Logic Engine

ITERATA EXPLAINS

A CRITICAL APPROACH TOWARDS THE SUBJECT

Logic Engine – LE

Logic Engine → the idea

The idea is the collection of a suitable preselection for specific research questions.

- The aim is to miss as little cases as possible. Not only the requested data is shown in the results, but also false-positive results are included in the preselection. A domain-expert decides the final case-set by looking at the results.
- Desired are only the “correct” data in the case-set and only a few false-positive cases. The domain expert checks the results to present a selection of patients e.g. to present it to a physician for a study.

- Preselection from BigData (All Data Container)
- Not all data is retrieved if not needed for the research question (e.g. only data about individuals >75 yrs)

The LogicEngine supports with logical evaluation and criteria the selection of data which can be used for further analysis

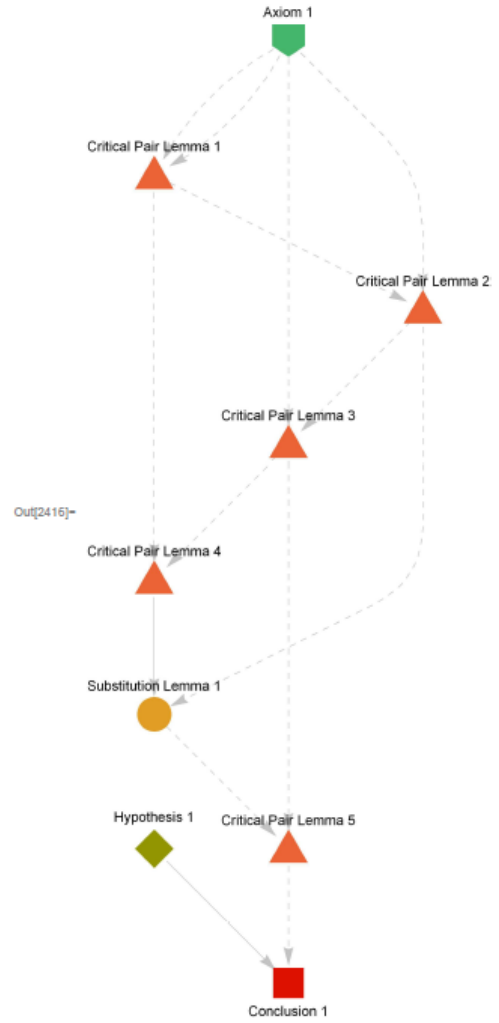
- Selection criteria: patients >75yrs, patients with specific diagnoses, medication, lab values, time periods, reports
- Connection of different criteria with AND-linkage: false if statement is false
- Connection of different criteria with OR-linkage: true if statement is true
- Implication: true if second statement is false but first one is true

Logic Engine – LE

Mathematica

In[2416]:=

```
proof["ProofGraph"]
```

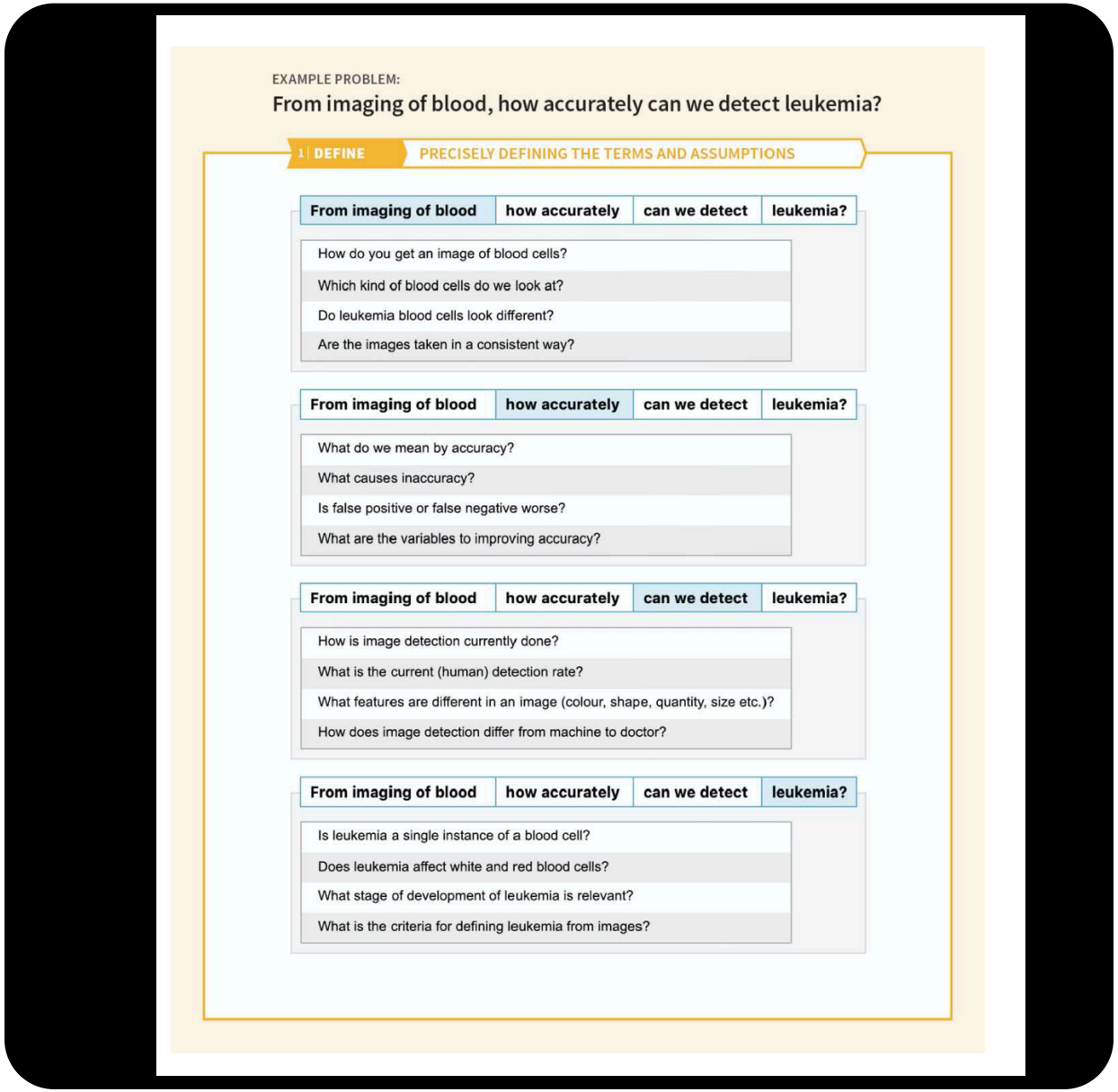


		Statement	Proof	
Axiom	1	`x = m[y, m[z, m[x, m[y, z]]]]`		
Hypothesis	1	`m[a, a] = m[b, b]`		
CriticalPairLemma	1	`x = m[y, m[m[z, m[x, y]], z]]`	Construct	{Axiom, 1}
			Orientation	-1
			9 total ›	
CriticalPairLemma	2	`m[m[x, y], m[z, y]] = m[x, z]`	Construct	{Axiom, 1}
			Orientation	-1
			9 total ›	
CriticalPairLemma	3	`m[x, y] = m[z, m[y, m[x, z]]]`	Construct	{Axiom, 1}
			Orientation	-1
			9 total ›	
CriticalPairLemma	4	`m[x, m[m[x, y], m[z, y]]] = z`	Construct	{CriticalPairLemma, 3}
			Orientation	-1
			9 total ›	
SubstitutionLemma	1	`m[x, m[x, y]] = y`	Input	{CriticalPairLemma, 4}
			Position	{1, 2}
			6 total ›	
CriticalPairLemma	5	`m[x, x] = m[y, y]`	Construct	{CriticalPairLemma, 3}
			Orientation	-1
			9 total ›	
Conclusion	1	True	Input	{Hypothesis, 1}
			Position	1
			6 total ›	

Logic Engine (LE) - Mathematica

The Maths fix

A first gold nugget



Logic Engine (LE) - Mathematica

EXAMPLE PROBLEM:

From imaging of blood, how accurately can we detect leukemia?

2 | ABSTRACT GETTING TO THE ABSTRACT FORM READY FOR COMPUTATION

Can we precisely define the features in the image that will indicate leukemia?

No → Use a machine learning approach

Yes → Use an algorithm to process the images

Do we have examples of leukemia cells?

No → Use unsupervised learning

Yes → Use supervised learning with set of training data

Construct computable form (in Wolfram Language using Classify)

```
In[•]:= Classify[
  <|"Leukemia Cells" → {<image>, <image>, <image>, <image>, <image>, <image>, <image>, <image>, ... ...},
  "Not Leukemia Cells" → {<image>, <image>, <image>, <image>, <image>, <image>, <image>, <image>, ... ...}|>]
```

Use Classify[ ] with images as input to automatically train a neural net to classify other images

EXAMPLE PROBLEM:

From imaging of blood, how accurately can we detect leukemia?

3 | COMPUTE EVALUATE TO BUILD THE CLASSIFIER FUNCTION

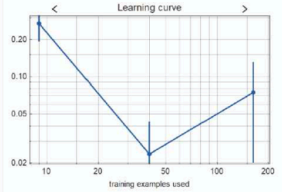
A) Run the classifier

```
In[•]:= testCell = Classify[
  <|"Leukemia Cells" → {<image>, <image>, <image>, <image>, <image>, <image>, <image>, <image>, ... ...},
  "Not Leukemia Cells" → {<image>, <image>, <image>, <image>, <image>, <image>, <image>, <image>, ... ...}|>]
```

Classify Progress

Time elapsed	56s
Training examples used	163 / 204 (80%)
Current best method	LogisticRegression
Current loss	0.0236

Learning curve



Stop

```
Out[•]= ClassifierFunction[ Input type: Image
                            Classes: Leukemia Cells, Not Leukemia Cells ]
```

B) Test the classifier on other images

```
In[•]:= testCell[<image>]
Out[•]= Not Leukemia Cells
```

```
In[•]:= testCell[<image>]
Out[•]= Leukemia Cells
```

Logic Engine (LE) - Mathematica

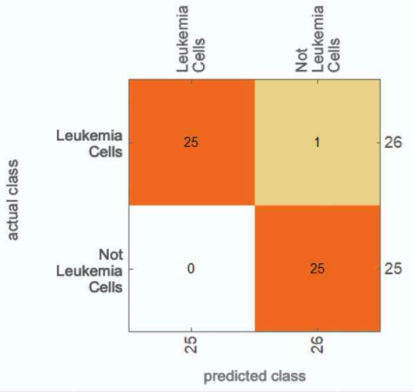
EXAMPLE PROBLEM:

From imaging of blood, how accurately can we detect leukemia?

4 | INTERPRET QUANTIFY AND QUALIFY RELIABILITY OF RESULTS

Classifier Measurements

Number of test examples	51
Accuracy	(98.0 ± 2.0)%
Accuracy baseline	(51. ± 7.)%
Geometric mean of probabilities	0.976 ± 0.024
Mean cross entropy	0.0248 ± 0.025
Single evaluation time	192. ms/example
Batch evaluation speed	4.95 examples/s
Rejection rate	0%



Apply ClassifierMeasurement[ ] function to gauge accuracy of results

Key Interpretation Options

How accurate was its performance?
What are the consequences of false positives? False negatives?
Are the detection rates better than a doctor's?
Is it good enough to supplement or even replace the doctor?
How sensitive is the classifier to image quality? To the amount of images?