Practical Normative Reasoning: Models and Challenges

Felipe Meneguzzi
School of Computer Science (FACIN)
Pontifical Catholic University of Rio Grande do Sul (PUCRS)
felipe.meneguzzi@pucrs.br

Outline

- Norms and Deontic logic
- Practical Normative Reasoning
  - Plan selection
  - Decision-theoretical reasoning

Why Norms?

- Autonomous agents in heterogenous societies act to achieve individual goals
- Multiple agents acting simultaneously will interfere with each other (negatively)
- Strategies will be either:
  - One against everyone else (game theory)
  - One-to-one coordination (expensive)
  - Normative systems

Norms

- Represent desirable behaviours for members of a society
  - "Soft-constraints" on behaviour
  - General expectation of behaviour
  - **Rewards** for compliance + **Sanctions** for non-compliance
- Traditionally represented through conditional rules of the form:

$$\langle \nu, \alpha, \epsilon \rangle$$

- ϵ → Expiration Condition
- α → Activation Condition
- ν → Norm condition (Deontic Formula)

Deontic Logic

- Alethic modal logic deals with **what is** (or could be)
- Deontic logic deals with **what should be**
- Most common deontic modalities:
  - Obligations - *Oq* - it is obligatory that q
  - Permissions - *Pq* - it is permitted that q
    $Pq \leftrightarrow \neg O \neg q$
  - Prohibitions - *Fq* - it is not permitted that q
    $Fq \leftrightarrow O \neg q$

Deontic Logic

- This talk **is not** about deontic logic
  - A lot of work still being done in logic
- For our purposes we *greatly simplify* things in terms of:
  - States we **want** agents to achieve
  - States we **do not want** agents to achieve

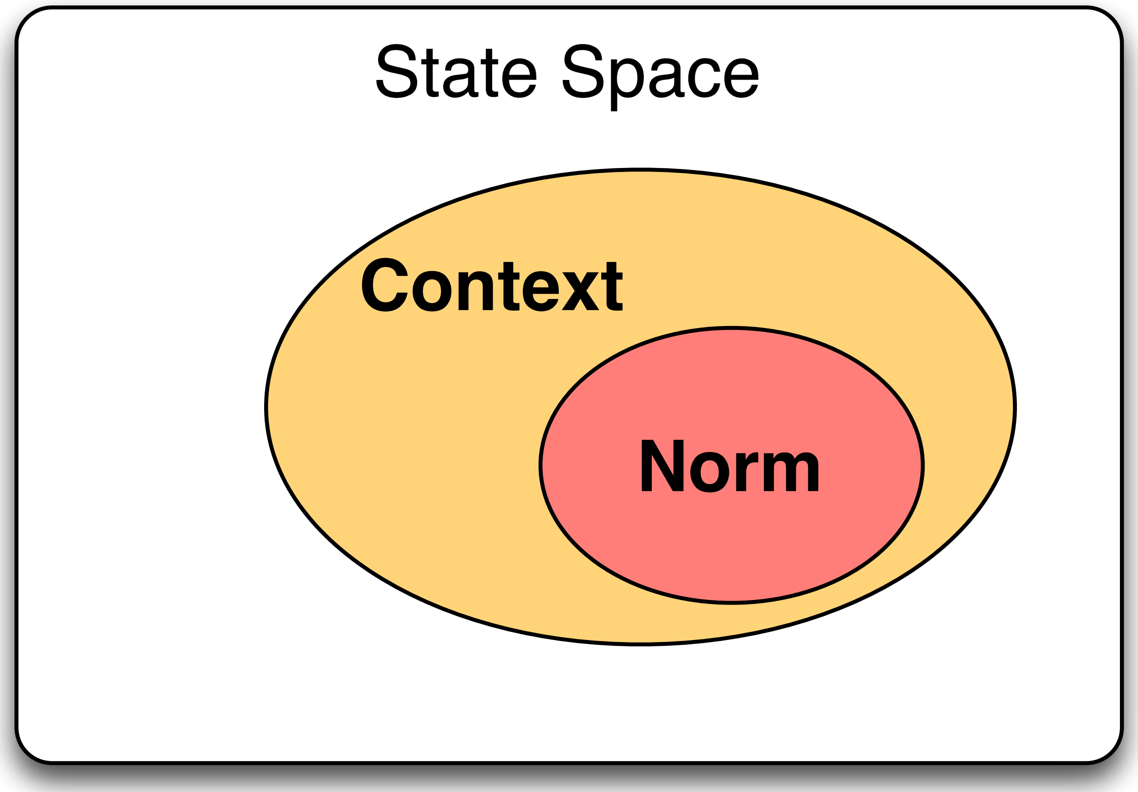
Traffic Light Example

$$\langle \mathbf{O}\, stop(A, P),$$
$$at(A, P) \wedge redlight(P),$$
$$\neg redlight(P) \rangle$$

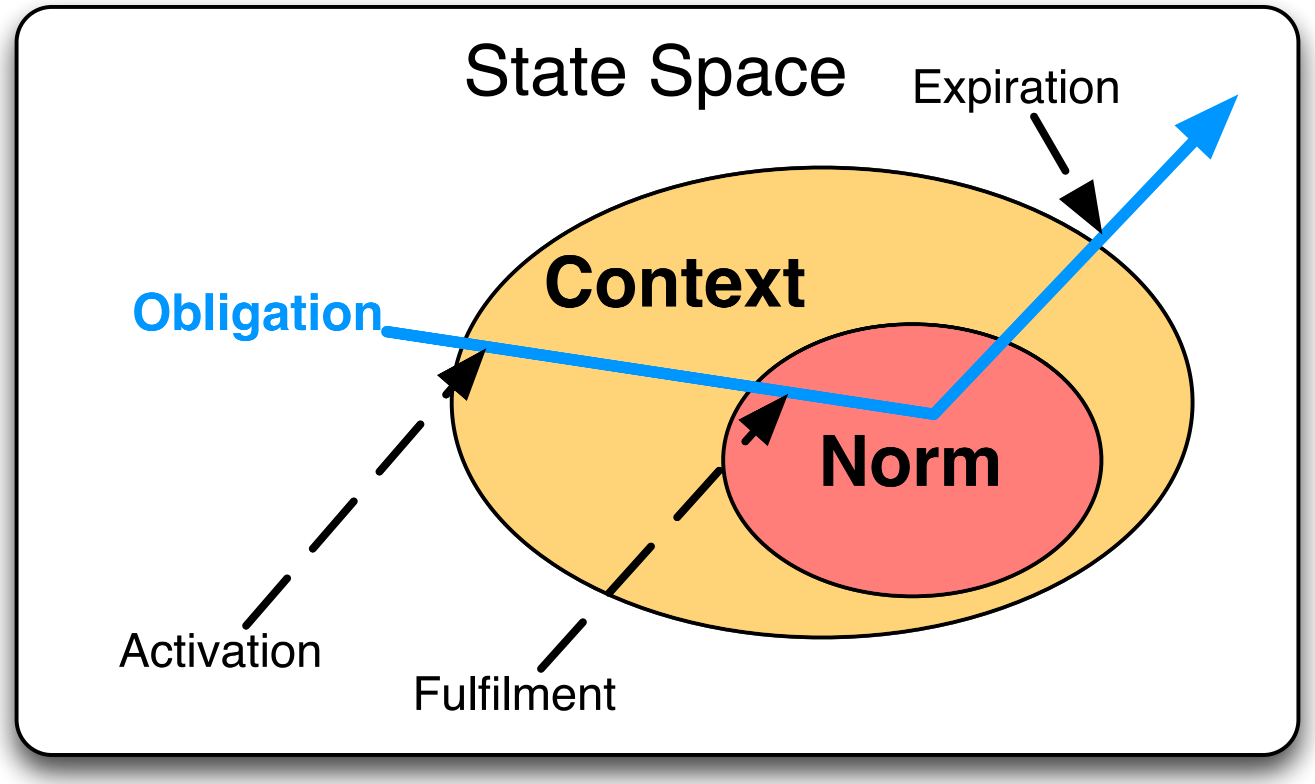
- Norm condition
- Activation condition
- Expiration condition

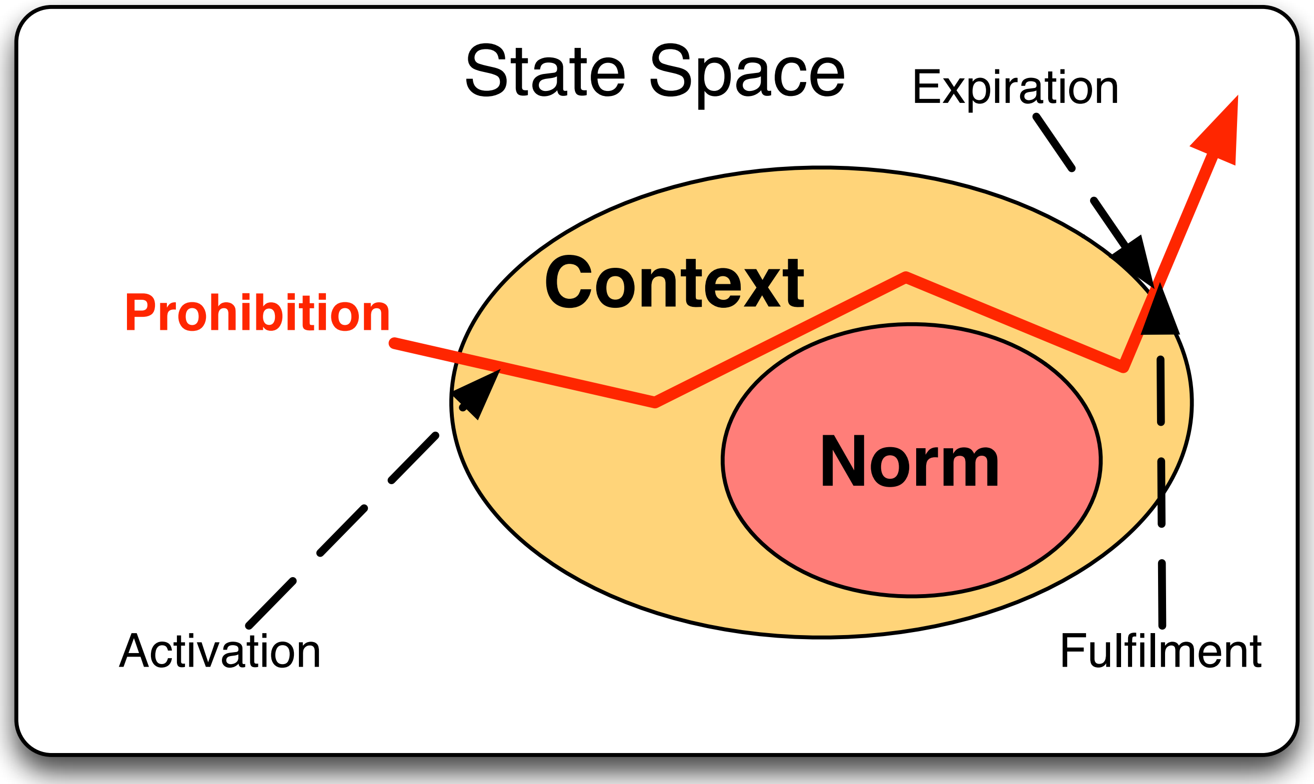
Norms and state-space

- Norm enforcement focuses on two sets of states
  - States between activation and expiration: **norm context**
  - States referred to by the **norm condition**
- Semantics of obligations sometimes differ

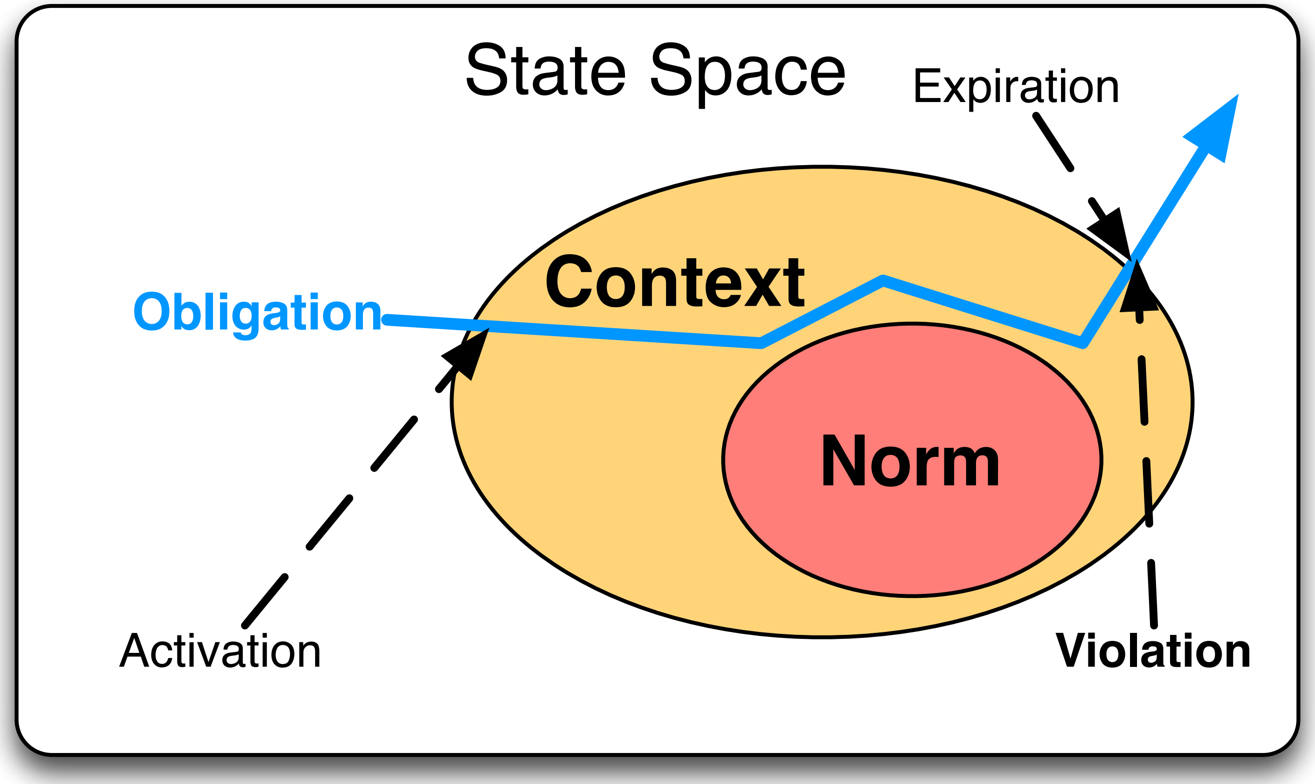


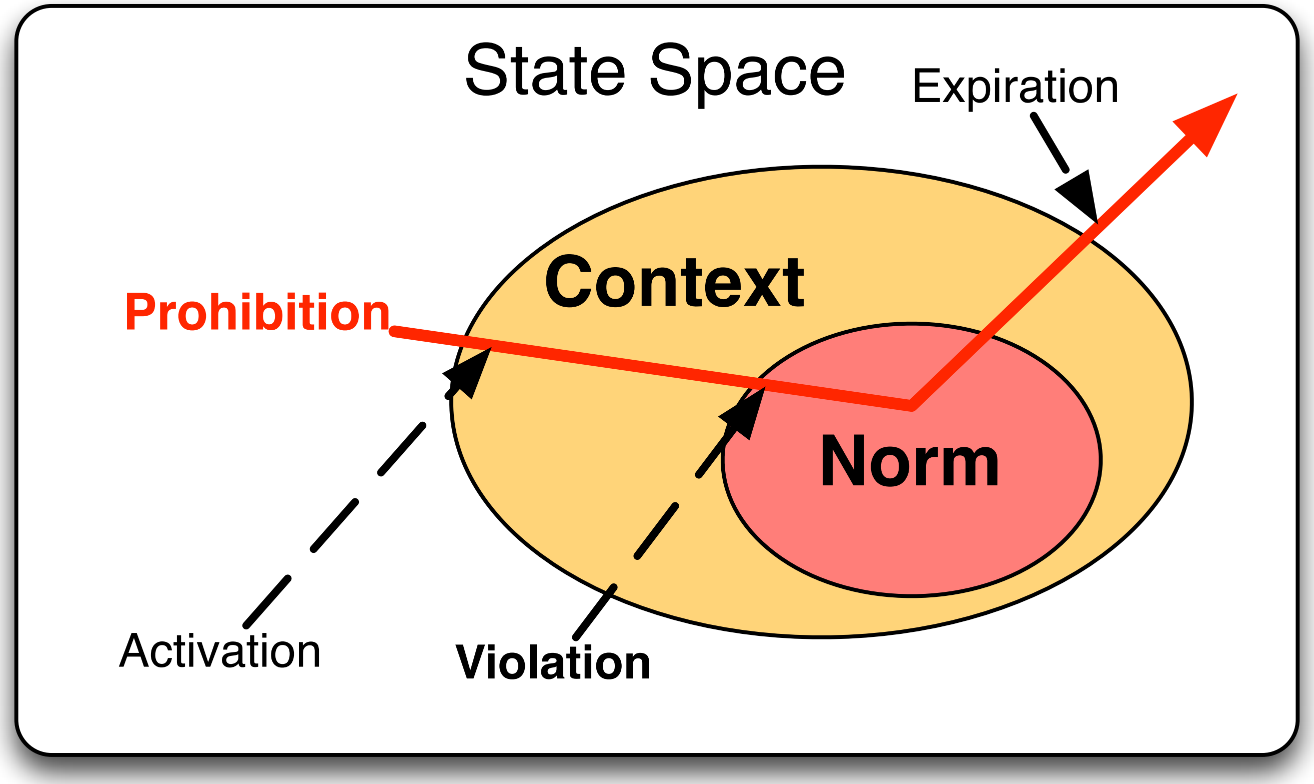
Norm Activation and Expiration





Norm Activation and Expiration





Practical Norm Reasoning

- Existing efforts largely focused
  - Logical aspects (deontic logic)
  - Macro-level (virtual organisations)
- Relatively few techniques for individual **agent behaviour**
  - Finite time/resources
  - Practical enforcement mechanisms

Practical Norm Reasoning

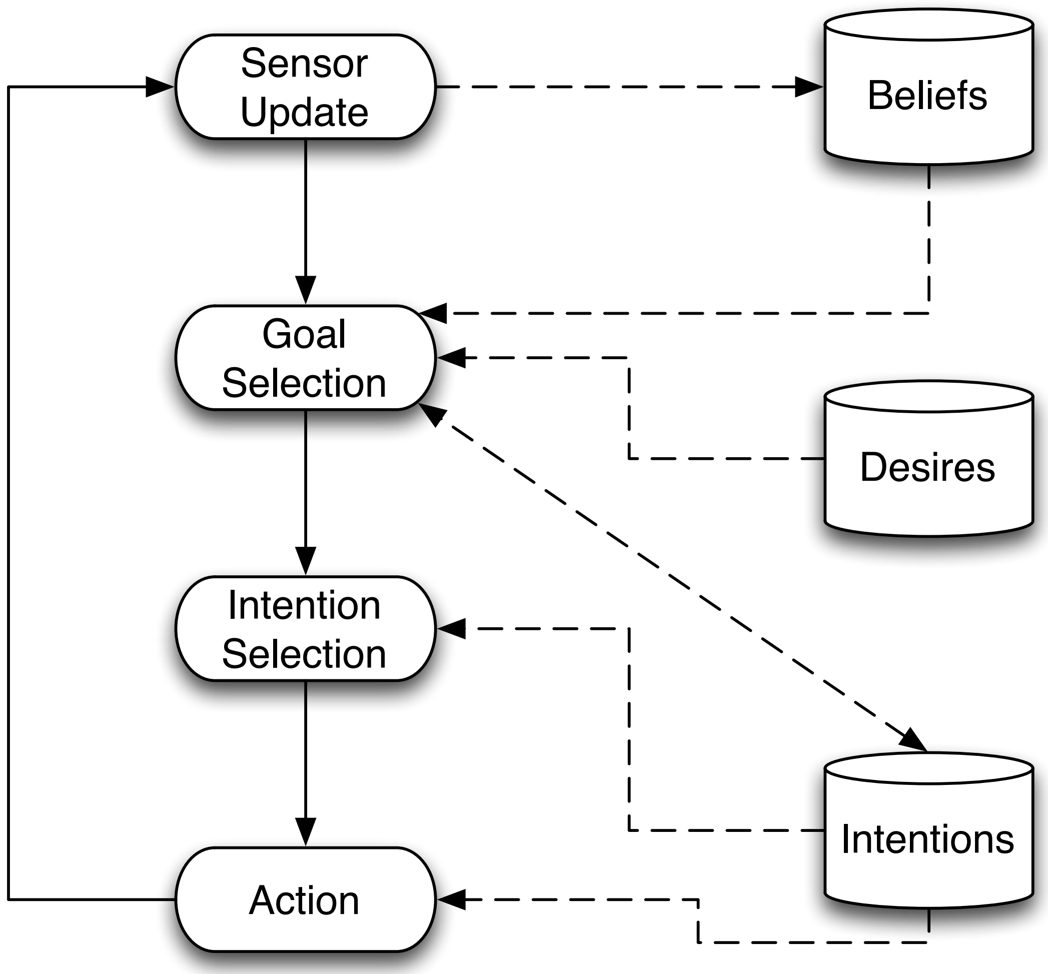
- How should an agent behave in a norm-driven society?
  - Norms as soft constraints
  - Dynamically changing sets of norms
  - Different enforcement mechanisms
  - Limited time/resources
- Depends on the assumptions on the environment

Environment Assumptions

- Deterministic/Stochastic
  - Plan selection
  - Decision theoretic planning
- Observable/Partially Observable
  - Norm inference / learning
- Explicitly multiagent
  - Reasoning about other agents/trust

Norms in the BDI model

- Assumption: deterministic, fully observable environments
- Reasoning within the BDI model
  - Beliefs - World model (from perception)
  - Desires - Overall objectives (from user)
  - Intentions - Committed objectives / plans (selected at runtime)
- Norms constrain intention selection



AgentSpeak(L)

- Most implementations of BDI systems are based on the Procedural Reasoning System (PRS)
- Later formalised in the AgentSpeak(L) programming language
- Agents are defined in terms of a **plan library** of **procedural plans** (reactive HTN methods) of the form:
  triggering_event : context <- body.

AgentSpeak(L)

- An AgentSpeak plan has the following general structure:
  triggering_event : context <- body.
- where:
  - the triggering event denotes the events that the plan is meant to handle;
  - the context represent the circumstances in which the plan can be used;
  - the body is the course of action to be used to handle the event if the context is believed true at the time a plan is being chosen to handle the event.

AgentSpeak(L) example

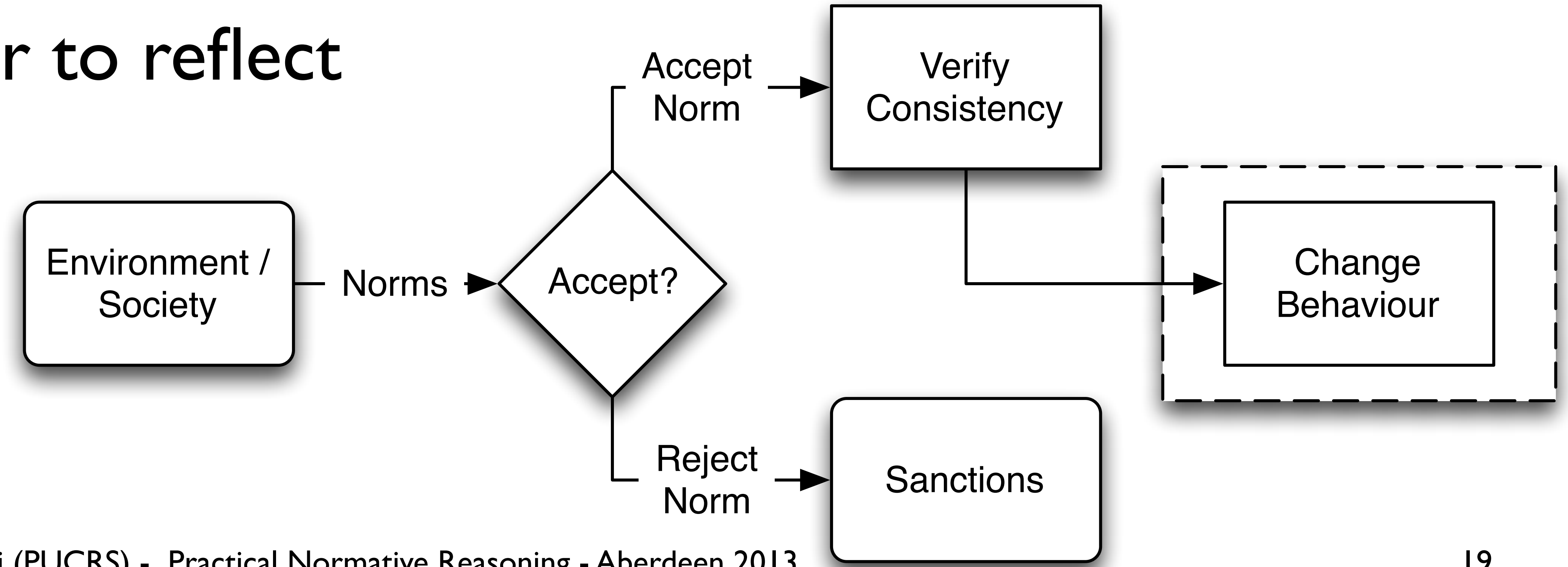
```
  +at(Pos) : gold(Pos)
  <- pickup(Pos).

+at(Pos) : gold(PosG)
  <- !goto(PosG).

+!goto(Pos) : at(PosA) & Pos < PosA
  <- move(left);
     !goto(Pos).

+!goto(Pos) : at(PosA) & Pos > PosA
  <- move(right);
     !goto(Pos).
```

BDI Normative reasoning

- Key processes:
  - Norm processing
  - Behaviour modification
  - Intention selection

BDI Behaviour Modification

- When new norms are perceived by the agent, it has to:
  - Detect normative conflicts (e.g. $\mathbf{O}p \wedge \mathbf{F}p$)
  - Decide whether to accept (and comply with) them
  - Change behaviour to reflect accepted norms



Norms and Goal Types

- We narrow norm types down to:
  - Obligations – agent must do/achieve something
  - Prohibitions – agent must not do/achieve something

Norm	Meaning
obligation(p)	add a goal to achieve state p, from Activation to Expiration.
obligation(a)	add a new plan with a Activation triggering event, and action a in its body.
prohibition(p)	prevent adoption of plans that bring about state p.
prohibition(a)	prevent adoption of plans that execute action a.

Motivating Example

AgentSpeak

```
+!cleanRoom(Room) : at(Room)
   <- +clean(Room).

+!clean(room1) : true
  <- +at(room1);
     !cleanRoom(room1).

+!clean(classifRoom) : true
  <- +at(classifRoom);
     !cleanRoom(classifRoom).

+cleanClassif : true
  <- !clean(classifRoom).
```

Norms

```
norm(time(4),
     time(20),
     obligation(clean(room1)))

norm(time(6),
     time(22),
   prohibition(at(classifRoom)))
```

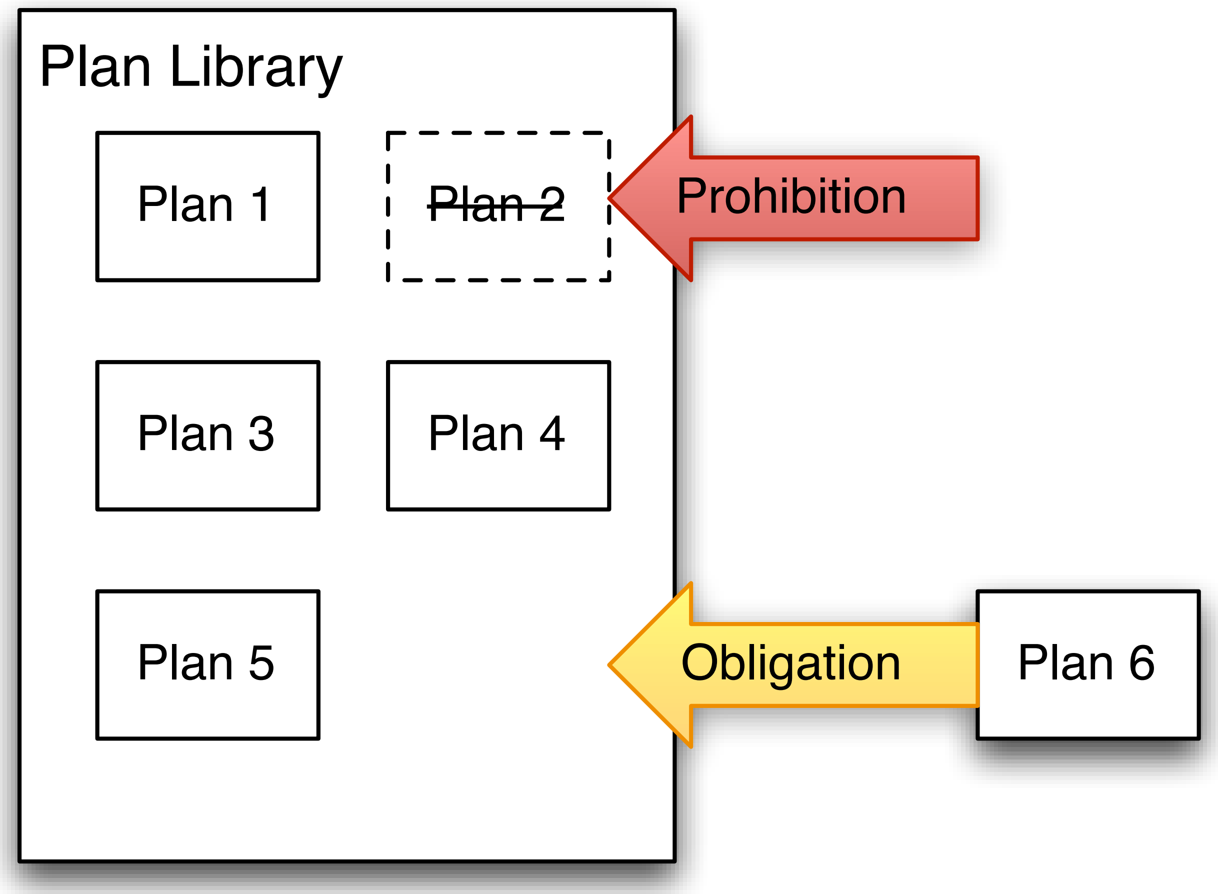
Expected Behaviour

- **time(4)**
- **time(6)**
- **cleanClassif**
- **time(20)**
- **time(22)**

- Adopt plan to clean room1
- Suppress plan to clean classifRoom
- No plan should be adopted
- Obligation to clean room1 expires
- Plan to clean classifRoom no longer suppressed

Norm Activation

- Obligations
  - Behaviours associated with obligations must be carried out when they become active
  - Activation condition becomes trigger for plans that achieve obligations
- Prohibitions
  - Behaviours associated with prohibitions must not be carried out when they become active
  - Activation conditions becomes trigger for plans that filter intentions and plan library



Norm Expiration

- When a norm expires, its effects in the plan library must be reversed
- Plans added for obligations can be removed
- Plans suppressed for prohibitions must be restored

Limitations and Refinement

- Only coarse control over agent behaviour is possible
  - Plans that affect prohibitions are completely removed
  - Plans created for obligations are not generic
- Finer grained approach
  - Restrict plan instantiation when selecting intentions

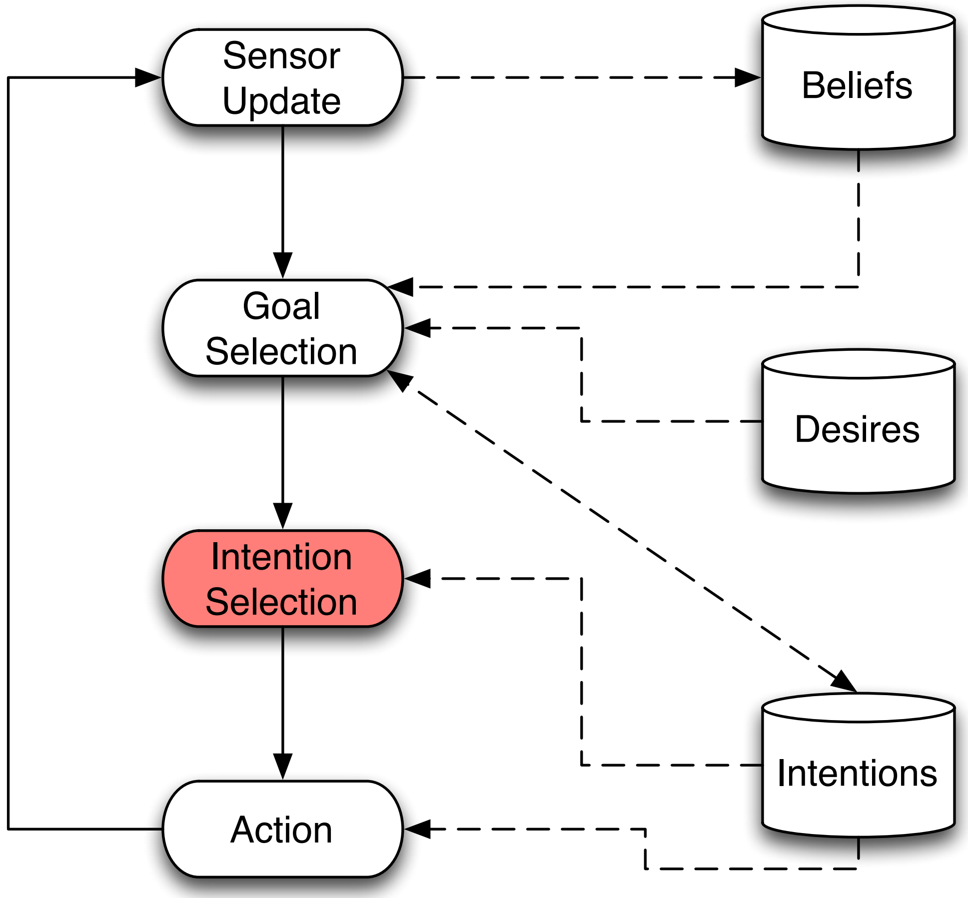
nu-BDI

- Norms constrain desirable states
- In AgentSpeak, plan library defines paths through the state space
- In nu-BDI norm condition is extended with a logical constraint

$$\mathsf{O}_\alpha \varphi \circ \Gamma \text{ (an obligation)}$$
$$\mathsf{F}_\alpha \varphi \circ \Gamma \text{ (a prohibition)}$$

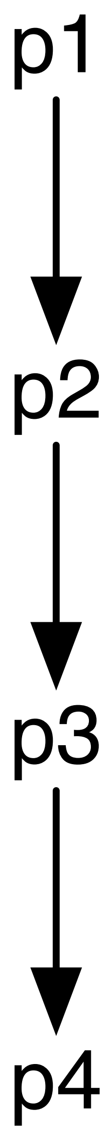
- Where Γ is a constraint formula:

$$\mathsf{O}_\alpha \varphi \circ (\gamma_1 \wedge \ldots \wedge \gamma_n)$$



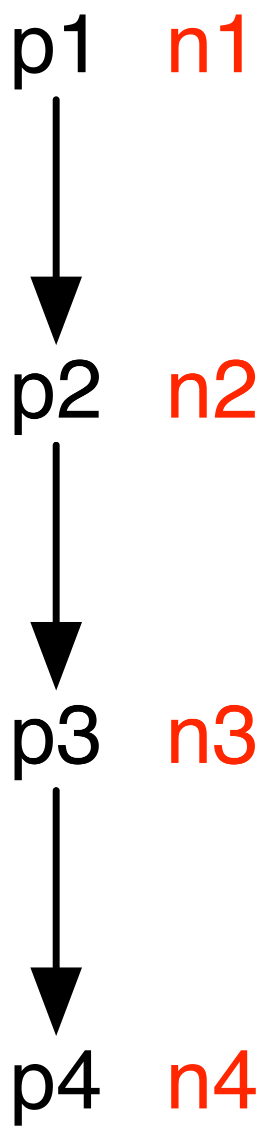
Annotating Plans

- General idea
  - Check applicable norms along possible execution paths
  - Consolidate restrictions detected in a path
  - Annotate plan with consolidated restrictions



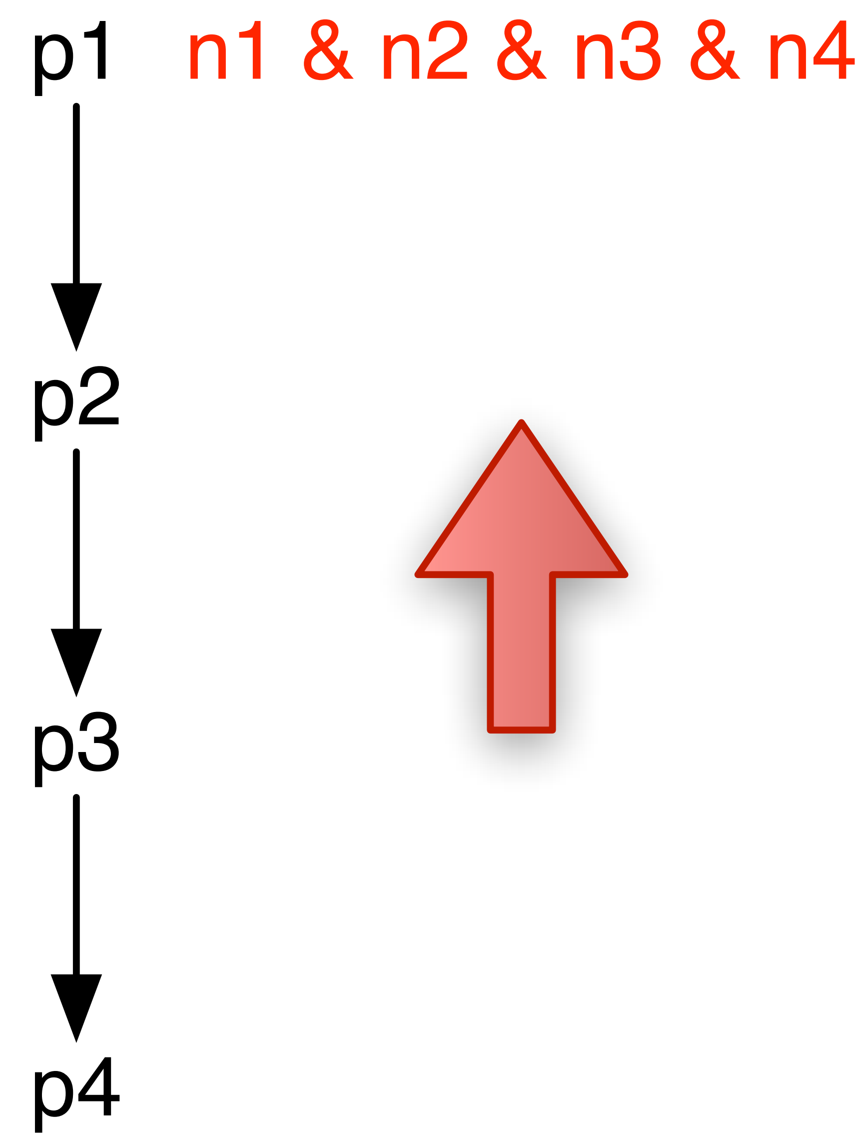
Annotating Plans

- General idea
  - Check applicable norms along possible execution paths
  - Consolidate restrictions detected in a path
  - Annotate plan with consolidated restrictions



Annotating Plans

- General idea
  - Check applicable norms along possible execution paths
  - Consolidate restrictions detected in a path
  - Annotate plan with consolidated restrictions



Example Plan and Norms

- Consider the following plan

$$\left\langle +level(X, medium), \begin{pmatrix} high\_risk(X) \wedge person(P) \wedge \\ at(P, X) \wedge \neg high\_risk(Y) \wedge \\ \neg high\_risk(Z) \wedge \neg(Y = Z) \end{pmatrix}, \begin{bmatrix} isolate(X), \\ evacuate(P, X, Y), \\ reroute(X, Z) \end{bmatrix} \right\rangle$$

- And the following **abstract norms**

$$\langle \mathsf{F}_A\, evacuate(P, X, Y), \neg safe(Y), safe(Y), 1 \rangle$$
$$\langle \mathsf{O}_A\, reroute(X, Z) \circ \{X + 1 \leq Z \leq X + 3\}, \neg safe(X), safe(X), 2 \rangle$$

- If the belief base entails both ¬safe(3) and ¬safe(6), we have the following **specific norms**

$$\langle \mathsf{F}_A\, evacuate(X, 3), \neg safe(3), safe(3), 1, ctr \rangle$$
$$\langle \mathsf{F}_A\, evacuate(X, 6), \neg safe(6), safe(6), 1, ctr \rangle$$
$$\langle \mathsf{O}_A\, reroute(3, Z) \circ \{4 \leq Z \leq 6\}, \neg safe(3), safe(3), 2, ctr \rangle$$
$$\langle \mathsf{O}_A\, reroute(6, Z) \circ \{7 \leq Z \leq 9\}, \neg safe(6), safe(6), 2, ctr \rangle$$

Extended Context Condition

- Given the norms

$$\langle \mathsf{F}_A\, evacuate(X,3), \neg safe(3), safe(3), 1, ctr\rangle$$
$$\langle \mathsf{F}_A\, evacuate(X,6), \neg safe(6), safe(6), 1, ctr\rangle$$
$$\langle \mathsf{O}_A\, reroute(3,Z) \circ \{4 \leq Z \leq 6\}, \neg safe(3), safe(3), 2, ctr\rangle$$
$$\langle \mathsf{O}_A\, reroute(6,Z) \circ \{7 \leq Z \leq 9\}, \neg safe(6), safe(6), 2, ctr\rangle$$

- And the plan steps

$$\begin{bmatrix} isolate(X), \\ evacuate(P,X,Y), \\ reroute(X,Z) \end{bmatrix} \Rightarrow \begin{bmatrix} isolate(X) \circ \top, \\ evacuate(X,Y) \circ \{Y \neq 3, Y \neq 6\}, \\ reroute(X,Z) \circ \{3 \leq Z \leq 5\} \end{bmatrix}$$

- We get an annotated plan

$$+level(X, medium), (high\_risk(X)),$$
$$\left\langle \begin{bmatrix} isolate(X) \circ \top, \\ evacuate(X,Y) \circ \{Y \neq 3, Y \neq 6\}, \\ reroute(X,Z) \circ \{3 \leq Z \leq 5\} \end{bmatrix}, \begin{Bmatrix} Y \neq 3, \\ Y \neq 6, \\ 3 \leq Z \leq 5 \end{Bmatrix} \right\rangle$$

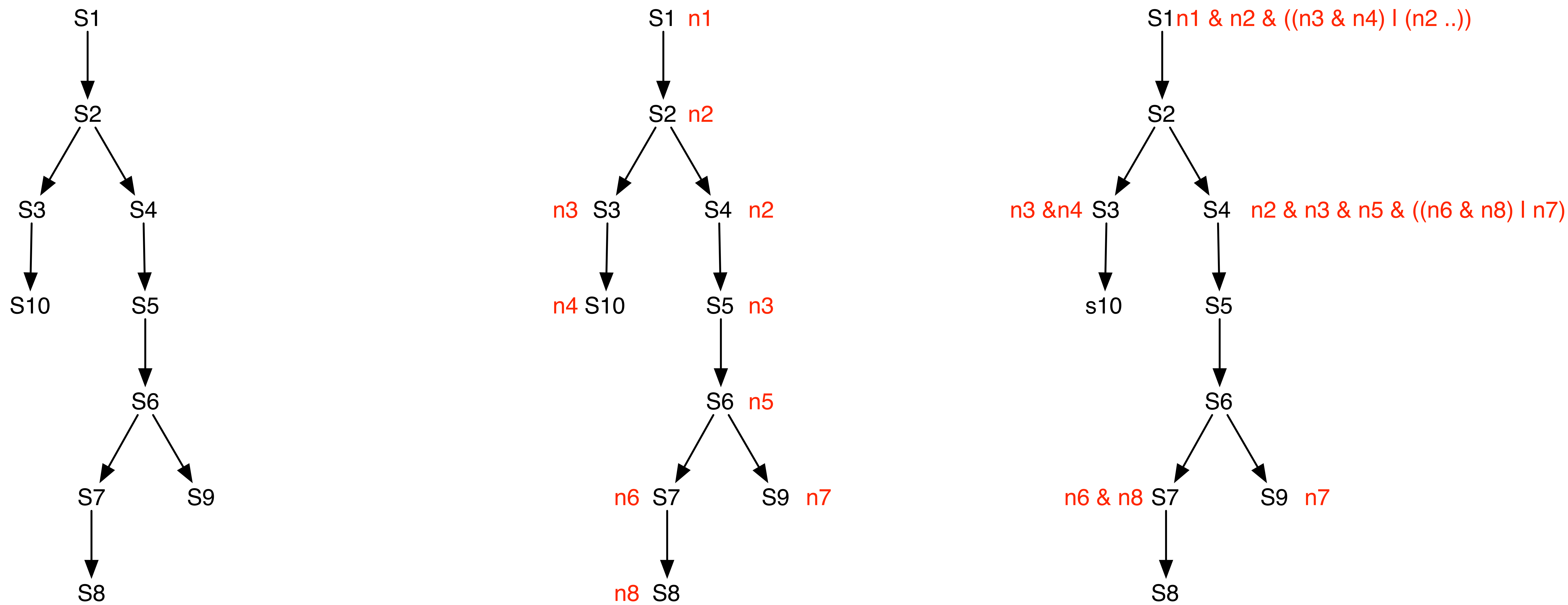
Choosing between plans

- The extended (normative) context condition can now be checked at plan instantiation time
  - If satisfiable, plan **can be** norm-compliant
  - If not, then **no plan** instance can be compliant
- Agent needs to choose least violating plan
  - Relaxing constraints (constraints $\rightarrow$ norms)

Challenges

- Determining the limitations of annotation mechanism
  - What can we guarantee?
  - How far can we look?
- Guarantees of norm properties w.r.t. a plan library
  - What norms can be followed by a plan library?
- Algebra of norms on plans

Challenges

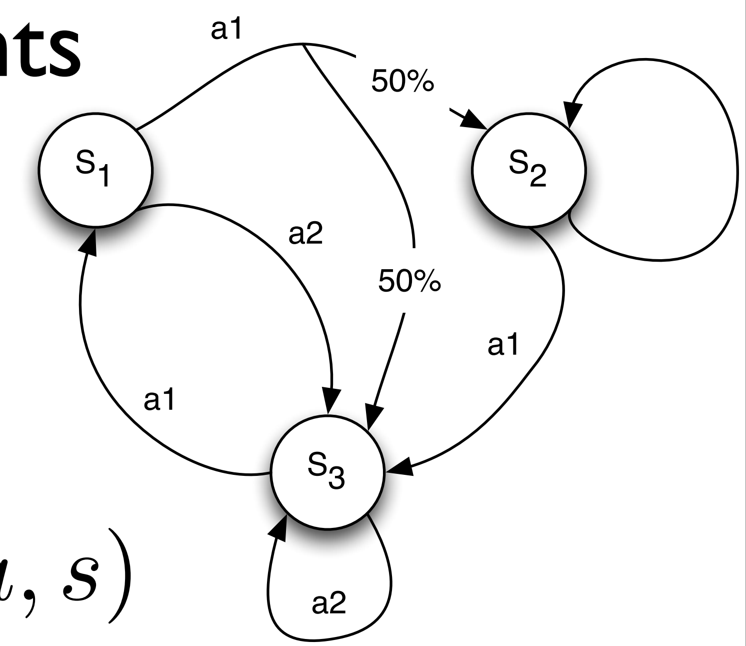

S1
S2
S3
S4
S10
S5
S6
S7
S9
S8
S1 n1
S2 n2
n3 S3
S4 n2
n4 S10
S5 n3
S6 n5
n6 S7
S9 n7
n8 S8
S1n1 & n2 & ((n3 & n4) I (n2 ..))
S2
n3 &n4 S3
S4 n2 & n3 & n5 & ((n6 & n8) I n7)
s10
S5
S6
n6 & n8 S7
S9 n7
S8

Alternative Approaches

- Planning with preferences
- Constraint Satisfaction planning

MDPs

- Assumption: Stochastic, Fully-Observable Environments
- Markov Decision Process (MDP) $\langle S, A, T, R \rangle$
  - Sets of states and actions
  - A *markovian* transition model $T(s', a, s) = P(s' \mid a, s)$
  - A reward function $R(s)$ $sometimes$ $R(s, a)$
- A solution to a MDP must specify what the agent should do for any state. Such a solution is called a **policy**



Optimal Policy

- A policy maps each state in the state-space to an action
- If this mapping selects the action that leads to the long-term **maximum reward**, then the policy is **optimal**
- This selection is done by calculating the **value** of a state

$$V(s) \leftarrow \left[\max_a \gamma \sum_{s'} P(s'|s,a) * V(s')\right] + R(s)$$

- And subsequently choosing the action that leads to the highest value $\pi(s) = \arg\max_a \sum_{s'} P(s'|s,a) * V(s')$

Normative MDP

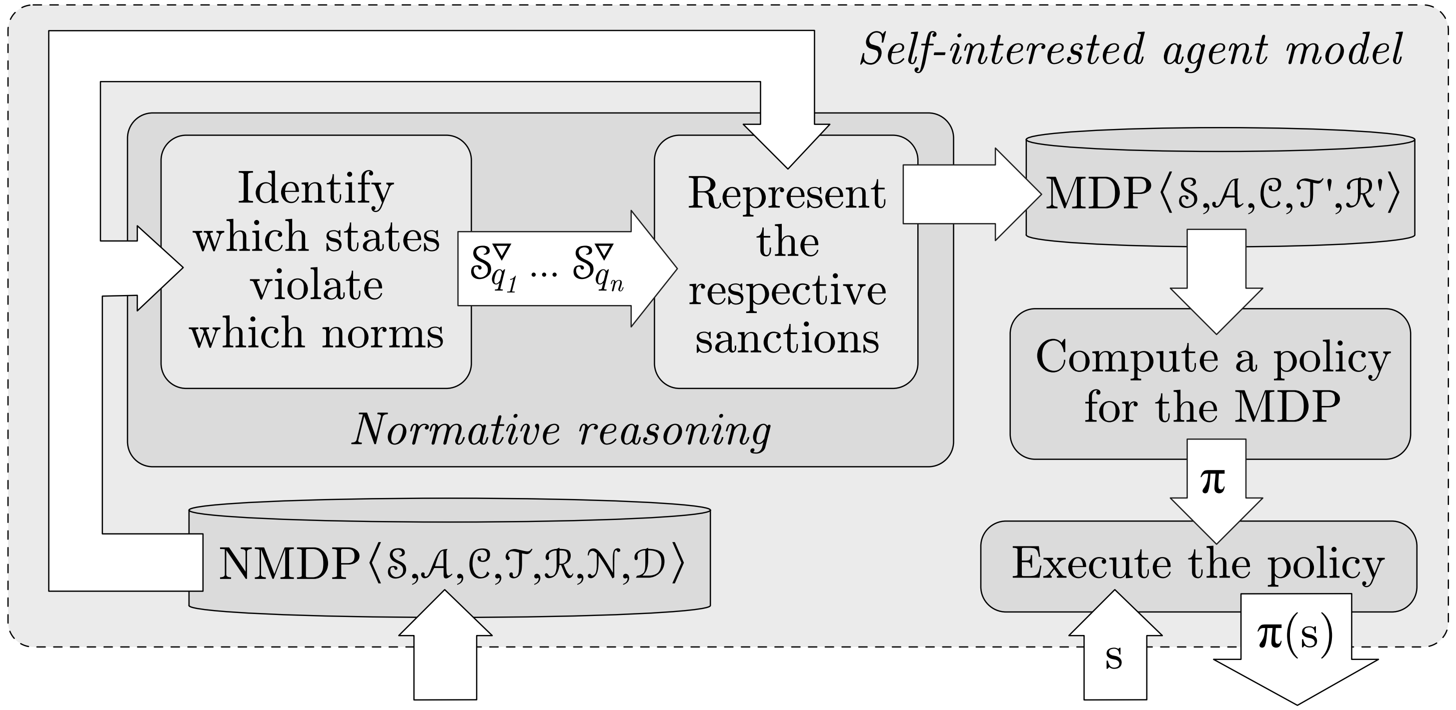
$\langle S, A, T, R, N, D \rangle$

- S → States
- A → Actions
- T → Transition Function
- R → Reward Function
- N → Norms → $\langle \nu, \chi, \varepsilon, \sigma \rangle$
  - ν → Deontic Modality
  - χ → Norm Context
  - ε → Norm Target
  - σ → Sanction → $\langle \rho, \phi \rangle$
    - ρ → Penalty
    - ϕ → Forced Transition
- D → Detection Model

- MDP extended with Norms
  - Norms include not only a penalty, but also a enforced transition
  - Detection model denotes the probability of a sanction being applied when norm is violated

NMDP Policies

- Current state of the art consists of compiling an NMDP into a traditional MDP
  - Removing actions that can transition to violating states Fully compliant behaviour
  - Merging sanctioning mechanism into transitions and rewards Selfish behaviour (asocial)



Limitations and Challenges

- Current state of the art generates agents that are either totally cautious or oblivious to violations
- Challenges in defining policy concepts and algorithms to strike a middle ground
- Many possible approaches

Regret Minimisation

- Potential approach, choose actions that minimise regret over sanctions
  - Keep track of sanctioned rewards separately from environment rewards

$$V(s) = R(s) + \max_a \gamma \sum_{s' \in S} T(s,a,s')V(s')$$

$$V_N(s) = R(s) + R_N(s) + \max_a \gamma \sum_{s' \in S} T(s,a,s')V_N(s')$$

  - And generate a policy that minimises regret between these two rewards

$$\pi^*_\delta = \arg\min_a \left(\sum_{s' \in S} T(s,a,s')V_N(s') - \sum_{s' \in S} T(s,a,s')V(s') \right)$$

Sanctioning Minimisation

- Alternatively, choose actions that minimise sanctions when these are inevitable
  - Keep track of sanction values separately

$$V_{|N|} = R_N(s) + \gamma \sum_{s' \in S} T(s, \pi^*_N(s), s')V_{|N|}(s')$$

  - And only maximise reward when sanctions are not present

$$\pi^*_{|N|} = \begin{cases} \arg\max_a \sum_{s' \in S} T(s, a, s')V_N(s') & \text{if } V_{|N|}(s') > 0 \\ \arg\min_a \sum_{s' \in S} T(s, a, s')V_{|N|} & \text{if } V_{|N|}(s') \leq 0 \end{cases}$$

Further Reading

- Meneguzzi et al. - Norm-based behaviour modification in BDI agents. AAMAS (1) 2009: 177-184.
- Meneguzzi et al. - Nu-BDI: Norm-aware BDI Agents. EUMAS 2012
- Fagundes, Ossowski and Meneguzzi - Norm enforcement in stochastic environments populated with self-interested agents. Under Review.