Computational Argumentation & AI at the Dutch Police

Methods in AI research

Floris Bex

Universiteit Utrecht

Previously on MAIR…

- AI is Sense - Reason – Act

- Building a dialogue system
  - Sense: what did the user say, which speech act did they use? Which preferences did the user express?
  - Reason: Determine dialogue state, perform restaurant lookup
  - Act: recommend a restaurant

Previously on MAIR…

- AI is Sense - Reason – Act

- Building a dialogue system
  - Sense: NLP (machine learning, keyword match)
  - Reason: state machine, lookup
  - Act: NLG (template)

Previously on MAIR…

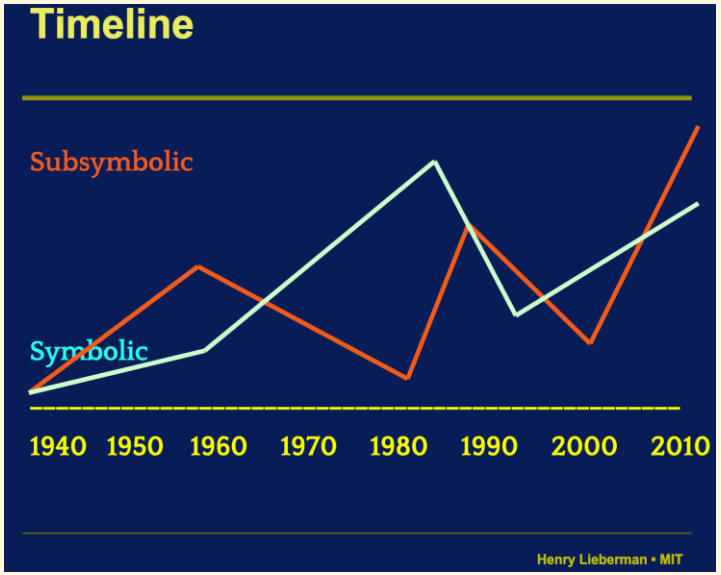
- The reasoning in the dialogue agent is fairly simple
  - Only "flat knowledge", no relations (e.g. there are different types of Mediterranean food: Italian, Spanish,…)

- More complex reasoning: logic in AI
  - e.g. $mediterranean \equiv italian \sqcup greek \sqcup spanish \sqcup turkish$
  - First-order logic, propositional logic, description logic

- FOL & DL are monotonic, absolute
  - Fine for many domains, maybe not for legal/police domain

Previously on MAIR…

- Symbolic vs. subsymbolic



- Combining symbolic with subsymbolic
  - Boxology of Design Patterns for Hybrid Learning and Reasoning Systems - van Harmelen & ten Teije, 2019

This lecture

- Computational Argumentation
  - Nonmonotonic logic & reasoning
- AI at the Dutch Police
  - Automated intake of criminal reports by citizens
  - Combining symbolic & subsymbolic
    - NLP, argumentation, dialogues

Nonmonotonic Reasoning

- Knowledge is often uncertain or incomplete
  - jump to conclusions under certain assumptions;
  - retract conclusions once they learn that an assumption was unwarranted.

- Nonmonotonic logics define consequence notions for reasoning in which information that is at one point accepted as true can at a later point be rejected.
  - **Exceptions:** If it is scheduled, there will be a lecture, unless the lecturer is ill.

Nonmonotonic reasoning

- Contradictory information
  - $p \wedge \neg p \vdash \bot$
  - principle of explosion: $\forall p \forall q : p \wedge \neg p \vdash q$

- Why not with "classical" propositional logic?
  - $Scheduled \rightarrow Lecture$ Default rule
  - $Scheduled \wedge Ill \rightarrow \neg Lecture$ Exception
  - $Scheduled \wedge Ill$ Inconsistent!

  - $Scheduled \wedge \neg Ill \rightarrow Lecture$ Default rule
  - $Scheduled \wedge Ill \rightarrow \neg Lecture$ Exception
  - $Scheduled$ Does not imply Lecture!

Argumentation

- Providing reasons for one's claims
- Giving counterarguments to claims

- Argumentation provides principled techniques for resolving inconsistency.
  - Or at least, sensible rules for deciding what to believe in the face of inconsistency.

Structured Arguments

- Arguments are reasons for conclusions
- Infer conclusions from premises using inference rules
- Example: classical logical proof
  - p, $p \rightarrow q$
  - q (Modus Ponens)

  - $p \wedge q$
  - p ($\wedge$-Elim)

Syntax of arguments - ASPIC+

- Arguments are trees
  - Nodes are wff of a logical language $\mathcal{L}$
  - Links are applications of inference rules
    - $\mathcal{R}_s$ = Strict rules (ϕ_1, ..., $\phi_n \rightarrow \phi$); or
    - $\mathcal{R}_d$= Defeasible rules (ϕ_1, ..., $\phi_n \Rightarrow \phi$)
  - Reasoning starts from a knowledge base $\mathcal{K} \subseteq \mathcal{L}$

ASPIC+ Argumentation systems (with symmetric negation)

- An argumentation system is a triple $AS = (\mathcal{L}, \mathcal{R}, n)$ where:
  - $\mathcal{L}$ is a logical language with negation ($\neg$)
  - $\mathcal{R} = \mathcal{R}_s \cup \mathcal{R}_d$ is a set of strict ($\phi_1, \ldots, \phi_n \rightarrow \phi$) and defeasible ($\phi_1, \ldots, \phi_n \Rightarrow \phi$) inference rules
  - n: $\mathcal{R}_d \rightarrow \mathcal{L}$ is a naming convention for defeasible rules

- An argumentation theory is a pair $AT = (AS, \mathcal{K})$ where AS is an argumentation system and $\mathcal{K}$ a knowledge base in AS.
  - A knowledge base in $AS = (\mathcal{L}, \mathcal{R}, n)$ is a set $\mathcal{K} \subseteq \mathcal{L}$ where $\mathcal{K}$ is a partition $\mathcal{K}_n \cup \mathcal{K}_p$ with:
    - $\mathcal{K}_n$ = necessary premises (axioms)
    - $\mathcal{K}_p$ = ordinary premises ("assumptions")

Arguments

- An argument A on the basis of an argumentation theory is:
  - ϕ if $\phi \in \mathcal{K}$ with
    - $\text{Prem}(A) = \{\phi\}$, $\text{Conc}(A) = \phi$, $\text{Sub}(A) = \{\phi\}$, $\text{DefRules}(A) = \varnothing$
  - $A_1, \ldots, A_n \rightarrow \phi$ if $A_1, \ldots, A_n$ are arguments such that there is a strict inference rule $\text{Conc}(A_1), \ldots, \text{Conc}(A_n) \rightarrow \phi$
    - $\text{Prem}(A) = \text{Prem}(A_1) \cup \ldots \cup \text{Prem}(A_n)$
    - $\text{Conc}(A) = \phi$
  - $A_1, \ldots, A_n \Rightarrow \phi$ if $A_1, \ldots, A_n$ are arguments such that there is a defeasible inference rule $\text{Conc}(A_1), \ldots, \text{Conc}(A_n) \Rightarrow \phi$
    - $\text{Prem}(A) = \text{Prem}(A_1) \cup \ldots \cup \text{Prem}(A_n)$
    - $\text{Conc}(A) = \phi$

Structured Arguments

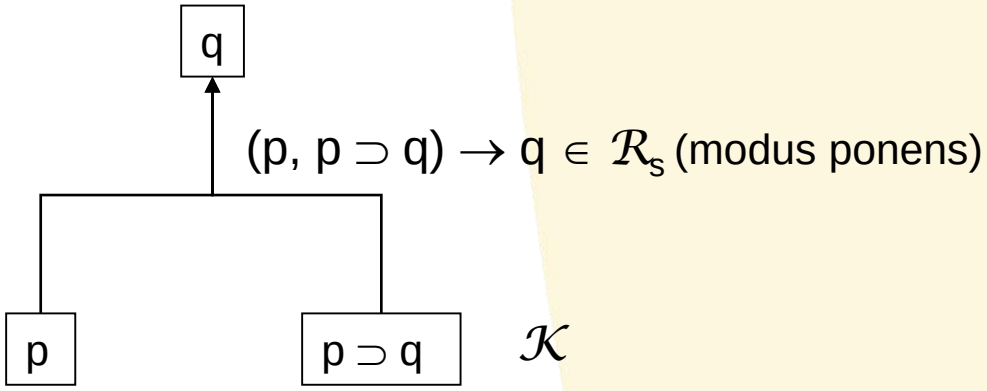
- Classical logical proof trees

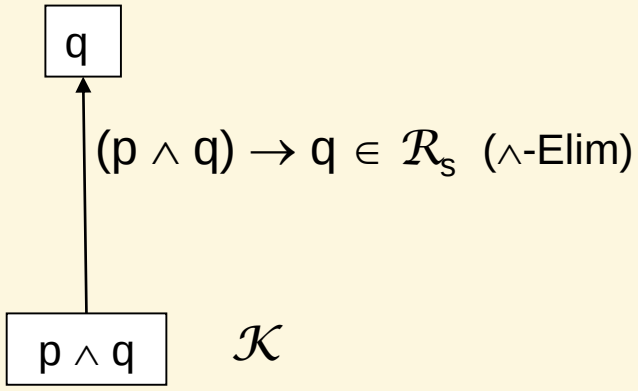
$\boxed{p}$ $\qquad$ $\dfrac{\boxed{p \supset q} \quad \mathcal{K}}{}$ $\qquad$ $\dfrac{\boxed{p \wedge q} \quad \mathcal{K}}{}$

- ($\supset$ is the classical logic implication $\rightarrow$)

Structured Arguments

- Classical logical proof trees





Structured Arguments

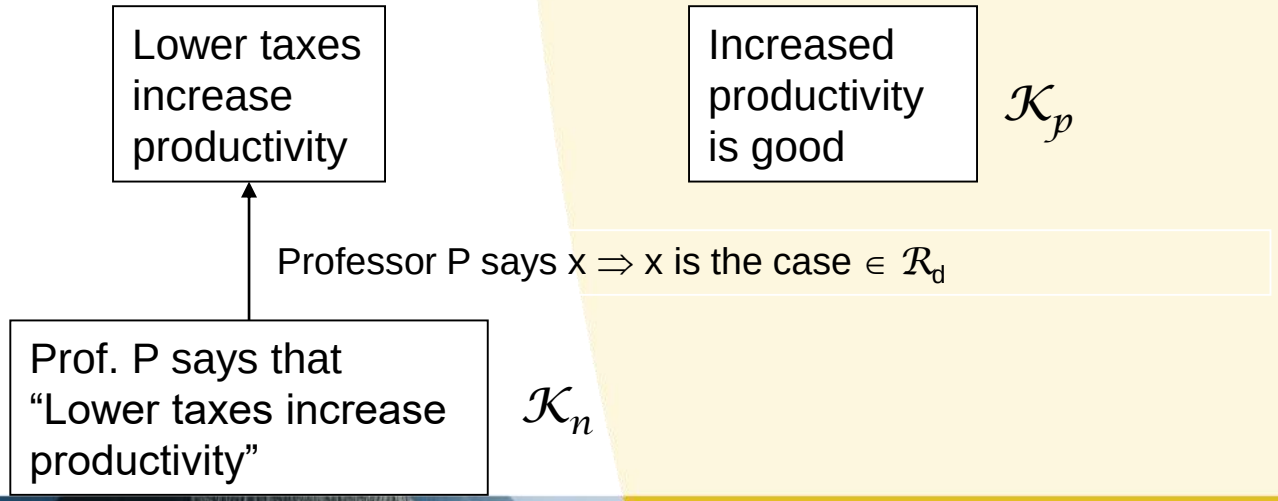
- Defeasible arguments

Increased productivity is good $\mathcal{K}_p$

Prof. P says that "Lower taxes increase productivity" $\mathcal{K}_n$

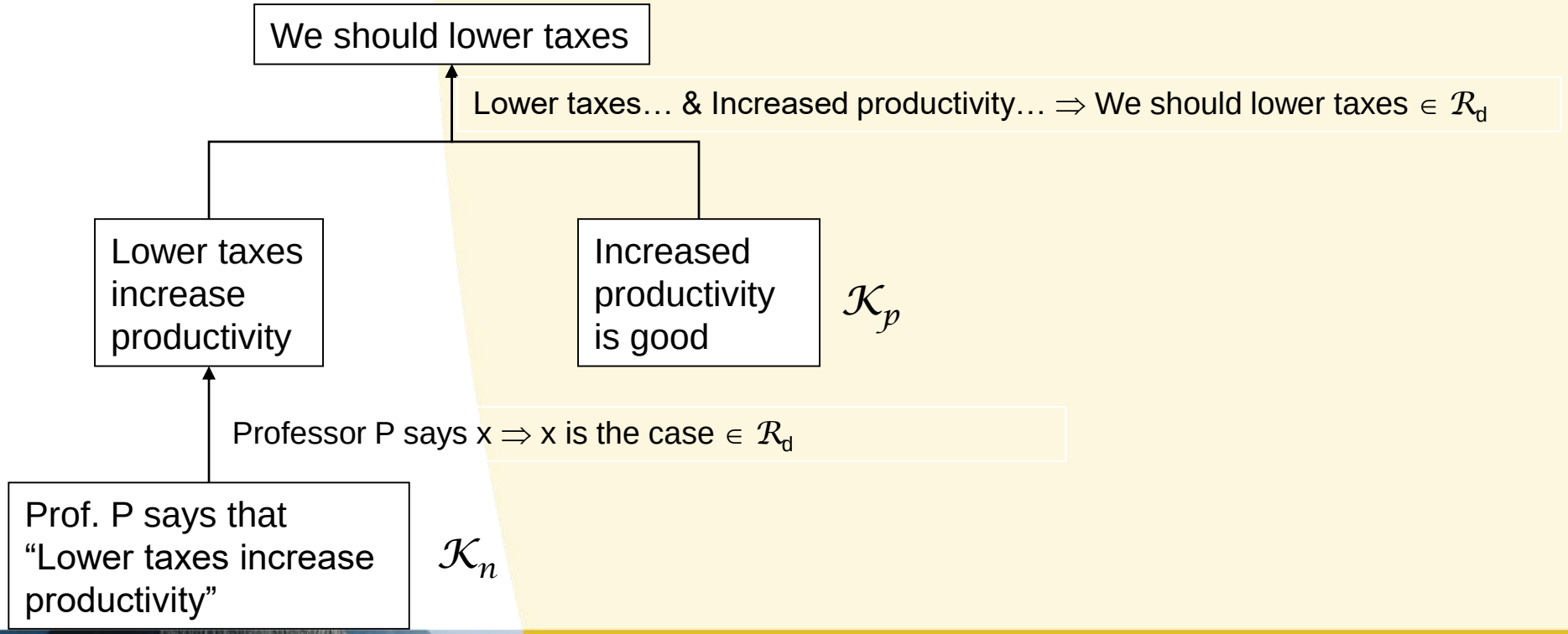
Structured Arguments

- Defeasible arguments



Structured Arguments

- Defeasible arguments



Rules

- Rules are based on typical rules or generalizations we use in everyday reasoning
  - Birds can (usually) fly
  - if expert *E* says *P*, we can usually believe *P*
  - If action A causes G and G is good (bad), then we should (not) do A

Rules and exceptions

- Birds can (usually) fly
  - Except penguins or ostriches!
- If expert *E* says *P*, we can usually believe *P*
  - Unless the expert is biased, or *E* is not an expert in the domain that *P* is in
- If action A causes G and G is good (bad), then we should (not) do A
  - Unless the circumstances are such that A does not cause G, or that we cannot reasonably perform A

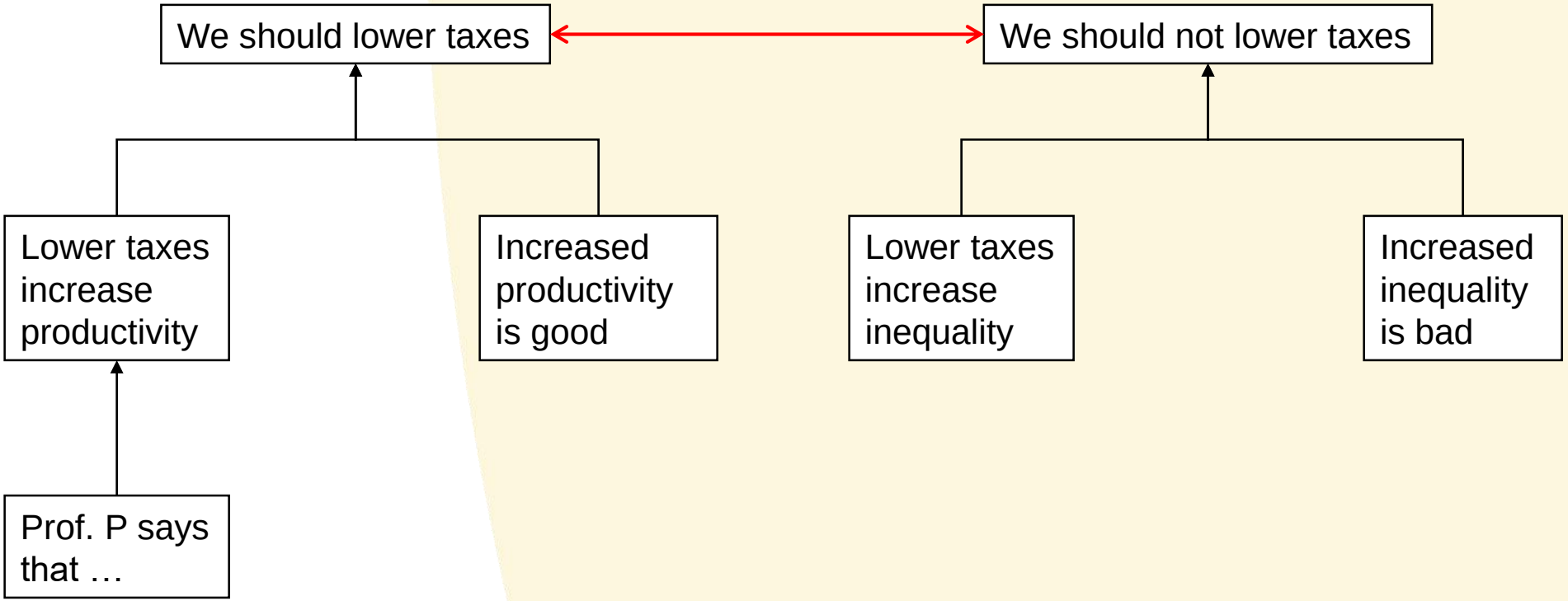
Counterarguments

- Arguments can *attack* each other
- Undercutting: providing an exception to the rule
  - Attack the inference
- Undermining
  - Attack the premise
- Rebutting
  - Attack a conclusion

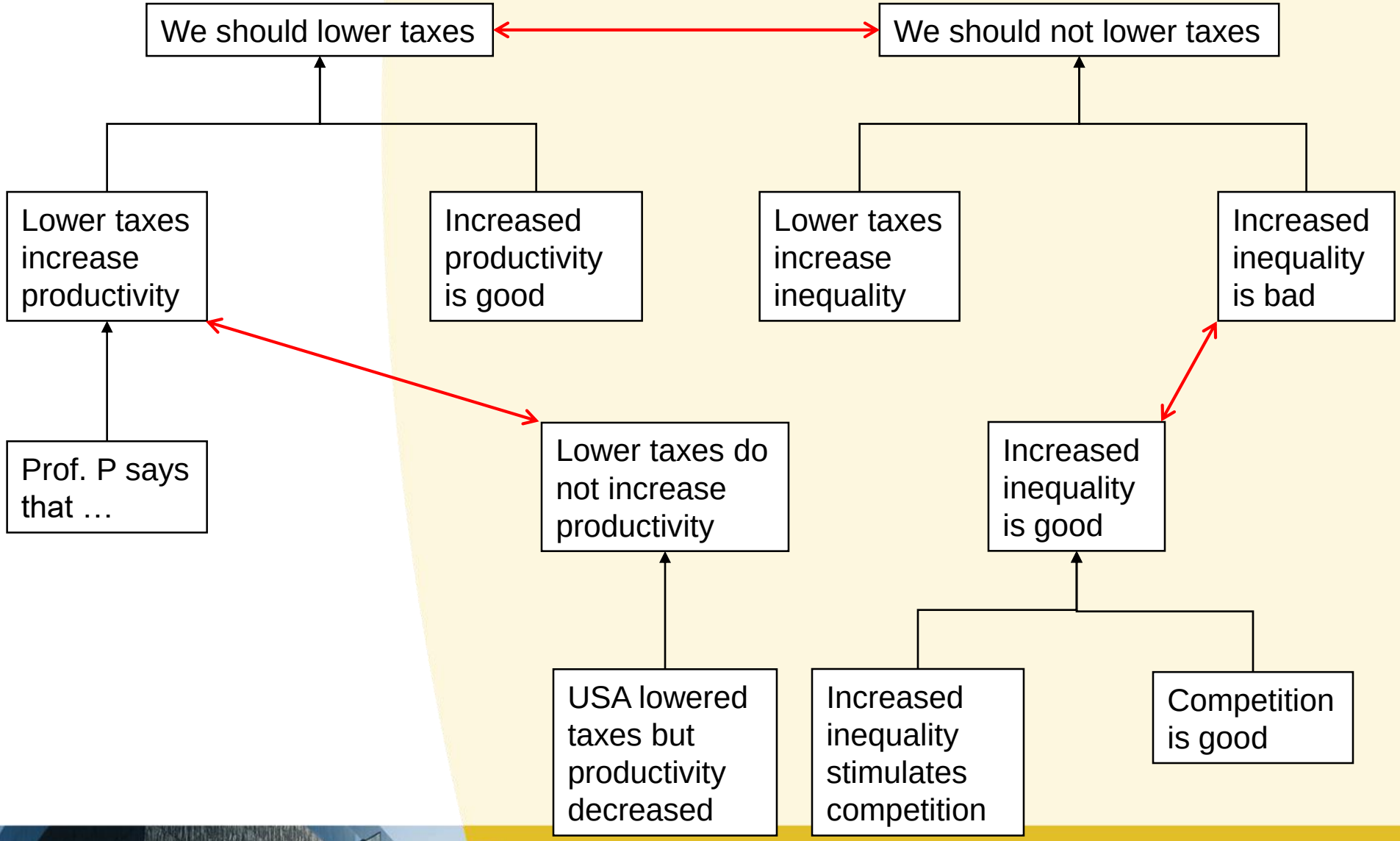
Attack

- *A* undermines *B* (on ϕ) if
  - Conc(*A*) = $\neg\phi$ for some $\phi \in$ Prem(*B*)/ $\mathcal{K}_n$;
- *A* rebuts *B* (on *B′*) if
  - Conc(*A*) = ¬Conc(*B′*) for some *B′* $\in$ Sub(*B)* with a defeasible top rule
- *A* undercuts *B* (on *B′*) if
  - Conc(*A*) = ¬n(*r*) ′for some *B′* $\in$ Sub(*B*) with defeasible top rule *r*
- *A* attacks *B* iff *A* undermines or rebuts or undercuts *B.*

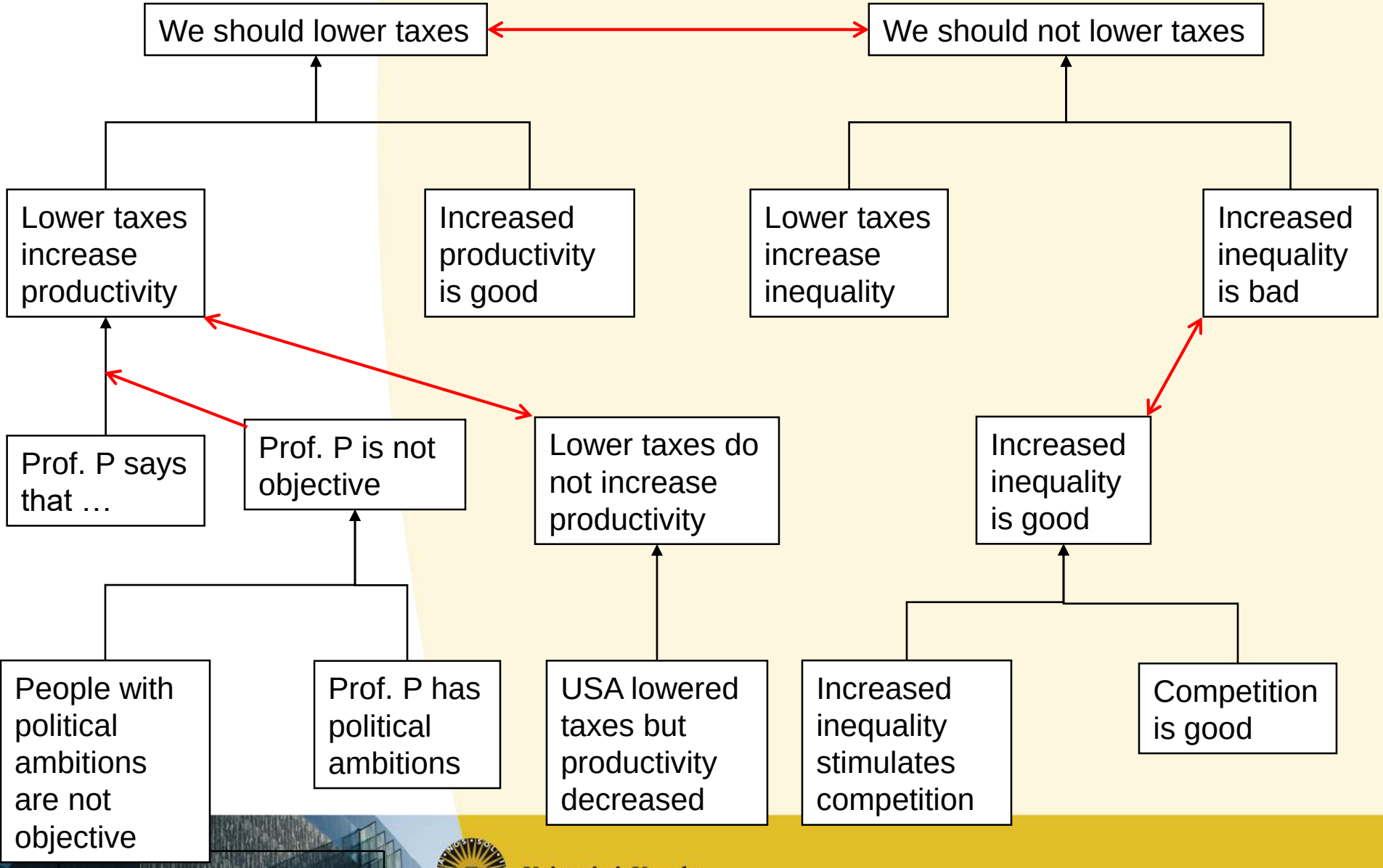
Rebuttal


We should lower taxes
We should not lower taxes
Lower taxes increase productivity
Increased productivity is good
Lower taxes increase inequality
Increased inequality is bad
Prof. P says that …

Undermining



Undercutting



Undercutting

- Attacks the application of the inference rule by providing an exception

- I see a red ball *so* there is a red ball
- I see a blue ball *so* there is no red ball (rebut)
- The ball is illuminated by a red light (undercut)
  - The conclusion (ball = red) might still be true, but the premise (I see red ball) is no good reason for this conclusion

Argumentation semantics

- FOL semantics are model-based
  - Is a formula true given the model of the world we have?

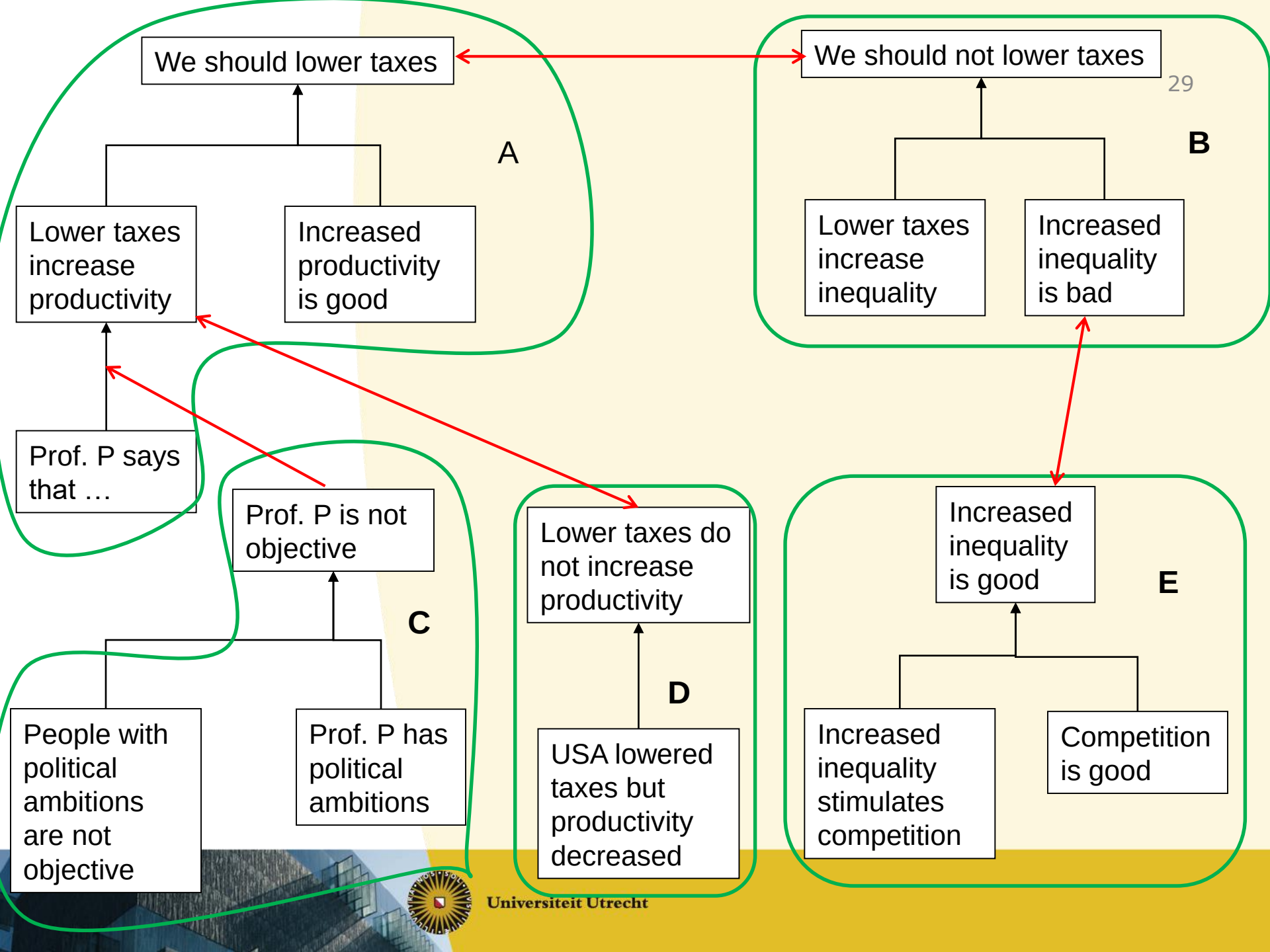
- Argumentation semantics are *dialectical*
  - Is an argument/proposition acceptable given all the (counter)arguments?
  - Given a set of arguments and attack relations, what are the consistent sets of arguments (extensions) we can reasonably accept?

Argumentation frameworks

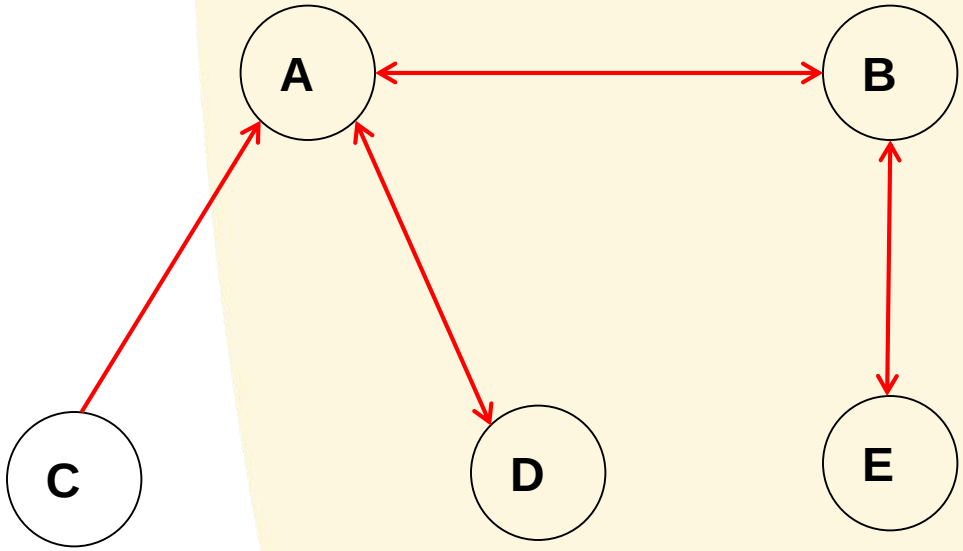
- An argumentation framework is a pair (*Args,D*) where *Args* is a set of arguments and *D* is the defeat relation on *Args*

- In abstract argumentation, we don't care what's inside the arguments, or where the attacks come from.
  - AF is a graph where nodes are arguments and (directed) edges attacks.

- Structured argumentation-> abstract argumentation
  - *Args* defined as the maximal arguments as defined by the argumentation theory
  - Defeat = Attack


We should lower taxes
A
Lower taxes increase productivity
Increased productivity is good
Prof. P says that …
We should not lower taxes
B
Lower taxes increase inequality
Increased inequality is bad
Prof. P is not objective
C
People with political ambitions are not objective
Prof. P has political ambitions
Lower taxes do not increase productivity
D
USA lowered taxes but productivity decreased
Increased inequality is good
E
Increased inequality stimulates competition
Competition is good

Abstract Argumentation



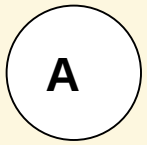
Admissible sets

- a set of arguments A is an *admissible extension* if
  - it is *conflict-free*: There are no arguments α and β in A, such that α attacks β.
  - the arguments in A are *acceptable* with respect to A: For all arguments α in A, such that there is an argument β that attacks α, there is an argument γ in A that attacks β.

- An admissible set of arguments is a preferred extension if it is a maximal (wrt set inclusion) admissible extension.

Preferred extension

- Preferred extension:

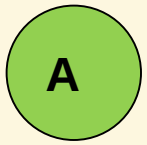


Conflict freeness: There are no arguments α and β in A, such that α attacks β.
Acceptability: For all arguments α in A, such that there is an argument β that attacks α, there is an argument γ in A that attacks β.

Preferred extension

Conflict freeness: There are no arguments α and β in A, such that α attacks β.
Acceptability: For all arguments α in A, such that there is an argument β that attacks α, there is an argument γ in A that attacks β.

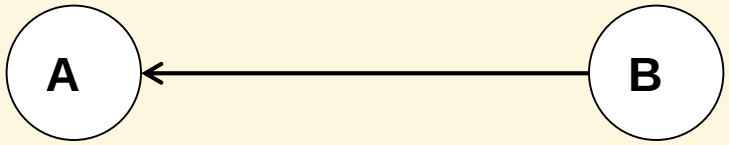
- Preferred extension: {A}
- Arguments that are not attacked are always in the preferred extension



Preferred extension

Conflict freeness: There are no arguments α and β in A, such that α attacks β.
Acceptability: For all arguments α in A, such that there is an argument β that attacks α, there is an argument γ in A that attacks β.

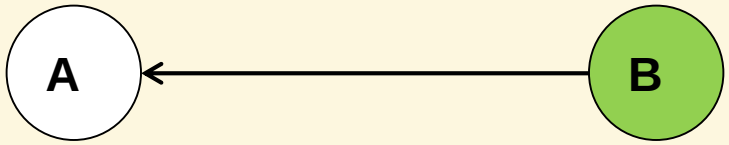
- Preferred extension:



Preferred extension

Conflict freeness: There are no arguments α and β in A, such that α attacks β.
Acceptability: For all arguments α in A, such that there is an argument β that attacks α, there is an argument γ in A that attacks β.

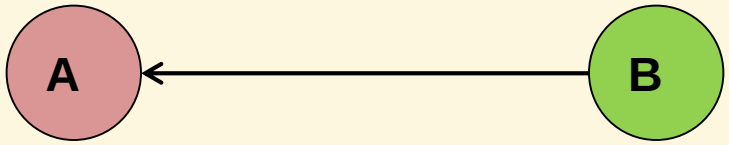
- Preferred extension: {B}



Preferred extension

Conflict freeness: There are no arguments α and β in A, such that α attacks β.
Acceptability: For all arguments α in A, such that there is an argument β that attacks α, there is an argument γ in A that attacks β.

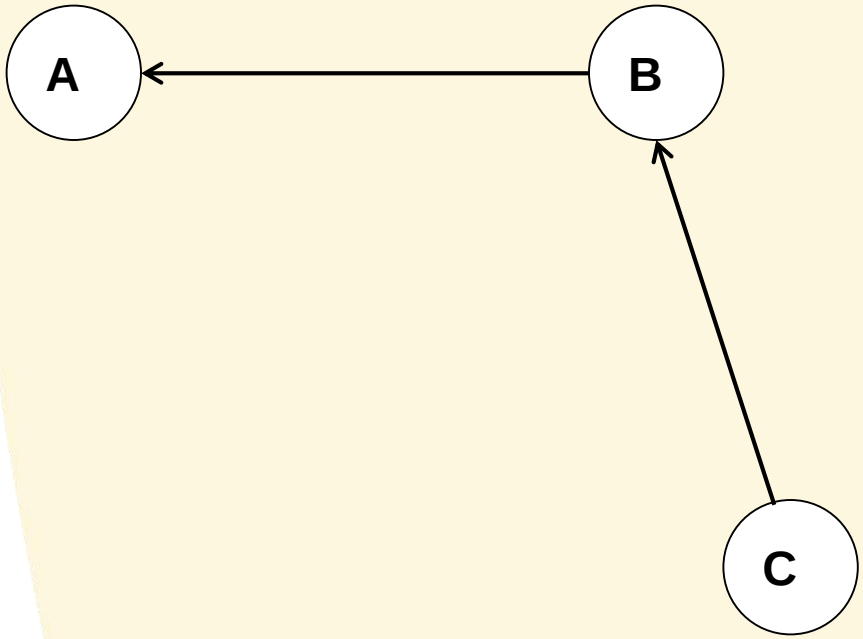
- Preferred extension: {B}
- Arguments that are attacked by an in argument are not in the preferred extension



Preferred extension

Conflict freeness: There are no arguments α and β in A, such that α attacks β.
Acceptability: For all arguments α in A, such that there is an argument β that attacks α, there is an argument γ in A that attacks β.

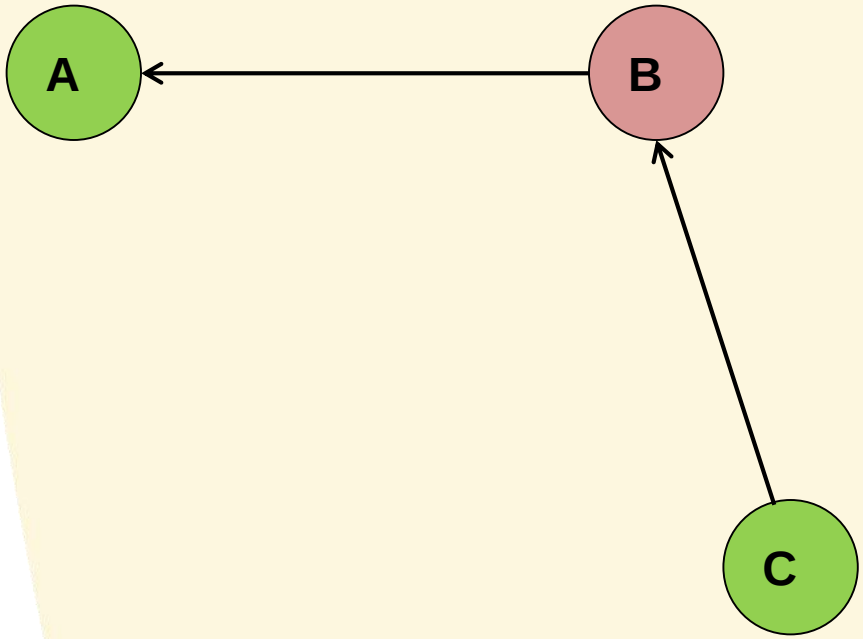
- Preferred extension:



Preferred extension

Conflict freeness: There are no arguments α and β in A, such that α attacks β.
Acceptability: For all arguments α in A, such that there is an argument β that attacks α, there is an argument γ in A that attacks β.

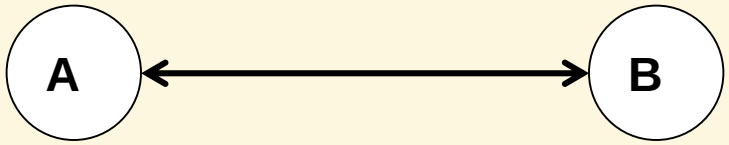
- Preferred extension: {A,C}



Multiple extensions

Conflict freeness: There are no arguments α and β in A, such that α attacks β.
Acceptability: For all arguments α in A, such that there is an argument β that attacks α, there is an argument γ in A that attacks β.

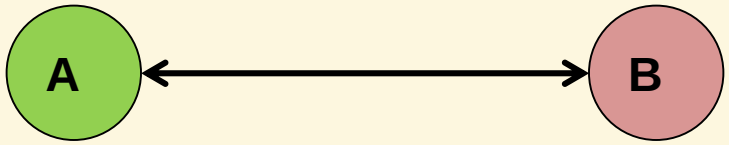
- Preferred extensions:



Multiple extensions

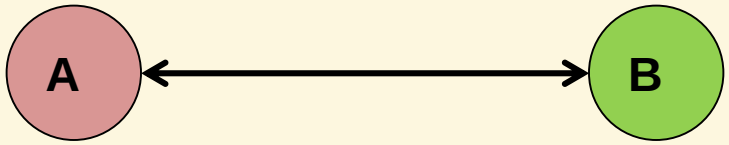
Conflict freeness: There are no arguments α and β in A, such that α attacks β.
Acceptability: For all arguments α in A, such that there is an argument β that attacks α, there is an argument γ in A that attacks β.

- Preferred extensions: {A},



Multiple extensions

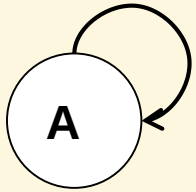
- Preferred extensions: {A}, {B}



Conflict freeness: There are no arguments α and β in A, such that α attacks β.
Acceptability: For all arguments α in A, such that there is an argument β that attacks α, there is an argument γ in A that attacks β.

No extension

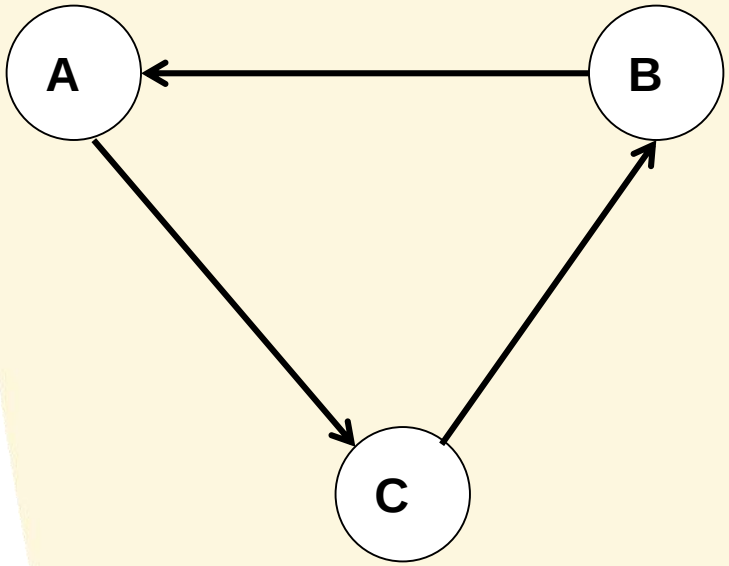
- Preferred extension: ∅



Conflict freeness: There are no arguments α and β in A, such that α attacks β.
Acceptability: For all arguments α in A, such that there is an argument β that attacks α, there is an argument γ in A that attacks β.

No extension

- Preferred extension: ∅
- Odd cycles

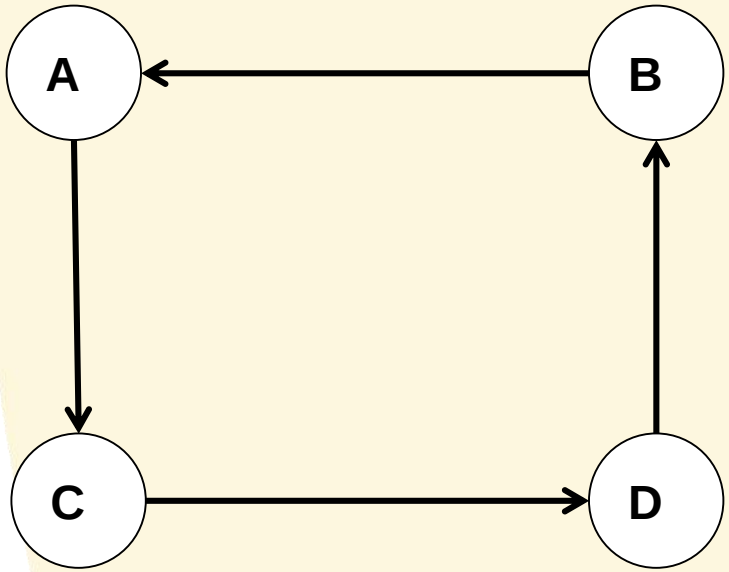


Conflict freeness: There are no arguments α and β in A, such that α attacks β.
Acceptability: For all arguments α in A, such that there is an argument β that attacks α, there is an argument γ in A that attacks β.

Multiple extensions

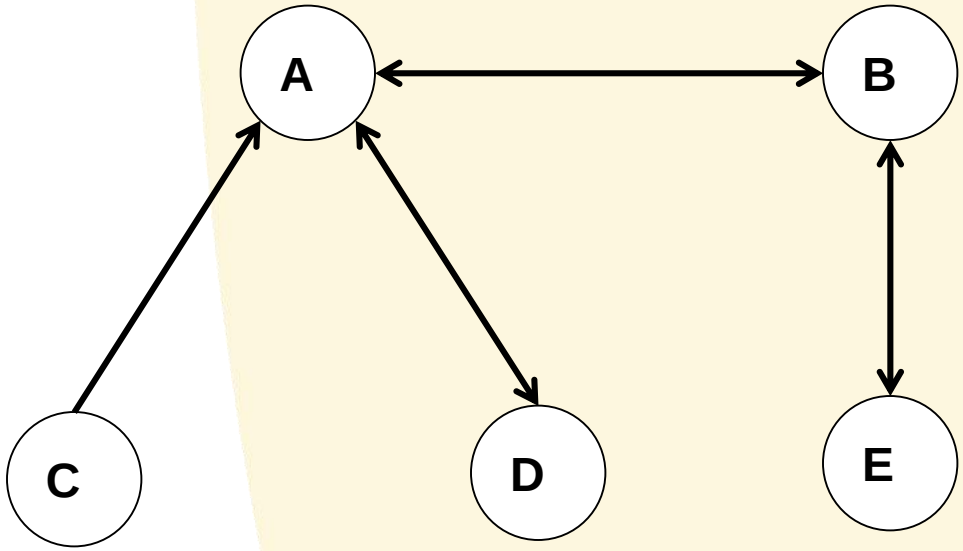
Conflict freeness: There are no arguments α and β in A, such that α attacks β.
Acceptability: For all arguments α in A, such that there is an argument β that attacks α, there is an argument γ in A that attacks β.

- What about even cycles?



Back to taxes

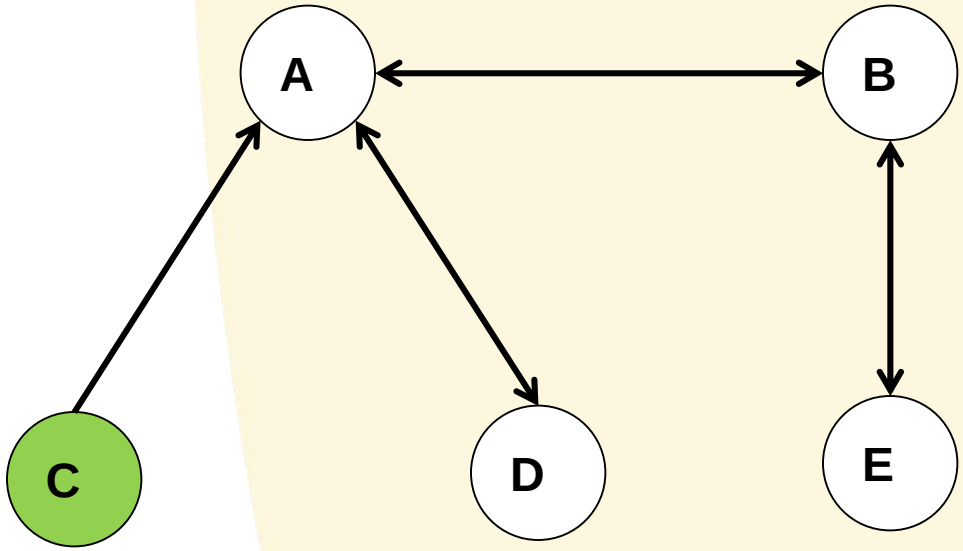
- Preferred extensions: {C



A: Lower taxes
B: Not lower taxes
C: Prof. P is not objective
D: Lower taxes do not increase productivity
E: Increased inequality is good

Back to taxes

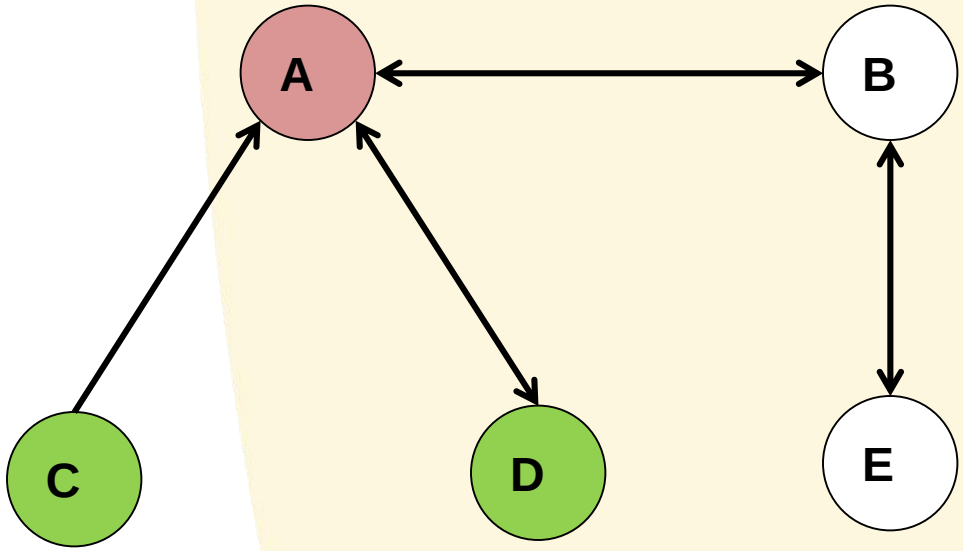
- Preferred extensions: {C



A: Lower taxes
B: Not lower taxes
C: Prof. P is not objective
D: Lower taxes do not increase productivity
E: Increased inequality is good

Back to taxes

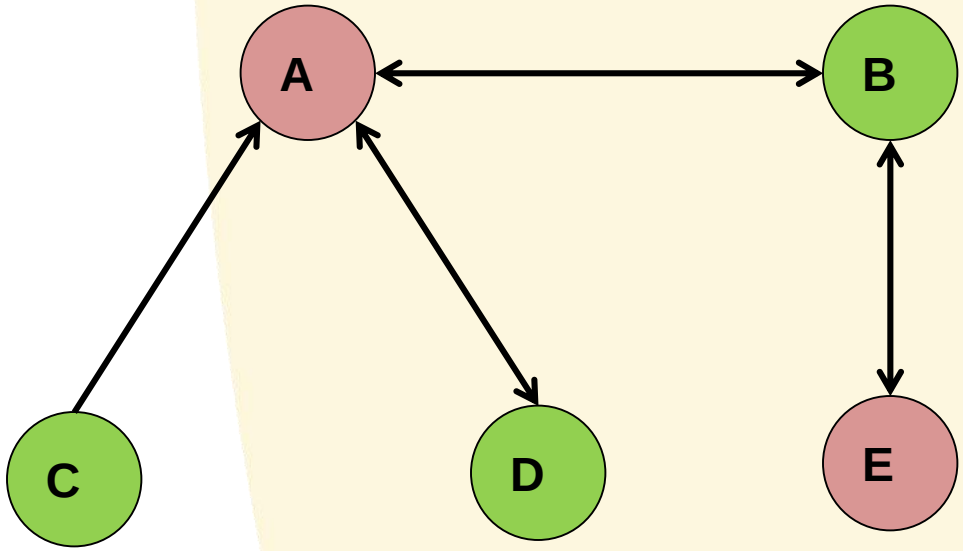
- Preferred extensions: {C,D



A: Lower taxes
B: Not lower taxes
C: Prof. P is not objective
D: Lower taxes do not increase productivity
E: Increased inequality is good

Back to taxes

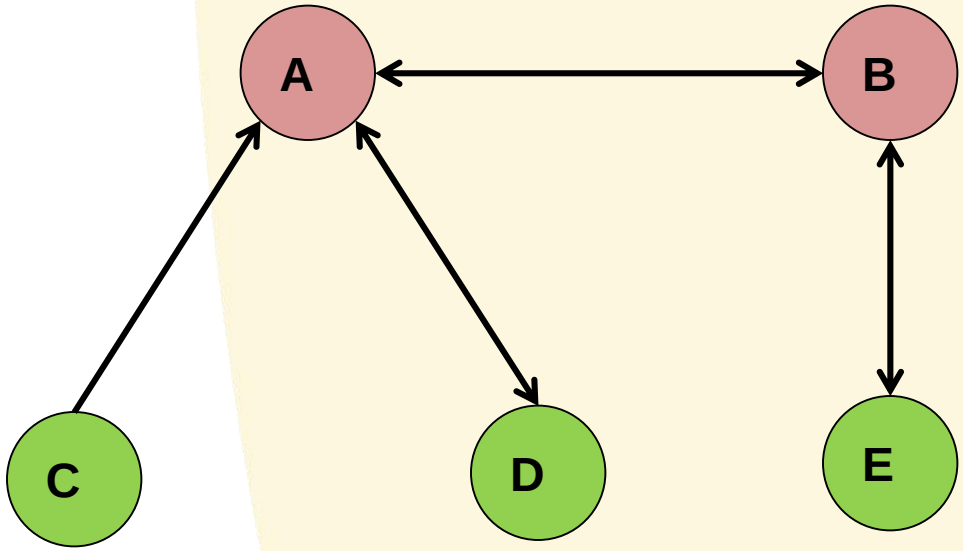
- Preferred extensions: {C,D,B},



A: Lower taxes
B: Not lower taxes
C: Prof. P is not objective
D: Lower taxes do not increase productivity
E: Increased inequality is good

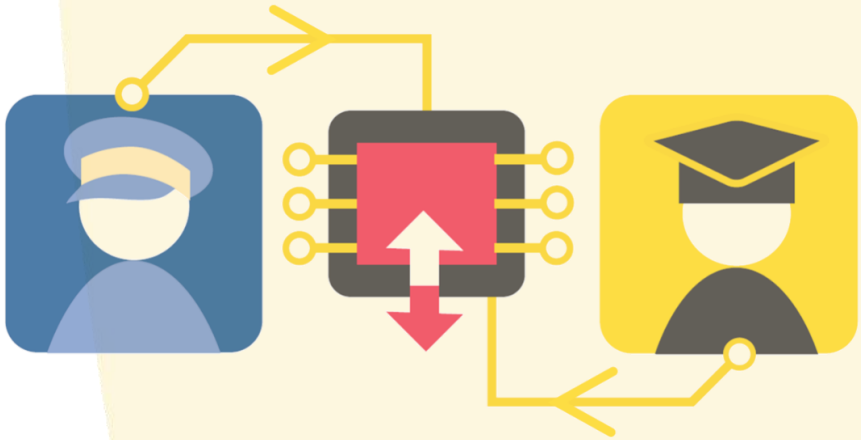
Back to taxes

- Preferred extensions: {C,D,B}, {C,D,E}

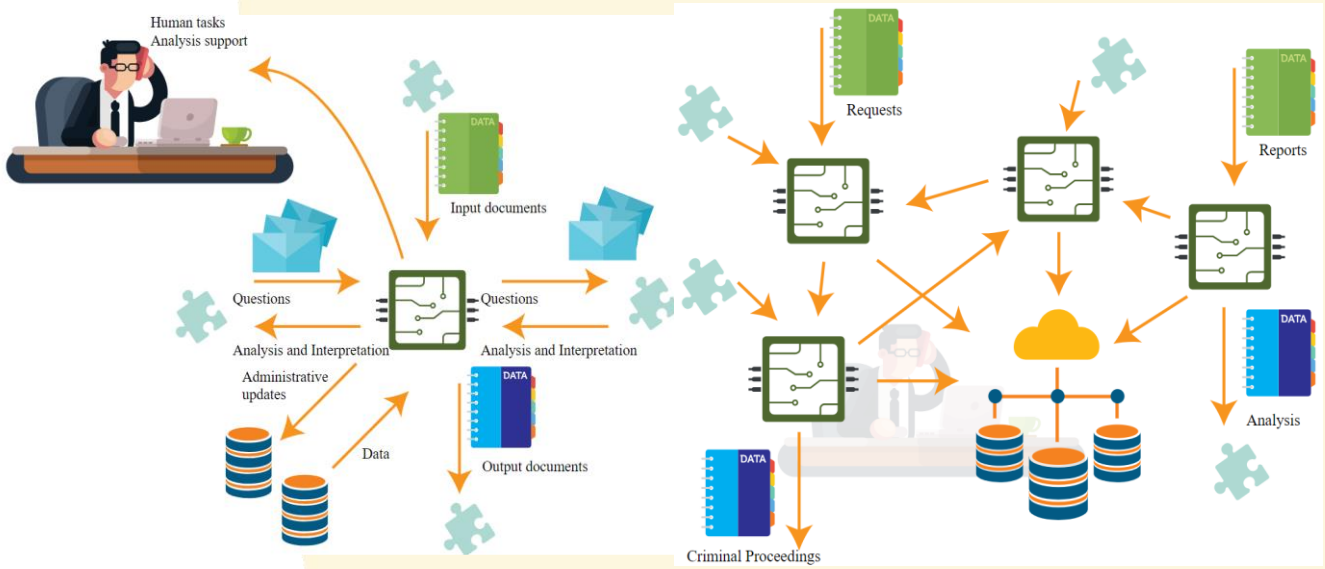


A: Lower taxes
B: Not lower taxes
C: Prof. P is not objective
D: Lower taxes do not increase productivity
E: Increased inequality is good

AI for the police



AI for the Dutch police – man-machine organisation

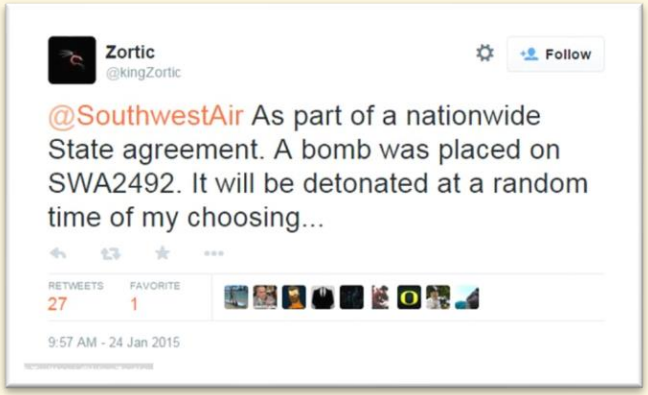


Requirements for police AI

- Accurate: Minimize Mistakes
- Transparent: Explanation of important decisions
- Controllable: Detect where errors are, meaningful human in the loop
- Contestable: Allow users to contest and influence decisions/conclusions
- Efficient: Minimize unnecessary actions

AI for the Dutch Police

- For some types of tasks machine learning is the best solution
  - Recognizing guns
  - Recognizing online threaths



AI for the Dutch police

- For other types of tasks, using only machine learning is not a good idea
  - (Autonomously) making (legally relevant) decisions based on input

- Automated handling of citizen crime reports

Online trade fraud

- Online trade fraud
  - Fraud on eBay, internet forums, fake websites
- 40,000 reports filed per year

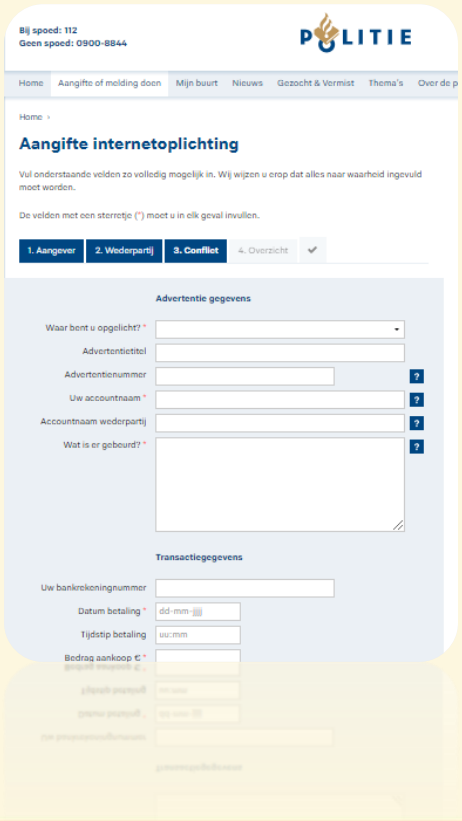
- Legal background: article 326 of Dutch Criminal Code
  - Take some good or money away from someone, while "*misleading through false contact details, deceptive tricks or an accumulation of lies*"

System for handling citizen reports on online trade fraud

- Submitted online
- Given report, decide:
  - Fraud
  - No fraud
  - More information needed → ask questions

- Transparent explanations!

- Check the video of ai-policing for demo

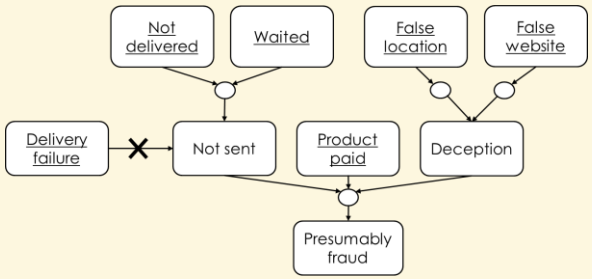


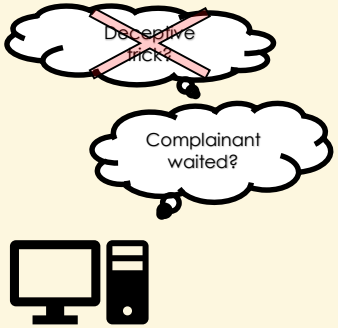
System architecture

False location

Paid

Not delivered





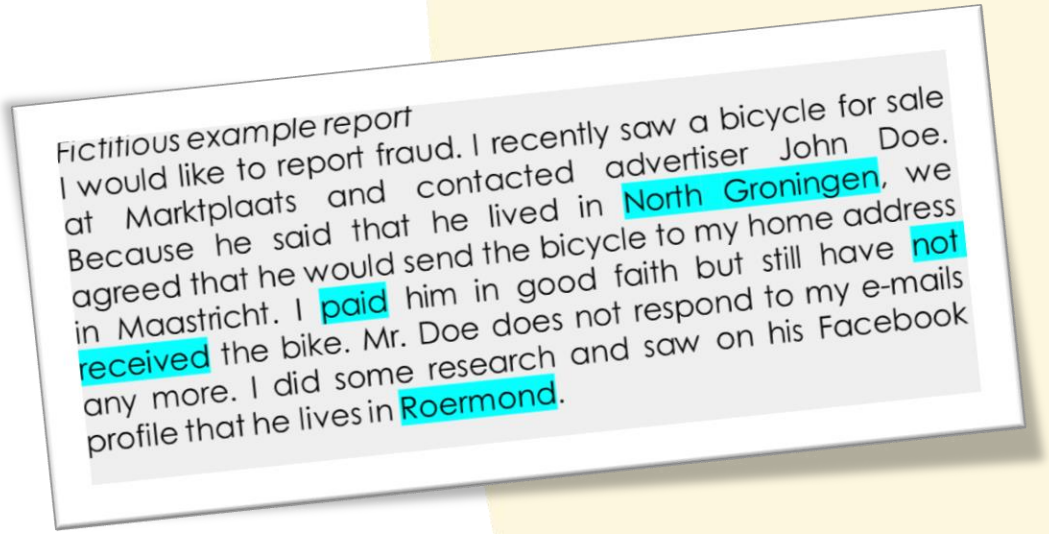
Information Extraction

Argumentation

Question Policy

Information extraction

- Extract observations from free text


Fictitious example report
I would like to report fraud. I recently saw a bicycle for sale at Marktplaats and contacted advertiser John Doe. Because he said that he lived in North Groningen, we agreed that he would send the bicycle to my home address in Maastricht. I paid him in good faith but still have not received the bike. Mr. Doe does not respond to my e-mails any more. I did some research and saw on his Facebook profile that he lives in Roermond.

False location

Paid

Not delivered

Information extraction

- Named Entity Recognition
  - Well-known Dutch NLP package (FROG)
    - trained on news articles
    - crime reports are different: bad punctuation, telegram style, misspellings, particular entities
  - Enriched version of NER module in Frog (92% precision, 82% recall)
- Relation extraction using LSTMs (94-99% accuracy)

- Classifying texts according to observations
  - Practical solution, ± 90% accuracy (lower recall)

Schraagen, Brinkhuis, Bex, (2017) Evaluation of Named Entity Recognition in Dutch online criminal complaints. *Computational Linguistics in the Netherlands Journal*, *7*.

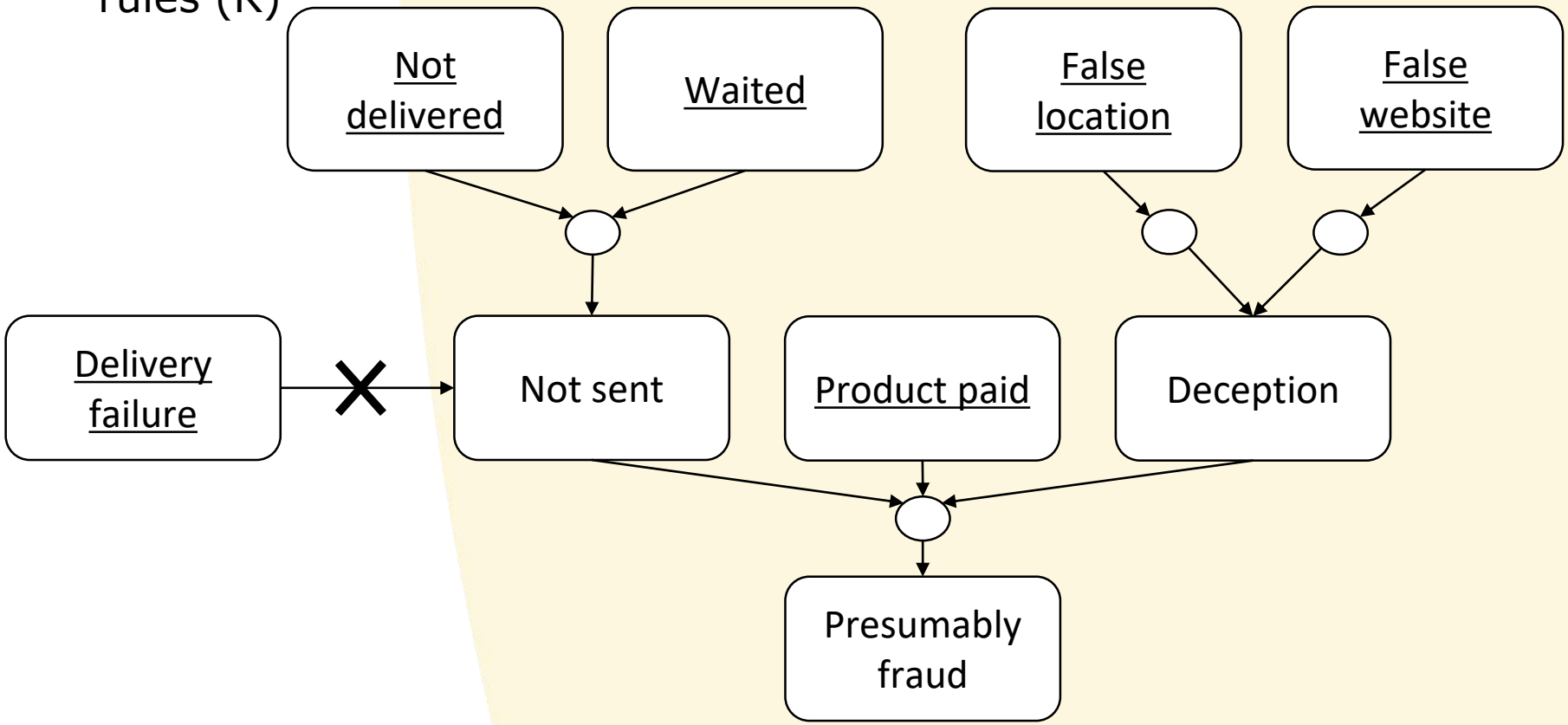
Schraagen, Bex (2019) Extraction of semantic relations in noisy user-generated law enforcement data, *Proceedings of the 13th IEEE International Conference on Semantic Computing (ICSC 2019)*.

Argumentation

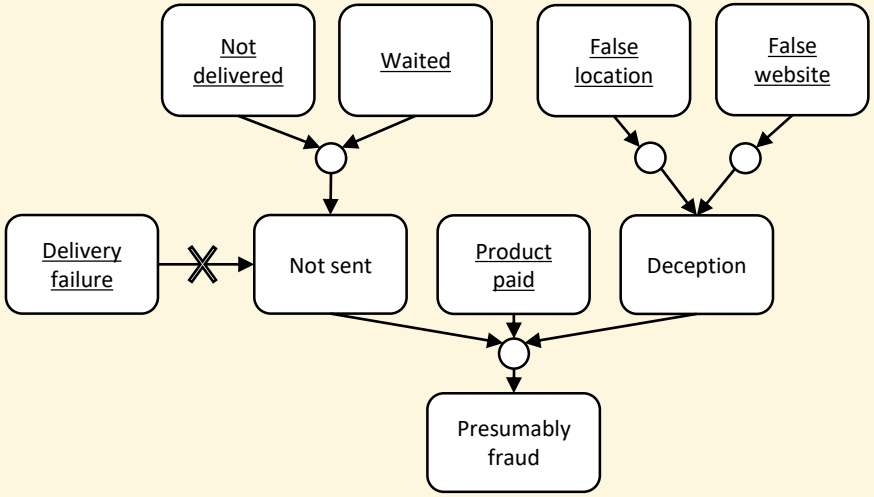
- Rule-based argumentation consisting of:
  - Facts observable in text of complaint
  - Inference rules
- Based on the law and expert knowledge
  - art 326 Criminal Code, case law, expert police knowledge
- Argumentation works very well for legal reasoning
  - Arguments with exceptions.

Argumentation theory example

- Full theory: 26 observables (possible elements of K), 46 inference rules (R)

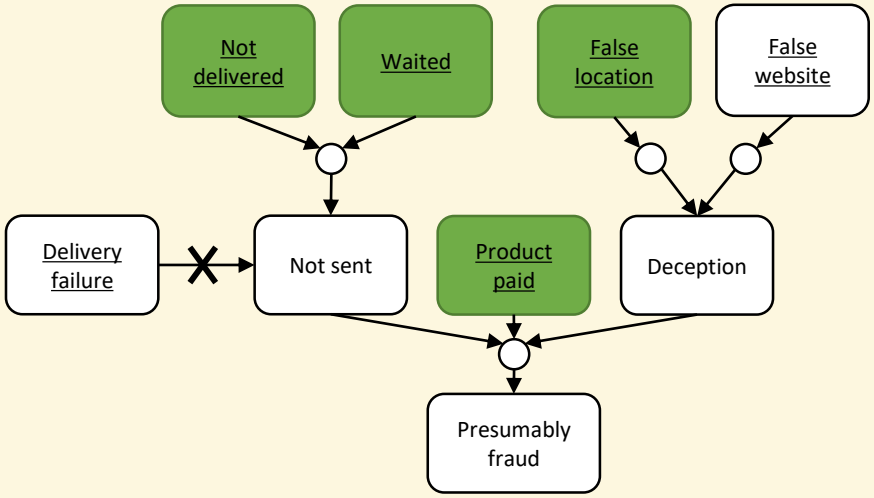


Argumentation example



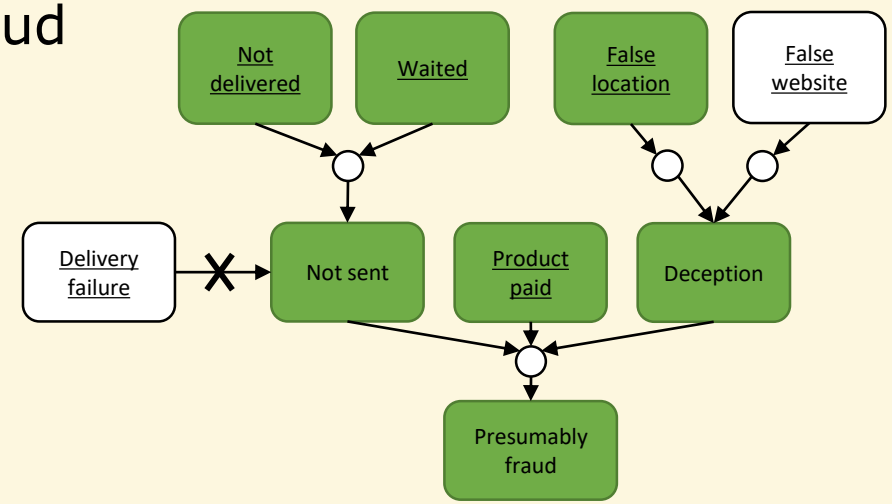
Argumentation example

- False location, paid, waited, not delivered



Argumentation example

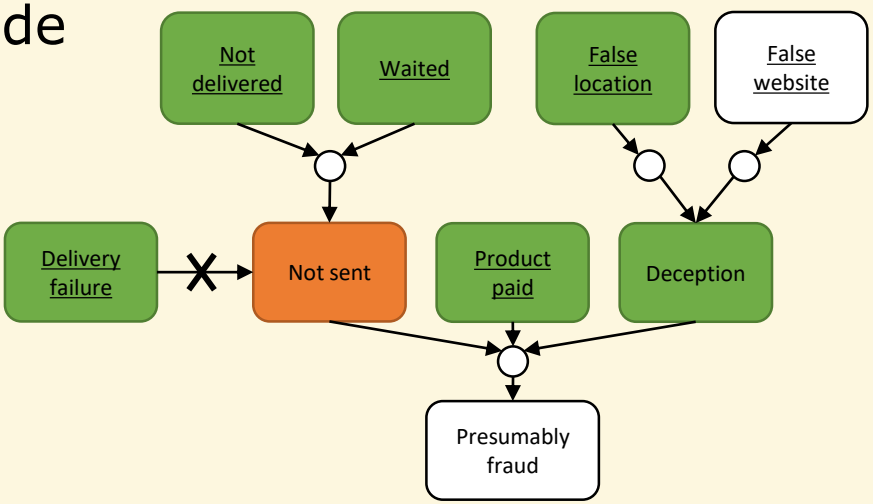
- False location, paid, waited, not delivered, not sent, deception, presumably fraud
- Not delivered ∧ waited → not sent
- Conclusion: presumably fraud



Argumentation example

- Delivery failure
  attacks not sent

- No longer possible to conclude presumably fraud



Dialogue

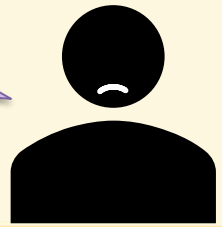
…. Not delivered ….
Paid ….
…. False location …

Did the counterparty use a false website?

No

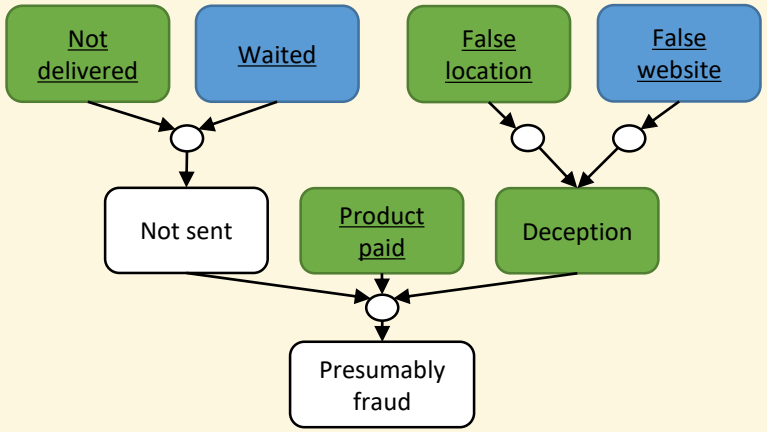
How long have you waited since he said he would send you the product?

I've waited for three weeks.



Question policy learning

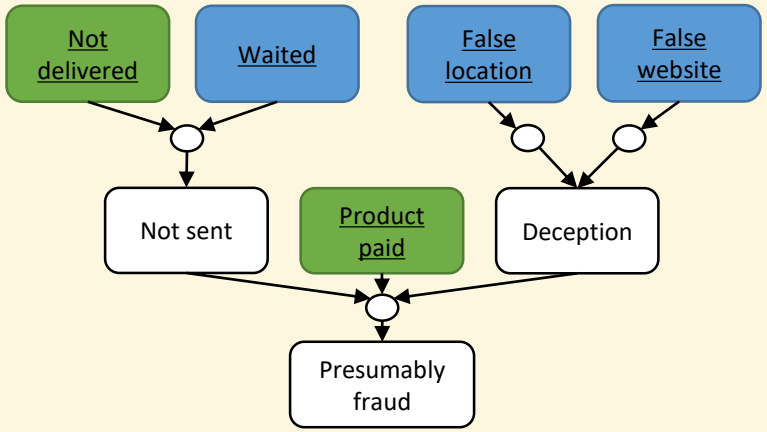
- Goal: Given observations, find next question(s)
- Efficiently reaching a conclusion
  - Only ask relevant questions
- Approximation algorithm for determining relevant questions



Testerink, Odekerken, Bex (2019) A Method for Efficient Argument-based Inquiry, *13th International Conference on Flexible Query Answering Systems (FQAS 2019)*. Lecture Notes in Artificial Intelligence.

Question policy learning

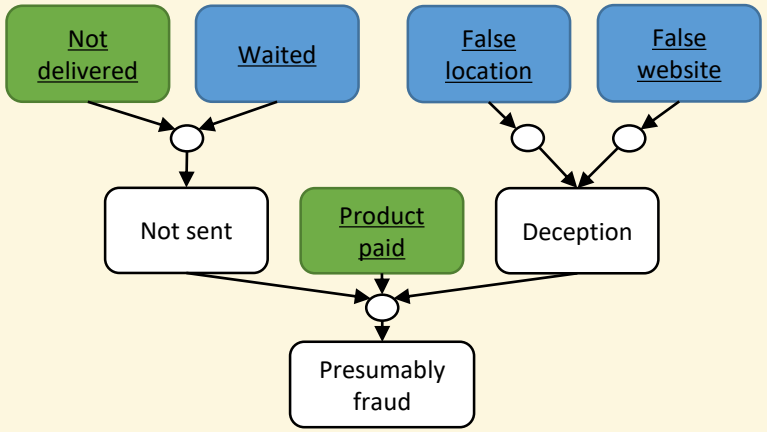
- Goal: Given observations, find next question(s)
- Efficiently reaching a conclusion
  - Include probabilities of user responses
  - Hypothesizing over all possible future questions?



Question policy learning

- Goal: Given observations, find next question(s)
- Efficiently reaching a conclusion
  - Include probabilities of user responses
  - Hypothesizing over all possible future questions?

- Idea: model the intake process as a Markov Decision Process, and apply reinforcement learning
  - reaching conclusion: high reward
  - Each question: small penalty

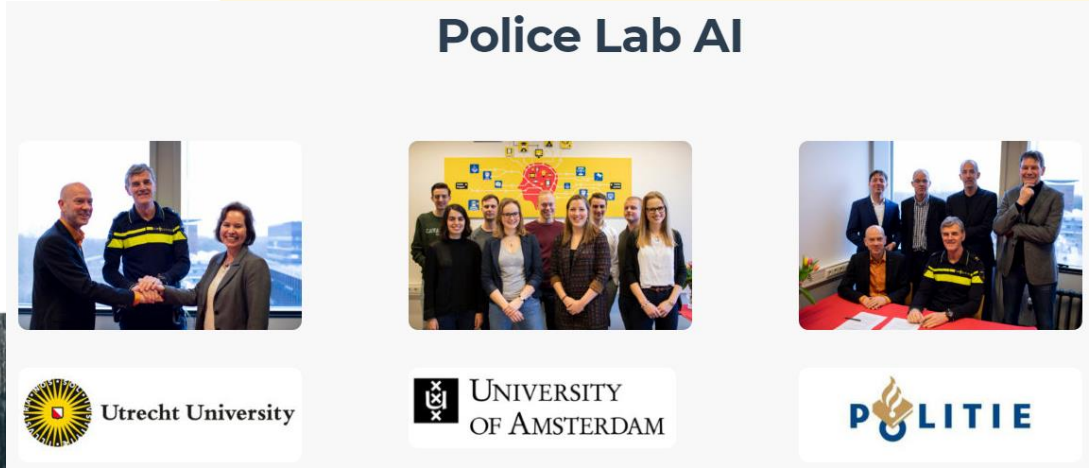


Conclusions

- Accurate, Transparent, Controllable, Contestable, Efficient AI
- Sub-symbolic machine learning is not always the best option!
  - But it is needed for interpreting the world around us
- Police AI
  - Sub-symbolic for interpreting text, images, etc. (“Sense”)
  - Symbolic for making decisions (“reason”)

Police lab AI

- NLP & OSINT (Open Source INTelligence)
  - Relevant information from news sites, Wikipedia, Twitter
- Computational Argumentation
  - Reasoning with scenario's, (legal) arguments
- Multi-agent systems
  - Communication between automated systems
- Forensics
  - Bayesian Networks for forensic reasoning



Study questions for the exam

- What is the key difference between classical logics and nonmonotonic logics?
  - Why can we not just use classical logic to model defeasible reasoning?
- In argumentation, what is the difference between a rebutting attack and an undercutting attack?
- Take an application of modus ponens in classical logic. Provide the knowledge base $\mathcal{K}$, the ruleset $\mathcal{R} = \mathcal{R}_s \cup \mathcal{R}_d$ and the argument A that captures this application.
- Take the argumentation framework from slide 30
  - Give the preferred extension(s)