Formalizing Commitments Using Action Languages

Tran Cao Son[1], Enrico Pontelli[1],
Chiaki Sakama[2]

[1]New Mexico State University

[2]Wakayama University

Commitments

- Commitments are an integral part of agent societies.

  - Related to agents' behavior and capabilities

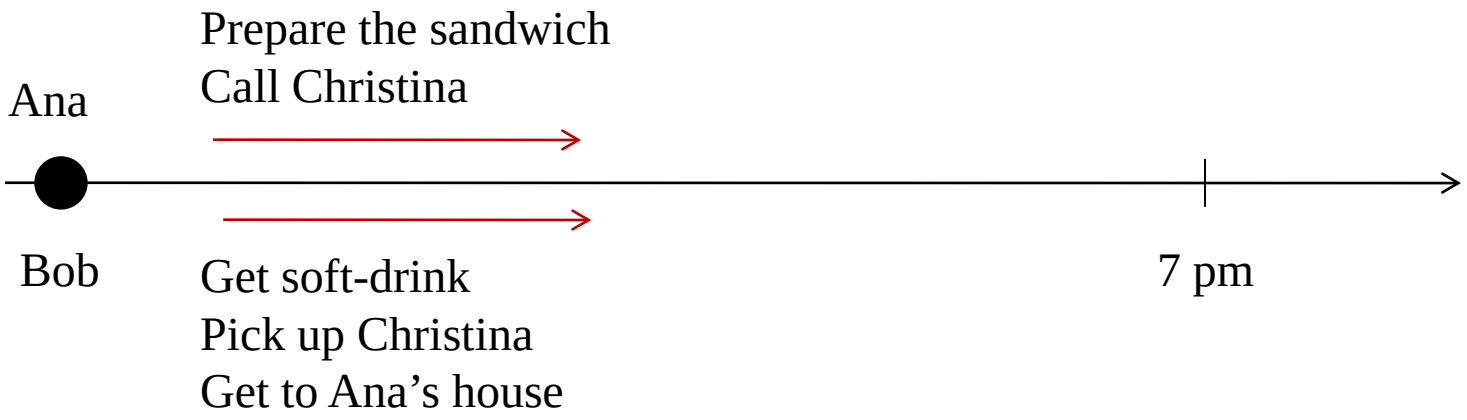
  - Often associated with time constraints in the future

  - Should be considered in conjunction with action and changes

Example

- ***Ana***: Do you want to do something tonight?
- ***Bob***: Sure, what do you want to do?
- ***Ana***: Let us have a pot-luck dinner with *Christina*. I will prepare some sandwiches and call *Christina*. But can you pick her up? Also, could you bring some soft-drinks?
- ***Bob***: Sure. How about 7pm?
- ***Ana***: Great.

Commitments



- Ana commits to prepare sandwich and call Christina.
- Bob commits to get soft-drink, picks up Christina, and both go to Ana's house.
- Reasoning about commitments strongly related to agents' behaviors and capabilities, time, deadlines.

Questions

- What does it mean for an agent
  - to satisfy a commitment?
  - to violate a commitment?
- How to specify commitment in a declarative manner?
- How to reason about commitment?

Goal and Contributions

- Investigate the use of action languages in formalizing commitments
- Contributions
  - An action language $\mathcal{L}^{mt}$ suitable for reasoning about commitment
  - Uniform view of reasoning about satisfaction, violation, achievement of commitments as reasoning about narratives and planning

Action Language $\mathcal{L}^{\mathrm{mt}}$

- Domain description
  - over set of agents $\mathcal{AG}$, sets of fluents $\mathcal{F}_{\mathrm{i}}$, sets of actions $\mathcal{A}_{\mathrm{i}}$
  - set of statements encoding effects of actions
  - effects of actions: delay, override by latter actions
  - action precondition

$\mathcal{L}^{mt}$ Example - Netbill Protocol

- Customer
  - request for a quote
  - accept a quote if quote has been received
  - send a payment if quote is accepted
  - cancel payment if check has not been deposited
- Merchant
  - send a quote
  - send the goods
  - send the receipt if payment is made

$\mathcal{L}^{mt}$ Syntax

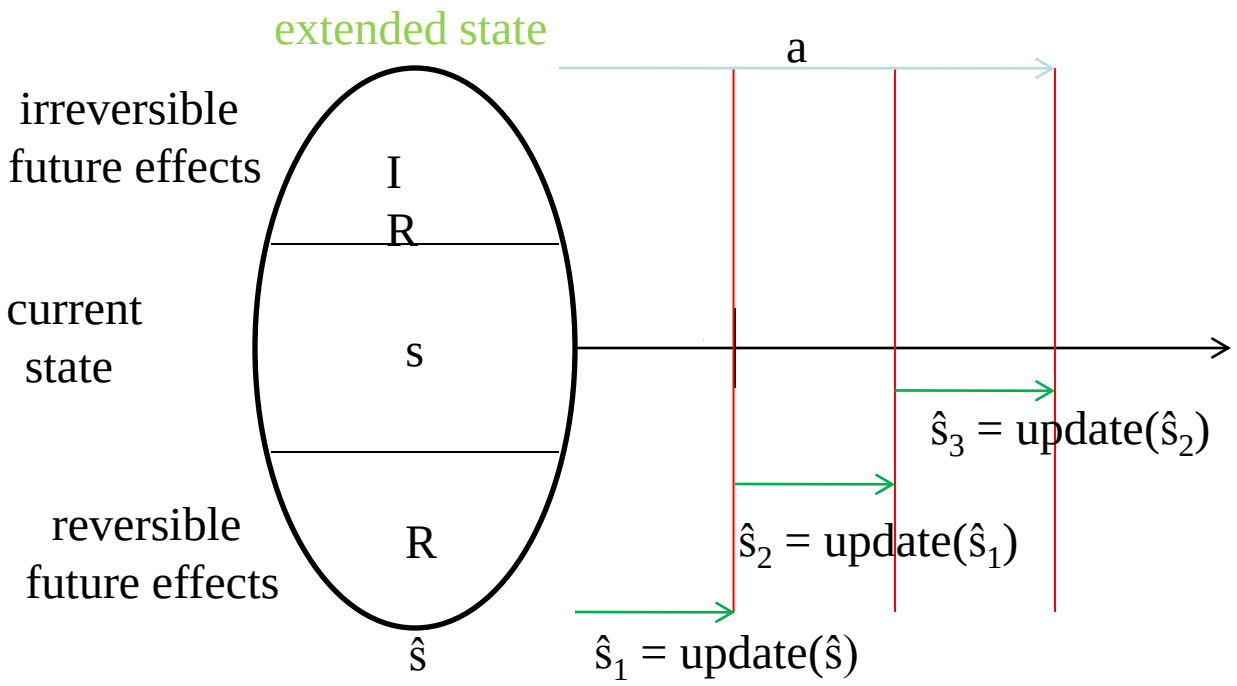
- Customer
  - sendPayment **starts** payment_process **reversible** paid$^{\vee[3,5]}$

    [sendPayment starts a reversible process that causes *paid* to be true within 3 to 5 unit of time]
  - cancelPayment **stops** payment_process

    [cancelPayment stops the payment process]
  - **impossible** {cancelPayment} **if** paid
  - ...
- Merchant
  - sendQuote **starts** quote_process **irreversible** quote$^{\vee[2,2]}$

    [sendQuote starts an irreversible process that causes quote to be true after 2 unit of time]
  - ...

$\mathcal{L}^{mt}$ Semantics transition function based approach



- update(ŝ): the state after one unit of time
- $\Phi(\hat{s}, \alpha)$: set of trajectories if α is executed in ŝ
- $\Phi(\hat{s}, \alpha, t) + t_1$: set of extended states t_1 unit from the time α is executed which starts after t unit from ŝ

Representing Observations

φ **at** s

α **between** s_1, s_2

α **occurs_at** s

$s_1 < s_2$

s **at** time

$\neg$paid **at** s_0

$\neg$accept **at** s_0

$\neg$quote **at** s_0

$\neg$good **at** s_0

sendRequet **occurs_at** s_0

sendAccept **occurs_at** s_1

sendGood **between** s_2 , s_3

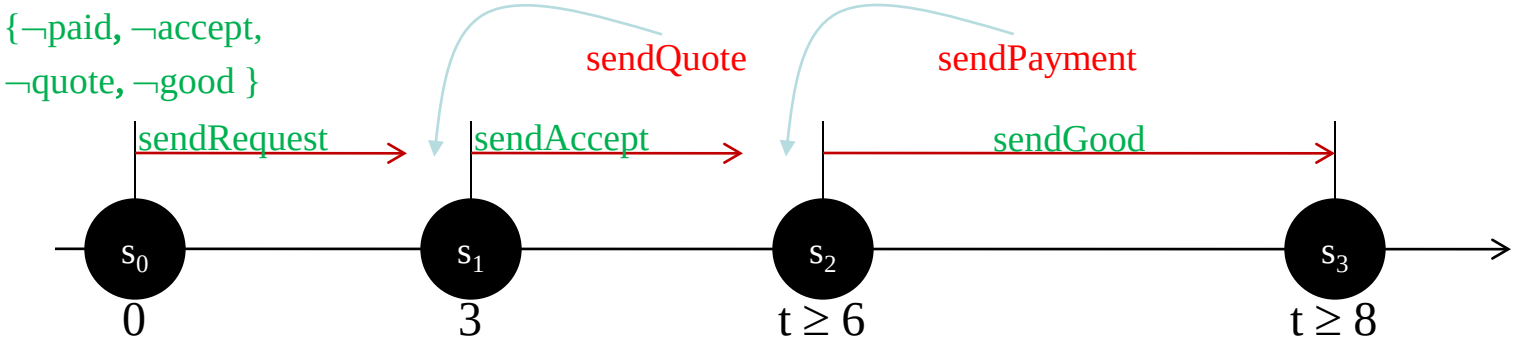
s_2 **at** 2

$s_0 < s_1 < s_2 < s_3$

$\mathcal{L}^{mt}$ Narratives

- Narrative: (D,O)
  - D: domain specification
  - O: observation
- Specifies a set of possible trajectories

$\neg$paid **at** s_0
$\neg$accept **at** s_0
$\neg$quote **at** s_0
$\neg$good **at** s_0
sendRequet **occurs_at** s_0
sendAccept **occurs_at** s_1
sendGood **between** s_2 , s_3
s_1 **at** 3
$s_0 < s_1 < s_2 < s_3$



Commitments in $\mathcal{L}^{mt}$

- Commitment: $c(x,y,\varphi,t_1,t_2)$: $x,y \in \mathcal{AG}$, φ - fluent formula, $t_1 \leq t_2$ - time points
  - c(Anna, Bob, sandwich, 0, 7)
  - c(Bob, Anna, arrived, 0, 7)
- Activities:
  - create(x,y,φ,t_1,t_2) [x commits to archive φ for y between t_1 and t_2]
  - discharge(x,y,φ) [x discharges the commitment φ towards y]
  - release(x,y,φ) [x releases y from the commitment φ]
  - assign(x,y,k,φ,t_1,t_2) [y transfers φ to another creditor]
  - delegate(x,y,k,φ,t_1,t_2) [x delegates φ to another debtor]
  - cancel$(x,y,\varphi,\psi,t_1,t_2)$ [x cancels φ and generate a new ψ]

Manipulation of Commitments

- Enabling statements

  [φ|a] **triggers** commitment_activity

- Examples:
  - Merchant agrees to send the receipt between 1 and 3 working days after receiving payment

    paid **triggers** c(m,c,receipt,1,3)
  - Customer agrees to pay between 5 working days after sending the acceptance of the quote

    sendAccept **triggers** create(c,m,paid,1,5)

Domain with Commitments

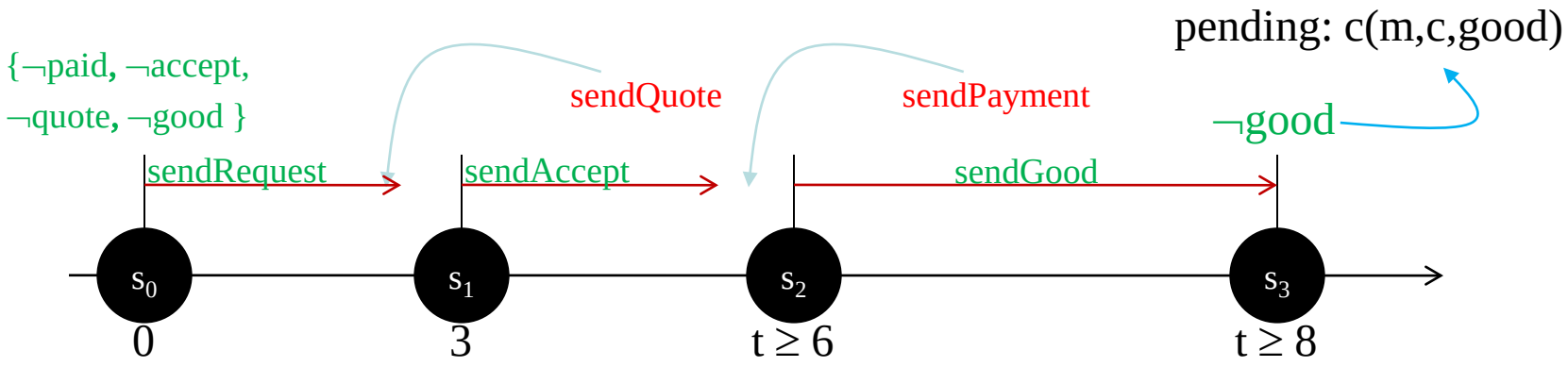
- M = (D,C) – D is a domain description and C is a set of trigger statements
- Semantics of M = (D,C) is specified by translating M to a $\mathcal{L}^{mt}$ domain M*
  - a **triggers** create(x,y,φ,t_1,t_2) is translated into
    - a **causes** c(x,y,φ)
    - a **starts** c(x,y,φ) **reversible** done(x,y,φ)$^{\vee[t_1,t_2]}$ where done(x,y,φ) means c(x,y,φ) needs to be realized.
  - …

Commitments w.r.t a Sequence of States

- M = (D,C) and $[u_0, \ldots, u_n]$ is a sequence of states
- c(x,y,φ) is
  - satisfied in u_n if $u_n \models \neg c(x,y,\varphi)$
  - violated in u_n if $u_n \models c(x,y,\varphi) \wedge done(x,y,\varphi)$
  - pending in u_n if $u_n \models c(x,y,\varphi)$ and $u_n \not\models done(x,y,\varphi)$

Commitments wrt. a Narrative

- (D,O,C) – (D,C) is a domain with commitments and O is a set of observations
- (D,O,C) – specifies a set of trajectories ➔ enable the reasoning about satisfied, violated, or pending commitments
- E.g., sendGood triggers create(m,c,good,3,7)



Planning for Pending Commitments

- Given M = (D,O,C)
- Identification of pending commitments:

  $M \models c(x,y,\varphi)$ and $M \not\models done(x,y,\varphi)$

- How to satisfy $c(x,y,\varphi)$?

  Planning to achieve $\neg c(x,y,\varphi)$

Related Work

- $\mathcal{L}^{mt}$ first action language with
  - static causal laws [Action language B, Gelfond & Lifschitz]
  - time and deadlines [ADC, Baral et al.]
  - observations [L, Baral et al.]
  - multi-agent domains
  - reversible and irreversible effects
- Commitments
  - basic commitment ontology as in [Mallaya & Huhns, Singh]
  - protocols different from [Giordano et al., Yolum & Singh]

Conclusions

- Introduced an action language for multi-agent domain with time, deadline, reversible and irreversible effects, and observations.
- Achieved reasoning about commitments and protocols through entailment relation between narratives and queries.
- Our next goal is applying multiagent planning techniques in generating plans to satisfy pending commitments.