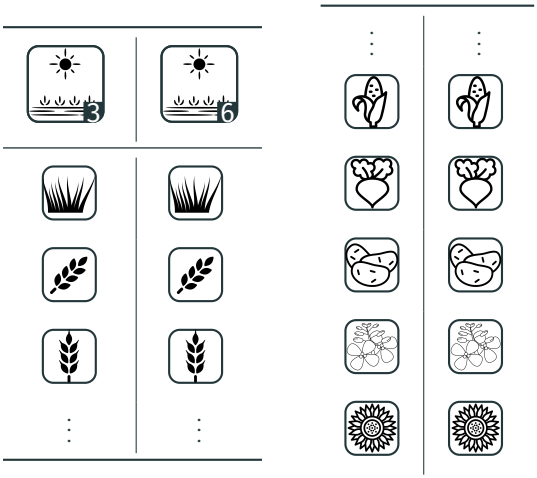
UCLouvain

The extensional constraint

WHAT IS AN EXTENSIONAL REPRESENTATION?

Intensional representation

Intensional representation



Extensional representation

Big question time!

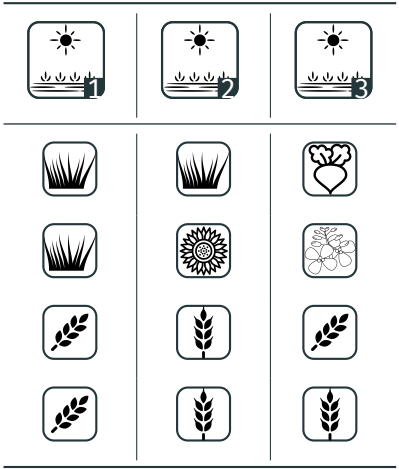
Big question time!

Why should we use extensional representation instead of intensional ones?

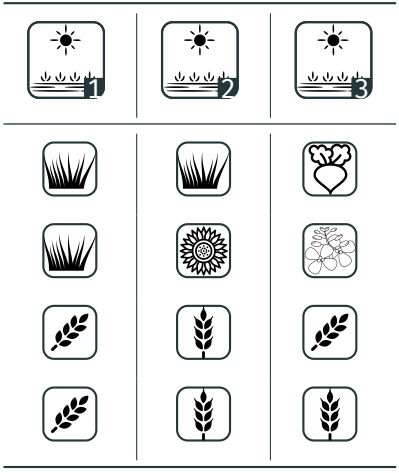
Big question time!

Why should we use extensional representation instead of intensional ones?

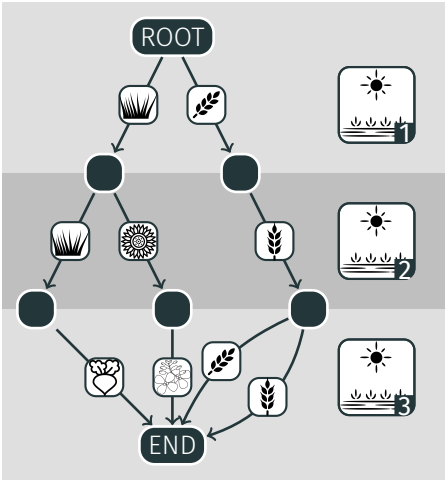
They can modelize any intensional formulas



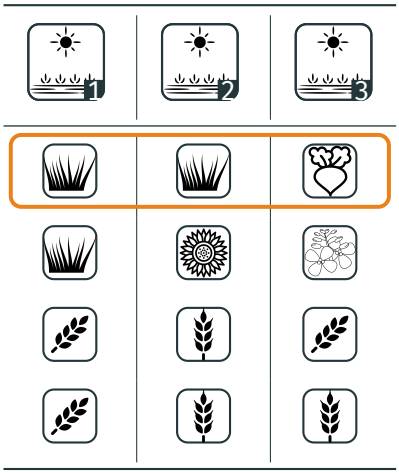
Tables



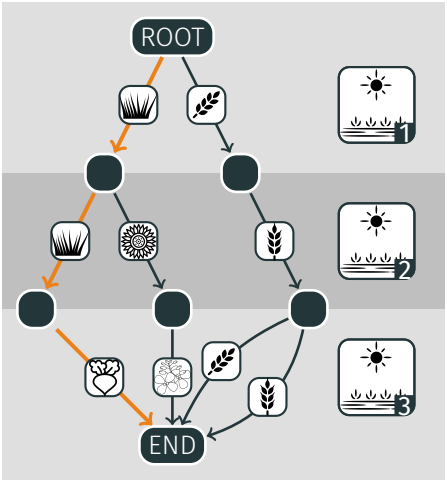
Tables



Diagrams (MDDs, MVDs,...)



Tables



Diagrams (MDDs, MVDs,...)

Table or diagram?

It depends!

Tables are more straightforward to update

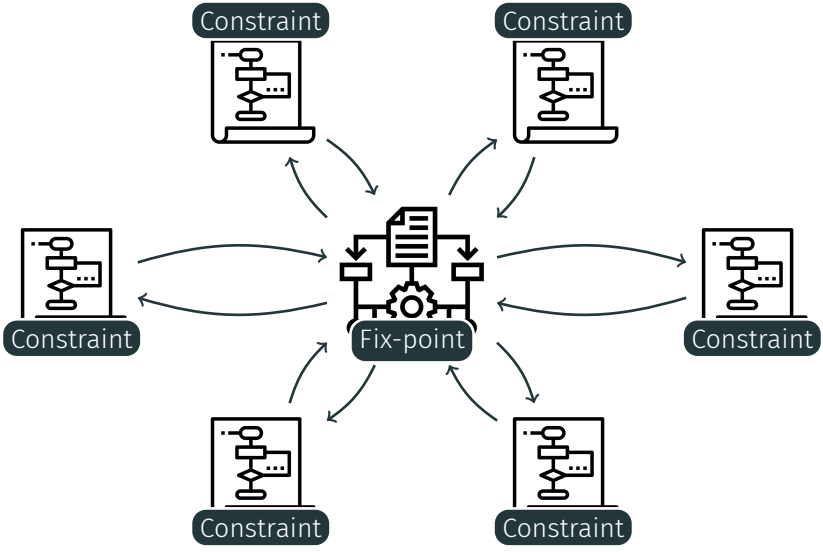
Diagrams can have a huge compression ratio

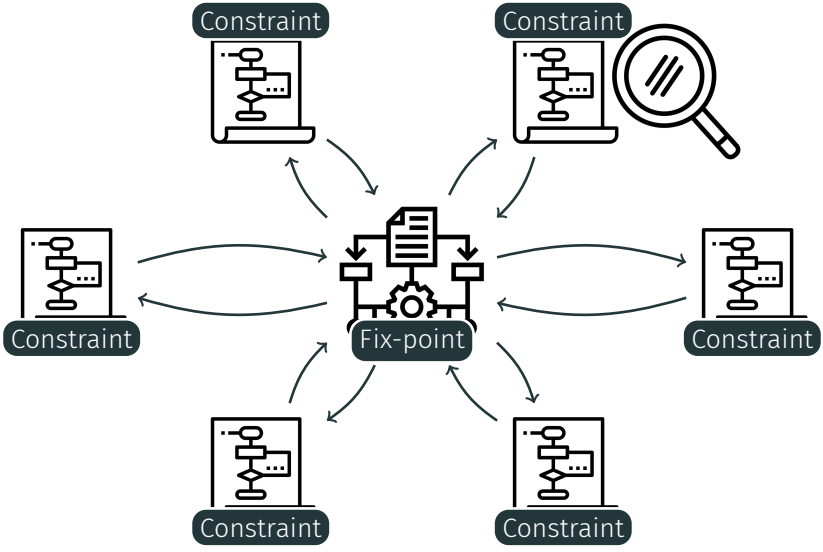
It depends on what you represent

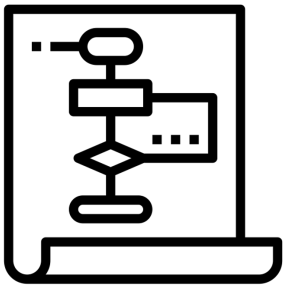
UCLouvain

The extensional constraint

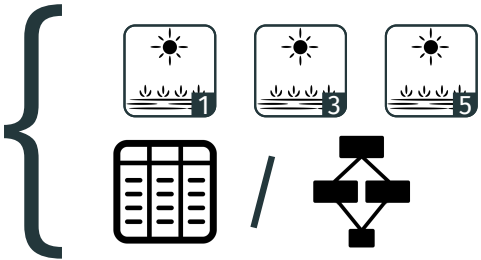
WHAT IS AN EXTENSIONAL CONSTRAINT?


Constraint
Constraint
Constraint
Fix-point
Constraint
Constraint
Constraint


Constraint
Constraint
Constraint
Fix-point
Constraint
Constraint
Constraint

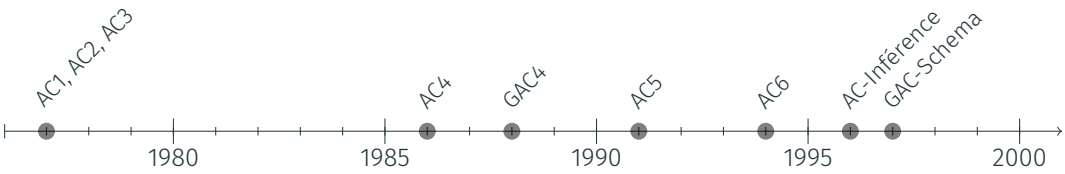


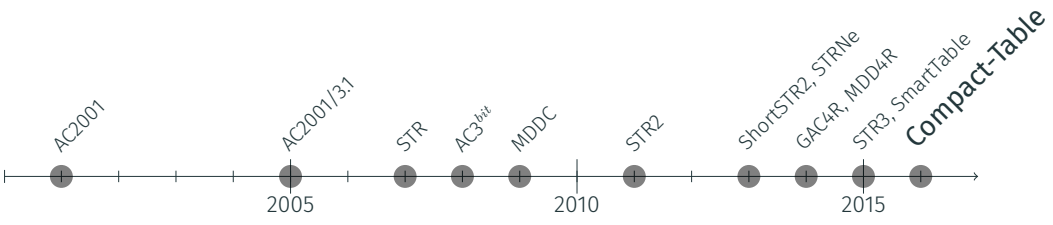
Input:



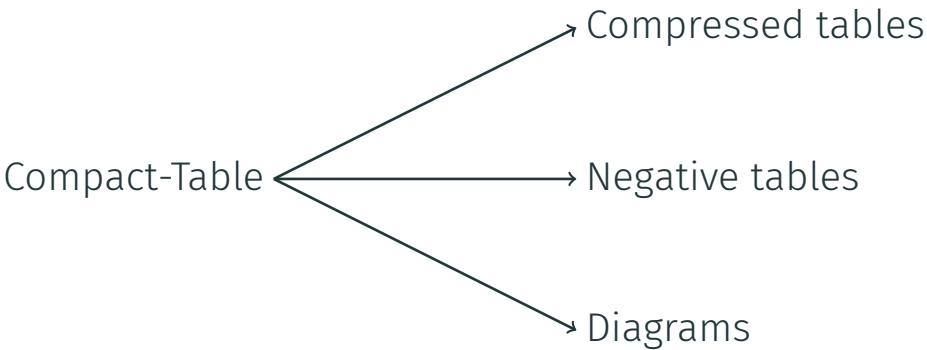
Goal:

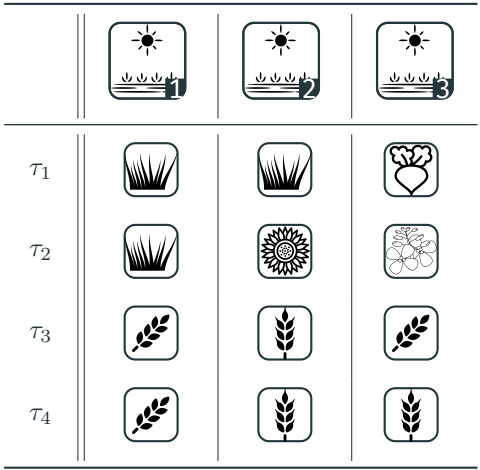
Ensure solution belong to


AC1, AC2, AC3
AC4
GAC4
AC5
AC6
AC-Inférence
GAC-Schema
1980
1985
1990
1995
2000


AC2001
AC2001/3.1
STR
AC3bit
MDDC
STR2
ShortSTR2, STRNe
GAC4R, MDD4R
STR3, SmartTable
Compact-Table
2005
2010
2015

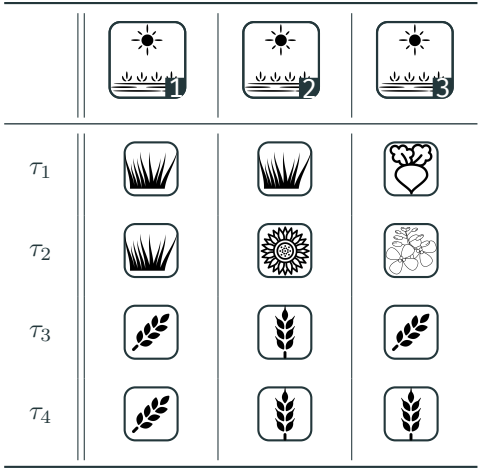
CONTRIBUTION OF THIS THESIS

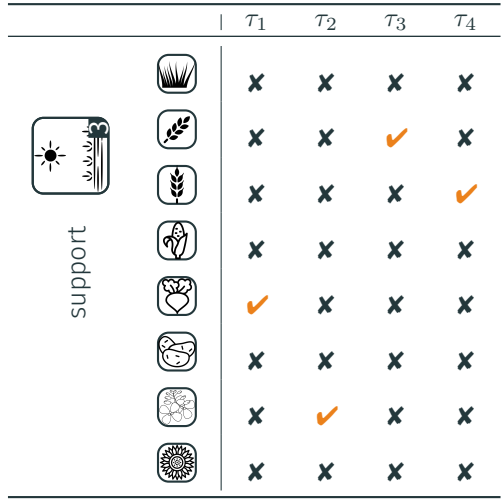

Compact-Table
Compressed tables
Negative tables
Diagrams



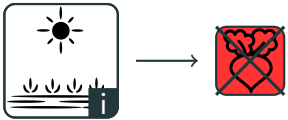
Current table:

τ_1	τ_2	τ_3	τ_4
✔	✔	✔	✔


1
2
3
τ_1
τ_2
τ_3
τ_4

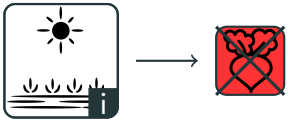

τ_1 τ_2 τ_3 τ_4
3
support

How do we update current table after



?

How do we update current table after



?

Supports help identify which tuples are no more valid

UCLouvain

	τ_1	τ_2	τ_3	τ_4	τ_5	τ_6	τ_7	τ_8
current table	✔	✔	✘	✔	✔	✔	✘	✘
support of removed value	✔	✘	✔	✔	✘	✘	✘	✔
intersection	✔	✘	✘	✔	✘	✘	✘	✘

⟵ tuples now invalid

UCLouvain

	τ_1	τ_2	τ_3	τ_4	τ_5	τ_6	τ_7	τ_8
current table	✔	✔	✘	✔	✔	✔	✘	✘
support of removed value	✔	✘	✔	✔	✘	✘	✘	✔
intersection	✔	✘	✘	✔	✘	✘	✘	✘

← tuples now invalid

	τ_1	τ_2	τ_3	τ_4	τ_5	τ_6	τ_7	τ_8
current table	✔	✔	✘	✔	✔	✔	✘	✘
intersection	✔	✘	✘	✔	✘	✘	✘	✘
removal	✘	✔	✘	✘	✔	✔	✘	✘

← new value of current table

UCLouvain

	τ_1	τ_2	τ_3	τ_4	τ_5	τ_6	τ_7	τ_8
current table	✔	✔	✘	✔	✔	✔	✘	✘
support of removed value	✔	✘	✔	✔	✘	✘	✘	✔
intersection	✔	✘	✘	✔	✘	✘	✘	✘

← tuples now invalid

	τ_1	τ_2	τ_3	τ_4	τ_5	τ_6	τ_7	τ_8
current table	✔	✔	✘	✔	✔	✔	✘	✘
intersection	✔	✘	✘	✔	✘	✘	✘	✘
removal	✘	✔	✘	✘	✔	✔	✘	✘

← new value of current table

Incremental update

UCLouvain

	τ_1	τ_2	τ_3	τ_4	τ_5	τ_6	τ_7	τ_8
support value 1	✔	✘	✘	✔	✔	✘	✘	✘
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
support value n	✘	✘	✔	✘	✘	✘	✘	✔
union	✔	✔	✔	✔	✔	✔	✘	✔

remaining values in the domain of the variable

← tuples valid for the given variable

UCLouvain

	τ_1	τ_2	τ_3	τ_4	τ_5	τ_6	τ_7	τ_8
support value 1	✔	✘	✘	✔	✔	✘	✘	✘
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
support value n	✘	✘	✔	✘	✘	✘	✘	✔
union	✔	✔	✔	✔	✔	✔	✘	✔

remaining values in the domain of the variable

← tuples valid for the given variable

	τ_1	τ_2	τ_3	τ_4	τ_5	τ_6	τ_7	τ_8
current table	✔	✔	✘	✔	✔	✔	✘	✘
union	✔	✔	✔	✔	✔	✔	✘	✔
intersection	✔	✔	✘	✔	✔	✔	✘	✘

← new value of current table

UCLouvain

	τ_1	τ_2	τ_3	τ_4	τ_5	τ_6	τ_7	τ_8
support value 1	✔	✘	✘	✔	✔	✘	✘	✘
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
support value n	✘	✘	✔	✘	✘	✘	✘	✔
union	✔	✔	✔	✔	✔	✔	✘	✔

remaining values in the domain of the variable

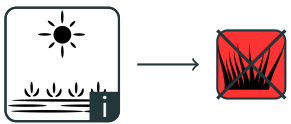
← tuples valid for the given variable

	τ_1	τ_2	τ_3	τ_4	τ_5	τ_6	τ_7	τ_8
current table	✔	✔	✘	✔	✔	✔	✘	✘
union	✔	✔	✔	✔	✔	✔	✘	✔
intersection	✔	✔	✘	✔	✔	✔	✘	✘

new value of current table ←

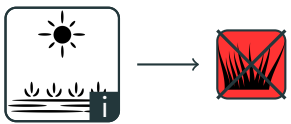
Reset update

How do we know if



?

How do we know if

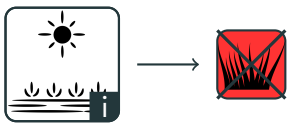


?

No tuples with the value anymore

UCLouvain

How do we know if



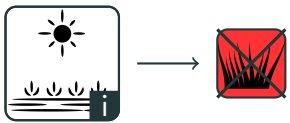
?

No tuples with the value anymore

	τ_1	τ_2	τ_3	τ_4	τ_5	τ_6	τ_7	τ_8
current table	✔	✔	✘	✔	✔	✔	✘	✘
support	✔	✘	✔	✔	✘	✘	✘	✔
intersection	✔	✘	✘	✔	✘	✘	✘	✘

UCLouvain

How do we know if

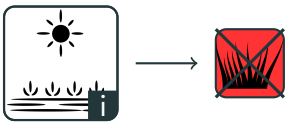


?

No tuples with the value anymore

	τ_1	τ_2	τ_3	τ_4	τ_5	τ_6	τ_7	τ_8
current table	✔	✔	✘	✔	✔	✔	✘	✘
support	✔	✘	✔	✔	✘	✘	✘	✔
intersection	✔	✘	✘	✔	✘	✘	✘	✘

if $|\text{intersection}| = 0$



POSITIVE TABLE CONSTRAINTS

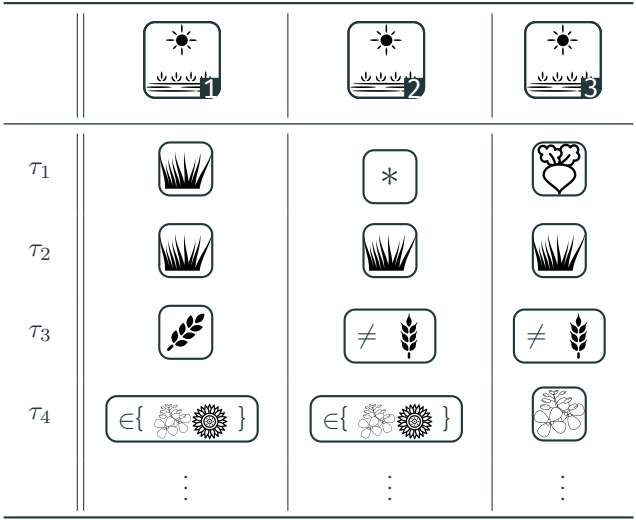
			
τ_1			
τ_2			
τ_3			
τ_4			
	⋮	⋮	⋮

UCLouvain

			
τ_1			
τ_2			
τ_3			
τ_4			
	⋮	⋮	⋮

single value

UCLouvain

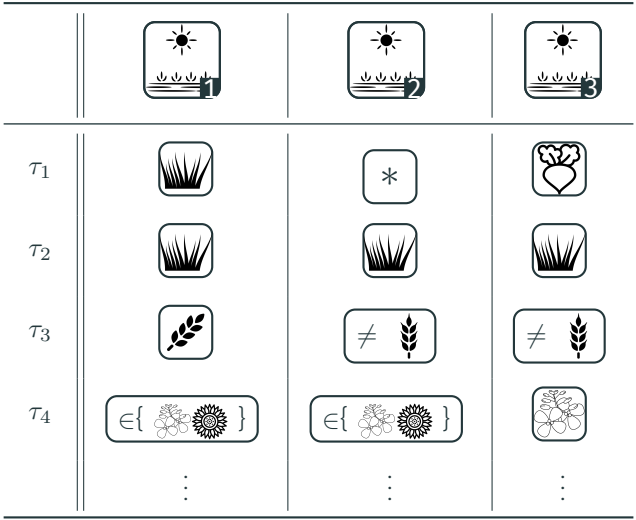


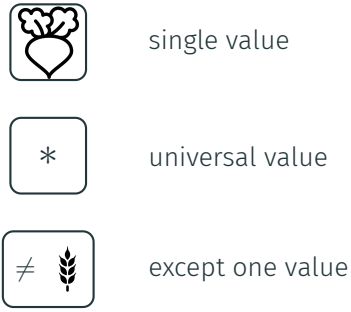
single value

$*$

universal value

UCLouvain



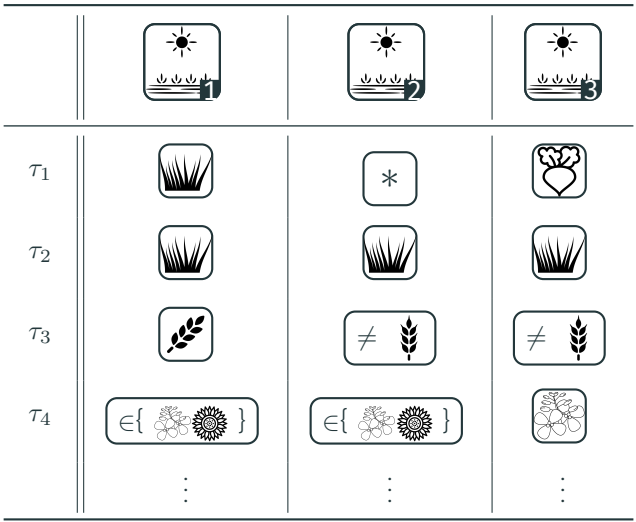


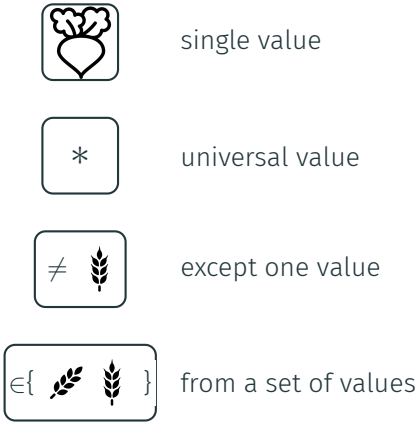
single value

universal value

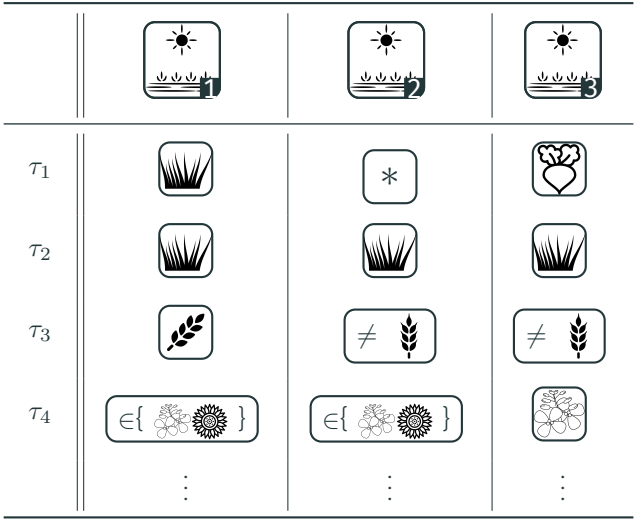
except one value

UCLouvain


1
2
3
τ_1
τ_2
τ_3
τ_4
$*$
$\neq$
$\in\{$ $\}$


single value
$*$
universal value
$\neq$
except one value
$\in\{$ $\}$
from a set of values

UCLouvain



single value

universal value

except one value

from a set of values

bounded values
(only with ordered domains)

How to adapt the update for

?

How to adapt the update for

?

If $|\text{dom}| = 1 \rightarrow$ reset update

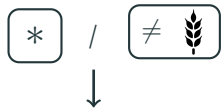
How to adapt the update for

?

If $|\text{dom}| = 1 \rightarrow$ reset update
If $|\text{dom}| > 1 \rightarrow$ stays valid

UCLouvain

Incremental update

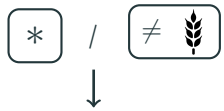


	τ_1	τ_2	τ_3	τ_4	τ_5	τ_6	τ_7	τ_8
current table	✔	✔	✘	✔	✔	✔	✘	✘
support of removed value	✔	✘	✔	✔	✘	✘	✘	✔
intersection	✔	✘	✘	✔	✘	✘	✘	✘

⟵ tuples now invalid

UCLouvain

Incremental update



	τ_1	τ_2	τ_3	τ_4	τ_5	τ_6	τ_7	τ_8
current table	✔	✔	✘	✔	✔	✔	✘	✘
exclusive support of removed value	✔	✘	✔	✘	✘	✘	✘	✔
intersection	✔	✘	✘	✘	✘	✘	✘	✘

⟵ tuples now invalid

How to adapt the update for

$\boxed{\leq j}$ $\boxed{\geq k}$

?

How to adapt the update for

$\boxed{\leq j}$ $\boxed{\geq k}$

?

If min(dom) $>$ j $\rightarrow$ $\boxed{\leq j}$ not supported

How to adapt the update for

$\boxed{\leq j}$ $\boxed{\geq k}$

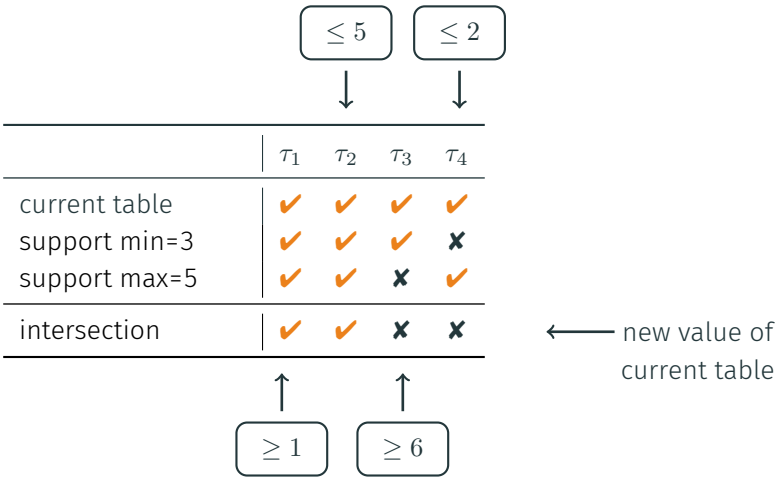
?

If min(dom) $>$ j $\rightarrow$ $\boxed{\leq j}$ not supported

If max(dom) $<$ k $\rightarrow$ $\boxed{\geq k}$ not supported

UCLouvain

Incremental update



What are the problems here?

What are the problems here?

Problem 1: the number of different sets

$$\sum_{k=1}^{n-1} \frac{n!}{(n-k)!k!}$$

$$\sum_{k=1}^{8-1} \frac{8!}{(8-k)!k!} = 256$$

What are the problems here?

Problem 2: incrementally detecting unsupported set requires individual counters for each set

Reset update is the only choice

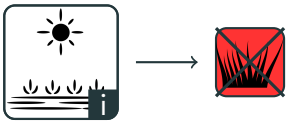
NEGATIVE TABLE CONSTRAINTS

			
τ_1			
τ_2			
τ_3			
τ_4			

UCLouvain

	1	2	3
τ_1			
τ_2			
τ_3			
τ_4			

How do we know if

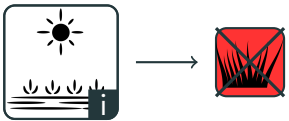


?

UCLouvain

	1	2	3
τ_1			
τ_2			
τ_3			
τ_4			

How do we know if



?

All combinations should be in the table

UCLouvain

	τ_1	τ_2	τ_3	τ_4	τ_5	τ_6	τ_7	τ_8
current table	✔	✔	✘	✔	✔	✔	✘	✘
support	✔	✘	✔	✔	✘	✘	✘	✔
intersection	✔	✘	✘	✔	✘	✘	✘	✘

UCLouvain

	τ_1	τ_2	τ_3	τ_4	τ_5	τ_6	τ_7	τ_8
current table	✔	✔	✘	✔	✔	✔	✘	✘
support	✔	✘	✔	✔	✘	✘	✘	✔
intersection	✔	✘	✘	✔	✘	✘	✘	✘

if |intersection| = |possible combinations |

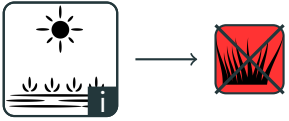
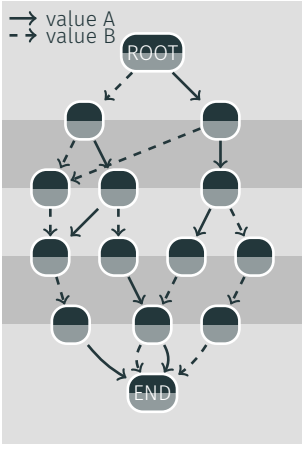


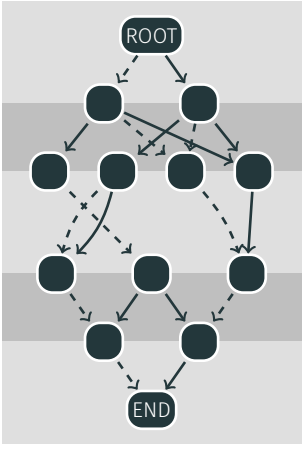
DIAGRAM CONSTRAINTS

UCLouvain

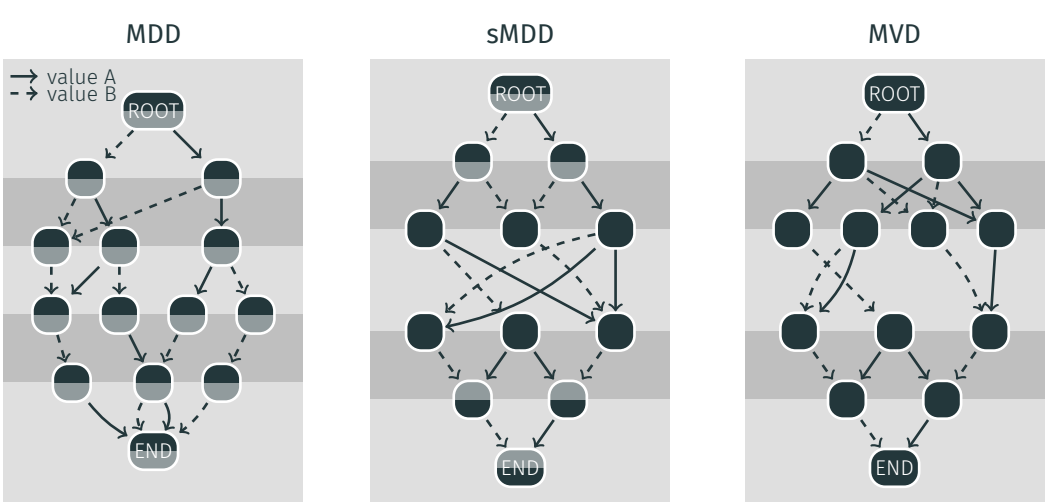
MDD

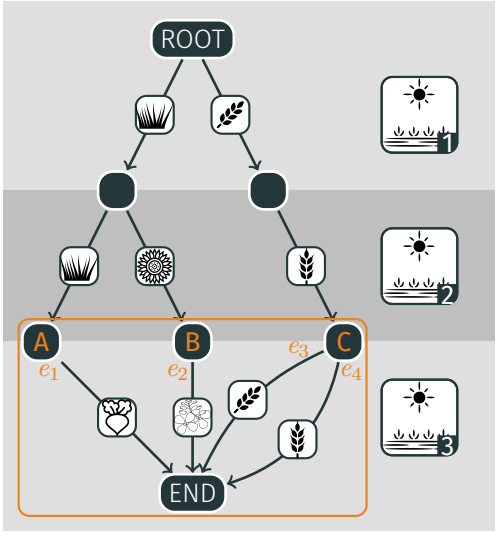


MVD

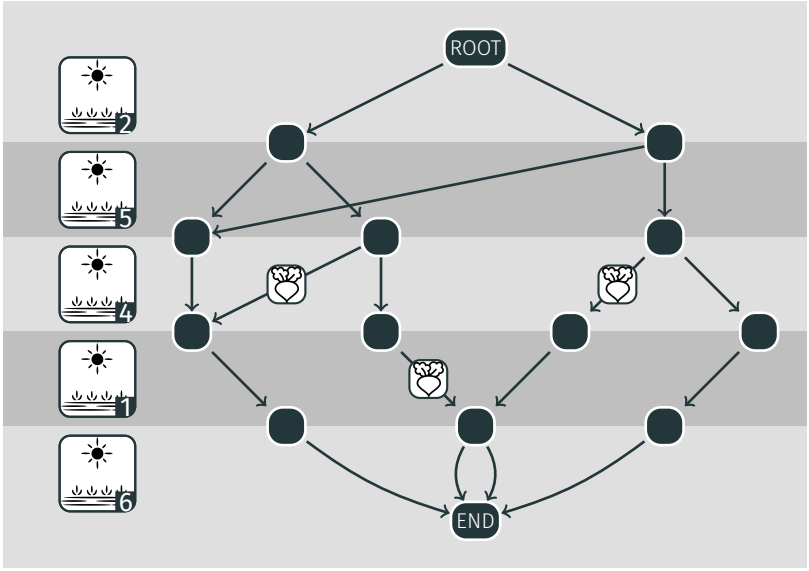


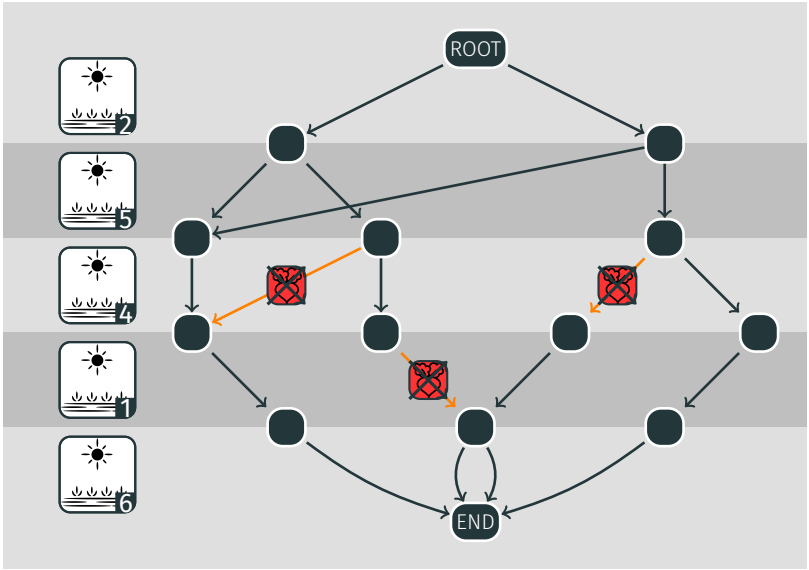
in-nd & out-nd in-nd & out-d

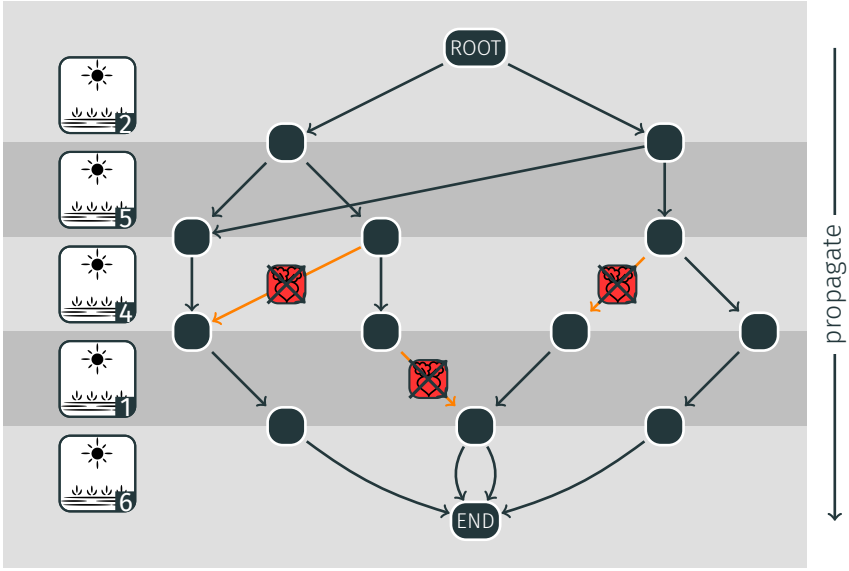

MDD
sMDD
MVD
value A
value B
ROOT
END
in-nd & out-nd
in-nd & out-d
in-d & out-nd

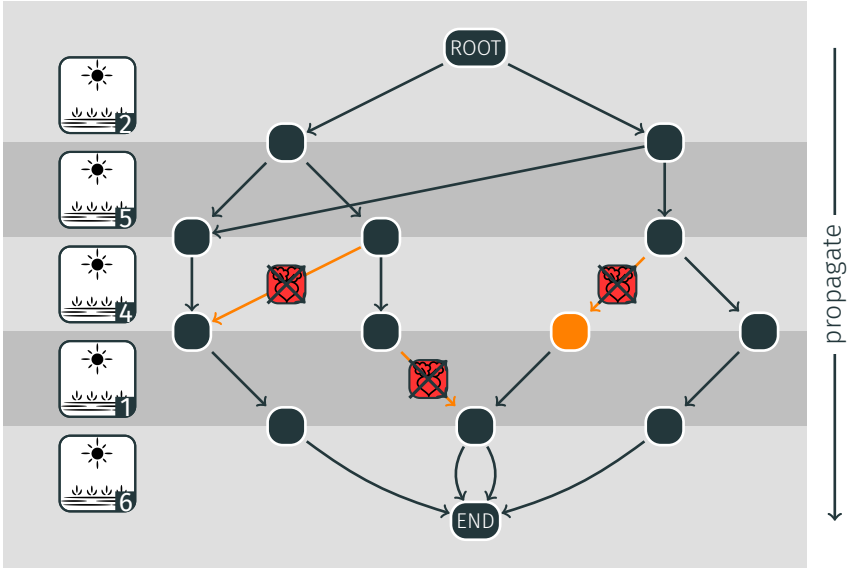


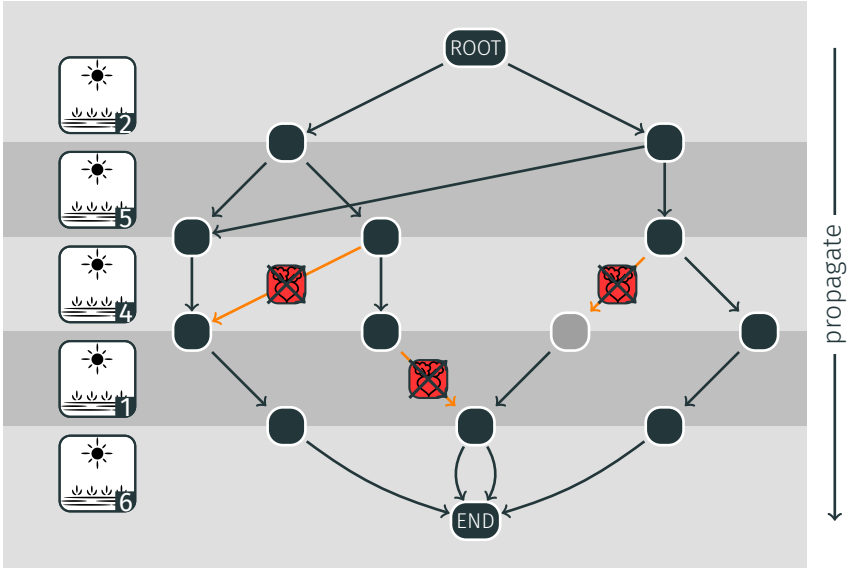
	e_1	e_2	e_3	e_4
current arc	✔	✔	✔	✔
support	✔	✘	✘	✘
support	✘	✔	✘	✘
support	✘	✘	✔	✘
support	✘	✘	✘	✔
arc out of A	✔	✘	✘	✘
arc out of B	✘	✔	✘	✘
arc out of C	✘	✘	✔	✔
arc into OUT	✔	✔	✔	✔

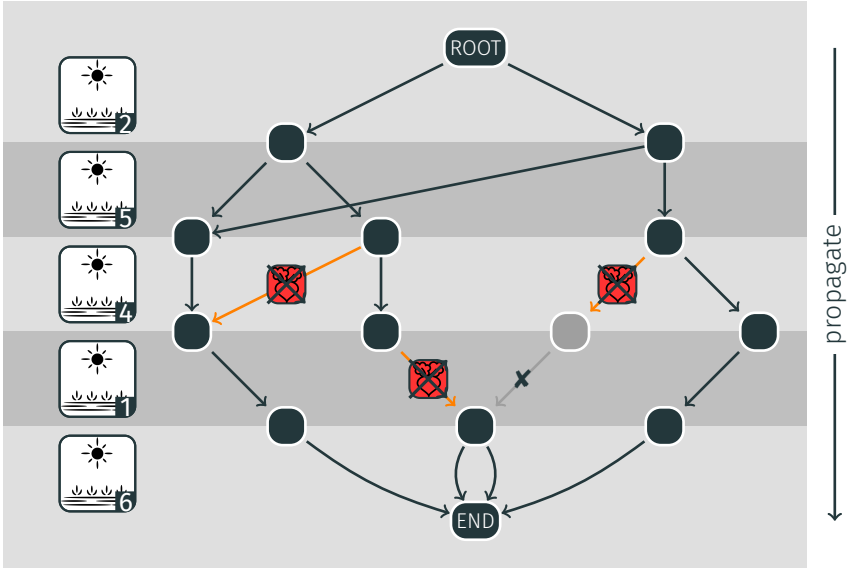

ROOT
2
5
4
1
6
END

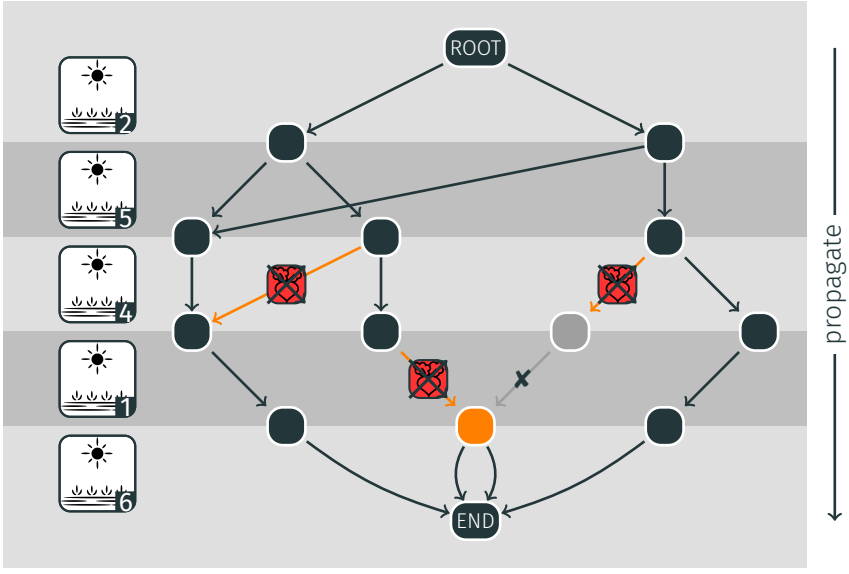

ROOT
2
5
4
1
6
END

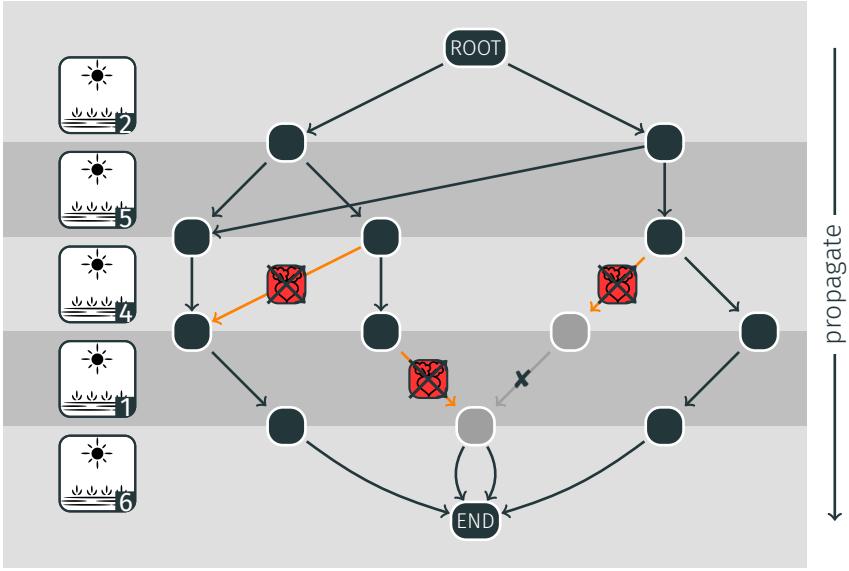

ROOT
END
2
5
4
1
6
propagate

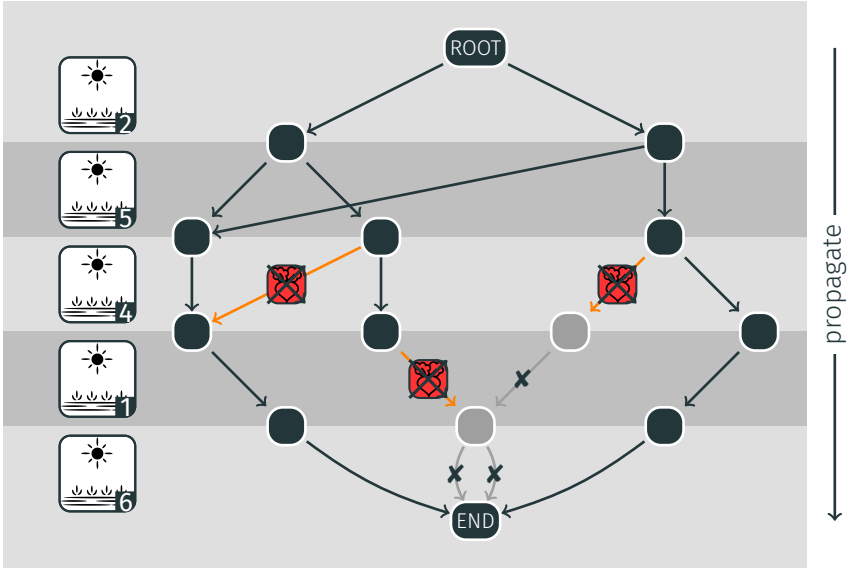

ROOT
2
5
4
1
6
END
propagate

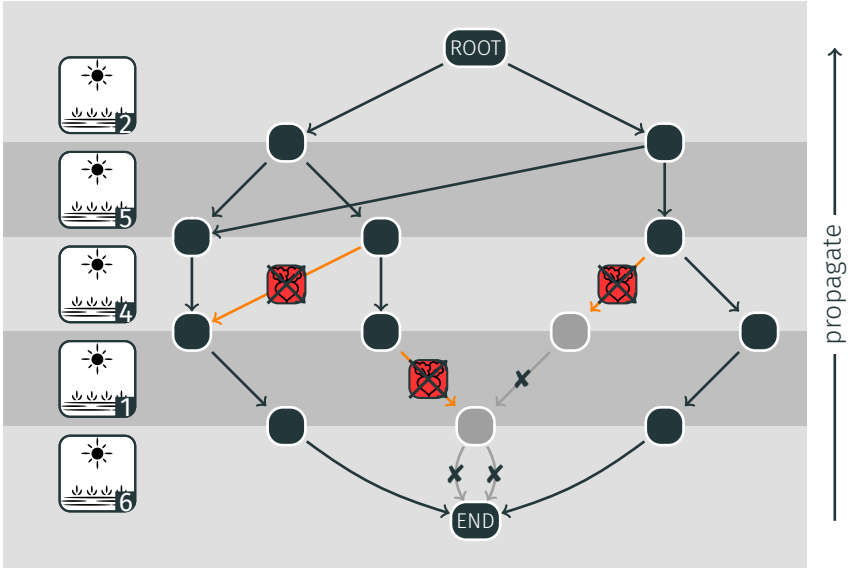

ROOT
END
2
5
4
1
6
propagate

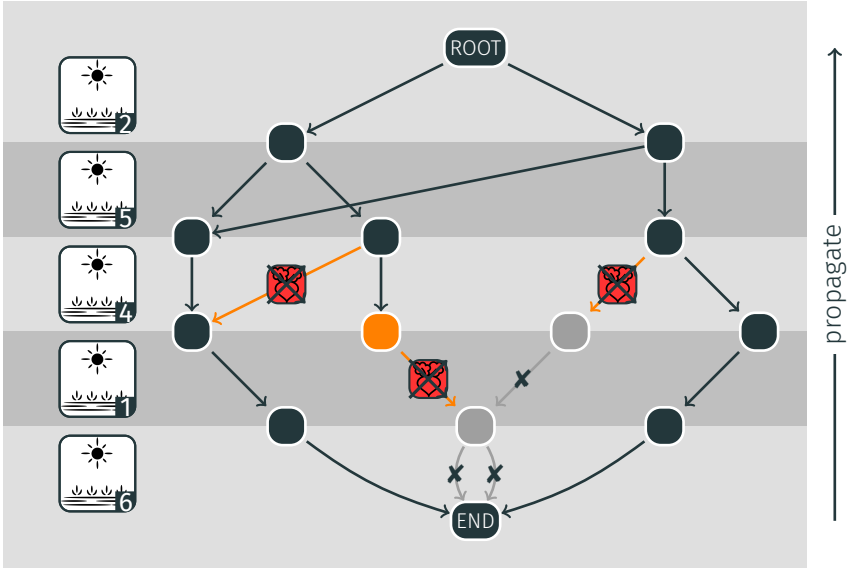

ROOT
2
5
4
1
6
END
propagate

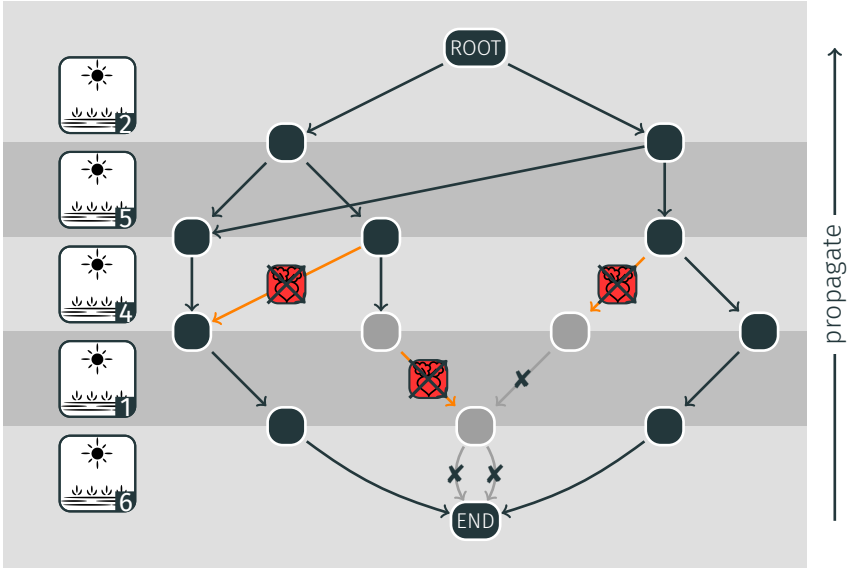

ROOT
2
5
4
1
6
END
propagate

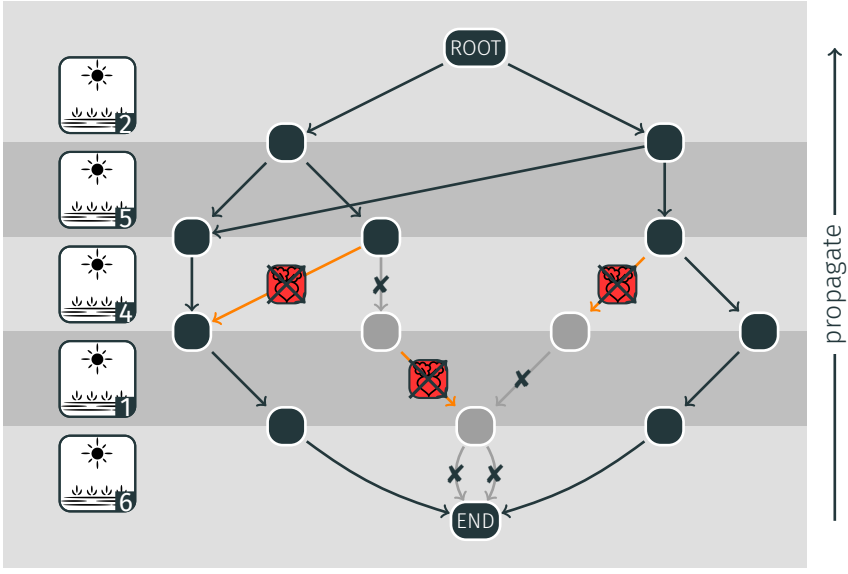

ROOT
2
5
4
1
6
END
propagate

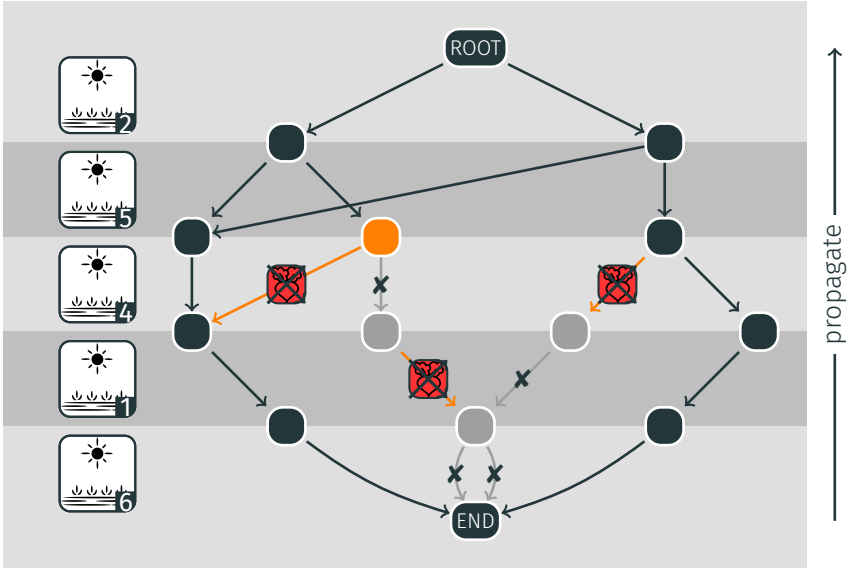

ROOT
2
5
4
1
6
END
propagate

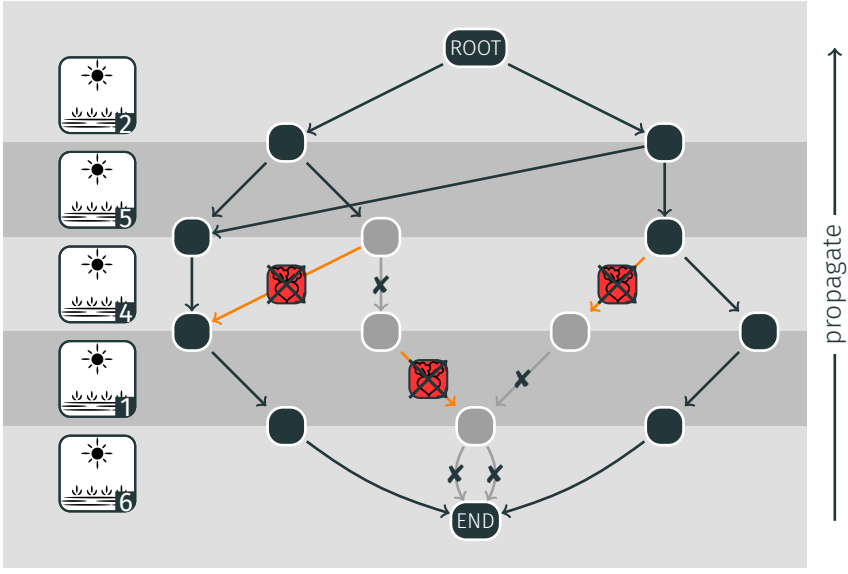

ROOT
2
5
4
1
6
END
propagate

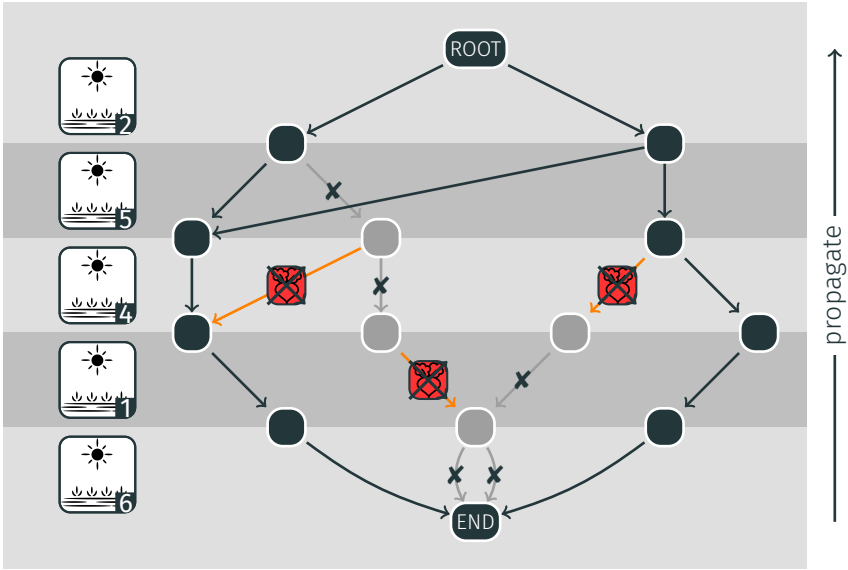

ROOT
2
5
4
1
6
END
propagate

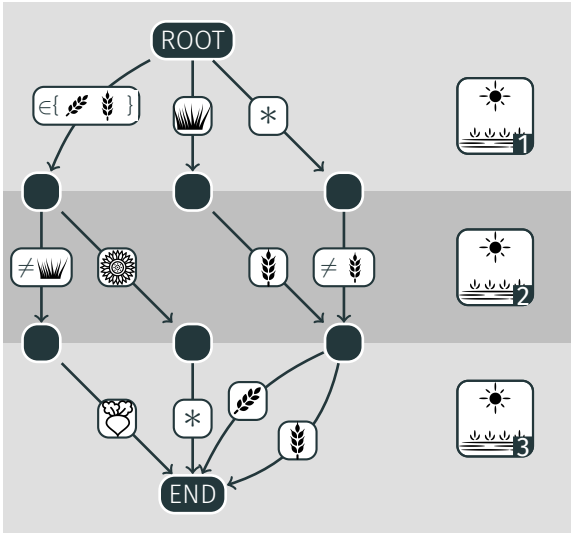

ROOT
2
5
4
1
6
END
propagate


ROOT
2
5
4
1
6
END
propagate


ROOT
2
5
4
1
6
END
propagate


ROOT
2
5
4
1
6
END
propagate


ROOT
2
5
4
1
6
END
propagate



single value

universal value

except one value

from a set of values

bounded values
(only with ordered domains)

CONCLUSION

Multiples contributions:

- a new diagram structure: `semi Multi-Valued Decision Diagram` (`sMDD`)
- a greedy compression algorithm for basic smart tables
- algorithms to create `sMDD`, `basic smart MDD`, `basic smart sMDD` and `basic smart MVD` from tables
- multiples extensions of `Compact-Table`: CT*, CT^{bs}, CT$_{neg}$, CT$^*_{neg}$, CD, CD^{bs}
- 4 publications: `AAAI17`, `CP2017`, `IJCAI18`, `CPAIOR19`

Extensional constraints are key modelization tools in CP

Good modelization tools are useless without good algorithms for the constraints

Thank you for listening!

Any questions?

Icons come from the Noun Project (`thenounproject.com`), graphists: Vineta Rendon, dojey, Rutmer Zijlstra, Becris, Eucalyp, Ben Davis, Symbolon, ffabio44, kareemovic3000, Handicon, Wan HD, Jolan Soens, Vectors Point, P Thanga Vignesh, IconMark, Alice Design