Specification and Analysis of Contracts
Lecture 6
Challenges in Defining a Good Language for Contracts

Gerardo Schneider

gerardo@ifi.uio.no

http://folk.uio.no/gerardo/

Department of Informatics,
University of Oslo

SEFM School, Oct. 27 - Nov. 7, 2008
Cape Town, South Africa

Plan of the Course

1. Introduction
2. Components, Services and Contracts
3. Background: Modal Logics 1
4. Background: Modal Logics 2
5. Deontic Logic
6. Challenges in Defining a Good Contract language
7. Specification of 'Deontic' Contracts ($\mathcal{CL}$)
8. Verification of 'Deontic' Contracts
9. Conflict Analysis of 'Deontic' Contracts
10. Other Analysis of 'Deontic' Contracts and Summary

Plan

1. An 'Ideal' Language for Contracts

2. The Language of Discourse

3. Difficulties in defining a good formal language for contracts

Plan

1. An 'Ideal' Language for Contracts

2. The Language of Discourse

3. Difficulties in defining a good formal language for contracts

Uses of a 'deontic' contract language

1. Service-oriented architectures
2. Component-based development
3. Fault-tolerant systems;
4. Compensable actions (long transactions);
5. Regulatory systems

Uses of a 'deontic' contract language

1. Service-oriented architectures
2. Component-based development
3. Fault-tolerant systems;
4. Compensable actions (long transactions);
5. Regulatory systems

- We have seen 1 and 2
- Both 3 and 4:
  - A (*mandatory*) behavior will not necessarily be respected due to failures
  - When a failure occurs, backtracking is needed to a previous state where an *alternative* behavior must be enforced
  - This is very much what CTDs and CTPs do
  - Sometimes we need to specify *exceptions*
- Regulatory systems are normative systems containing regulation and policies rich on
  - Intra and inter cross references
  - Primary obligations and exceptional cases

An 'Ideal' formal language for contracts

We call OPP-logic a logic containing the following:

- Modalities for obligation, permission and prohibition
  - Defined over complex actions (Kleene star, sequences, choices, concurrency, negation, complement)
- Nested CTDs and CTPs
- Temporal (causal) aspects
- Nested exceptions
- Real-time aspects
- References to other expressions or clauses
- Invariants
- (Bounded) fairness constraints
- Introspection/reflection

An 'Ideal' formal language for contracts

A proposal...

- In what follows we will propose an 'ideal' language for specifying contracts
  - We will discuss issues related to the OPP-logic
- We will concentrate on the problems of a good interpretation (semantics)
- More questions than answers!

Plan

1. An 'Ideal' Language for Contracts

2. The Language of Discourse

3. Difficulties in defining a good formal language for contracts

The language of discourse

Actions

- We assume a set of *simple actions* SimpAction as for instance *pay*, *send*, etc.

Actions

$$\text{Action} ::= \varepsilon \mid \text{Any} \mid \text{SimpAction} \mid \text{SimpAction(Param)} \mid \text{Action \& Action} \mid \overline{\text{Action}}$$

The language of discourse

Actions

- We assume a set of *simple actions* SimpAction as for instance *pay*, *send*, etc.

Actions

$$\text{Action} ::= \varepsilon \mid \text{Any} \mid \text{SimpAction} \mid \text{SimpAction}(\text{Param}) \mid \text{Action} \,\&\, \text{Action} \mid \overline{\text{Action}}$$

Example

$pay(200), \; pay \,\&\, sendAck$

We will use lower-case Latin letters, $a, b, c, \ldots$ to denote basic actions

The language of discourse

Expressions over actions

- Reason about causality, sequentiality, choice, concurrency and repetition

Compound Actions

$$\begin{array}{rcl}\mathsf{CompAction} & ::= & \mathsf{Action} \,|\, \neg\ \mathsf{CompAction} \,|\, \mathsf{CompAction}^* \\ & & |\, \mathsf{CompAction}\ +\ \mathsf{CompAction} \,|\, \mathsf{CompAction}\ \&\ \mathsf{CompAction} \\ & & |\, \mathsf{CompAction}\ .\ \mathsf{CompAction}\end{array}$$

The language of discourse

Expressions over actions

- Reason about causality, sequentiality, choice, concurrency and repetition

Compound Actions

$$\begin{aligned} \text{CompAction} \quad ::= \quad & \text{Action} \mid \neg\ \text{CompAction} \mid \text{CompAction}^* \\ & \mid \text{CompAction}\ +\ \text{CompAction} \mid \text{CompAction}\ \&\ \text{CompAction} \\ & \mid \text{CompAction}\ .\ \text{CompAction} \end{aligned}$$

Example

$$(\mathit{keepPromise} + (\overline{\mathit{keepPromise}}\ .\ (\mathit{pay}(200) + (\mathit{notify}\ .\ \mathit{pay}(400)))))^*$$

The language of discourse

Deontic operators

- At least the deontic notions of obligation, permission and prohibition

Simple Deontic Contracts

$$\text{SimpContract} \quad ::= \quad \mathbb{Y} \,|\, \mathbb{N} \,|\, \mathbb{P}(\text{CompAction}) \,|\, \mathbb{F}(\text{CompAction}) \,|\, \mathbb{O}(\text{CompAction})$$

The language of discourse

Deontic operators

- At least the deontic notions of obligation, permission and prohibition

Simple Deontic Contracts

$$\text{SimpContract} \quad ::= \quad \mathbb{Y} \,|\, \mathbb{N} \,|\, \mathbb{P}(\text{CompAction}) \,|\, \mathbb{F}(\text{CompAction}) \,|\, \mathbb{O}(\text{CompAction})$$

Example

$\mathbb{O}(\mathit{keepPromise})$, $\mathbb{F}(\mathit{notify}\ .\ \overline{\mathit{pay}(400)})$

The language of discourse

Default contracts

- Normal vs exceptional behavior
  - Contrary-to-duties
  - Contrary-to-prohibitions
  - Exceptions

Compound Contracts

$$\begin{aligned}\mathsf{CompContract} \quad ::= \quad & \mathsf{SimpContract} \,|\, \mathsf{CTD}(\mathsf{CompAction},\ \mathsf{CompContract}) \\ & |\, \mathsf{CTP}(\mathsf{CompAction},\ \mathsf{CompContract}) \\ & |\, \mathsf{CompAction}\ \textit{unless}\ \mathsf{CompContract}\end{aligned}$$

The language of discourse

Default contracts

- Normal vs exceptional behavior
  - Contrary-to-duties
  - Contrary-to-prohibitions
  - Exceptions

Compound Contracts

$$\begin{array}{rcl} \text{CompContract} & ::= & \text{SimpContract} \,|\, \text{CTD(CompAction, CompContract)} \\ & & |\, \text{CTP(CompAction, CompContract)} \\ & & |\, \text{CompAction } \textit{unless} \text{ CompContract} \end{array}$$

Example

$CTD(keepPromise, \mathbb{O}(pay(200) + (notify\ .\ pay(400))))$

The language of discourse

Expressions over contracts

- Temporal operators over contracts
- Based on regular expressions

Expressions Over Contracts

$$\begin{aligned}\mathsf{Contract} \quad ::= \quad & \mathsf{CompContract} \mid \neg\mathsf{Contract} \mid \mathsf{Contract}^{*} \mid \mathsf{Contract} + \mathsf{Contract} \\ & \mid \mathsf{Contract}\ \&\ \mathsf{Contract} \mid \mathsf{Contract}\ .\ \mathsf{Contract} \\ & \mid \mathsf{CompAction?}\ .\ \mathsf{Contract}\end{aligned}$$

The language of discourse

Expressions over contracts

- Temporal operators over contracts
- Based on regular expressions

Expressions Over Contracts

$$\begin{aligned}\text{Contract} \quad ::= \quad & \text{CompContract} \mid \neg\text{Contract} \mid \text{Contract}^* \mid \text{Contract} + \text{Contract} \\ & \mid \text{Contract} \,\&\, \text{Contract} \mid \text{Contract} \;.\; \text{Contract} \\ & \mid \text{CompAction?} \;.\; \text{Contract}\end{aligned}$$

Example

$CTD(keepPromise, \mathbb{O}(pay(200) + (notify\ .\ pay(400))) \ \&\ \mathbb{F}(sendFalseInf))$

Plan

1. An 'Ideal' Language for Contracts

2. The Language of Discourse

3. Difficulties in defining a good formal language for contracts

Sequences over contracts vs contracts over sequences

- $\mathbb{F}(a.b)$ and $\mathbb{F}(a).\mathbb{F}(b)$ are different
  - Should we interpret $\mathbb{F}(a.b)$ as $a?.\mathbb{F}(b)$?
- What about $\mathbb{O}(a.b)$ and $\mathbb{O}(a).\mathbb{O}(b)$?
  - They may be equal if only interested on the normal behavior
  - In the presence of a *contract break* (e.g. not doing a) they should be different
    - We could add an exception or CTD to each step in the second case
- We could also interpret the sequential operator '.' inside and outside the modalities as *external* and *internal*

Sequences over contracts vs contracts over sequences

- $\mathbb{F}(a.b)$ and $\mathbb{F}(a).\mathbb{F}(b)$ are different
  - Should we interpret $\mathbb{F}(a.b)$ as $a?.\mathbb{F}(b)$?
- What about $\mathbb{O}(a.b)$ and $\mathbb{O}(a).\mathbb{O}(b)$?
  - They may be equal if only interested on the normal behavior
  - In the presence of a *contract break* (e.g. not doing a) they should be different
    - We could add an exception or CTD to each step in the second case
- We could also interpret the sequential operator '.' inside and outside the modalities as *external* and *internal*

Sequences over contracts vs contracts over sequences

- $\mathbb{F}(a.b)$ and $\mathbb{F}(a).\mathbb{F}(b)$ are different
  - Should we interpret $\mathbb{F}(a.b)$ as $a?.\mathbb{F}(b)$?
- What about $\mathbb{O}(a.b)$ and $\mathbb{O}(a).\mathbb{O}(b)$?
  - They may be equal if only interested on the normal behavior
  - In the presence of a *contract break* (e.g. not doing a) they should be different
    - We could add an exception or CTD to each step in the second case
- We could also interpret the sequential operator '.' inside and outside the modalities as *external* and *internal*

Causality

- Let us consider CTD(α, C)
    - An obligation to perform α is enacted, but if it is not, contract C has to be satisfied
- Two different views of the operator:
    1. C must hold as soon as (or one time unit after) the initial obligation is broken
    2. The choice between performing the obligations or the alternative contract C as soon as the CTD is enacted
- Problems with the second interpretation
    - Ex: CTD(Any.a, $\mathbb{O}(b)$)
    - An initial action b may satisfy the CTD or not –there is no way we can know this until we get the second set of events

Causality

- Let us consider CTD(α, C)
  - An obligation to perform α is enacted, but if it is not, contract C has to be satisfied

- Two different views of the operator:
  1. C must hold as soon as (or one time unit after) the initial obligation is broken
  2. The choice between performing the obligations or the alternative contract C as soon as the CTD is enacted

- Problems with the second interpretation
  - Ex: CTD($\mathrm{Any}.a, \mathbb{O}(b)$)
  - An initial action b may satisfy the CTD or not –there is no way we can know this until we get the second set of events

Causality

- Let us consider CTD(α, C)
  - An obligation to perform α is enacted, but if it is not, contract C has to be satisfied
- Two different views of the operator:
  1. C must hold as soon as (or one time unit after) the initial obligation is broken
  2. The choice between performing the obligations or the alternative contract C as soon as the CTD is enacted
- Problems with the second interpretation
  - Ex: CTD($\mathsf{Any}.a, \mathbb{O}(b)$)
  - An initial action b may satisfy the CTD or not –there is no way we can know this until we get the second set of events

Breaking an obligation

Example

The law of a country says that: 'You are obliged to hand in Form A on Monday and Form B on Tuesday, unless officials stop you from doing so.'

On Monday, John spent a day on the beach, thus not handing in Form A. On Tuesday at 00:00 he was arrested, and brought to justice on Wednesday.

The police argue: 'To satisfy his obligation the defendant had to hand in Form A on Monday, which he did not. Hence he should be found guilty.'

But John's lawyer argues back: 'But to satisfy the obligation the defendant had to hand in Form B on Tuesday, which he was stopped from doing by officials. He is hence innocent.'

Breaking an obligation

- Who is right?
- Formalizing the primary obligation in the law, we get $\mathbb{O}(a.b)$, where a represents handling Form A on Monday and b handling Form B on Tuesday
- When is the obligation to be considered violated — upon the lack of action a, or at the end of two consecutive actions?
- It will depend on whether we model the above with CTDs or “unless”, and what is the formal semantics

Breaking an obligation

- Who is right?
- Formalizing the primary obligation in the law, we get $\mathbb{O}(a.b)$, where a represents handling Form A on Monday and b handling Form B on Tuesday
- When is the obligation to be considered violated — upon the lack of action a, or at the end of two consecutive actions?
- It will depend on whether we model the above with CTDs or "unless", and what is the formal semantics

Breaking an obligation

- Who is right?
- Formalizing the primary obligation in the law, we get $\mathbb{O}(a.b)$, where a represents handling Form A on Monday and b handling Form B on Tuesday
- When is the obligation to be considered violated — upon the lack of action a, or at the end of two consecutive actions?
- It will depend on whether we model the above with CTDs or “unless”, and what is the formal semantics

Breaking an obligation

- Who is right?
- Formalizing the primary obligation in the law, we get $\mathbb{O}(a.b)$, where a represents handling Form A on Monday and b handling Form B on Tuesday
- When is the obligation to be considered violated — upon the lack of action a, or at the end of two consecutive actions?
- It will depend on whether we model the above with CTDs or “unless”, and what is the formal semantics

CTDs and sequences of actions

- Let us consider CTD$(a, \mathbb{O}(b))$
- Does this correspond to an obligation to do a, which *if violated*, will then set up an obligation to perform a b? Or,
- Can the b be performed immediately to satisfy the contract
- In other words, does the sequence of actions $(\overline{a}\&\overline{b}).b$ satisfy the contract? What about $\overline{a}\&b.\overline{a}$?

CTDs and sequences of actions

- Let us consider CTD$(a, \mathbb{O}(b))$
- Does this correspond to an obligation to do a, which *if violated*, will then set up an obligation to perform a b? Or,
- Can the b be performed immediately to satisfy the contract
- In other words, does the sequence of actions $(\bar{a}\&\bar{b}).b$ satisfy the contract? What about $\bar{a}\&b.\bar{a}$?

CTDs and sequences of actions

- Let us consider CTD$(a, \mathbb{O}(b))$
- Does this correspond to an obligation to do a, which *if violated*, will then set up an obligation to perform a b? Or,
- Can the b be performed immediately to satisfy the contract
- In other words, does the sequence of actions $(\bar{a}\&\bar{b}).b$ satisfy the contract? What about $\bar{a}\&b.\bar{a}$?

CTDs and sequences of actions

- Let us consider $\mathrm{CTD}(a, \mathbb{O}(b))$
- Does this correspond to an obligation to do a, which *if violated*, will then set up an obligation to perform a b? Or,
- Can the b be performed immediately to satisfy the contract
- In other words, does the sequence of actions $(\bar{a}\&\bar{b}).b$ satisfy the contract? What about $\bar{a}\&b.\bar{a}$?

Choice of obligations vs obligations of choices

- $\mathbb{O}(a+b)$ –Two possible interpretations:
  - angelic vs demonic (internal vs external) choice
- Similarly for $\mathbb{O}(a) + \mathbb{O}(b)$

Choice of obligations vs obligations of choices

- $\mathbb{O}(a + b)$ –Two possible interpretations:
  - angelic vs demonic (internal vs external) choice
- Similarly for $\mathbb{O}(a) + \mathbb{O}(b)$

Example (Contract between Peter and John)

Contract 1: 'On the 1st of May, John will either (i) be obliged to sell 100 shares at \$1 each; or (ii) be obliged to sell 50 shares at the market price.'

Contract 2: 'On the 1st of May, John will be obliged either (i) to sell 100 shares at \$1 each; or (ii) to sell 50 shares at the market price.'

Choice of obligations vs obligations of choices

- $\mathbb{O}(a + b)$ –Two possible interpretations:
  - angelic vs demonic (internal vs external) choice
- Similarly for $\mathbb{O}(a) + \mathbb{O}(b)$

Example (Contract between Peter and John)

Contract 1: 'On the 1st of May, John will either (i) be obliged to sell 100 shares at $1 each; or (ii) be obliged to sell 50 shares at the market price.'

Contract 2: 'On the 1st of May, John will be obliged either (i) to sell 100 shares at $1 each; or (ii) to sell 50 shares at the market price.'

- While in contract 1 the choice of which obligation to enact lies with Peter, in the latter one obligation is enacted, and it is up to John to decide how to discharge it
- Peter should prefer the first contract, whereas John should prefer the second

Choice of obligations vs obligations of choices

- Contrast $\mathbb{O}(a + b)$ with $\mathbb{F}(a + b)$
- Here the 'internal' and 'external' choices are inverted
  - It seems like the choice inside a forbidden operator becomes an internal choice, not an external one
- Possible interpretations
  - $\mathbb{F}(a + b)$ to be $(\mathbb{F}(a) \wedge \neg\mathbb{P}(b)) + (\mathbb{F}(b) \wedge \neg\mathbb{P}(a))$ ('if you are forbidden to do one, you are not forbidden to do the other')
  - $\mathbb{F}(a + b)$ to be defined as $\neg\mathbb{P}(a) \wedge \neg\mathbb{P}(b)$ ('both actions are forbidden')

Choice of obligations vs obligations of choices

- Contrast $\mathbb{O}(a+b)$ with $\mathbb{F}(a+b)$
- Here the 'internal' and 'external' choices are inverted
  - It seems like the choice inside a forbidden operator becomes an internal choice, not an external one
- Possible interpretations
  - $\mathbb{F}(a+b)$ to be $(\mathbb{F}(a) \wedge \neg\mathbb{P}(b)) + (\mathbb{F}(b) \wedge \neg\mathbb{P}(a))$ ('if you are forbidden to do one, you are not forbidden to do the other')
  - $\mathbb{F}(a+b)$ to be defined as $\neg\mathbb{P}(a) \wedge \neg\mathbb{P}(b)$ ('both actions are forbidden')

Choice of obligations vs obligations of choices

- Contrast $\mathbb{O}(a+b)$ with $\mathbb{F}(a+b)$
- Here the 'internal' and 'external' choices are inverted
  - It seems like the choice inside a forbidden operator becomes an internal choice, not an external one
- Possible interpretations
  - $\mathbb{F}(a+b)$ to be $(\mathbb{F}(a) \wedge \neg\mathbb{P}(b)) + (\mathbb{F}(b) \wedge \neg\mathbb{P}(a))$ ('if you are forbidden to do one, you are not forbidden to do the other')
  - $\mathbb{F}(a+b)$ to be defined as $\neg\mathbb{P}(a) \wedge \neg\mathbb{P}(b)$ ('both actions are forbidden')

The moment of choice and the moment of contract satisfaction

- An important issue is *when* the choice is made
- Is $\text{Any?}.\mathbb{O}(a) + \text{Any?}.\mathbb{O}(b)$ equal to $\text{Any?}.(\mathbb{O}(a) + \mathbb{O}(b))$?
    - Choice may be immediate or delayed
- What about $\mathbb{O}(a + a.c).\mathbb{O}(d)$?
    - After an a, we don't know whether the first contract has been satisfied –It depends on whether we get a $c.d$, or a d
- Similarly for $\mathbb{O}(a + a.c).\mathbb{O}(c)$
    - it is non-deterministic whether the action sequence $a.c.c$ satisfied the contract after the first two, or three symbols

The moment of choice and the moment of contract satisfaction

- An important issue is *when* the choice is made
- Is $\text{Any?}.\mathbb{O}(a) + \text{Any?}.\mathbb{O}(b)$ equal to $\text{Any?}.(\mathbb{O}(a) + \mathbb{O}(b))$?
  - Choice may be immediate or delayed
- What about $\mathbb{O}(a + a.c).\mathbb{O}(d)$?
  - After an a, we don't know whether the first contract has been satisfied –It depends on whether we get a $c.d$, or a d
- Similarly for $\mathbb{O}(a + a.c).\mathbb{O}(c)$
  - it is non-deterministic whether the action sequence $a.c.c$ satisfied the contract after the first two, or three symbols

The moment of choice and the moment of contract satisfaction

- An important issue is *when* the choice is made
- Is Any?.$\mathbb{O}(a)$ + Any?.$\mathbb{O}(b)$ equal to Any?.$(\mathbb{O}(a) + \mathbb{O}(b))$?
  - Choice may be immediate or delayed
- What about $\mathbb{O}(a + a.c).\mathbb{O}(d)$?
  - After an a, we don't know whether the first contract has been satisfied –It depends on whether we get a $c.d$, or a d
- Similarly for $\mathbb{O}(a + a.c).\mathbb{O}(c)$
  - it is non-deterministic whether the action sequence $a.c.c$ satisfied the contract after the first two, or three symbols

The moment of choice and the moment of contract satisfaction

- An important issue is *when* the choice is made
- Is $\text{Any?}.\mathbb{O}(a) + \text{Any?}.\mathbb{O}(b)$ equal to $\text{Any?}.(\mathbb{O}(a) + \mathbb{O}(b))$?
  - Choice may be immediate or delayed
- What about $\mathbb{O}(a + a.c).\mathbb{O}(d)$?
  - After an a, we don't know whether the first contract has been satisfied –It depends on whether we get a $c.d$, or a d
- Similarly for $\mathbb{O}(a + a.c).\mathbb{O}(c)$
  - it is non-deterministic whether the action sequence $a.c.c$ satisfied the contract after the first two, or three symbols

Choice and CTDs

- CTD(a, b) + CTD(c, d) may be broken if one performs an a but no c (and no d to compensate)
  - It depends on the interpretation on whether we first choose and then apply the CTD (**xor**) or if both CTDs are enforced before choosing
- What about CTD$(a, \mathbb{O}(b))$ + CTD$(b, \mathbb{O}(a))$
  - With a **xor** interpretation: No way to satisfy the contract!
  - If non-determinism is allowed, interpreting + as a choice is also problematic

Choice and CTDs

- $\mathrm{CTD}(a, b) + \mathrm{CTD}(c, d)$ may be broken if one performs an a but no c (and no d to compensate)
  - It depends on the interpretation on whether we first choose and then apply the CTD (**xor**) or if both CTDs are enforced before choosing
- What about $\mathrm{CTD}(a, \mathbb{O}(b)) + \mathrm{CTD}(b, \mathbb{O}(a))$
  - With a **xor** interpretation: No way to satisfy the contract!
  - If non-determinism is allowed, interpreting $+$ as a choice is also problematic

Contract of repetitions and repetition of contracts

- Are $\mathbb{O}(a^*)$ and $\mathbb{O}(a)^*$ equivalent?
- $\mathbb{O}(a^*)$ is intuitively equivalent to $\mathbb{O}(\varepsilon + a + a.a + a.a.a + \dots)$
  - A number of actions a are to be performed — the choice regarding the number of repetitions is external (decided by the entity bound by the contract)
- $\mathbb{O}(a)^*$ is intuitively equivalent to $\mathbb{Y} + \mathbb{O}(a) + \mathbb{O}(a).\mathbb{O}(a) + \mathbb{O}(a).\mathbb{O}(a).\mathbb{O}(a) + \dots$
  - The choice regarding the number of repetitions is internal, and thus imposed

Contract of repetitions and repetition of contracts

- Are $\mathbb{O}(a^*)$ and $\mathbb{O}(a)^*$ equivalent?
- $\mathbb{O}(a^*)$ is intuitively equivalent to $\mathbb{O}(\varepsilon + a + a.a + a.a.a + \dots)$
  - A number of actions a are to be performed — the choice regarding the number of repetitions is external (decided by the entity bound by the contract)
- $\mathbb{O}(a)^*$ is intuitively equivalent to $\mathbb{Y} + \mathbb{O}(a) + \mathbb{O}(a).\mathbb{O}(a) + \mathbb{O}(a).\mathbb{O}(a).\mathbb{O}(a) + \dots$
  - The choice regarding the number of repetitions is internal, and thus imposed

Contract of repetitions and repetition of contracts

- Are $\mathbb{O}(a^*)$ and $\mathbb{O}(a)^*$ equivalent?
- $\mathbb{O}(a^*)$ is intuitively equivalent to $\mathbb{O}(\varepsilon + a + a.a + a.a.a + \dots)$
  - A number of actions a are to be performed — the choice regarding the number of repetitions is external (decided by the entity bound by the contract)
- $\mathbb{O}(a)^*$ is intuitively equivalent to $\mathbb{Y} + \mathbb{O}(a) + \mathbb{O}(a).\mathbb{O}(a) + \mathbb{O}(a).\mathbb{O}(a).\mathbb{O}(a) + \dots$
  - The choice regarding the number of repetitions is internal, and thus imposed

Unbounded repetition

- Example: 'If John uses the service, then he is bound to eventually pay'
  - Written as $s?.\mathbb{O}(\text{Any}^*.p)$
- Problems:
  - No bound is placed on how long John takes to pay his dues
  - A formal semantics of the logic over infinite sequences enables to decide whether or not John has satisfied the contract
  - Looking at finite sequences, one requires the use of a three-valued logic to differentiate between the contract being violated, satisfied, and the third situation when it may still be satisfied in the future
- In practice, it seems more natural to have only bounded iteration

Unbounded repetition

- Example: ‘If John uses the service, then he is bound to eventually pay’
  - Written as $s?.\mathbb{O}(\mathrm{Any}^{*}.p)$
- Problems:
  - No bound is placed on how long John takes to pay his dues
  - A formal semantics of the logic over infinite sequences enables to decide whether or not John has satisfied the contract
  - Looking at finite sequences, one requires the use of a three-valued logic to differentiate between the contract being violated, satisfied, and the third situation when it may still be satisfied in the future
- In practice, it seems more natural to have only bounded iteration

Unbounded repetition

- Example: ‘If John uses the service, then he is bound to eventually pay’
  - Written as $s?.\mathbb{O}(\mathrm{Any}^*.p)$
- Problems:
  - No bound is placed on how long John takes to pay his dues
  - A formal semantics of the logic over infinite sequences enables to decide whether or not John has satisfied the contract
  - Looking at finite sequences, one requires the use of a three-valued logic to differentiate between the contract being violated, satisfied, and the third situation when it may still be satisfied in the future
- In practice, it seems more natural to have only bounded iteration

Real-time aspects

- Most contracts include some timing aspects
  - Deadlines, timeouts, durations, etc.
- Challenges
  - Should we associate time with the modalities, clauses, actions, or with all of them?
  - Is an interval-based necessary to reason about the beginning and end of an action?
  - Would the semantics be given by enriched timed automata with deontic notions or labelled Kripke structure enriched with time?
- A good solution would be to use *clocks* with freezing quantifiers and resets

Real-time aspects

- Most contracts include some timing aspects
  - Deadlines, timeouts, durations, etc.
- Challenges
  - Should we associate time with the modalities, clauses, actions, or with all of them?
  - Is an interval-based necessary to reason about the beginning and end of an action?
  - Would the semantics be given by enriched timed automata with deontic notions or labelled Kripke structure enriched with time?
- A good solution would be to use *clocks* with freezing quantifiers and resets

Real-time aspects

- Most contracts include some timing aspects
  - Deadlines, timeouts, durations, etc.
- Challenges
  - Should we associate time with the modalities, clauses, actions, or with all of them?
  - Is an interval-based necessary to reason about the beginning and end of an action?
  - Would the semantics be given by enriched timed automata with deontic notions or labelled Kripke structure enriched with time?
- A good solution would be to use *clocks* with freezing quantifiers and resets

Reference to other expressions

- How to analyze cross-references (intra- and inter-contract)?
- A *nominal* logic or simply annotations on clauses and contracts may be needed to be able to refer to other clauses
- The analysis of cross-references could be analyzed with standard existing techniques on graph

Reference to other expressions

- How to analyze cross-references (intra- and inter-contract)?
- A *nominal* logic or simply annotations on clauses and contracts may be needed to be able to refer to other clauses
- The analysis of cross-references could be analyzed with standard existing techniques on graph

Reference to other expressions

- How to analyze cross-references (intra- and inter-contract)?
- A *nominal* logic or simply annotations on clauses and contracts may be needed to be able to refer to other clauses
- The analysis of cross-references could be analyzed with standard existing techniques on graph

Introspection - reflection

- Contract introspection: The capability of having conditions which depend on which obligations, permissions and prohibitions are active
  - Ex: 'Whenever you are obliged to pay, you are also obliged to produce identification'
- A contract may contain references to itself, i.e. be *reflexive*
  - Ex: A clause may state that a party may has the *power* to change other clauses, or even to cancel the contract

Introspection - reflection

- Contract introspection: The capability of having conditions which depend on which obligations, permissions and prohibitions are active
  - Ex: 'Whenever you are obliged to pay, you are also obliged to produce identification'
- A contract may contain references to itself, i.e. be *reflexive*
  - Ex: A clause may state that a party may has the *power* to change other clauses, or even to cancel the contract

Fairness and invariants

- Invariants
  - An obligation which is always enabled
  - Always being forbidden to do something
- Fairness
  - Ex: “any infinitely often enabled process should be infinitely often taken”
  - More realistic: Bounded fairness

Fairness and invariants

- Invariants
    - An obligation which is always enabled
    - Always being forbidden to do something
- Fairness
    - Ex: "any infinitely often enabled process should be infinitely often taken"
    - More realistic: Bounded fairness

Concurrency

- True concurrency seems natural in many contracts
  - 'You are obliged to sit-down and remain silent': $\mathbb{O}(s\&r)$
- How to handle violations?
  - Not doing any (nor both) actions.
- Other problems:
  - Does $\mathbb{O}(s) \wedge \mathbb{O}(r)$ entails $\mathbb{O}(s\&r)$, or are they equivalent?
  - Conjunction is interpreted as branching in Dynamic logic

Concurrency

- True concurrency seems natural in many contracts
  - 'You are obliged to sit-down and remain silent': $\mathbb{O}(s\&r)$
- How to handle violations?
  - Not doing any (nor both) actions.
- Other problems:
  - Does $\mathbb{O}(s) \wedge \mathbb{O}(r)$ entails $\mathbb{O}(s\&r)$, or are they equivalent?
  - Conjunction is interpreted as branching in Dynamic logic

Concurrency

- True concurrency seems natural in many contracts
  - 'You are obliged to sit-down and remain silent': $\mathbb{O}(s\&r)$
- How to handle violations?
  - Not doing any (nor both) actions.
- Other problems:
  - Does $\mathbb{O}(s) \wedge \mathbb{O}(r)$ entails $\mathbb{O}(s\&r)$, or are they equivalent?
  - Conjunction is interpreted as branching in Dynamic logic

Conditional contracts

- ‘Unless the service is disabled, John is obliged to pay in the next time unit’ versus ‘John is obliged to pay in the next time unit, unless the service is disabled now.’
- How to formalize it?
  - $\bar{d}?.\mathbb{O}(p)$, $\mathbb{O}(\bar{d}.p + d)$ (or even $\text{Any}.\mathbb{O}(p)$ *unless* $d?$)
- Problems
  - Subjects, objects and actors:
    - Automatically, actions which are not performed by the party being obliged to do something, are conditions
    - What happens when the condition includes actions under the control of the party
  - How to deal with implicit *otherwise* cases
    - We may introduce a conditional operator: $\mathbb{O}(\varepsilon \lhd d \rhd p)$
  - Does the condition take time?
    - What is the meaning of $\bar{d}?.d$ (if d is not present in the current time unit, then ensure it is)

Conditional contracts

- 'Unless the service is disabled, John is obliged to pay in the next time unit' versus 'John is obliged to pay in the next time unit, unless the service is disabled now.'
- How to formalize it?
  - $\bar{d}?.\mathbb{O}(p)$, $\mathbb{O}(\bar{d}.p + d)$ (or even $\mathrm{Any}.\mathbb{O}(p)$ *unless* d?)
- Problems
  - Subjects, objects and actors:
    - Automatically, actions which are not performed by the party being obliged to do something, are conditions
    - What happens when the condition includes actions under the control of the party
  - How to deal with implicit *otherwise* cases
    - We may introduce a conditional operator: $\mathbb{O}(\varepsilon \triangleleft d \triangleright p)$
  - Does the condition take time?
    - What is the meaning of $\bar{d}?.d$ (if d is not present in the current time unit, then ensure it is)

Conditional contracts

- ‘Unless the service is disabled, John is obliged to pay in the next time unit’ versus ‘John is obliged to pay in the next time unit, unless the service is disabled now.’
- How to formalize it?
  - $\bar{d}?.\mathbb{O}(p)$, $\mathbb{O}(\bar{d}.p + d)$ (or even $\text{Any}.\mathbb{O}(p)$ *unless* d?)
- Problems
  - Subjects, objects and actors:
    - Automatically, actions which are not performed by the party being obliged to do something, are conditions
    - What happens when the condition includes actions under the control of the party
  - How to deal with implicit *otherwise* cases
    - We may introduce a conditional operator: $\mathbb{O}(\varepsilon \triangleleft d \triangleright p)$
  - Does the condition take time?
    - What is the meaning of $\bar{d}?.d$ (if d is not present in the current time unit, then ensure it is)

Final Remarks

Not Easy!!

- Clearly good syntax is not enough
- Defining the semantics is challenging!
- Very important to set the application domain and give the intended semantics
    - Crucial to prove the language/logic preserves the desired properties
- A lot of research to do to obtain a clean,useful contract language!

Final Remarks

Not Easy!!

- Clearly good syntax is not enough
- Defining the semantics is challenging!
- Very important to set the application domain and give the intended semantics
  - Crucial to prove the language/logic preserves the desired properties
- A lot of research to do to obtain a clean,useful contract language!

Next lecture

- We will see the contract language $\mathcal{CL}$
- It does not solve all the problems pointed out in this lecture, but advances the state of the art...

Further Reading

- G. Pace, and G. Schneider. **Challenges in the specification of full contracts.** Accepted at iFM'09. To appear in LNCS.