Probabilistic Argumentation Systems

C. Schneuwly

Department of Informatics
University of Fribourg, Switzerland

Seminar in Theoretical Computer Science
about Probabilistic Expert Systems

WS 2004 / 2005

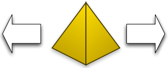
24.03.2005

Outline

Outline

Propositional Sentences

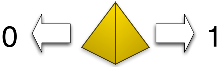
- Propositions: Statements that can be either true or false

0  1

- Impossible statement: $\perp$, the one which is always true: $\top$
- Let $P = \{p_1, \ldots, p_n\}$ be a finite set of propositions
- The $p_i \in P$ are called atomic formulas or atoms

Propositional Sentences

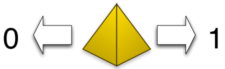
- Propositions: Statements that can be either true or false



- Impossible statement: $\perp$, the one which is always true: $\top$
- Let $P = \{p_1, \ldots, p_n\}$ be a finite set of propositions
- The $p_i \in P$ are called atomic formulas or atoms

Propositional Sentences

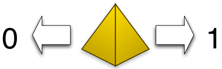
- Propositions: Statements that can be either true or false



- Impossible statement: $\perp$, the one which is always true: $\top$
- Let $P = \{p_1, \ldots, p_n\}$ be a finite set of propositions
- The $p_i \in P$ are called atomic formulas or atoms

Propositional Sentences

- Propositions: Statements that can be either true or false



- Impossible statement: $\perp$, the one which is always true: $\top$
- Let $P = \{p_1, \ldots, p_n\}$ be a finite set of propositions
- The $p_i \in P$ are called atomic formulas or atoms

Propositional Sentences

Compound formulas are build by:

- atoms, $\perp$ and $\top$ are formulas
- if γ is a formula, then $\neg\gamma$ is a formula
- if γ and δ are formulas, then $(\gamma \wedge \delta)$, $(\gamma \vee \delta)$, $(\gamma \rightarrow \delta)$ and $(\gamma \leftrightarrow \delta)$ are formulas

- The set $\mathcal{L}_P$ of all formulas is called propositional language over P
- A formula $\gamma \in \mathcal{L}_P$ is also called propositional sentence

Propositional Sentences

Compound formulas are build by:

- atoms, $\perp$ and $\top$ are formulas
- if γ is a formula, then $\neg\gamma$ is a formula
- if γ and δ are formulas, then $(\gamma \wedge \delta)$, $(\gamma \vee \delta)$, $(\gamma \rightarrow \delta)$ and $(\gamma \leftrightarrow \delta)$ are formulas

- The set $\mathcal{L}_P$ of all formulas is called propositional language over P
- A formula $\gamma \in \mathcal{L}_P$ is also called propositional sentence

Propositional Sentences

Compound formulas are build by:

- atoms, $\perp$ and $\top$ are formulas
- if γ is a formula, then $\neg\gamma$ is a formula
- if γ and δ are formulas, then $(\gamma \wedge \delta)$, $(\gamma \vee \delta)$, $(\gamma \rightarrow \delta)$ and $(\gamma \leftrightarrow \delta)$ are formulas

- The set $\mathcal{L}_P$ of all formulas is called propositional language over P
- A formula $\gamma \in \mathcal{L}_P$ is also called propositional sentence

Propositional Sentences

Compound formulas are build by:

- atoms, $\perp$ and $\top$ are formulas
- if γ is a formula, then $\neg\gamma$ is a formula
- if γ and δ are formulas, then $(\gamma \wedge \delta)$, $(\gamma \vee \delta)$, $(\gamma \rightarrow \delta)$ and $(\gamma \leftrightarrow \delta)$ are formulas

- The set $\mathcal{L}_P$ of all formulas is called propositional language over P
- A formula $\gamma \in \mathcal{L}_P$ is also called propositional sentence

Propositional Sentences

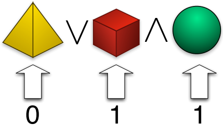
Compound formulas are build by:

- atoms, $\perp$ and $\top$ are formulas
- if γ is a formula, then $\neg\gamma$ is a formula
- if γ and δ are formulas, then $(\gamma \wedge \delta)$, $(\gamma \vee \delta)$, $(\gamma \rightarrow \delta)$ and $(\gamma \leftrightarrow \delta)$ are formulas

- The set $\mathcal{L}_P$ of all formulas is called propositional language over P
- A formula $\gamma \in \mathcal{L}_P$ is also called propositional sentence

Semantics

The meaning of a propositional sentence:

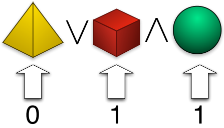


- An assignment of truth values to P is called interpretation
- N_P denotes the set of all 2^n interpretations

γ	δ	$\bot$	$\top$	$\neg\gamma$	$\gamma \wedge \delta$	$\gamma \vee \delta$	$\gamma \rightarrow \delta$	$\gamma \leftrightarrow \delta$
0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0
1	0	0	1	0	0	1	0	0
1	1	0	1	0	1	1	1	1

Semantics

The meaning of a propositional sentence:

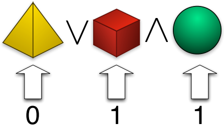


- An assignment of truth values to P is called interpretation
- N_P denotes the set of all 2^n interpretations

γ	δ	$\bot$	$\top$	$\neg\gamma$	$\gamma \wedge \delta$	$\gamma \vee \delta$	$\gamma \rightarrow \delta$	$\gamma \leftrightarrow \delta$
0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0
1	0	0	1	0	0	1	0	0
1	1	0	1	0	1	1	1	1

Semantics

The meaning of a propositional sentence:

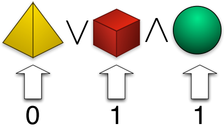


- An assignment of truth values to P is called interpretation
- N_P denotes the set of all 2^n interpretations

γ	δ	$\bot$	$\top$	$\neg\gamma$	$\gamma \wedge \delta$	$\gamma \vee \delta$	$\gamma \rightarrow \delta$	$\gamma \leftrightarrow \delta$
0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0
1	0	0	1	0	0	1	0	0
1	1	0	1	0	1	1	1	1

Semantics

The meaning of a propositional sentence:



- An assignment of truth values to P is called interpretation
- N_P denotes the set of all 2^n interpretations

γ	δ	$\bot$	$\top$	$\neg\gamma$	$\gamma \wedge \delta$	$\gamma \vee \delta$	$\gamma \rightarrow \delta$	$\gamma \leftrightarrow \delta$
0	0	0	1	1	0	0	1	1
0	1	0	1	1	0	1	1	0
1	0	0	1	0	0	1	0	0
1	1	0	1	0	1	1	1	1

Logical Consequences

- An interpretation x is called a model of γ if γ evaluates to 1
- The set of all models of γ is denoted by $N_P(\gamma) \subseteq N_P$
- If $N_P(\gamma) \neq \emptyset$ then γ is called satisfiable

Entailment Relation

- δ is a logical consequence of $\gamma \Leftrightarrow N_P(\gamma) \subseteq N_P(\delta)$
- we write $\gamma \models \delta$

- γ and δ are logical equivalent $(\gamma \equiv \delta) \Leftrightarrow N_P(\gamma) = N_P(\delta)$

Logical Consequences

- An interpretation x is called a model of γ if γ evaluates to 1
- The set of all models of γ is denoted by $N_P(\gamma) \subseteq N_P$
- If $N_P(\gamma) \neq \emptyset$ then γ is called satisfiable

Entailment Relation

- δ is a logical consequence of $\gamma \Leftrightarrow N_P(\gamma) \subseteq N_P(\delta)$
- we write $\gamma \models \delta$

- γ and δ are logical equivalent $(\gamma \equiv \delta) \Leftrightarrow N_P(\gamma) = N_P(\delta)$

Logical Consequences

- An interpretation x is called a model of γ if γ evaluates to 1
- The set of all models of γ is denoted by $N_P(\gamma) \subseteq N_P$
- If $N_P(\gamma) \neq \emptyset$ then γ is called satisfiable

Entailment Relation

- δ is a logical consequence of $\gamma \Leftrightarrow N_P(\gamma) \subseteq N_P(\delta)$
- we write $\gamma \models \delta$

- γ and δ are logical equivalent $(\gamma \equiv \delta) \Leftrightarrow N_P(\gamma) = N_P(\delta)$

Logical Consequences

- An interpretation x is called a model of γ if γ evaluates to 1
- The set of all models of γ is denoted by $N_P(\gamma) \subseteq N_P$
- If $N_P(\gamma) \neq \emptyset$ then γ is called satisfiable

Entailment Relation

- δ is a logical consequence of $\gamma \Leftrightarrow N_P(\gamma) \subseteq N_P(\delta)$
- we write $\gamma \models \delta$

- γ and δ are logical equivalent $(\gamma \equiv \delta) \Leftrightarrow N_P(\gamma) = N_P(\delta)$

Logical Consequences

- An interpretation x is called a model of γ if γ evaluates to 1
- The set of all models of γ is denoted by $N_P(\gamma) \subseteq N_P$
- If $N_P(\gamma) \neq \emptyset$ then γ is called satisfiable

Entailment Relation

- δ is a logical consequence of $\gamma \Leftrightarrow N_P(\gamma) \subseteq N_P(\delta)$
- we write $\gamma \models \delta$

- γ and δ are logical equivalent $(\gamma \equiv \delta) \Leftrightarrow N_P(\gamma) = N_P(\delta)$

Logical Consequences

- An interpretation x is called a model of γ if γ evaluates to 1
- The set of all models of γ is denoted by $N_P(\gamma) \subseteq N_P$
- If $N_P(\gamma) \neq \emptyset$ then γ is called satisfiable

Entailment Relation

- δ is a logical consequence of $\gamma \Leftrightarrow N_P(\gamma) \subseteq N_P(\delta)$
- we write $\gamma \models \delta$

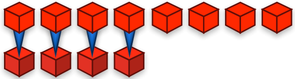
- γ and δ are logical equivalent $(\gamma \equiv \delta) \Leftrightarrow N_P(\gamma) = N_P(\delta)$

Sub-Languages

- For a subset $Q \subseteq P$ we call $\mathcal{L}_Q$ sub-language of $\mathcal{L}_P$
- If $x \in N_P$ then $x^{\downarrow Q} \in N_Q$ denotes the projection of x to Q
- More generally: $N_P^{\downarrow Q} = \{x^{\downarrow Q} : x \in N_P\}$
- If $x \in N_Q$ then $x^{\uparrow P} \in N_P$ denotes the extension of x to P, $x^{\uparrow P} = \{y \in N_P : y^{\downarrow Q} = x\}$

Sub-Languages

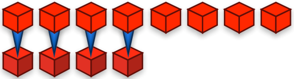
- For a subset $Q \subseteq P$ we call $\mathcal{L}_Q$ sub-language of $\mathcal{L}_P$
- If $x \in N_P$ then $x^{\downarrow Q} \in N_Q$ denotes the projection of x to Q



- More generally: $N_P^{\downarrow Q} = \{x^{\downarrow Q} : x \in N_P\}$
- If $x \in N_Q$ then $x^{\uparrow P} \in N_P$ denotes the extension of x to P, $x^{\uparrow P} = \{y \in N_P : y^{\downarrow Q} = x\}$

Sub-Languages

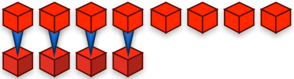
- For a subset $Q \subseteq P$ we call $\mathcal{L}_Q$ sub-language of $\mathcal{L}_P$
- If $x \in N_P$ then $x^{\downarrow Q} \in N_Q$ denotes the projection of x to Q



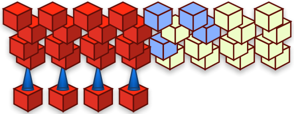
- More generally: $N_P^{\downarrow Q} = \{x^{\downarrow Q} : x \in N_P\}$
- If $x \in N_Q$ then $x^{\uparrow P} \in N_P$ denotes the extension of x to P, $x^{\uparrow P} = \{y \in N_P : y^{\downarrow Q} = x\}$

Sub-Languages

- For a subset $Q \subseteq P$ we call $\mathcal{L}_Q$ sub-language of $\mathcal{L}_P$
- If $x \in N_P$ then $x^{\downarrow Q} \in N_Q$ denotes the projection of x to Q



- More generally: $N_P^{\downarrow Q} = \{x^{\downarrow Q} : x \in N_P\}$
- If $x \in N_Q$ then $x^{\uparrow P} \in N_P$ denotes the extension of x to P, $x^{\uparrow P} = \{y \in N_P : y^{\downarrow Q} = x\}$



Sub-Languages

Let $\gamma, \delta \in \mathcal{L}_P$ and $x \in N_Q$, $Q = \{q_1, \dots, q_m\} \subseteq P$

- $\gamma_{Q \leftarrow x}$ denotes the formula obtained from gamma
  - by replacing each occurrence of q_i by $\perp$ if $x_i = 0$
  - by replacing each occurrence of q_i by $\top$ if $x_i = 1$
- $N_P(\gamma_{Q \leftarrow x}) = N_P(\gamma) \cap x^{\uparrow P}$
- We call x model of δ relative to γ if $\gamma_{Q \leftarrow x} \models \delta$ and write $x \models_\gamma \delta$

Sub-Languages

Let $\gamma, \delta \in \mathcal{L}_P$ and $x \in N_Q$, $Q = \{q_1, \ldots, q_m\} \subseteq P$

- $\gamma_{Q \leftarrow x}$ denotes the formula obtained from gamma
  - by replacing each occurrence of q_i by $\perp$ if $x_i = 0$
  - by replacing each occurrence of q_i by $\top$ if $x_i = 1$
- $N_P(\gamma_{Q \leftarrow x}) = N_P(\gamma) \cap x^{\uparrow P}$
- We call x model of δ relative to γ if $\gamma_{Q \leftarrow x} \models \delta$ and write $x \models_\gamma \delta$

Sub-Languages

Let $\gamma, \delta \in \mathcal{L}_P$ and $x \in N_Q$, $Q = \{q_1, \ldots, q_m\} \subseteq P$

- $\gamma_{Q \leftarrow x}$ denotes the formula obtained from gamma
  - by replacing each occurrence of q_i by $\perp$ if $x_i = 0$
  - by replacing each occurrence of q_i by $\top$ if $x_i = 1$
- $N_P(\gamma_{Q \leftarrow x}) = N_P(\gamma) \cap x^{\uparrow P}$
- We call x model of δ relative to γ if $\gamma_{Q \leftarrow x} \models \delta$ and write $x \models_\gamma \delta$

Sub-Languages

Let $\gamma, \delta \in \mathcal{L}_P$ and $x \in N_Q$, $Q = \{q_1, \ldots, q_m\} \subseteq P$

- $\gamma_{Q \leftarrow x}$ denotes the formula obtained from gamma
  - by replacing each occurrence of q_i by $\perp$ if $x_i = 0$
  - by replacing each occurrence of q_i by $\top$ if $x_i = 1$
- $N_P(\gamma_{Q \leftarrow x}) = N_P(\gamma) \cap x^{\uparrow P}$
- We call x model of δ relative to γ if $\gamma_{Q \leftarrow x} \models \delta$ and write $x \models_\gamma \delta$

Outline

1. Propositional Argumentation Systems
   - Propositional Logic
   - Argumentation Systems
   - Probabilistic Argumentation Systems

2. Argumentation Systems on Set Constraint Logic
   - Set Constraint Logic
   - Constraint-Based Argumentation Systems
   - Introducing Probabilities

Propositional Argumentation System

Definition

Let A and P be two disjoint sets of propositions. If $\xi \in \mathcal{L}_{A \cup P}$, then we call $\mathcal{AS}_P = (\xi, P, A)$ propositional argumentation system.

Example

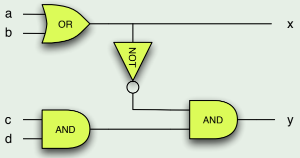
A: assumptions that components work
P: propositions in system description
ξ: system description

Propositional Argumentation System

Definition

Let A and P be two disjoint sets of propositions. If $\xi \in \mathcal{L}_{A \cup P}$, then we call $\mathcal{AS}_P = (\xi, P, A)$ propositional argumentation system.

Example



A: assumptions that components work
P: propositions in system description
ξ: system description

Scenarios

- The set N_A is of particular interest
- Interpretations $s \in N_A$ are called scenarios

Definition

Let $\xi \in \mathcal{L}_{A \cup P}$. A scenario $s \in N_A$ is called

- inconsistent relative to $\xi \Leftrightarrow s \models_\xi \perp$
- consistent relative to ξ else

- The set of all inconsistent scenarios is denoted by $I_A(\xi)$
- The set of all consistent scenarios is denoted by $C_A(\xi)$
- $C_A(\xi) = N_A - I_A(\xi)$

Scenarios

- The set N_A is of particular interest
- Interpretations $s \in N_A$ are called scenarios

Definition

Let $\xi \in \mathcal{L}_{A \cup P}$. A scenario $s \in N_A$ is called

- inconsistent relative to $\xi \Leftrightarrow s \models_\xi \perp$
- consistent relative to ξ else

- The set of all inconsistent scenarios is denoted by $I_A(\xi)$
- The set of all consistent scenarios is denoted by $C_A(\xi)$
- $C_A(\xi) = N_A - I_A(\xi)$

Scenarios

- The set N_A is of particular interest
- Interpretations $s \in N_A$ are called scenarios

Definition

Let $\xi \in \mathcal{L}_{A \cup P}$. A scenario $s \in N_A$ is called

- inconsistent relative to $\xi \Leftrightarrow s \models_\xi \perp$
- consistent relative to ξ else

- The set of all inconsistent scenarios is denoted by $I_A(\xi)$
- The set of all consistent scenarios is denoted by $C_A(\xi)$
- $C_A(\xi) = N_A - I_A(\xi)$

Scenarios

- The set N_A is of particular interest
- Interpretations $s \in N_A$ are called scenarios

Definition

Let $\xi \in \mathcal{L}_{A \cup P}$. A scenario $s \in N_A$ is called

- inconsistent relative to $\xi \Leftrightarrow s \models_{\xi} \perp$
- consistent relative to ξ else

- The set of all inconsistent scenarios is denoted by $I_A(\xi)$
- The set of all consistent scenarios is denoted by $C_A(\xi)$
- $C_A(\xi) = N_A - I_A(\xi)$

Scenarios

- The set N_A is of particular interest
- Interpretations $s \in N_A$ are called scenarios

Definition

Let $\xi \in \mathcal{L}_{A \cup P}$. A scenario $s \in N_A$ is called

- inconsistent relative to $\xi \Leftrightarrow s \models_{\xi} \perp$
- consistent relative to ξ else

- The set of all inconsistent scenarios is denoted by $I_A(\xi)$
- The set of all consistent scenarios is denoted by $C_A(\xi)$
- $C_A(\xi) = N_A - I_A(\xi)$

Scenarios

- The set N_A is of particular interest
- Interpretations $s \in N_A$ are called scenarios

Definition

Let $\xi \in \mathcal{L}_{A \cup P}$. A scenario $s \in N_A$ is called

- inconsistent relative to $\xi \Leftrightarrow s \models_\xi \perp$
- consistent relative to ξ else

- The set of all inconsistent scenarios is denoted by $I_A(\xi)$
- The set of all consistent scenarios is denoted by $C_A(\xi)$
- $C_A(\xi) = N_A - I_A(\xi)$

Supporting Scenarios

- Now, a second propositional sentence $h \in \mathcal{L}_{\mathcal{A} \cup \mathcal{P}}$, called hypothesis, is given
- A scenario $s \in N_A$ is called a
  - quasi-supporting scenario for h relative to $\xi \Leftrightarrow s \models_{\xi} h$
  - supporting scenario for h relative to $\xi \Leftrightarrow s \models_{\xi} h$ and $s \not\models_{\xi} \perp$
  - possibly supporting scenario for h relative to $\xi \Leftrightarrow s \not\models_{\xi} \neg h$
- $QS_A(h, \xi)$: quasi-supporting scenarios for h relative to ξ
- $SP_A(h, \xi)$: supporting scenarios for h relative to ξ
- $PS_A(h, \xi)$: possibly supporting scenarios for h relative to ξ

Supporting Scenarios

- Now, a second propositional sentence $h \in \mathcal{L}_{\mathcal{A} \cup \mathcal{P}}$, called hypothesis, is given
- A scenario $s \in N_A$ is called a
  - quasi-supporting scenario for h relative to $\xi \Leftrightarrow s \models_\xi h$
  - supporting scenario for h relative to $\xi \Leftrightarrow s \models_\xi h$ and $s \not\models_\xi \perp$
  - possibly supporting scenario for h relative to $\xi \Leftrightarrow s \not\models_\xi \neg h$
- $QS_A(h, \xi)$: quasi-supporting scenarios for h relative to ξ
- $SP_A(h, \xi)$: supporting scenarios for h relative to ξ
- $PS_A(h, \xi)$: possibly supporting scenarios for h relative to ξ

Supporting Scenarios

- Now, a second propositional sentence $h \in \mathcal{L}_{\mathcal{A} \cup \mathcal{P}}$, called hypothesis, is given
- A scenario $s \in N_A$ is called a
  - quasi-supporting scenario for h relative to $\xi \Leftrightarrow s \models_\xi h$
  - supporting scenario for h relative to $\xi \Leftrightarrow s \models_\xi h$ and $s \not\models_\xi \perp$
  - possibly supporting scenario for h relative to $\xi \Leftrightarrow s \not\models_\xi \neg h$
- $QS_A(h, \xi)$: quasi-supporting scenarios for h relative to ξ
- $SP_A(h, \xi)$: supporting scenarios for h relative to ξ
- $PS_A(h, \xi)$: possibly supporting scenarios for h relative to ξ

Supporting Scenarios

- Now, a second propositional sentence $h \in \mathcal{L}_{\mathcal{A} \cup \mathcal{P}}$, called hypothesis, is given
- A scenario $s \in N_A$ is called a
  - quasi-supporting scenario for h relative to $\xi \Leftrightarrow s \models_\xi h$
  - supporting scenario for h relative to $\xi \Leftrightarrow s \models_\xi h$ and $s \not\models_\xi \perp$
  - possibly supporting scenario for h relative to $\xi \Leftrightarrow s \not\models_\xi \neg h$
- $QS_A(h, \xi)$: quasi-supporting scenarios for h relative to ξ
- $SP_A(h, \xi)$: supporting scenarios for h relative to ξ
- $PS_A(h, \xi)$: possibly supporting scenarios for h relative to ξ

Supporting Scenarios

- Now, a second propositional sentence $h \in \mathcal{L}_{\mathcal{A} \cup \mathcal{P}}$, called hypothesis, is given
- A scenario $s \in N_A$ is called a
  - quasi-supporting scenario for h relative to $\xi \Leftrightarrow s \models_\xi h$
  - supporting scenario for h relative to $\xi \Leftrightarrow s \models_\xi h$ and $s \not\models_\xi \perp$
  - possibly supporting scenario for h relative to $\xi \Leftrightarrow s \not\models_\xi \neg h$
- $QS_A(h, \xi)$: quasi-supporting scenarios for h relative to ξ
- $SP_A(h, \xi)$: supporting scenarios for h relative to ξ
- $PS_A(h, \xi)$: possibly supporting scenarios for h relative to ξ

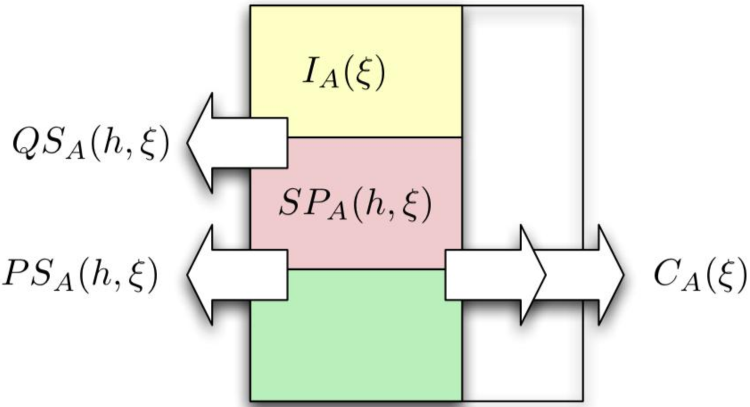
Supporting Scenarios

- Now, a second propositional sentence $h \in \mathcal{L}_{\mathcal{A} \cup \mathcal{P}}$, called hypothesis, is given
- A scenario $s \in N_A$ is called a
  - quasi-supporting scenario for h relative to $\xi \Leftrightarrow s \models_\xi h$
  - supporting scenario for h relative to $\xi \Leftrightarrow s \models_\xi h$ and $s \not\models_\xi \perp$
  - possibly supporting scenario for h relative to $\xi \Leftrightarrow s \not\models_\xi \neg h$
- $QS_A(h, \xi)$: quasi-supporting scenarios for h relative to ξ
- $SP_A(h, \xi)$: supporting scenarios for h relative to ξ
- $PS_A(h, \xi)$: possibly supporting scenarios for h relative to ξ

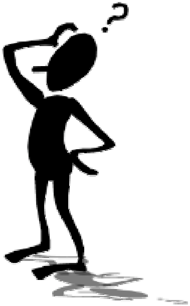
Supporting Scenarios

- Now, a second propositional sentence $h \in \mathcal{L}_{\mathcal{A} \cup \mathcal{P}}$, called hypothesis, is given
- A scenario $s \in N_A$ is called a
  - quasi-supporting scenario for h relative to $\xi \Leftrightarrow s \models_\xi h$
  - supporting scenario for h relative to $\xi \Leftrightarrow s \models_\xi h$ and $s \not\models_\xi \perp$
  - possibly supporting scenario for h relative to $\xi \Leftrightarrow s \not\models_\xi \neg h$
- $QS_A(h, \xi)$: quasi-supporting scenarios for h relative to ξ
- $SP_A(h, \xi)$: supporting scenarios for h relative to ξ
- $PS_A(h, \xi)$: possibly supporting scenarios for h relative to ξ

N_A



Example



Outline

1. Propositional Argumentation Systems
   - Propositional Logic
   - Argumentation Systems
   - Probabilistic Argumentation Systems
2. Argumentation Systems on Set Constraint Logic
   - Set Constraint Logic
   - Constraint-Based Argumentation Systems
   - Introducing Probabilities

Assigning Probabilities

- We link every assumption $a_i \in A$ to a prior probability π_i
- The π_i are assumed to be stochastically independent

Definition

A probabilistic argumentation system is a quadruple $\mathcal{PAS}_P = (\xi, P, A, \Pi)$, where $\Pi = \{\pi_1, \ldots, \pi_m\}$ denotes the set of probabilities assigned to the assumptions $a_i \in A$.

Assigning Probabilities

- We link every assumption $a_i \in A$ to a prior probability π_i
- The π_i are assumed to be stochastically independent

Definition

A probabilistic argumentation system is a quadruple $\mathcal{PAS}_P = (\xi, P, A, \Pi)$, where $\Pi = \{\pi_1, \ldots, \pi_m\}$ denotes the set of probabilities assigned to the assumptions $a_i \in A$.

Assigning Probabilities

- We link every assumption $a_i \in A$ to a prior probability π_i
- The π_i are assumed to be stochastically independent

Definition

A probabilistic argumentation system is a quadruple $\mathcal{PAS}_P = (\xi, P, A, \Pi)$, where $\Pi = \{\pi_1, \ldots, \pi_m\}$ denotes the set of probabilities assigned to the assumptions $a_i \in A$.

Degree of Support and Possibility

- Let $s = \{x_1, \ldots, x_m\}$ be a scenario in N_A
- The prior probability of s is determined by

$$p(s) = \prod_{i=1}^{m} \pi_i^{x_i} \cdot (1 - \pi_i)^{(1-x_i)}$$

- For $S \subseteq N_A$ we define

$$p(S) = \sum_{s \in S} p(s)$$

Degree of Support and Possibility

- Let $s = \{x_1, \ldots, x_m\}$ be a scenario in N_A
- The prior probability of s is determined by

$$p(s) = \prod_{i=1}^{m} \pi_i^{x_i} \cdot (1 - \pi_i)^{(1-x_i)}$$

- For $S \subseteq N_A$ we define

$$p(S) = \sum_{s \in S} p(s)$$

Degree of Support and Possibility

- Let $s = \{x_1, \ldots, x_m\}$ be a scenario in N_A
- The prior probability of s is determined by

$$p(s) = \prod_{i=1}^{m} \pi_i^{x_i} \cdot (1 - \pi_i)^{(1 - x_i)}$$

- For $S \subseteq N_A$ we define

$$p(S) = \sum_{s \in S} p(s)$$

Degree of Support and Possibility

- For $h \in \mathcal{L}_{A \cup P}$ we call $dqs(h, \xi) = p(QS_A(h, \xi))$ degree of quasi-support
- But inconsistent scenarios "are not allowed", i.e.

$$p'(s) = p(s|C_A(\xi)) = \begin{cases} p(s)/p(C_A(\xi)), & \text{if } s \in C_A(\xi), \\ 0, & \text{otherwise.} \end{cases}$$

- $dsp(h, \xi) = p'(SP_A(h, \xi))$ is called degree of support
- $dps(h, \xi) = p'(PS_A(h, \xi))$ is called degree of possibility

Degree of Support and Possibility

- For $h \in \mathcal{L}_{A \cup P}$ we call $dqs(h, \xi) = p(QS_A(h, \xi))$ degree of quasi-support
- But inconsistent scenarios "are not allowed", i.e.

$$p'(s) = p(s | C_A(\xi)) = \begin{cases} p(s)/p(C_A(\xi)), & \text{if } s \in C_A(\xi), \\ 0, & \text{otherwise.} \end{cases}$$

- $dsp(h, \xi) = p'(SP_A(h, \xi))$ is called degree of support
- $dps(h, \xi) = p'(PS_A(h, \xi))$ is called degree of possibility

Degree of Support and Possibility

- For $h \in \mathcal{L}_{A \cup P}$ we call $dqs(h, \xi) = p(QS_A(h, \xi))$ degree of quasi-support
- But inconsistent scenarios "are not allowed", i.e.

$$p'(s) = p(s|C_A(\xi)) = \begin{cases} p(s)/p(C_A(\xi)), & \text{if } s \in C_A(\xi), \\ 0, & \text{otherwise.} \end{cases}$$

- $dsp(h, \xi) = p'(SP_A(h, \xi))$ is called degree of support
- $dps(h, \xi) = p'(PS_A(h, \xi))$ is called degree of possibility

Degree of Support and Possibility

- For $h \in \mathcal{L}_{A \cup P}$ we call $dqs(h, \xi) = p(QS_A(h, \xi))$ degree of quasi-support
- But inconsistent scenarios "are not allowed", i.e.

$$p'(s) = p(s|C_A(\xi)) = \begin{cases} p(s)/p(C_A(\xi)), & \text{if } s \in C_A(\xi), \\ 0, & \text{otherwise.} \end{cases}$$

- $dsp(h, \xi) = p'(SP_A(h, \xi))$ is called degree of support
- $dps(h, \xi) = p'(PS_A(h, \xi))$ is called degree of possibility

Degree of Support and Possibility

Whenever $\xi \neq \bot$:

- $dps(\bot, \xi) = dsp(\bot, \xi) = 0$
- $dps(\top, \xi) = dsp(\top, \xi) = 1$
- $h_1 \models h_2 \Rightarrow dsp(h_1, \xi) \leq dsp(h_2, \xi), dps(h_1, \xi) \leq dps(h_2, \xi)$
- $h_1 \equiv h_2 \Rightarrow dsp(h_1, \xi) = dsp(h_2, \xi), dps(h_1, \xi) = dps(h_2, \xi)$
- $dsp(h, \xi) \leq dps(h, \xi)$

Degree of Support and Possibility

Whenever $\xi \neq \perp$:

- $dps(\perp, \xi) = dsp(\perp, \xi) = 0$
- $dps(\top, \xi) = dsp(\top, \xi) = 1$
- $h_1 \models h_2 \Rightarrow dsp(h_1, \xi) \leq dsp(h_2, \xi), dps(h_1, \xi) \leq dps(h_2, \xi)$
- $h_1 \equiv h_2 \Rightarrow dsp(h_1, \xi) = dsp(h_2, \xi), dps(h_1, \xi) = dps(h_2, \xi)$
- $dsp(h, \xi) \leq dps(h, \xi)$

Degree of Support and Possibility

Whenever $\xi \neq \bot$:

- $dps(\bot, \xi) = dsp(\bot, \xi) = 0$
- $dps(\top, \xi) = dsp(\top, \xi) = 1$
- $h_1 \models h_2 \Rightarrow dsp(h_1, \xi) \leq dsp(h_2, \xi), dps(h_1, \xi) \leq dps(h_2, \xi)$
- $h_1 \equiv h_2 \Rightarrow dsp(h_1, \xi) = dsp(h_2, \xi), dps(h_1, \xi) = dps(h_2, \xi)$
- $dsp(h, \xi) \leq dps(h, \xi)$

Degree of Support and Possibility

Whenever $\xi \neq \bot$:

- $dps(\bot, \xi) = dsp(\bot, \xi) = 0$
- $dps(\top, \xi) = dsp(\top, \xi) = 1$
- $h_1 \models h_2 \Rightarrow dsp(h_1, \xi) \leq dsp(h_2, \xi)$, $dps(h_1, \xi) \leq dps(h_2, \xi)$
- $h_1 \equiv h_2 \Rightarrow dsp(h_1, \xi) = dsp(h_2, \xi)$, $dps(h_1, \xi) = dps(h_2, \xi)$
- $dsp(h, \xi) \leq dps(h, \xi)$

Degree of Support and Possibility

Whenever $\xi \neq \bot$:

- $dps(\bot, \xi) = dsp(\bot, \xi) = 0$
- $dps(\top, \xi) = dsp(\top, \xi) = 1$
- $h_1 \models h_2 \Rightarrow dsp(h_1, \xi) \leq dsp(h_2, \xi)$, $dps(h_1, \xi) \leq dps(h_2, \xi)$
- $h_1 \equiv h_2 \Rightarrow dsp(h_1, \xi) = dsp(h_2, \xi)$, $dps(h_1, \xi) = dps(h_2, \xi)$
- $dsp(h, \xi) \leq dps(h, \xi)$

Outline

Frames and Constraints

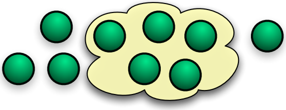
- Given a finite set of variables $V = \{v_1, \dots, v_n\}$
- Every $v \in V$ has possible values out of Θ_v, its frame
- An expression $< v \in X >$, $X \subseteq \Theta$, is called set constraint
- An assignment is a set constraint $< v \in \{\theta_i\} >$, $\theta_i \in \Theta_v$

Frames and Constraints

- Given a finite set of variables $V = \{v_1, \ldots, v_n\}$
- Every $v \in V$ has possible values out of Θ_v, its frame
- An expression $< v \in X >$, $X \subseteq \Theta$, is called set constraint
- An assignment is a set constraint $< v \in \{\theta_i\} >$, $\theta_i \in \Theta_v$

Frames and Constraints

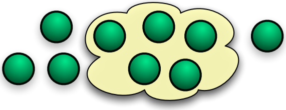
- Given a finite set of variables $V = \{v_1, \ldots, v_n\}$
- Every $v \in V$ has possible values out of Θ_v, its frame
- An expression $< v \in X >$, $X \subseteq \Theta$, is called set constraint



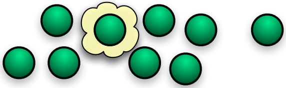
- An assignment is a set constraint $< v \in \{\theta_i\} >$, $\theta_i \in \Theta_v$

Frames and Constraints

- Given a finite set of variables $V = \{v_1, \ldots, v_n\}$
- Every $v \in V$ has possible values out of Θ_v, its frame
- An expression $< v \in X >$, $X \subseteq \Theta$, is called set constraint



- An assignment is a set constraint $< v \in \{\theta_i\} >$, $\theta_i \in \Theta_v$



SCL-Formulas

- set constraints, $\perp$ and $\top$ are SCL-formulas
- if γ is a SCL-formula, then $\neg\gamma$ is a SCL-formula
- ff γ and δ are SCL-formulas, then $(\gamma \wedge \delta)$, $(\gamma \vee \delta)$, $(\gamma \rightarrow \delta)$ and $(\gamma \leftrightarrow \delta)$ are SCL-formulas

SCL-Formulas

- set constraints, $\perp$ and $\top$ are SCL-formulas
- if γ is a SCL-formula, then $\neg\gamma$ is a SCL-formula
- ff γ and δ are SCL-formulas, then $(\gamma \wedge \delta)$, $(\gamma \vee \delta)$, $(\gamma \rightarrow \delta)$ and $(\gamma \leftrightarrow \delta)$ are SCL-formulas

SCL-Formulas

- set constraints, $\perp$ and $\top$ are SCL-formulas
- if γ is a SCL-formula, then $\neg\gamma$ is a SCL-formula
- ff γ and δ are SCL-formulas, then $(\gamma \wedge \delta)$, $(\gamma \vee \delta)$, $(\gamma \rightarrow \delta)$ and $(\gamma \leftrightarrow \delta)$ are SCL-formulas

SCL-Formulas

- Assigning a value to every $v \in V$ is called interpretation
- The set of all possible interpretations is denoted by N_V
- An interpretation is in fact a point $x = \{x_1, \ldots, x_n\}$ in N_V
- For a fixed interpretation x, the truth value of $< v_i \in X >$ is 1 whenever $x_i \in X$ and 0 otherwise
- The truth value of a formula is determined like for propositional logic

SCL-Formulas

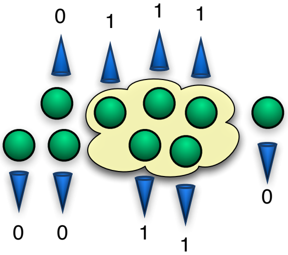
- Assigning a value to every $v \in V$ is called interpretation
- The set of all possible interpretations is denoted by N_V
- An interpretation is in fact a point $x = \{x_1, \ldots, x_n\}$ in N_V
- For a fixed interpretation x, the truth value of $< v_i \in X >$ is 1 whenever $x_i \in X$ and 0 otherwise
- The truth value of a formula is determined like for propositional logic

SCL-Formulas

- Assigning a value to every $v \in V$ is called interpretation
- The set of all possible interpretations is denoted by N_V
- An interpretation is in fact a point $x = \{x_1, \ldots, x_n\}$ in N_V
- For a fixed interpretation x, the truth value of $< v_i \in X >$ is 1 whenever $x_i \in X$ and 0 otherwise
- The truth value of a formula is determined like for propositional logic

SCL-Formulas

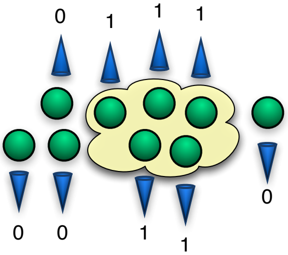
- Assigning a value to every $v \in V$ is called interpretation
- The set of all possible interpretations is denoted by N_V
- An interpretation is in fact a point $x = \{x_1, \ldots, x_n\}$ in N_V
- For a fixed interpretation x, the truth value of $< v_i \in X >$ is 1 whenever $x_i \in X$ and 0 otherwise



- The truth value of a formula is determined like for propositional logic

SCL-Formulas

- Assigning a value to every $v \in V$ is called interpretation
- The set of all possible interpretations is denoted by N_V
- An interpretation is in fact a point $x = \{x_1, \ldots, x_n\}$ in N_V
- For a fixed interpretation x, the truth value of $< v_i \in X >$ is 1 whenever $x_i \in X$ and 0 otherwise



- The truth value of a formula is determined like for propositional logic

SCL-Formulas

- $N(\gamma) \subseteq N_V$ denotes all interpretations for which γ is true
- $\gamma \models \delta$ if, and only if, $N(\gamma) \subseteq N(\delta)$
- $\gamma \equiv \delta$ if, and only if, $N(\gamma) = N(\delta)$
- Let $\gamma \in \mathcal{L}_V$ and $x \in N_Q$ with $Q \subseteq V$. $\gamma_{Q \leftarrow x}$ is the formula obtained by replacing each set constraint $< v_i \in X >$ by $\top$ if $x_i \in X$ and by $\perp$ otherwise
- For $\delta \in \mathcal{L}_V$ then $x \models_\gamma \delta$ means $\gamma_{Q \leftarrow x} \models \delta$

SCL-Formulas

- $N(\gamma) \subseteq N_V$ denotes all interpretations for which γ is true
- $\gamma \models \delta$ if, and only if, $N(\gamma) \subseteq N(\delta)$
- $\gamma \equiv \delta$ if, and only if, $N(\gamma) = N(\delta)$
- Let $\gamma \in \mathcal{L}_V$ and $x \in N_Q$ with $Q \subseteq V$. $\gamma_{Q \leftarrow x}$ is the formula obtained by replacing each set constraint $< v_i \in X >$ by $\top$ if $x_i \in X$ and by $\perp$ otherwise
- For $\delta \in \mathcal{L}_V$ then $x \models_\gamma \delta$ means $\gamma_{Q \leftarrow x} \models \delta$

SCL-Formulas

- $N(\gamma) \subseteq N_V$ denotes all interpretations for which γ is true
- $\gamma \models \delta$ if, and only if, $N(\gamma) \subseteq N(\delta)$
- $\gamma \equiv \delta$ if, and only if, $N(\gamma) = N(\delta)$
- Let $\gamma \in \mathcal{L}_V$ and $x \in N_Q$ with $Q \subseteq V$. $\gamma_{Q \leftarrow x}$ is the formula obtained by replacing each set constraint $< v_i \in X >$ by $\top$ if $x_i \in X$ and by $\perp$ otherwise
- For $\delta \in \mathcal{L}_V$ then $x \models_\gamma \delta$ means $\gamma_{Q \leftarrow x} \models \delta$

SCL-Formulas

- $N(\gamma) \subseteq N_V$ denotes all interpretations for which γ is true
- $\gamma \models \delta$ if, and only if, $N(\gamma) \subseteq N(\delta)$
- $\gamma \equiv \delta$ if, and only if, $N(\gamma) = N(\delta)$
- Let $\gamma \in \mathcal{L}_V$ and $x \in N_Q$ with $Q \subseteq V$. $\gamma_{Q \leftarrow x}$ is the formula obtained by replacing each set constraint $< v_i \in X >$ by $\top$ if $x_i \in X$ and by $\perp$ otherwise
- For $\delta \in \mathcal{L}_V$ then $x \models_\gamma \delta$ means $\gamma_{Q \leftarrow x} \models \delta$

SCL-Formulas

- $N(\gamma) \subseteq N_V$ denotes all interpretations for which γ is true
- $\gamma \models \delta$ if, and only if, $N(\gamma) \subseteq N(\delta)$
- $\gamma \equiv \delta$ if, and only if, $N(\gamma) = N(\delta)$
- Let $\gamma \in \mathcal{L}_V$ and $x \in N_Q$ with $Q \subseteq V$. $\gamma_{Q \leftarrow x}$ is the formula obtained by replacing each set constraint $< v_i \in X >$ by $\top$ if $x_i \in X$ and by $\perp$ otherwise
- For $\delta \in \mathcal{L}_V$ then $x \models_\gamma \delta$ means $\gamma_{Q \leftarrow x} \models \delta$

Outline

Constraint-Based Argumentation Systems

Definition

Let $V = \{v_1, \ldots, v_n\}$ and $E = \{e_1, \ldots, e_m\}$ be two sets of variables. If $\xi \in \mathcal{L}_{V \cup E}$ then we call $\mathcal{AS}_{\mathcal{C}} = (\xi, V, E)$ constraint-based argumentation system.

- The elements of E are called environmental variables
- One can introduce in the same way than for propositional logic the notions consistent/inconsistent, quasi-supporting, supporting and possibly supporting scenarios

Constraint-Based Argumentation Systems

Definition

Let $V = \{v_1, \ldots, v_n\}$ and $E = \{e_1, \ldots, e_m\}$ be two sets of variables. If $\xi \in \mathcal{L}_{V \cup E}$ then we call $\mathcal{AS}_{\mathcal{C}} = (\xi, V, E)$ constraint-based argumentation system.

- The elements of E are called environmental variables
- One can introduce in the same way than for propositional logic the notions consistent/inconsistent, quasi-supporting, supporting and possibly supporting scenarios

Outline

Probabilistic Argumentation Systems

- Suppose every Θ_{e_i} is finite for $e_i \in E$
- Let $\pi_{ij} = p(e_i = \theta_{ij})$ with $\theta_i \in \Theta_{e_i}$ and $\sum_j \pi_{ij} = 1$
- The probability distribution assigned to e_i is denoted by π_i

Definition

We call $\mathcal{PAS}_C(\xi, V, E, \Pi)$ with $\Pi = \{\pi_1, \ldots, \pi_m\}$ probabilistic constraint-based argumentation system.

Probabilistic Argumentation Systems

- Suppose every Θ_{e_i} is finite for $e_i \in E$
- Let $\pi_{ij} = p(e_i = \theta_{ij})$ with $\theta_i \in \Theta_{e_i}$ and $\sum_j \pi_{ij} = 1$
- The probability distribution assigned to e_i is denoted by π_i

Definition

We call $\mathcal{PAS}_C(\xi, V, E, \Pi)$ with $\Pi = \{\pi_1, \ldots, \pi_m\}$ probabilistic constraint-based argumentation system.

Probabilistic Argumentation Systems

- Suppose every Θ_{e_i} is finite for $e_i \in E$
- Let $\pi_{ij} = p(e_i = \theta_{ij})$ with $\theta_i \in \Theta_{e_i}$ and $\sum_j \pi_{ij} = 1$
- The probability distribution assigned to e_i is denoted by π_i

Definition

We call $\mathcal{PAS}_C(\xi, V, E, \Pi)$ with $\Pi = \{\pi_1, \ldots, \pi_m\}$ probabilistic constraint-based argumentation system.

Probabilistic Argumentation Systems

- Suppose every Θ_{e_i} is finite for $e_i \in E$
- Let $\pi_{ij} = p(e_i = \theta_{ij})$ with $\theta_i \in \Theta_{e_i}$ and $\sum_j \pi_{ij} = 1$
- The probability distribution assigned to e_i is denoted by π_i

Definition

We call $\mathcal{PAS}_C(\xi, V, E, \Pi)$ with $\Pi = \{\pi_1, \ldots, \pi_m\}$ probabilistic constraint-based argumentation system.

Probabilistic Argumentation Systems

- Let $s = (\theta_{1j}, \ldots, \theta_{mj})$ be a particular scenario in N_E. The probability of s is

$$p(s) = \prod_{i=1}^{m} p(e_i = \theta_{ij}) = \prod_{i=1}^{m} \pi_{ij}.$$

- The probability of $S \subseteq N_E$ is then $p(S) = \sum_{s \in S} p(s)$
- Degree of quasi-support / degree of support / degree of possibility can be defined like for the propositional case

Probabilistic Argumentation Systems

- Let $s = (\theta_{1j}, \ldots, \theta_{mj})$ be a particular scenario in N_E. The probability of s is

$$p(s) = \prod_{i=1}^{m} p(e_i = \theta_{ij}) = \prod_{i=1}^{m} \pi_{ij}.$$

- The probability of $S \subseteq N_E$ is then $p(S) = \sum_{s \in S} p(s)$
- Degree of quasi-support / degree of support / degree of possibility can be defined like for the propositional case

Probabilistic Argumentation Systems

- Let $s = (\theta_{1j}, \ldots, \theta_{mj})$ be a particular scenario in N_E. The probability of s is

$$p(s) = \prod_{i=1}^{m} p(e_i = \theta_{ij}) = \prod_{i=1}^{m} \pi_{ij}.$$

- The probability of $S \subseteq N_E$ is then $p(S) = \sum_{s \in S} p(s)$
- Degree of quasi-support / degree of support / degree of possibility can be defined like for the propositional case


Ende