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

 $($ \Box $)$ $($ \Box $)$

 $\mathbf{y} = \mathbf{z}$. The \mathbf{y}

Outline

[Propositional Argumentation Systems](#page-2-0)

- **[Propositional Logic](#page-2-0)**
- **[Argumentation Systems](#page-30-0)**
- [Probabilistic Argumentation Systems](#page-48-0)

[Argumentation Systems on Set Constraint Logic](#page-64-0)

- **[Set Constraint Logic](#page-64-0)**
- **[Constraint-Based Argumentation Systems](#page-82-0)**
- **•** [Introducing Probabilities](#page-85-0)

[Propositional Logic](#page-2-0) [Argumentation Systems](#page-30-0) [Probabilistic Argumentation Systems](#page-48-0)

Outline

[Propositional Argumentation Systems](#page-2-0)

- **[Propositional Logic](#page-2-0)**
- **[Argumentation Systems](#page-30-0)**
- **[Probabilistic Argumentation Systems](#page-48-0)**

2 [Argumentation Systems on Set Constraint Logic](#page-64-0)

- [Set Constraint Logic](#page-64-0)
- [Constraint-Based Argumentation Systems](#page-82-0) \bullet
- **•** [Introducing Probabilities](#page-85-0)

 $\left\{ \begin{array}{ccc} \square & \rightarrow & \left\langle \bigoplus \gamma & \right\rangle & \left\langle \begin{array}{ccc} \exists & \gamma & \gamma & \exists \gamma \end{array} \right\rangle \end{array} \right.$

[Propositional Logic](#page-2-0) [Argumentation Systems](#page-30-0) [Probabilistic Argumentation Systems](#page-48-0)

Propositional Sentences

- Propositions: Statements that can be either true or false 0 1
- Impossible statement: ⊥, the one which is always true: T
- 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 Logic](#page-2-0) [Argumentation Systems](#page-30-0) [Probabilistic Argumentation Systems](#page-48-0)

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

 $\left\{ \begin{array}{ccc} \square & \rightarrow & \left\langle \bigoplus \gamma & \right\rangle & \left\langle \begin{array}{ccc} \exists & \gamma & \gamma & \exists \gamma \end{array} \right\rangle \end{array} \right.$

[Propositional Logic](#page-2-0) [Argumentation Systems](#page-30-0) [Probabilistic Argumentation Systems](#page-48-0)

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

 $\left\{ \begin{array}{ccc} \square & \rightarrow & \left\langle \bigoplus \gamma & \right\rangle & \left\langle \begin{array}{ccc} \exists & \gamma & \gamma & \exists \gamma \end{array} \right\rangle \end{array} \right.$

[Propositional Logic](#page-2-0) [Argumentation Systems](#page-30-0) [Probabilistic Argumentation Systems](#page-48-0)

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

(ロ) (伊)

[Propositional Logic](#page-2-0) [Argumentation Systems](#page-30-0) [Probabilistic Argumentation Systems](#page-48-0)

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
- \bullet 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

 4 ロ > 4 何 > 4 ミ > 4 ミ >

[Propositional Logic](#page-2-0) [Argumentation Systems](#page-30-0) [Probabilistic Argumentation Systems](#page-48-0)

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
- \bullet 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

 4 ロ > 4 何 > 4 ミ > 4 ミ >

[Propositional Logic](#page-2-0) [Argumentation Systems](#page-30-0) [Probabilistic Argumentation Systems](#page-48-0)

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
- \bullet 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

 4 ロ } 4 6 } 4 \geq } 4 \geq }

[Propositional Logic](#page-2-0) [Argumentation Systems](#page-30-0) [Probabilistic Argumentation Systems](#page-48-0)

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
- \bullet 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 Logic](#page-2-0) [Argumentation Systems](#page-30-0) [Probabilistic Argumentation Systems](#page-48-0)

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
- \bullet 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

[Propositional Logic](#page-2-0) [Argumentation Systems](#page-30-0) [Probabilistic Argumentation Systems](#page-48-0)

The meaning of a propositional sentence:

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

Semantics

[Propositional Logic](#page-2-0) [Argumentation Systems](#page-30-0) [Probabilistic Argumentation Systems](#page-48-0)

The meaning of a propositional sentence:

 $\left\{ \begin{array}{ccc} 1 & 0 & 0 \\ 0 & 1 & 0 \end{array} \right.$

 \bullet An assignment of truth values to P is called interpretation

 N_P denotes the set of all 2^n interpretations

Semantics

The meaning of a propositional sentence:

0 1 1 0 1 1 0 1 0 0 1 0 0

 \bullet An assignment of truth values to P is called interpretation

 $\gamma \,\|\, \delta \,\| \perp \,\| \top \,\| \neg \gamma \,\| \, \gamma \wedge \delta \,\| \, \gamma \vee \delta \,\| \, \gamma \rightarrow \delta \,\| \, \gamma \leftrightarrow \delta$ 0 0 0 1 1 0 0 1 1

1 | 1 | 0 1 0 1 1 1 1 1

[Propositional Logic](#page-2-0) [Argumentation Systems](#page-30-0) [Probabilistic Argumentation Systems](#page-48-0)

 N_P denotes the set of all 2^n interpretations

Semantics

The meaning of a propositional sentence:

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

[Propositional Logic](#page-2-0) [Argumentation Systems](#page-30-0) [Probabilistic Argumentation Systems](#page-48-0)

[Propositional Logic](#page-2-0) [Argumentation Systems](#page-30-0) [Probabilistic Argumentation Systems](#page-48-0)

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

- \bullet 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 $(\gamma \equiv \delta) \Leftrightarrow N_P(\gamma) = N_P(\delta)$

[Propositional Logic](#page-2-0) [Argumentation Systems](#page-30-0) [Probabilistic Argumentation Systems](#page-48-0)

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

- \bullet 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

- \bullet 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

- \bullet 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

- \bullet 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

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

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

 4 ロ } 4 6 } 4 \geq } 4 \geq }

[Propositional Logic](#page-2-0) [Argumentation Systems](#page-30-0) [Probabilistic Argumentation Systems](#page-48-0)

• 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, $x^{\uparrow P} = \{y \in N_P : y^{\downarrow Q} = x\}$

K ロ ▶ K 伊 ▶ K ヨ ▶ K ヨ ▶

[Propositional Logic](#page-2-0) [Argumentation Systems](#page-30-0) [Probabilistic Argumentation Systems](#page-48-0)

• 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 AMMAM
- 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, $x^{\uparrow P} = \{y \in N_P : y^{\downarrow Q} = x\}$

 4 ロ } 4 6 } 4 \geq } 4 \geq }

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

 $\left\{ \begin{array}{ccc} \square & \rightarrow & \left\langle \bigoplus \gamma & \right\rangle & \left\langle \begin{array}{ccc} \exists & \gamma & \gamma & \exists \gamma \end{array} \right\rangle \end{array} \right.$

[Propositional Logic](#page-2-0) [Argumentation Systems](#page-30-0) [Probabilistic Argumentation Systems](#page-48-0)

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

- \bullet $\gamma_{\Omega \leftarrow x}$ 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 $x_i = 1$

$$
\quad \bullet \ \ N_P(\gamma_{\mathsf{Q} \leftarrow {\mathsf{x}}}) = N_P(\gamma) \cap {\mathsf{x}}^\uparrow
$$

We call **x** model of δ relative to γ if $\gamma_{\mathsf{Q}\leftarrow\mathbf{x}}\models\delta$ and write $x \models_{\gamma} \delta$

[Propositional Logic](#page-2-0) [Argumentation Systems](#page-30-0) [Probabilistic Argumentation Systems](#page-48-0)

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

- \bullet $\gamma_{\Omega \leftarrow x}$ 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 $x_i = 1$
- $N_P(\gamma_{\mathsf{Q}\leftarrow{\mathsf{x}}})=N_P(\gamma)\cap{\mathsf{x}}^{\uparrow P}$

We call **x** model of δ relative to γ if $\gamma_{\mathsf{Q}\leftarrow\mathbf{x}}\models\delta$ and write $x \models_{\gamma} \delta$

[Propositional Logic](#page-2-0) [Argumentation Systems](#page-30-0) [Probabilistic Argumentation Systems](#page-48-0)

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

- \bullet $\gamma_{\Omega \leftarrow x}$ 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 $x_i = 1$

$$
\bullet \ \mathsf{N}_\mathsf{P}(\gamma_{\mathsf{Q} \leftarrow \mathsf{x}}) = \mathsf{N}_\mathsf{P}(\gamma) \cap \mathsf{x}^{\uparrow \mathsf{P}}
$$

We call **x** model of δ relative to γ if $\gamma_{\mathsf{Q}\leftarrow\mathbf{x}}\models\delta$ and write $x \models_{\gamma} \delta$

[Propositional Logic](#page-2-0) [Argumentation Systems](#page-30-0) [Probabilistic Argumentation Systems](#page-48-0)

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

- \bullet $\gamma_{\Omega \leftarrow x}$ 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 $x_i = 1$

$$
\bullet \ \mathsf{N}_P(\gamma_{\mathsf{Q} \leftarrow {\mathsf{x}}}) = \mathsf{N}_P(\gamma) \cap {\mathsf{x}}^{\uparrow P}
$$

We call $\boldsymbol{\mathsf{x}}$ model of δ relative to γ if $\gamma_{\mathsf{Q}\leftarrow\boldsymbol{\mathsf{x}}}\models\delta$ and write $x \models_{\gamma} \delta$

Outline

[Propositional Logic](#page-2-0) [Argumentation Systems](#page-30-0) [Probabilistic Argumentation Systems](#page-48-0)

[Propositional Argumentation Systems](#page-2-0)

- **[Propositional Logic](#page-2-0)**
- **[Argumentation Systems](#page-30-0)**
- **[Probabilistic Argumentation Systems](#page-48-0)**

2 [Argumentation Systems on Set Constraint Logic](#page-64-0)

- [Set Constraint Logic](#page-64-0)
- [Constraint-Based Argumentation Systems](#page-82-0)
- **•** [Introducing Probabilities](#page-85-0)

 $\left\{ \begin{array}{ccc} \square & \rightarrow & \left\langle \bigoplus \gamma & \right\rangle & \left\langle \begin{array}{ccc} \exists & \gamma & \gamma & \exists \gamma \end{array} \right\rangle \end{array} \right.$

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 $AS_P = (\xi, P, A)$ propositional argumentation system.

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

 $\left\{ \begin{array}{ccc} \square & \rightarrow & \left\langle \bigoplus \gamma & \right\rangle & \left\langle \begin{array}{ccc} \exists & \gamma & \gamma & \exists \gamma \end{array} \right\rangle \end{array} \right.$

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 $AS_P = (\xi, P, A)$ propositional argumentation system.

Example OR NOT AND AND a b c d x y A: assumptions that components work P: propositions in system description ξ: system description

(ロ) (伊)

Scenarios

[Propositional Logic](#page-2-0) [Argumentation Systems](#page-30-0) [Probabilistic Argumentation Systems](#page-48-0)

\bullet The set N_A is of particular interest

• Interpretations $s \in N_A$ are called scenarios

- Let $\xi \in \mathcal{L}_{A\cup P}$. A scenario $s \in N_A$ is called
	- **o** inconsistent relative to $\xi \Leftrightarrow$ s \models _{$\xi \perp$}
	- \bullet 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 $C_A(\xi)$
	- \bullet $C_A(\xi) = N_A I_A(\xi)$

 $\left\{ \begin{array}{ccc} \square & \rightarrow & \left\langle \bigoplus \gamma & \right\rangle & \left\langle \begin{array}{ccc} \exists & \gamma & \gamma & \exists \gamma \end{array} \right\rangle \end{array} \right.$

Scenarios

[Propositional Logic](#page-2-0) [Argumentation Systems](#page-30-0) [Probabilistic Argumentation Systems](#page-48-0)

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

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

- **o** inconsistent relative to $\xi \Leftrightarrow$ s \models _{$\xi \perp$}
- \bullet 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 $C_A(\xi)$
- \bullet $C_A(\xi) = N_A I_A(\xi)$

 $\left\{ \begin{array}{ccc} \pm & \pm & \pm \end{array} \right.$ and $\left\{ \begin{array}{ccc} \pm & \pm & \pm \end{array} \right.$ and $\left\{ \begin{array}{ccc} \pm & \pm & \pm \end{array} \right.$

Scenarios

[Propositional Logic](#page-2-0) [Argumentation Systems](#page-30-0) [Probabilistic Argumentation Systems](#page-48-0)

- \bullet 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 _f \bot
- \bullet 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 $C_A(\xi)$
- \bullet $C_A(\xi) = N_A I_A(\xi)$

 $(1 - 1)$ $(1 - 1)$ $(1 - 1)$ $(1 - 1)$ $(1 - 1)$ $(1 - 1)$ $(1 - 1)$
Scenarios

[Propositional Logic](#page-2-0) [Argumentation Systems](#page-30-0) [Probabilistic Argumentation Systems](#page-48-0)

- \bullet 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 _f \bot
- **o** 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 $C_{\Lambda}(\xi)$
- \bullet $C_A(\xi) = N_A I_A(\xi)$

≮ロト ⊀何ト ⊀ ヨト ⊀ ヨト

Scenarios

[Propositional Logic](#page-2-0) [Argumentation Systems](#page-30-0) [Probabilistic Argumentation Systems](#page-48-0)

- \bullet 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 _f \bot
- **o** 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 $C_A(\xi)$
- \bullet $C_A(\xi) = N_A I_A(\xi)$

Scenarios

[Propositional Logic](#page-2-0) [Argumentation Systems](#page-30-0) [Probabilistic Argumentation Systems](#page-48-0)

- \bullet 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 _f \bot
- **o** 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 $C_A(\xi)$
- \bullet $C_A(\xi) = N_A I_A(\xi)$

[Propositional Logic](#page-2-0) [Argumentation Systems](#page-30-0) [Probabilistic Argumentation Systems](#page-48-0)

Supporting Scenarios

- Now, a second propositional sentence $h \in \mathcal{L}_{A\cup P}$, 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** $s \not\models_{\xi} \bot$
	- **possibly supporting scenario for h relative to** $\xi \Leftrightarrow s \not\models_{\xi} \neg h$
- \bullet QS_A(h, ξ): quasi-supporting scenarios for h relative to ξ
- \bullet SP_A(h, ξ): supporting scenarios for h relative to ξ
- **PS**_A(h, ξ): possibly supporting scenarios for h relative to ξ

 $\left\{ \begin{array}{ccc} \pm & \pm & \pm \end{array} \right.$ and $\left\{ \begin{array}{ccc} \pm & \pm & \pm \end{array} \right.$ and $\left\{ \begin{array}{ccc} \pm & \pm & \pm \end{array} \right.$

Supporting Scenarios

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

 $\left\{ \begin{array}{ccc} \pm & \pm & \pm \end{array} \right.$ and $\left\{ \begin{array}{ccc} \pm & \pm & \pm \end{array} \right.$ and $\left\{ \begin{array}{ccc} \pm & \pm & \pm \end{array} \right.$

Supporting Scenarios

- Now, a second propositional sentence $h \in \mathcal{L}_{A\cup P}$, 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** $s \not\models_{\xi} \bot$
	- **possibly supporting scenario for h relative to** $\xi \Leftrightarrow s \not\models_{\xi} \neg h$
- \bullet QS_A(h, ξ): quasi-supporting scenarios for h relative to ξ
- \bullet SP_A(h, ξ): 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 $h \in \mathcal{L}_{A\cup P}$, 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** $s \not\models_{\xi} \bot$
	- **possibly supporting scenario for h relative to** $\xi \Leftrightarrow s \not\models_{\xi} \neg h$
- \bullet QS_A(h, ξ): quasi-supporting scenarios for h relative to ξ
- \bullet SP_A(h, ξ): supporting scenarios for h relative to ξ
- **PS**_A(h, ξ): possibly supporting scenarios for h relative to ξ

Supporting Scenarios

- Now, a second propositional sentence $h \in \mathcal{L}_{A\cup P}$, 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** $s \not\models_{\xi} \bot$
	- **possibly supporting scenario for h relative to** $\xi \Leftrightarrow s \not\models_{\xi} \neg h$
- \bullet QS_A(h, ξ): quasi-supporting scenarios for h relative to ξ
- \bullet SP_A(h, ξ): 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 $h \in \mathcal{L}_{A\cup P}$, 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** $s \not\models_{\xi} \bot$
	- **possibly supporting scenario for h relative to** $\xi \Leftrightarrow s \not\models_{\xi} \neg h$
- \bullet QS_A(h, ξ): quasi-supporting scenarios for h relative to ξ
- \bullet SP_A(h, ξ): supporting scenarios for h relative to ξ
- **PS**_A(h, ξ): possibly supporting scenarios for h relative to ξ

Supporting Scenarios

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

≮ロト ⊀何ト ⊀ ヨト ⊀ ヨト

[Argumentation Systems](#page-30-0)

N_A

L<mark>yan</mark>
Universita
Frirurgenti

[Argumentation Systems](#page-30-0)

メロメメ 御き メミメメ ミメ

 \Rightarrow

Outline

[Propositional Logic](#page-2-0) [Argumentation Systems](#page-30-0) [Probabilistic Argumentation Systems](#page-48-0)

[Propositional Argumentation Systems](#page-2-0)

- **[Propositional Logic](#page-2-0)**
- **[Argumentation Systems](#page-30-0)**
- [Probabilistic Argumentation Systems](#page-48-0)

2 [Argumentation Systems on Set Constraint Logic](#page-64-0)

- [Set Constraint Logic](#page-64-0)
- [Constraint-Based Argumentation Systems](#page-82-0)
- **•** [Introducing Probabilities](#page-85-0)

[Propositional Logic](#page-2-0) [Argumentation Systems](#page-30-0) [Probabilistic Argumentation Systems](#page-48-0)

Assigning Probabilities

• We link every assumption $a_i \in A$ to a prior probability π_i

 \bullet The π_i are assumed to be stochastically independent

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

[Propositional Logic](#page-2-0) [Argumentation Systems](#page-30-0) [Probabilistic Argumentation Systems](#page-48-0)

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

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

[Propositional Logic](#page-2-0) [Argumentation Systems](#page-30-0) [Probabilistic Argumentation Systems](#page-48-0)

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 $PAS_P = (\xi, P, A, \Pi)$, where $\Pi = {\pi_1, \ldots, \pi_m}$ denotes the set of probabilities assigned to the assumptions $a_i \in A$.

 $($ \Box $)$ $($ \Box $)$

[Propositional Logic](#page-2-0) [Argumentation Systems](#page-30-0) [Probabilistic Argumentation Systems](#page-48-0)

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

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

 \bullet For $S \subseteq N_A$ we define

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

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

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

 \bullet For $S \subseteq N_A$ we define

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

- 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

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

• 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.

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

 $\text{dsp}(h,\xi) = p'(\text{SP}_A(h,\xi))$ is called degree of support $dps(h, \xi) = p'(PS_A(h, \xi))$ is called degree of 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.

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

 $\text{dsp}(h,\xi) = p'(\text{SP}_A(h,\xi))$ is called degree of support $dps(h, \xi) = p'(PS_A(h, \xi))$ is called degree of 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.

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

 $\text{dsp}(h,\xi) = p'(\text{SP}_A(h,\xi))$ is called degree of support $dps(h, \xi) = p'(PS_A(h, \xi))$ is called degree of 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.

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

- $\text{dsp}(h,\xi) = p'(\text{SP}_A(h,\xi))$ is called degree of support
- $dps(h, \xi) = p'(PS_A(h, \xi))$ is called degree of possibility

[Propositional Logic](#page-2-0) [Argumentation Systems](#page-30-0) [Probabilistic Argumentation Systems](#page-48-0)

Degree of Support and Possibility

Whenever $\xi \neq \perp$:

$$
\bullet \ \textit{dps}(\bot,\xi) = \textit{dsp}(\bot,\xi) = 0
$$

•
$$
dps(\top, \xi) = dsp(\top, \xi) = 1
$$

•
$$
h_1 \models h_2 \Rightarrow dsp(h_1, \xi) \le dsp(h_2, \xi)
$$
, $dps(h_1, \xi) \le dbs(h_2, \xi)$

•
$$
h_1 \equiv h_2 \Rightarrow dsp(h_1, \xi) = dsp(h_2, \xi)
$$
, $dps(h_1, \xi) = dps(h_2, \xi)$

• dsp(h, ξ) \leq dps(h, ξ)

[Propositional Logic](#page-2-0) [Argumentation Systems](#page-30-0) [Probabilistic Argumentation Systems](#page-48-0)

Degree of Support and Possibility

Whenever $\xi \neq \perp$:

o dps(\bot , ξ) = dsp(\bot , ξ) = 0

$$
\bullet \ \textit{dps}(T,\xi) = \textit{dsp}(T,\xi) = 1
$$

- \bullet $h_1 \models h_2 \Rightarrow$ dsp(h_1, ξ) < dsp(h_2, ξ), dps(h_1, ξ) < dps(h_2, ξ)
- $h_1 \equiv h_2 \Rightarrow dsp(h_1, \xi) = dsp(h_2, \xi)$, dps(h_1, ξ) = dps(h_2, ξ)

• dsp(h, ξ) < dps(h, ξ)

[Propositional Logic](#page-2-0) [Argumentation Systems](#page-30-0) [Probabilistic Argumentation Systems](#page-48-0)

Degree of Support and Possibility

Whenever $\xi \neq \perp$:

- o dps(\bot, ξ) = dsp(\bot, ξ) = 0
- dps $(\top, \xi) =$ dsp $(\top, \xi) = 1$
- $h_1 \models h_2 \Rightarrow \text{dsp}(h_1, \xi) \leq \text{dsp}(h_2, \xi)$, $\text{dps}(h_1, \xi) \leq \text{dps}(h_2, \xi)$
- $h_1 \equiv h_2 \Rightarrow dsp(h_1, \xi) = dsp(h_2, \xi)$, dps(h_1, ξ) = dps(h_2, ξ)

• dsp(h, ξ) < dps(h, ξ)

[Propositional Logic](#page-2-0) [Argumentation Systems](#page-30-0) [Probabilistic Argumentation Systems](#page-48-0)

Degree of Support and Possibility

Whenever $\xi \neq \perp$:

- o dps(\bot, ξ) = dsp(\bot, ξ) = 0
- dps $(\top, \xi) =$ dsp $(\top, \xi) = 1$
- $h_1 \models h_2 \Rightarrow \text{dsp}(h_1, \xi) \leq \text{dsp}(h_2, \xi)$, $\text{dps}(h_1, \xi) \leq \text{dps}(h_2, \xi)$
- $h_1 \equiv h_2 \Rightarrow dsp(h_1, \xi) = dsp(h_2, \xi)$, $dsp(h_1, \xi) = dsp(h_2, \xi)$

• dsp(h, ξ) < dps(h, ξ)

[Propositional Logic](#page-2-0) [Argumentation Systems](#page-30-0) [Probabilistic Argumentation Systems](#page-48-0)

Degree of Support and Possibility

Whenever $\xi \neq \perp$:

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

K ロ ⊁ K 何 ≯ K ヨ ⊁ K ヨ ⊁

[Set Constraint Logic](#page-64-0) [Constraint-Based Argumentation Systems](#page-82-0) [Introducing Probabilities](#page-85-0)

Outline

[Propositional Argumentation Systems](#page-2-0)

- **[Propositional Logic](#page-2-0)**
- **[Argumentation Systems](#page-30-0)**
- **[Probabilistic Argumentation Systems](#page-48-0)**

2 [Argumentation Systems on Set Constraint Logic](#page-64-0)

- [Set Constraint Logic](#page-64-0)
- [Constraint-Based Argumentation Systems](#page-82-0)
- **•** [Introducing Probabilities](#page-85-0)

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 $\lt v \in X > X \subseteq \Theta$, is called set constraint
- An assignment is a set constraint $\langle v \in \{\theta_i\} > \theta_i \in \Theta_v$

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 $\lt v \in X > X \subseteq \Theta$, is called set constraint
- An assignment is a set constraint $\langle v \in \{\theta_i\} > \theta_i \in \Theta_v$

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 $\lt v \in X >$, $X \subseteq \Theta$, is called set constraint

• An assignment is a set constraint $v \in \{\theta_i\} > \theta_i \in \Theta_v$

 $(1,1)$ $(1,1)$ $(1,1)$ $(1,1)$ $(1,1)$ $(1,1)$ $(1,1)$ $(1,1)$ $(1,1)$ $(1,1)$ $(1,1)$ $(1,1)$

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 $\lt v \in X >$, $X \subseteq \Theta$, is called set constraint

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

 $($ \Box $)$ $($ \Box $)$

SCL-Formulas

[Set Constraint Logic](#page-64-0) [Constraint-Based Argumentation Systems](#page-82-0) [Introducing Probabilities](#page-85-0)

• set constraints, ⊥ and ⊤ 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

 4 ロ > 4 6 > 4 \geq > 4 \geq \geq

SCL-Formulas

[Set Constraint Logic](#page-64-0) [Constraint-Based Argumentation Systems](#page-82-0) [Introducing Probabilities](#page-85-0)

- set constraints, ⊥ and ⊤ 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

 4 ロ > 4 6 > 4 \geq > 4 \geq \geq

SCL-Formulas

[Set Constraint Logic](#page-64-0) [Constraint-Based Argumentation Systems](#page-82-0) [Introducing Probabilities](#page-85-0)

- set constraints, ⊥ and ⊤ are SCL-formulas
- if γ is a SCL-formula, then $\neg \gamma$ is a SCL-formula
- **•** if γ and δ are SCL-formulas, then $(\gamma \wedge \delta)$, $(\gamma \vee \delta)$, $(\gamma \rightarrow \delta)$ and ($\gamma \leftrightarrow \delta$) are SCL-formulas
[Set Constraint Logic](#page-64-0) [Constraint-Based Argumentation Systems](#page-82-0) [Introducing Probabilities](#page-85-0)

Assigning a value to every $v \in V$ is called interpretation

-
-
-
-

K ロ ▶ K 御 ▶ K 君 ▶ K 君 ▶ .

[Set Constraint Logic](#page-64-0) [Constraint-Based Argumentation Systems](#page-82-0) [Introducing Probabilities](#page-85-0)

Assigning a value to every $v \in V$ is called interpretation

- The set of all possible interpretations is denoted by N_V
-
-
-

K ロ ⊁ K 御 ⊁ K 君 ⊁ K 君 ⊁

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
-
-

[Set Constraint Logic](#page-64-0) [Constraint-Based Argumentation Systems](#page-82-0)

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 $y_i \in X > i$
	- 1 whenever $x_i \in X$ and 0 otherwise

The truth value of a formula is determined like for propositional logic **K ロ ▶ K 御 ▶ K ヨ ▶ K ヨ**

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 $y_i \in X > i$
	- 1 whenever $x_i \in X$ and 0 otherwise

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

[Propositional Argumentation Systems](#page-2-0) [Argumentation Systems on Set Constraint Logic](#page-64-0)

SCL-Formulas

[Set Constraint Logic](#page-64-0) [Constraint-Based Argumentation Systems](#page-82-0) [Introducing Probabilities](#page-85-0)

- $N(\gamma) \subset N_V$ denotes all interpretations for which γ is true
- $\bullet \ \gamma \models \delta$ if, and only if, $N(\gamma) \subseteq N(\delta)$
- $\gamma \equiv \delta$ if, and only if, $N(\gamma) = N(\delta)$
- Let $\gamma\in\mathcal{L}_\mathcal{V}$ and $\mathsf{x}\in\mathcal{N}_\mathsf{Q}$ with $\mathsf{Q}\subseteq\mathsf{V}.$ $\gamma_{\mathsf{Q}\leftarrow\mathsf{x}}$ is the formula obtained by replacing each set constraint $\langle v_i \in X \rangle$ by T if $x_i \in X$ and by \perp otherwise
- For $\delta\in{\cal L}_{\sf V}$ then ${\sf x}\models_\gamma\delta$ means $\gamma_{\sf Q\leftarrow{\sf x}}\models\delta$

[Set Constraint Logic](#page-64-0) [Constraint-Based Argumentation Systems](#page-82-0) [Introducing Probabilities](#page-85-0)

- \bullet $N(\gamma) \subseteq N_V$ denotes all interpretations for which γ is true
- $\bullet \ \gamma \models \delta$ if, and only if, $N(\gamma) \subset N(\delta)$
- $\gamma \equiv \delta$ if, and only if, $N(\gamma) = N(\delta)$
- Let $\gamma\in\mathcal{L}_\mathcal{V}$ and $\mathsf{x}\in\mathcal{N}_\mathsf{Q}$ with $\mathsf{Q}\subseteq\mathsf{V}.$ $\gamma_{\mathsf{Q}\leftarrow\mathsf{x}}$ is the formula obtained by replacing each set constraint $\langle v_i \in X \rangle$ by T if $x_i \in X$ and by \perp otherwise
- For $\delta\in{\cal L}_{\sf V}$ then ${\sf x}\models_\gamma\delta$ means $\gamma_{\sf Q\leftarrow{\sf x}}\models\delta$

[Set Constraint Logic](#page-64-0) [Constraint-Based Argumentation Systems](#page-82-0) [Introducing Probabilities](#page-85-0)

- \bullet $N(\gamma) \subseteq N_V$ denotes all interpretations for which γ is true
- $\bullet \ \gamma \models \delta$ if, and only if, $N(\gamma) \subset N(\delta)$
- $\bullet \ \gamma \equiv \delta$ if, and only if, $N(\gamma) = N(\delta)$
- Let $\gamma\in\mathcal{L}_\mathcal{V}$ and $\mathsf{x}\in\mathcal{N}_\mathsf{Q}$ with $\mathsf{Q}\subseteq\mathsf{V}.$ $\gamma_{\mathsf{Q}\leftarrow\mathsf{x}}$ is the formula obtained by replacing each set constraint $\langle v_i \in X \rangle$ by T if $x_i \in X$ and by \perp otherwise
- For $\delta\in{\cal L}_{\sf V}$ then ${\sf x}\models_\gamma\delta$ means $\gamma_{\sf Q\leftarrow{\sf x}}\models\delta$

 4 ロ } 4 6 } 4 \geq } 4 \geq }

[Set Constraint Logic](#page-64-0) [Constraint-Based Argumentation Systems](#page-82-0) [Introducing Probabilities](#page-85-0)

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

[Set Constraint Logic](#page-64-0) [Constraint-Based Argumentation Systems](#page-82-0) [Introducing Probabilities](#page-85-0)

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

[Propositional Argumentation Systems](#page-2-0) [Argumentation Systems on Set Constraint Logic](#page-64-0) **[Set Constraint Logic](#page-64-0)** [Constraint-Based Argumentation Systems](#page-82-0) [Introducing Probabilities](#page-85-0)

Outline

- **[Propositional Argumentation Systems](#page-2-0)**
	- **[Propositional Logic](#page-2-0)**
	- **[Argumentation Systems](#page-30-0)**
	- **[Probabilistic Argumentation Systems](#page-48-0)**

[Argumentation Systems on Set Constraint Logic](#page-64-0)

- [Set Constraint Logic](#page-64-0)
- **[Constraint-Based Argumentation Systems](#page-82-0)**
- **•** [Introducing Probabilities](#page-85-0)

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 $\xi \in \mathcal{L}_{V \cup F}$ then we call $\mathcal{A}\mathcal{S}_{\mathcal{C}} = (\xi, V, E)$ 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

≮ロト ⊀何ト ⊀ ヨト ⊀ ヨト

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 $\mathcal{A}\mathcal{S}_{\mathcal{C}} = (\xi, V, E)$ constraint-based argumentation system.

- \bullet 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

≮ロト ⊀何ト ⊀ ヨト ⊀ ヨト

[Propositional Argumentation Systems](#page-2-0) [Argumentation Systems on Set Constraint Logic](#page-64-0) [Set Constraint Logic](#page-64-0) [Introducing Probabilities](#page-85-0)

Outline

- **[Propositional Argumentation Systems](#page-2-0)**
	- **[Propositional Logic](#page-2-0)**
	- **[Argumentation Systems](#page-30-0)**
	- **[Probabilistic Argumentation Systems](#page-48-0)**

[Argumentation Systems on Set Constraint Logic](#page-64-0)

- [Set Constraint Logic](#page-64-0)
- **[Constraint-Based Argumentation Systems](#page-82-0)**
- **•** [Introducing Probabilities](#page-85-0)

Probabilistic Argumentation Systems

Suppose every $\Theta_{\bm{e}_i}$ is finite for $\bm{e}_i \in E$

- Let $\pi_{ij} = \rho(e_i = \theta_{ij})$ with $\theta_i \in \Theta_{e_i}$ and $\sum_j \pi_{ij} = 1$
- The probability distribution assigned to \boldsymbol{e}_i is denoted by π_i

We call $PAS_C(\xi, V, E, \Pi)$ with $\Pi = {\pi_1, \dots, \pi_m}$ probabilistic

Probabilistic Argumentation Systems

- Suppose every $\Theta_{\bm{e}_i}$ is finite for $\bm{e}_i \in E$
- Let $\pi_{ij}=\pmb{\rho}(\pmb{e}_i=\theta_{ij})$ with $\theta_i\in\Theta_{\pmb{e}_i}$ and $\sum_j\pi_{ij}=1$
- The probability distribution assigned to \boldsymbol{e}_i is denoted by π_i

We call $PAS_C(\xi, V, E, \Pi)$ with $\Pi = {\pi_1, \dots, \pi_m}$ probabilistic

[Set Constraint Logic](#page-64-0) [Introducing Probabilities](#page-85-0)

Probabilistic Argumentation Systems

- Suppose every $\Theta_{\bm{e}_i}$ is finite for $\bm{e}_i \in E$
- Let $\pi_{ij}=\pmb{\rho}(\pmb{e}_i=\theta_{ij})$ with $\theta_i\in\Theta_{\pmb{e}_i}$ and $\sum_j\pi_{ij}=1$
- The probability distribution assigned to \boldsymbol{e}_i is denoted by π_i

We call $PAS_C(\xi, V, E, \Pi)$ with $\Pi = {\pi_1, \dots, \pi_m}$ probabilistic

Probabilistic Argumentation Systems

- Suppose every $\Theta_{\bm{e}_i}$ is finite for $\bm{e}_i \in E$
- Let $\pi_{ij}=\pmb{\rho}(\pmb{e}_i=\theta_{ij})$ with $\theta_i\in\Theta_{\pmb{e}_i}$ and $\sum_j\pi_{ij}=1$
- The probability distribution assigned to \boldsymbol{e}_i is denoted by π_i

Definition

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

≮ロト ⊀伊ト ⊀ ヨト ⊀ ヨト

Probabilistic Argumentation Systems

Let $\mathsf{s}=(\theta_{1j},\ldots,\theta_{mj})$ 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 $\mathcal{S} \subseteq \mathcal{N}_{E}$ is then $p(\mathcal{S}) = \sum_{s \in \mathcal{S}} p(s)$
- Degree of quasi-support / degree of support / degree of possibility can be defined like for the propositional case

Probabilistic Argumentation Systems

Let $\mathsf{s}=(\theta_{1j},\ldots,\theta_{mj})$ 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 $\mathcal{S} \subseteq \mathcal{N}_E$ is then $\mathit{p}(\mathcal{S}) = \sum_{s \in \mathcal{S}} \mathit{p}(s)$
- Degree of quasi-support / degree of support / degree of possibility can be defined like for the propositional case

Probabilistic Argumentation Systems

Let $\mathsf{s}=(\theta_{1j},\ldots,\theta_{mj})$ 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 $\mathcal{S} \subseteq \mathcal{N}_E$ is then $\mathit{p}(\mathcal{S}) = \sum_{s \in \mathcal{S}} \mathit{p}(s)$
- Degree of quasi-support / degree of support / degree of possibility can be defined like for the propositional case

 $(0,1)$ $(0,1)$ $(1,1)$ $(1,1)$ $(1,1)$ $(1,1)$ $(1,1)$ $(1,1)$ $(1,1)$ $(1,1)$ $(1,1)$ $(1,1)$ $(1,1)$ $(1,1)$

[Set Constraint Logic](#page-64-0) [Introducing Probabilities](#page-85-0)

K ロ ▶ K 御 ▶ K 君 ▶ K 君 ▶ .

唐