On obligation and normative ability

Navid Talebanfard

Universiteit van Amsterdam

Jan 25, 2010

Normative alternating-time temporal logic

- Intended to capture the coalitional abilities in normative systems.

Normative alternating-time temporal logic

- Intended to capture the coalitional abilities in normative systems.
- The goal is to formulate statements of the form $\langle\langle \eta : C \rangle\rangle \phi$ with the interpretation that the coalition C can bring about ϕ if all the agents in that coalition conform to the rules in η.

Normative alternating-time temporal logic

- Intended to capture the coalitional abilities in normative systems.
- The goal is to formulate statements of the form $\langle\langle \eta : C \rangle\rangle \phi$ with the interpretation that the coalition C can bring about ϕ if all the agents in that coalition conform to the rules in η.
- This logic can be used to model deontic expressions in a normative system.

Normative alternating-time temporal logic

- Intended to capture the coalitional abilities in normative systems.
- The goal is to formulate statements of the form $\langle\langle \eta : C \rangle\rangle \phi$ with the interpretation that the coalition C can bring about ϕ if all the agents in that coalition conform to the rules in η.
- This logic can be used to model deontic expressions in a normative system.
- It can also be used to reason about social contracts.

Action-based alternating transition systems (AATS)

- Q is a finite, non-empty set of *states*,

Action-based alternating transition systems (AATS)

- Q is a finite, non-empty set of *states*,
- $q_0 \in Q$ is the *inital state*,

Action-based alternating transition systems (AATS)

- Q is a finite, non-empty set of *states*,
- $q_0 \in Q$ is the *inital state*,
- $Ag = \{1, \ldots, n\}$ is a finite, non-empty set of *agents*,

Action-based alternating transition systems (AATS)

- Q is a finite, non-empty set of *states*,
- $q_0 \in Q$ is the *inital state*,
- $Ag = \{1, \ldots, n\}$ is a finite, non-empty set of *agents*,
- Ac_i is a finite, non-empty set *actions*, for each $i \in Ag$, where $Ac_i \cap Ac_j = \emptyset$ for all $i \neq j \in Ag$,

Action-based alternating transition systems (AATS)

- Q is a finite, non-empty set of *states*,
- $q_0 \in Q$ is the *inital state*,
- $Ag = \{1, \ldots, n\}$ is a finite, non-empty set of *agents*,
- Ac_i is a finite, non-empty set *actions*, for each $i \in Ag$, where $Ac_i \cap Ac_j = \emptyset$ for all $i \neq j \in Ag$,
- $\rho : Ac_{Ag} \rightarrow 2^Q$ is an *action precondition function*. $\rho(\alpha)$ denotes the states in which α may be executed,

Action-based alternating transition systems (AATS)

- Q is a finite, non-empty set of *states*,
- $q_0 \in Q$ is the *inital state*,
- $Ag = \{1, \ldots, n\}$ is a finite, non-empty set of *agents*,
- Ac_i is a finite, non-empty set *actions*, for each $i \in Ag$, where $Ac_i \cap Ac_j = \emptyset$ for all $i \neq j \in Ag$,
- $\rho : Ac_{Ag} \rightarrow 2^Q$ is an *action precondition function*. $\rho(\alpha)$ denotes the states in which α may be executed,
- $\tau : Q \times J_{Ag} \rightarrow Q$ is a partial *system transition function*,

Action-based alternating transition systems (AATS)

- Q is a finite, non-empty set of *states*,
- $q_0 \in Q$ is the *inital state*,
- $Ag = \{1, \ldots, n\}$ is a finite, non-empty set of *agents*,
- Ac_i is a finite, non-empty set *actions*, for each $i \in Ag$, where $Ac_i \cap Ac_j = \emptyset$ for all $i \neq j \in Ag$,
- $\rho : Ac_{Ag} \rightarrow 2^Q$ is an *action precondition function*. $\rho(\alpha)$ denotes the states in which α may be executed,
- $\tau : Q \times J_{Ag} \rightarrow Q$ is a partial *system transition function*,
- Φ is a finite, non-empty set of *atomic propositions*,

Action-based alternating transition systems (AATS)

- Q is a finite, non-empty set of *states*,
- $q_0 \in Q$ is the *inital state*,
- $Ag = \{1, \ldots, n\}$ is a finite, non-empty set of *agents*,
- Ac_i is a finite, non-empty set *actions*, for each $i \in Ag$, where $Ac_i \cap Ac_j = \emptyset$ for all $i \neq j \in Ag$,
- $\rho : Ac_{Ag} \to 2^Q$ is an *action precondition function*. $\rho(\alpha)$ denotes the states in which α may be executed,
- $\tau : Q \times J_{Ag} \to Q$ is a partial *system transition function*,
- Φ is a finite, non-empty set of *atomic propositions*,
- $\pi : Q \to 2^\Phi$ is an interpretation function: $\pi(q)$ is the set of atomic propositions which are satisfied in q.

Coherence constraints on AATSs

- *Non-triviality*. Agents always have an available action.

$$\forall q \in Q, \forall i \in Ag, \exists \alpha \in Ac_i \text{ s.t. } q \in \rho(\alpha)$$

Coherence constraints on AATSs

- *Non-triviality.* Agents always have an available action.

$$\forall q \in Q, \forall i \in Ag, \exists \alpha \in Ac_i \text{ s.t. } q \in \rho(\alpha)$$

- *Consistency.* ρ and τ agree on actions that may be performed:

$$\forall q, \forall j \in J_{Ag}, (q, j) \in \mathrm{dom}\tau \text{ iff } \forall i \in Ag, q \in \rho(j_i)$$

Coherence constraints on AATSs

- *Non-triviality*. Agents always have an available action.

$$\forall q \in Q, \forall i \in Ag, \exists \alpha \in Ac_i \text{ s.t. } q \in \rho(\alpha)$$

- *Consistency*. ρ and τ agree on actions that may be performed:

$$\forall q, \forall j \in J_{Ag}, (q, j) \in \text{dom}\tau \text{ iff } \forall i \in Ag, q \in \rho(j_i)$$

- The set of sequences on Q is denoted by Q^* and the set of non-empty sequences is denoted by Q^+.

Strategies

- The set of available actions for agent i at state q is denoted by

$$options(i, q) = \{\alpha | \alpha \in Ac_i, q \in \rho(\alpha)\}$$

Strategies

- The set of available actions for agent i at state q is denoted by

$$options(i, q) = \{\alpha | \alpha \in Ac_i, q \in \rho(\alpha)\}$$

- A *strategy* for agent i is a function $\sigma_i : Q \to Ac_i$ that satisfies $\sigma_i(q) \in options(i, q)$.

Strategies

- The set of available actions for agent i at state q is denoted by

$$options(i, q) = \{\alpha | \alpha \in Ac_i, q \in \rho(\alpha)\}$$

- A *strategy* for agent i is a function $\sigma_i : Q \rightarrow Ac_i$ that satisfies $\sigma_i(q) \in options(i, q)$.
- A *strategy profile* for a coalition $C = \{a_1, \ldots, a_k\} \subseteq Ag$ is a k-tuple $\langle \sigma_1, \ldots, \sigma_k \rangle$ of strategies.

Strategies

- The set of available actions for agent i at state q is denoted by

$$options(i, q) = \{\alpha | \alpha \in Ac_i, q \in \rho(\alpha)\}$$

- A *strategy* for agent i is a function $\sigma_i : Q \rightarrow Ac_i$ that satisfies $\sigma_i(q) \in options(i, q)$.
- A *strategy profile* for a coalition $C = \{a_1, \ldots, a_k\} \subseteq Ag$ is a k-tuple $\langle \sigma_1, \ldots, \sigma_k \rangle$ of strategies.
- Σ_C is the set of all strategy profiles for coalition C.

Strategies

- The set of available actions for agent i at state q is denoted by

$$options(i, q) = \{\alpha | \alpha \in Ac_i, q \in \rho(\alpha)\}$$

- A *strategy* for agent i is a function $\sigma_i : Q \to Ac_i$ that satisfies $\sigma_i(q) \in options(i, q)$.
- A *strategy profile* for a coalition $C = \{a_1, \ldots, a_k\} \subseteq Ag$ is a k-tuple $\langle \sigma_1, \ldots, \sigma_k \rangle$ of strategies.
- Σ_C is the set of all strategy profiles for coalition C.
- For $\sigma_C \in \Sigma_C$, σ^i_C denotes the i's component of σ_C.

Strategies

- The set of available actions for agent i at state q is denoted by

$$options(i, q) = \{\alpha | \alpha \in Ac_i, q \in \rho(\alpha)\}$$

- A *strategy* for agent i is a function $\sigma_i : Q \to Ac_i$ that satisfies $\sigma_i(q) \in options(i, q)$.
- A *strategy profile* for a coalition $C = \{a_1, \ldots, a_k\} \subseteq Ag$ is a k-tuple $\langle \sigma_1, \ldots, \sigma_k \rangle$ of strategies.
- Σ_C is the set of all strategy profiles for coalition C.
- For $\sigma_C \in \Sigma_C$, σ_C^i denotes the i's component of σ_C.
- $out(\sigma_C, q)$ denotes the set of possible states that the coalition C following σ_C from state a can reach:

$$out(\sigma_C, q) = \{q' | \tau(q, j) = q' \text{ where } (q, j) \in \text{ dom}\tau, \sigma_C^i = j_i \text{ for } i \in C\}$$

Computations

- A *computation* is infinite sequence $\lambda \in Q^{+}$.

Computations

- A *computation* is infinite sequence $\lambda \in Q^{+}$.
- For $u \in \mathbb{N}$, $\lambda[u]$ denotes the state indexed by u in λ ($\lambda[0]$ is the first element).

Computations

- A *computation* is infinite sequence $\lambda \in Q^+$.
- For $u \in \mathbb{N}$, $\lambda[u]$ denotes the state indexed by u in λ ($\lambda[0]$ is the first element).
- $\lambda[0, u]$ and $\lambda[u, \infty]$ denote the finite prefix $q_0, \ldots, q_u$ and the infintie suffix $q_u, q_{u+1}, \ldots$ of λ respectively.

Computations

- A *computation* is infinite sequence $\lambda \in Q^{+}$.
- For $u \in \mathbb{N}$, $\lambda[u]$ denotes the state indexed by u in λ ($\lambda[0]$ is the first element).
- $\lambda[0, u]$ and $\lambda[u, \infty]$ denote the finite prefix $q_0, \ldots, q_u$ and the infintie suffix $q_u, q_{u+1}, \ldots$ of λ respectively.
- A *q-computation* is a sequence λ with $\lambda[0] = q$.

Computations

- A *computation* is infinite sequence $\lambda \in Q^+$.
- For $u \in \mathbb{N}$, $\lambda[u]$ denotes the state indexed by u in λ ($\lambda[0]$ is the first element).
- $\lambda[0, u]$ and $\lambda[u, \infty]$ denote the finite prefix $q_0, \ldots, q_u$ and the infintie suffix $q_u, q_{u+1}, \ldots$ of λ respectively.
- A *q-computation* is a sequence λ with $\lambda[0] = q$.
- $comp(\sigma_C, q)$ is the set of possible runs from state q if every agent in C follows strategy σ_C

$$comp(\sigma_C, q) = \{\lambda | \lambda[0] = q, \forall u \in \mathbb{N} : \lambda[u+1] \in out(\sigma_C, \lambda[u])\}$$

Normative systems

- A *normative system* is a function $\eta : Ac_{Ag} \rightarrow 2^{Q}$.

Normative systems

- A *normative system* is a function $\eta : Ac_{Ag} \rightarrow 2^Q$.
- α is forbidden in q if $q \in \eta(\alpha)$.

Normative systems

- A *normative system* is a function $\eta : Ac_{Ag} \rightarrow 2^Q$.
- α is forbidden in q if $q \in \eta(\alpha)$.
- Those actions that are forbidden by nature are also forbidden by the normative system:

$$\forall \alpha \in Ac_{Ag} : (Q \setminus \rho(\alpha)) \subseteq \eta(\alpha)$$

Normative systems

- A *normative system* is a function $\eta : Ac_{Ag} \rightarrow 2^{Q}$.
- α is forbidden in q if $q \in \eta(\alpha)$.
- Those actions that are forbidden by nature are also forbidden by the normative system:

$$\forall \alpha \in Ac_{Ag} : (Q \setminus \rho(\alpha)) \subseteq \eta(\alpha)$$

- A strategy σ_i is η-*conformant* if it never selects a forbidden action:

$$conf(\sigma_i, \eta) \Leftrightarrow \forall q : q \notin \eta(\sigma_i(q))$$

$$conf(\sigma_C, \eta) \Leftrightarrow \forall i \in C : conf(\sigma^i_C, \eta)$$

Normative systems

- A *normative system* is a function $\eta : Ac_{Ag} \rightarrow 2^Q$.
- α is forbidden in q if $q \in \eta(\alpha)$.
- Those actions that are forbidden by nature are also forbidden by the normative system:

$$\forall \alpha \in Ac_{Ag} : (Q \setminus \rho(\alpha)) \subseteq \eta(\alpha)$$

- A strategy σ_i is η-*conformant* if it never selects a forbidden action:

$$conf(\sigma_i, \eta) \Leftrightarrow \forall q : q \notin \eta(\sigma_i(q))$$

$$conf(\sigma_C, \eta) \Leftrightarrow \forall i \in C : conf(\sigma_C^i, \eta)$$

- $\Sigma_C^\eta = \{\sigma_C \in \Sigma_C | conf(\sigma_C, \eta)\}$ is the set of η-conformant strategy profiles for C.

Operations on normative systems

- *Empty normative system:* $\forall \alpha \in Ac_{Ag} : \eta_{\perp}(\alpha) = Q \setminus \rho(\alpha)$.

Operations on normative systems

- *Empty normative system:* $\forall \alpha \in Ac_{Ag} : \eta_{\perp}(\alpha) = Q \setminus \rho(\alpha)$.
- *Trivial normative system:* $\forall \alpha \in Ac_{Ag} : \eta_{\top}(\alpha) = Q$.

Operations on normative systems

- *Empty normative system:* $\forall \alpha \in Ac_{Ag} : \eta_{\perp}(\alpha) = Q \setminus \rho(\alpha)$.
- *Trivial normative system:* $\forall \alpha \in Ac_{Ag} : \eta_{\top}(\alpha) = Q$.
- $$\eta \sqcap \eta'(\alpha) = \eta(\alpha) \cap \eta'(\alpha)$$
  $$\eta \sqcup \eta'(\alpha) = \eta(\alpha) \cup \eta'(\alpha)$$

Operations on normative systems

- *Empty normative system:* $\forall \alpha \in Ac_{Ag} : \eta_{\perp}(\alpha) = Q \setminus \rho(\alpha)$.
- *Trivial normative system:* $\forall \alpha \in Ac_{Ag} : \eta_{\top}(\alpha) = Q$.
- $$\eta \sqcap \eta'(\alpha) = \eta(\alpha) \cap \eta'(\alpha)$$
  $$\eta \sqcup \eta'(\alpha) = \eta(\alpha) \cup \eta'(\alpha)$$
- $$\eta \sqcup \eta_{\perp} = \eta, \eta \sqcup \eta_{\top} = \eta_{\top}$$
  $$\eta \sqcap \eta_{\top} = \eta, \eta \sqcap \eta_{\perp} = \eta_{\perp}$$

Relationships between normative systems

- $\eta \preceq \eta' \Leftrightarrow \forall \alpha \in Ac_{Ag} : \eta(\alpha) \subseteq \eta'(\alpha)$ (η is less restrictive than η').

Relationships between normative systems

- $\eta \preceq \eta' \Leftrightarrow \forall \alpha \in Ac_{Ag} : \eta(\alpha) \subseteq \eta'(\alpha)$ (η is less restrictive than η').
- A normative system is *non-trivial* if under that system every agent has some actions available at every state:

$$\forall q \in Q \exists j \in J_{Ag} \forall i \in Ag : q \notin \eta(j_i)$$

Relationships between normative systems

- $\eta \preceq \eta' \Leftrightarrow \forall \alpha \in Ac_{Ag} : \eta(\alpha) \subseteq \eta'(\alpha)$ (η is less restrictive than η').
- A normative system is *non-trivial* if under that system every agent has some actions available at every state:

$$\forall q \in Q \exists j \in J_{Ag} \forall i \in Ag : q \notin \eta(j_i)$$

Theorem

Let S be an AATS, and let η and η' be non-trivial normative systems on it. Then

$$\eta \preceq \eta' \Leftrightarrow \forall C \subseteq Ag : \Sigma_C^{\eta'} \subseteq \Sigma_C^{\eta}$$

The syntax of NATL (1)

- *State formulae:* are interpreted with respect to individual states.

The syntax of NATL (1)

- *State formulae:* are interpreted with respect to individual states.
- *Path formula:* are interpreted with respect to computations.

The syntax of NATL (2)

$$
\begin{aligned}
\langle state - fmla \rangle \quad : \quad & := \text{true} \\
| \quad & p \\
| \quad & \neg \langle state - fmla \rangle \\
| \quad & \langle state - fmla \rangle \vee \langle state - fmla \rangle \\
| \quad & \langle\langle \eta : C \rangle\rangle \langle path - fmla \rangle
\end{aligned}
$$

$$
\begin{aligned}
\langle path - fmla \rangle \quad : \quad & := \langle state - fmla \rangle \\
| \quad & \neg \langle path - fmla \rangle \\
| \quad & \langle path - fmla \rangle \vee \langle path - fmla \rangle \\
| \quad & \Diamond \langle path - fmla \rangle \\
| \quad & \Box \langle path - fmla \rangle \\
| \quad & \langle path - fmla \rangle \mathcal{U} \langle path - fmla \rangle
\end{aligned}
$$

The semantics of NATL (1)

$S, q \models \text{true}$
$S, q \models p$ iff $p \in \pi(q)(p \in \Phi)$
$S, q \models \neg\phi$ iff $S, q \not\models \phi$
$S, q \models \phi \vee \psi$ iff $S, q \models \psi$
$S, q \models \langle\langle \eta, C \rangle\rangle \phi$ iff $\exists \sigma_C \in \Sigma_C^\eta$, s.t $\forall \lambda \in comp(\sigma_C, q)$, we have
$S, \lambda \mid\models \phi$.

The semantics of NATL (2)

$S, \lambda \mid\models \phi$ iff $S, \lambda[0] \models \phi$ (ϕ is a state formula)
$S, \lambda \mid\models \neg\phi$ iff $S, \lambda \mid\not\models \phi$
$S, \lambda \mid\models \phi \vee \psi$ iff $S, \lambda \mid\models \phi$ or $S, \lambda \mid\models \psi$
$S, \lambda \mid\models \bigcirc\phi$ iff $S, \lambda[1, \infty] \mid\models \psi$
$S, \lambda \mid\models \Diamond\phi$ iff $\exists u \in \mathbb{N}$, we have $S, \lambda[u, \infty] \mid\models \phi$
$S, \lambda \mid\models \Box\phi$ iff $\forall u \in \mathbb{N}$ we have $S, \lambda[u, \infty] \mid\models \phi$
$S, \lambda \mid\models \phi\mathcal{U}\psi$ iff $\exists u \in \mathbb{N}$ s.t $S, \lambda[u, \infty] \mid\models \psi$, and $\forall v$ s.t.
$0 \leq v < u : S, \lambda[v, \infty] \mid\models \phi$

Permission and obligation

- ϕ is permissible within a normative system η if the grand coalition can achieve ϕ: $\mathbf{P}_\eta \phi = \langle\langle \eta : C \rangle\rangle \phi$

Permission and obligation

- ϕ is permissible within a normative system η if the grand coalition can achieve ϕ: $\mathbf{P}_\eta \phi = \langle\langle \eta : C \rangle\rangle \phi$
- ϕ is obligatory within η if it is inevitable if the grand coalition conform to η: $\mathbf{O}_\eta \phi = \neg \mathbf{P}_\eta \neg \phi$.

Permission and obligation

- ϕ is permissible within a normative system η if the grand coalition can achieve ϕ: $\mathbf{P}_\eta \phi = \langle\langle \eta : C \rangle\rangle \phi$
- ϕ is obligatory within η if it is inevitable if the grand coalition conform to η: $\mathbf{O}_\eta \phi = \neg \mathbf{P}_\eta \neg \phi$.

Theorem

Let S be an AATS, and let η, η' be arbitrary non-trivial normative systems over S such that $\eta \preceq \eta'$. Then:

Permission and obligation

- ϕ is permissible within a normative system η if the grand coalition can achieve ϕ: $\mathbf{P}_\eta \phi = \langle\langle \eta : C \rangle\rangle \phi$
- ϕ is obligatory within η if it is inevitable if the grand coalition conform to η: $\mathbf{O}_\eta \phi = \neg \mathbf{P}_\eta \neg \phi$.

Theorem

Let S be an AATS, and let η, η' be arbitrary non-trivial normative systems over S such that $\eta \preceq \eta'$. Then:

1. $S \models \langle\langle \eta' : C \rangle\rangle \phi \rightarrow \langle\langle \eta : C \rangle\rangle \phi$

Permission and obligation

- ϕ is permissible within a normative system η if the grand coalition can achieve ϕ: $\mathbf{P}_\eta \phi = \langle\langle \eta : C \rangle\rangle \phi$
- ϕ is obligatory within η if it is inevitable if the grand coalition conform to η: $\mathbf{O}_\eta \phi = \neg \mathbf{P}_\eta \neg \phi$.

Theorem

Let S be an AATS, and let η, η' be arbitrary non-trivial normative systems over S such that $\eta \preceq \eta'$. Then:

1. $S \models \langle\langle \eta' : C \rangle\rangle \phi \rightarrow \langle\langle \eta : C \rangle\rangle \phi$
2. $S \models \mathbf{P}_{\eta'} \phi \rightarrow \mathbf{P}_\eta \phi$

Permission and obligation

- ϕ is permissible within a normative system η if the grand coalition can achieve ϕ: $\mathbf{P}_\eta\phi = \langle\langle\eta : C\rangle\rangle\phi$
- ϕ is obligatory within η if it is inevitable if the grand coalition conform to η: $\mathbf{O}_\eta\phi = \neg\mathbf{P}_\eta\neg\phi$.

Theorem

Let S be an AATS, and let η, η' be arbitrary non-trivial normative systems over S such that $\eta \preceq \eta'$. Then:

1. $S \models \langle\langle\eta' : C\rangle\rangle\phi \rightarrow \langle\langle\eta : C\rangle\rangle\phi$
2. $S \models \mathbf{P}_{\eta'}\phi \rightarrow \mathbf{P}_\eta\phi$
3. $S \models \mathbf{O}_\eta\phi \rightarrow \mathbf{O}_{\eta'}\phi$

Social contracts (1)

- *Multi-agent system:* An AATS with a path-formula γ_i for each agent $i \in Ag$ representing its goal.

Social contracts (1)

- *Multi-agent system:* An AATS with a path-formula γ_i for each agent $i \in Ag$ representing its goal.
- *Social law:* A normative system with path formula Ψ representing the social goal.

Social contracts (1)

- *Multi-agent system:* An AATS with a path-formula γ_i for each agent $i \in Ag$ representing its goal.
- *Social law:* A normative system with path formula Ψ representing the social goal.
- *Social contract:* A multi-agent system with a social law on it.

Social contracts (2)

A social law $\langle \Psi, \eta \rangle$ over a multi-agent system is:

1. *globally effective:* if $S, q_0 \models \mathbf{O}_\eta \Psi$
2. *weakly globally effective:* if $S, q_0 \models \mathbf{P}_\eta \Psi$
3. *globally ineffective:* if $S, q_0 \models \mathbf{O}_\eta \neg\Psi$