User-friendly Probabilistic Decision Logic Modeling

Simon Vandevelde

28/08/2023

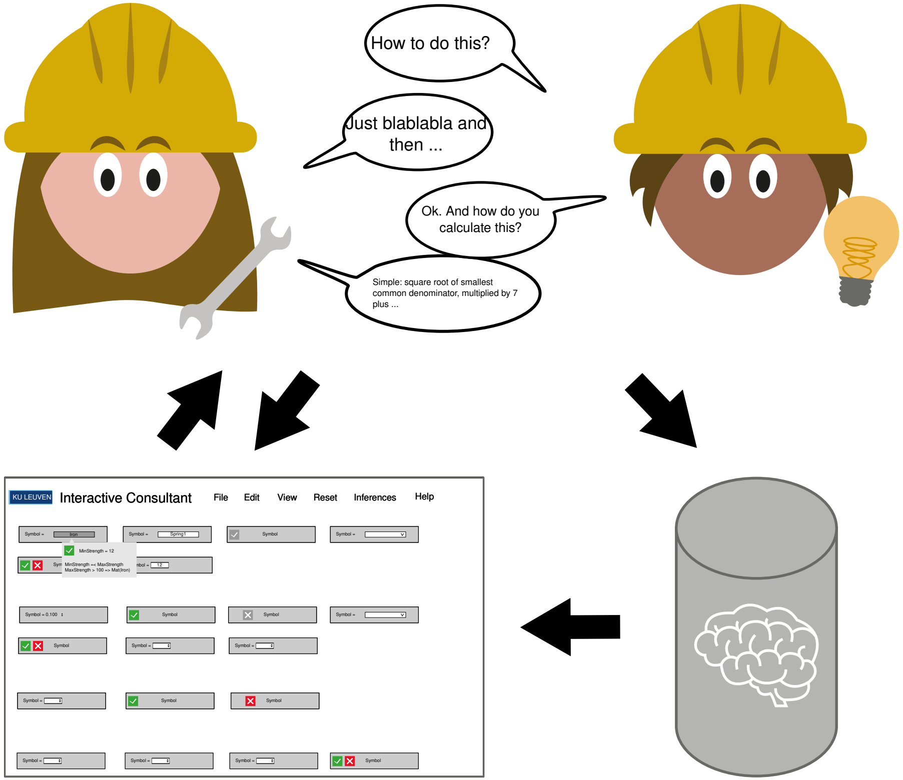
Aim of presentation

- Highlight role of uncertainty in decision making
- Show that DMN is a great foundation for user-friendly probabilistic modeling

Introduction

- Simon Vandevelde
- PhD student @ CS, KU Leuven university, Belgium
- Research on symbolic AI
- Specific focus on user-friendliness
  - How do we make knowledge-based systems more accessible?
- DMN is obviously of interest

Introduction


How to do this?
Just blablabla and then ...
Ok. And how do you calculate this?
Simple: square root of smallest common denominator, multiplied by 7 plus ...
KU LEUVEN
Interactive Consultant
File
Edit
View
Reset
Inferences
Help

Symbol = Iron

Symbol = Spring1

☑ Symbol

Symbol =

✓ MinStrength = 12

MinStrength =< MaxStrength
MaxStrength > 100 => Mat(Iron)

✓ ✗ Sym

mbol = 12

Symbol = 0.100

✓ Symbol

✗ Symbol

Symbol =

✓ ✗ Symbol

Symbol =

Symbol =

Symbol =

✓ Symbol

✗ Symbol

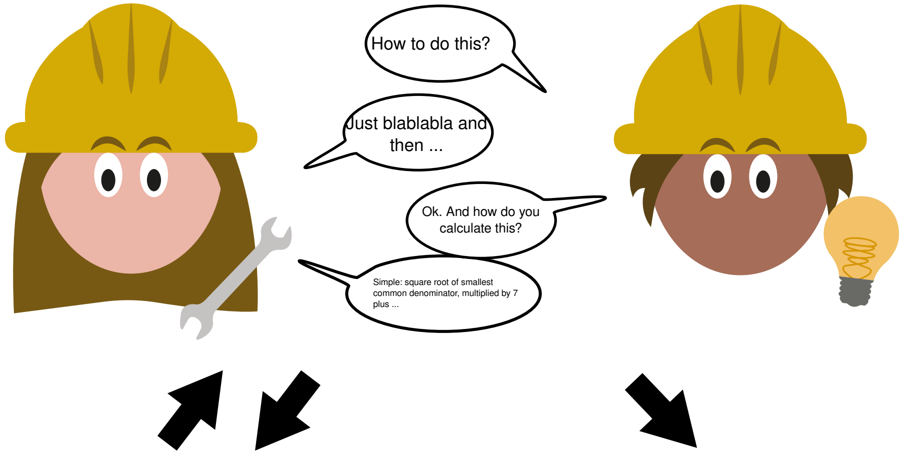
Symbol =

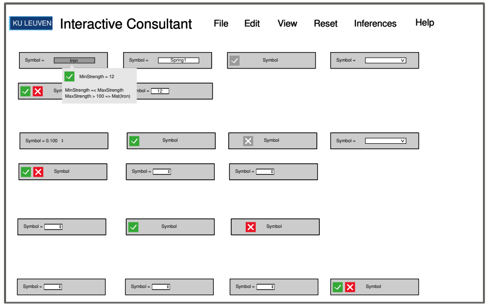
Symbol =

Symbol =

✓ ✗ Symbol

Introduction


How to do this?
Just blablabla and then ...
Ok. And how do you calculate this?
Simple: square root of smallest common denominator, multiplied by 7 plus ...



Symbol = Iron

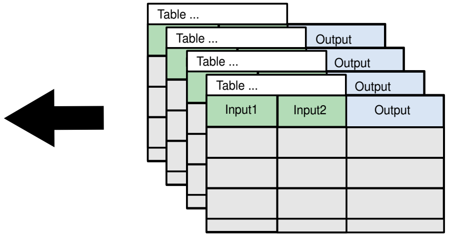
✓ MinStrength = 12

MinStrength =< MaxStrength
MaxStrength > 100 => Mat(Iron)

Symbol = Spring1

Symbol = 12

Symbol = 0.100



Input1	Input2	Output

Context

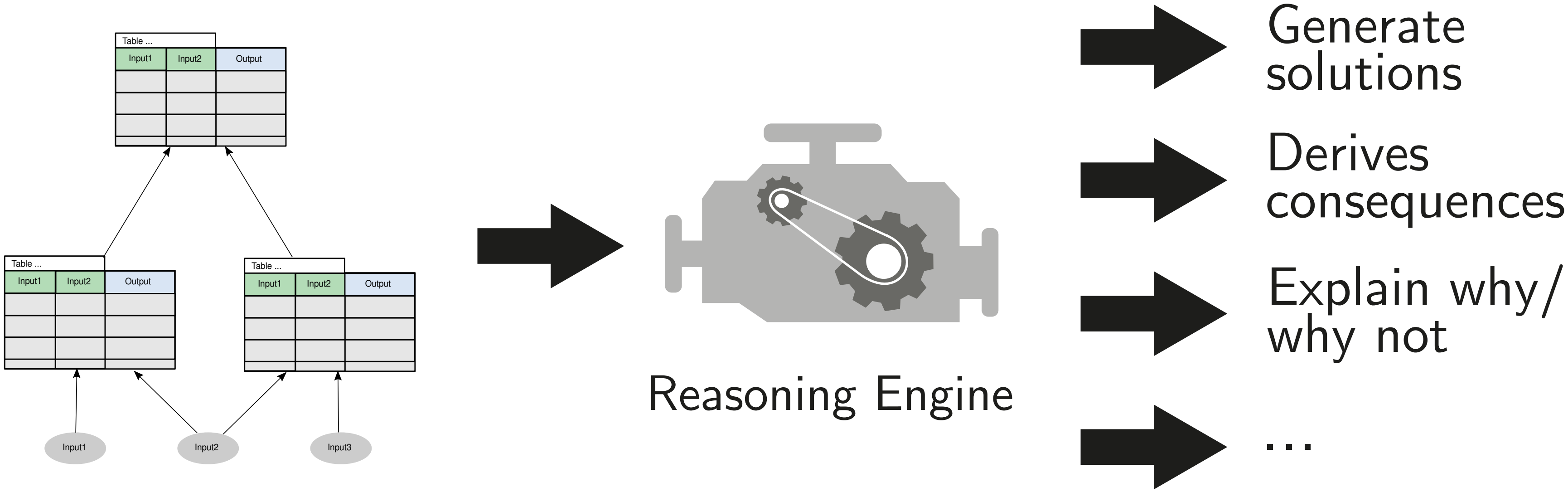
- DMN from a more theoretical viewpoint

Context

- DMN from a more theoretical viewpoint – but still practically applied

Context

- DMN from a more theoretical viewpoint – but still practically applied
- **DMN-IDP[1]:** Do more with DMN



Reasoning Engine

Generate solutions

Derives consequences

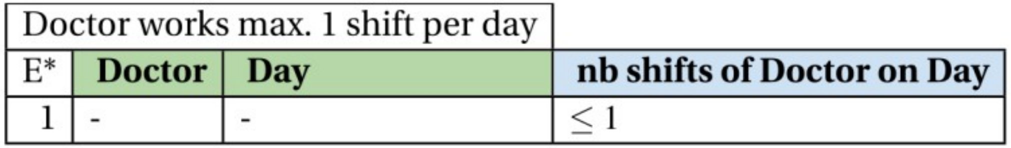
Explain why/ why not

...

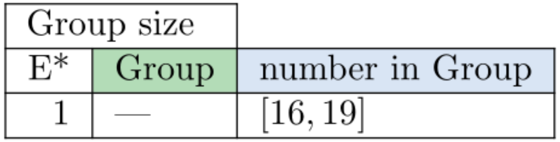
DecisionCAMP 2021: "Unlocking the full potential of DMN"

Context

- DMN from a more theoretical viewpoint – but still practically applied
- **cDMN**[2]**:** Extend DMN with constraints



Doctor works max. 1 shift per day			
E*	**Doctor**	**Day**	**nb shifts of Doctor on Day**
1	-	-	≤ 1

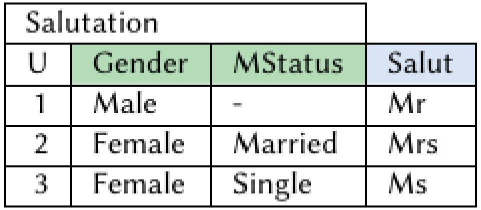


Group size		
E*	Group	number in Group
1	—	$[16, 19]$

DecisionCAMP 2020, "cDMN: Combining DMN with constraint reasoning"

Context

- DMN from a more theoretical viewpoint – but still practically applied
- How to properly handle unknown values?



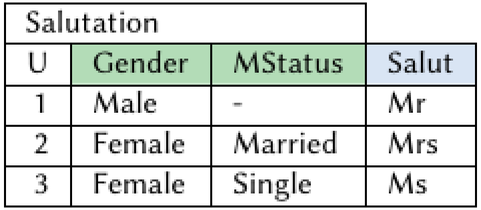
Salutation			
U	Gender	MStatus	Salut
1	Male	-	Mr
2	Female	Married	Mrs
3	Female	Single	Ms

(c) Salutation decision table

RuleML+RR, "On the Semantics of "Null" in DMN: Undefined is not Unknown", Djordje Markovic, Simon Vandevelde, Joost Vennekens, Marc Denecker
https://dmcommunity.org/challenge/challenge-aug-2016/ (Greeting a customer with unknown data)

Context

- DMN from a more theoretical viewpoint – but still practically applied
- How to properly handle unknown values?
- Undefined is not unknown!



Salutation			
U	Gender	MStatus	Salut
1	Male	-	Mr
2	Female	Married	Mrs
3	Female	Single	Ms

(c) Salutation decision table

RuleML+RR, “On the Semantics of "Null" in DMN: Undefined is not Unknown”, Djordje Markovic, Simon Vandevelde, Joost Vennekens, Marc Denecker
https://dmcommunity.org/challenge/challenge-aug-2016/ (Greeting a customer with unknown data)

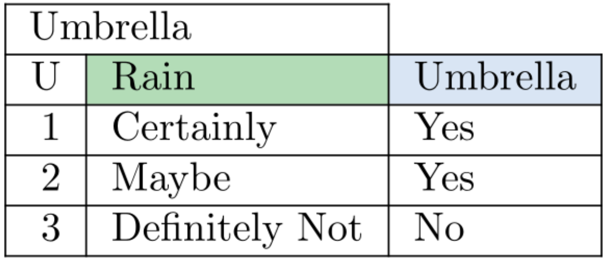
Context

- DMN from a more theoretical viewpoint – but still practically applied
- **pDMN:** how to elegantly incorporate probabilities?

Uncertainties

- Life is inherently uncertain
  - Will it rain?
  - How much traffic will there be?
  - Will the medicine be effective?
  - ...
- Massive shift to learning/predicting from data: these are all (un)certainties!

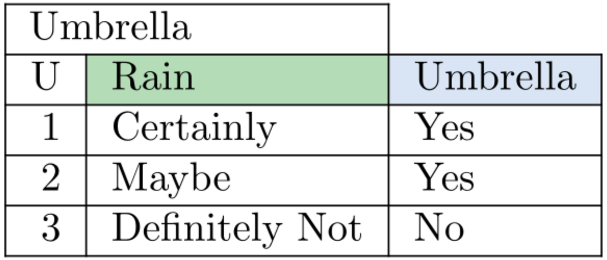
Probabilities can be “forced” into DMN



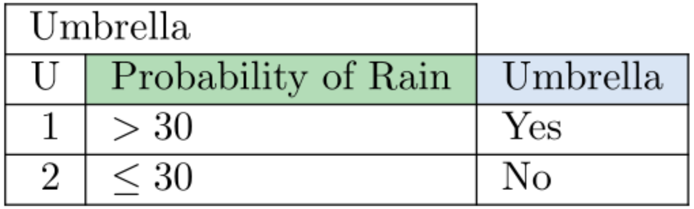
Umbrella

U	Rain	Umbrella
1	Certainly	Yes
2	Maybe	Yes
3	Definitely Not	No

Probabilities can be “forced” into DMN



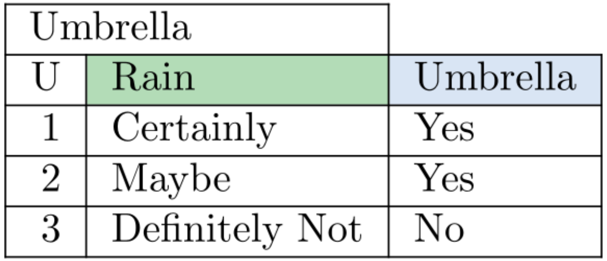
Umbrella		
U	Rain	Umbrella
1	Certainly	Yes
2	Maybe	Yes
3	Definitely Not	No



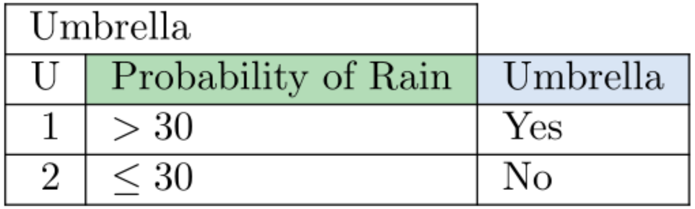
Umbrella		
U	Probability of Rain	Umbrella
1	> 30	Yes
2	≤ 30	No

Probabilities can be “forced” into DMN

- Very rigid
- Table “hides” the probability
- You cannot express probability on input value
  - E.g., a dice has a chance of 1/6 to be 1, 2, ...



Umbrella		
U	Rain	Umbrella
1	Certainly	Yes
2	Maybe	Yes
3	Definitely Not	No



Umbrella		
U	Probability of Rain	Umbrella
1	> 30	Yes
2	≤ 30	No

Goal

- Extend DMN with probabilities:
  - Elegantly, in DMN-like way
  - Presence of probability should be clear
  - Reason on them throughout entire model!
- Build an inference engine for this notation

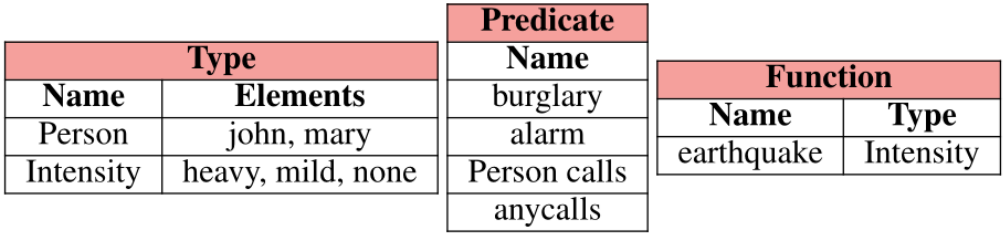
pDMN

- DMN extension for probabilistic logic
- DMN is a great foundation!
  - User-friendly, readable
  - Table-based format, very intuitive

Note: we do advocate for adding probabilities to official DMN standard, but want to point out that it forms an excellent foundation for a probabilistic notation.

pDMN: glossary

- Symbols can have arity n
- i.e., functions and predicates are possible
- Must be declared in glossary



Type	
Name	**Elements**
Person	john, mary
Intensity	heavy, mild, none

Predicate
Name
burglary
alarm
Person calls
anycalls

Function	
Name	**Type**
earthquake	Intensity

pDMN: decision tables

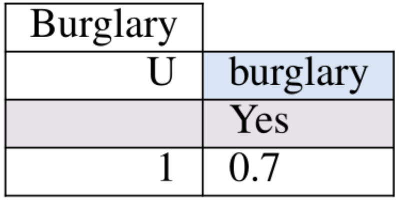
- pDMN extends decision tables with three new concepts

pDMN: decision tables

- pDMN extends decision tables with three new concepts
  - Probabilities

pDMN: decision tables

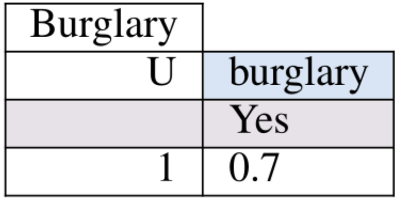
- pDMN extends decision tables with three new concepts
  - Probabilities



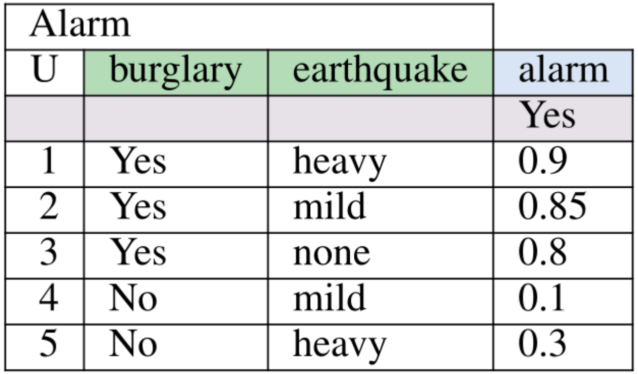
Burglary	
U	burglary
	Yes
1	0.7

pDMN: decision tables

- pDMN extends decision tables with three new concepts
  - Probabilities



Burglary	
U	burglary
	Yes
1	0.7



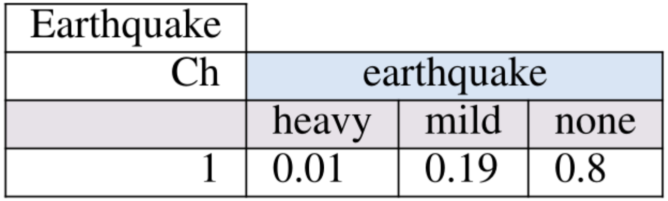
Alarm			
U	burglary	earthquake	alarm
			Yes
1	Yes	heavy	0.9
2	Yes	mild	0.85
3	Yes	none	0.8
4	No	mild	0.1
5	No	heavy	0.3

pDMN: decision tables

- pDMN extends decision tables with three new concepts
  - Probabilities
  - Ch(oice) hit policy

pDMN: decision tables

- pDMN extends decision tables with three new concepts
  - Probabilities
  - Ch(oice) hit policy

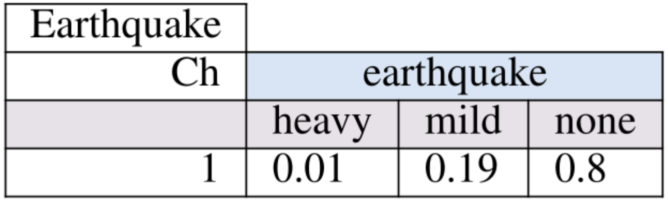


Earthquake

Ch	earthquake		
	heavy	mild	none
1	0.01	0.19	0.8

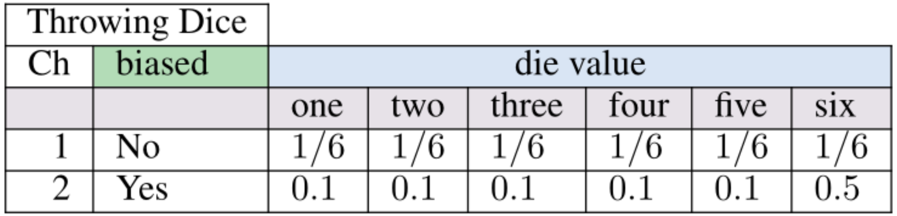
pDMN: decision tables

- pDMN extends decision tables with three new concepts
  - Probabilities
  - Ch(oice) hit policy



Earthquake

Ch	earthquake		
	heavy	mild	none
1	0.01	0.19	0.8



Throwing Dice

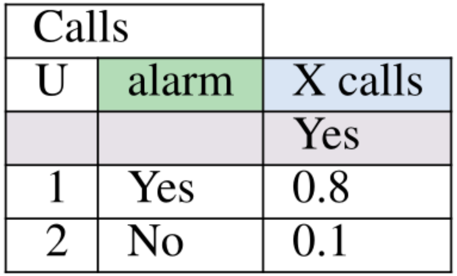
Ch	biased	die value					
		one	two	three	four	five	six
1	No	1/6	1/6	1/6	1/6	1/6	1/6
2	Yes	0.1	0.1	0.1	0.1	0.1	0.5

pDMN: decision tables

- pDMN extends decision tables with three new concepts
  - Probabilities
  - Ch(oice) hit policy
  - Quantification (for every)

pDMN: decision tables

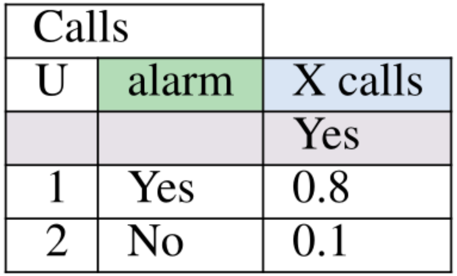
- pDMN extends decision tables with three new concepts
  - Probabilities
  - Ch(oice) hit policy
  - Quantification (for every)



Calls		
U	alarm	X calls
		Yes
1	Yes	0.8
2	No	0.1

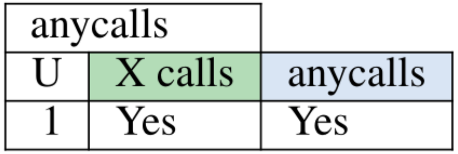
pDMN: decision tables

- pDMN extends decision tables with three new concepts
  - Probabilities
  - Ch(oice) hit policy
  - Quantification (for every)



Calls

U	alarm	X calls
		Yes
1	Yes	0.8
2	No	0.1

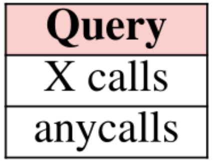


anycalls

U	X calls	anycalls
1	Yes	Yes

pDMN: query

- We want to calculate probability of a symbol
- Query table informs solver of symbols



Query
X calls
anycalls

Type	
Name	**Elements**
Person	john, mary
Intensity	heavy, mild, none

Predicate
Name
burglary
alarm
Person calls
anycalls

Function	
Name	**Type**
earthquake	Intensity

Burglary	
U	burglary
	Yes
1	0.7

Earthquake			
Ch	earthquake		
	heavy	mild	none
1	0.01	0.19	0.8

Calls		
U	alarm	X calls
		Yes
1	Yes	0.8
2	No	0.1

Alarm			
U	burglary	earthquake	alarm
			Yes
1	Yes	heavy	0.9
2	Yes	mild	0.85
3	Yes	none	0.8
4	No	mild	0.1
5	No	heavy	0.3

anycalls		
U	X calls	anycalls
1	Yes	Yes

Query
X calls
anycalls

Type	
Name	**Elements**
Person	john, mary
Intensity	heavy, mild, none

Predicate
Name
burglary
alarm
Person calls
anycalls

Function	
Name	**Type**
earthquake	Intensity

Burglary	
U	burglary
	Yes
1	0.7

Earthquake			
Ch	earthquake		
	heavy	mild	none
1	0.01	0.19	0.8

Calls		
U	alarm	X calls
		Yes
1	Yes	0.8
2	No	0.1

```
{
person_calls(mary): 0.45916,
person_calls(john): 0.45916,
anycalls: 0.550992,
}
```

Alarm			
U	burglary	earthquake	alarm
			Yes
1	Yes	heavy	0.9
2	Yes	mild	0.85
3	Yes	none	0.8
4	No	mild	0.1
5	No	heavy	0.3

anycalls		
U	X calls	anycalls
1	Yes	Yes

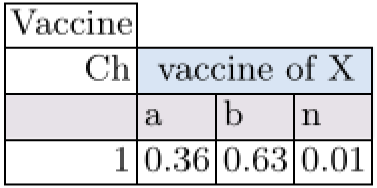
Query
X calls
anycalls

Covid example

- Someone came in contact with infected person
- You want to assess if they had high-risk contact

Covid example

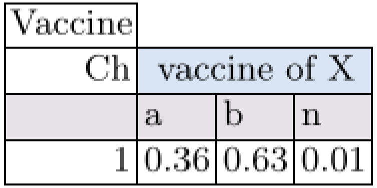
- Someone came in contact with infected person
- You want to assess if they had high-risk contact



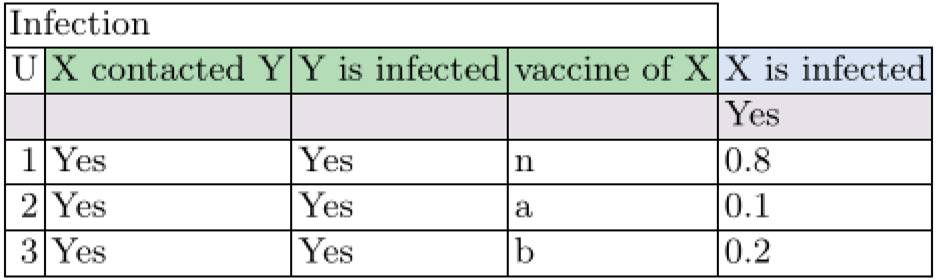
Vaccine			
Ch	vaccine of X		
	a	b	n
1	0.36	0.63	0.01

Covid example

- Someone came in contact with infected person
- You want to assess if they had high-risk contact



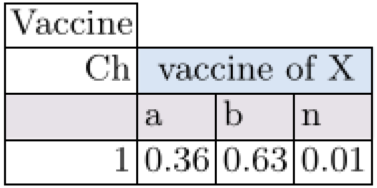
Vaccine			
Ch	vaccine of X		
	a	b	n
1	0.36	0.63	0.01



Infection				
U	X contacted Y	Y is infected	vaccine of X	X is infected
				Yes
1	Yes	Yes	n	0.8
2	Yes	Yes	a	0.1
3	Yes	Yes	b	0.2

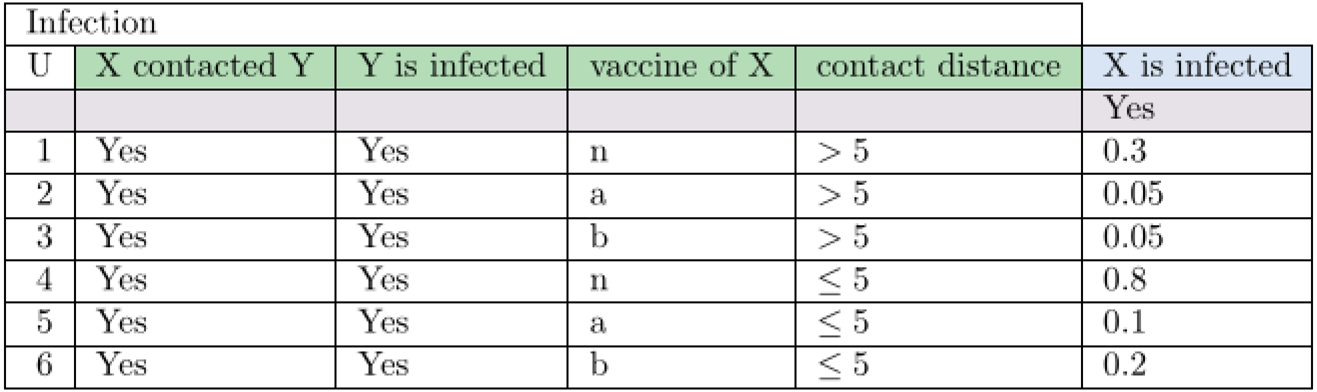
Covid example

- Someone came in contact with infected person
- You want to assess if they had high-risk contact



Vaccine

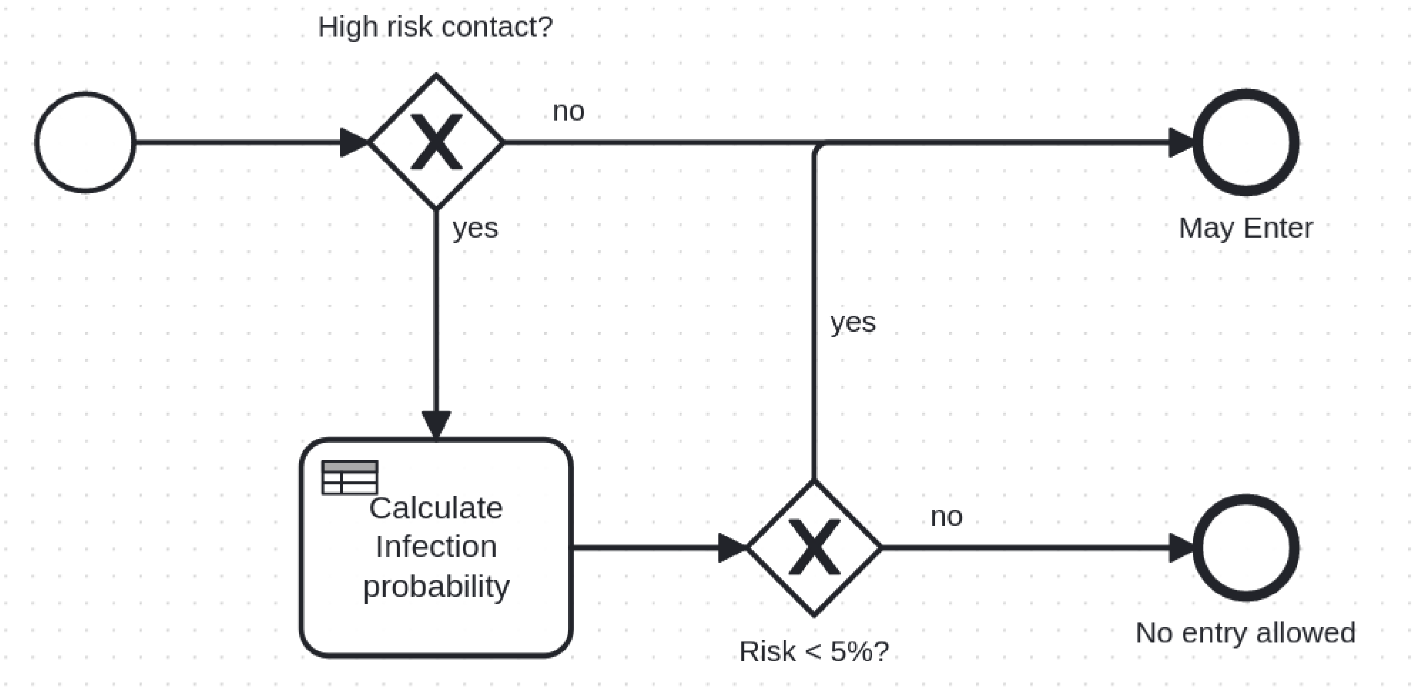
Ch	vaccine of X		
	a	b	n
1	0.36	0.63	0.01



Infection

U	X contacted Y	Y is infected	vaccine of X	contact distance	X is infected
					Yes
1	Yes	Yes	n	> 5	0.3
2	Yes	Yes	a	> 5	0.05
3	Yes	Yes	b	> 5	0.05
4	Yes	Yes	n	≤ 5	0.8
5	Yes	Yes	a	≤ 5	0.1
6	Yes	Yes	b	≤ 5	0.2

Covid example



pDMN engine

Uses ProbLog internally:

- pDMN tables are translated to ProbLog
- ProbLog calculates the probabilities

pDMN engine

Uses ProbLog internally:

- pDMN tables are translated to ProbLog
- ProbLog calculates the probabilities

```
% Vaccine
0.36::vaccine_of_Person(X, a);0.63::vaccine_of_Person(X, b);0.01::vaccine_of_Person(X, n) :- person(X).


% Infection
0.8::person_is_infected(X) :- person_contacted_Person(X, Y), person_is_infected(Y), vaccine_of_Person(X, n), person(X), person(Y).
0.1::person_is_infected(X) :- person_contacted_Person(X, Y), person_is_infected(Y), vaccine_of_Person(X, a), person(X), person(Y).
0.2::person_is_infected(X) :- person_contacted_Person(X, Y), person_is_infected(Y), vaccine_of_Person(X, b), person(X), person(Y).
```

Thank you

More info:

- https://cdmn.readthedocs.io/en/latest/pdmn.html
- https://gitlab.com/EAVISE/cdmn/pdmn
- Vandevelde, S., Verreet, V., De Raedt, L., & Vennekens, J. (2021). A table-based representation for probabilistic logic: Preliminary results. ArXiv Preprint ArXiv:2110.01909.
- www.simonvandevelde.be