Temporal Dynamic Description Logic

Liang Chang

 changl.guet@gmail.com
 liang.chang@manchester.ac.uk
Temporal description logic (TDL)

- For capturing temporal aspects of concepts in ontologies.
  \[ \neg \text{Doctor} \sqcap \Diamond \text{Doctor} \sqsubseteq \Diamond (\text{PHDStudent} \sqcap \neg \text{Doctor} \sqcap (\text{PHDStudent} \cup \text{Doctor})) \]

\[ \text{PHDStudent} \sqsubseteq \exists \text{hasSup} . \text{Doctor} \]
Temporal description logic (TDL)

- For capturing temporal aspects of concepts in ontologies.

\[ \neg \text{Doctor} \sqcap \Diamond \text{Doctor} \sqsubseteq \Diamond (\text{PHDStudent} \sqcap \neg \text{Doctor} \sqcap (\text{PHDStudent} \cup \text{Doctor})) \]

\[ \text{PHDStudent} \sqsubseteq \exists \text{hasSup}.\text{Doctor} \]

- Two-dimensional logics [GKWZ03]
  - Temporal description logic
  - Dynamic description logic
  - …
Temporal description logic (TDL)

- For capturing temporal aspects of concepts in ontologies.

$$\neg \text{Doctor} \sqcap \Diamond \text{Doctor} \sqsubseteq \Diamond (\text{PHDStudent} \sqcap \neg \text{Doctor} \sqcap (\text{PHDStudent} \cup \text{Doctor}))$$

$$\text{PHDStudent} \sqsubseteq \exists \text{hasSup}.\text{Doctor}$$

- Two-dimensional logics [GKWZ03]
  - Temporal description logic
  - Dynamic description logic

\[ \text{joint?} \]
Different temporal extensions of DLs

• Explicit notion of time or implicit time

• Interval-based notion of time or point-based time
  – External representation of time or internal representation

• Linear time or branching time
Different temporal extensions

- Varying DL component: DL-Lite, EL, ALC, SHOIQ, …

- Different choice for applying temporal operators: concepts, TBox axioms, ABox assertions
  - $\neg$Doctor $\sqcap$ $\Diamond$Doctor $\sqsubseteq$ $\Diamond$(PHDStudent $\cup$ Doctor)
  - $\Diamond$□(Citizen $\sqsubseteq$ HASVote)
  - PHDStudent(Jack) $\wedge$ $\Diamond$(PHDStudent(Jack) $\cup$ Doctor(Jack))

- Additional constraints on concepts and roles: rigid concepts, rigid roles

- Interpretation domains: expanding, constant
Different temporal extensions

- **Explicit** notion of time or **implicit** time
- **Interval-based** notion of time or **point-based** time
  - External representation of time or internal representation
- **Linear** time or **branching** time

---

Dozens of combinations!

- Varying DL component: DL-Lite, EL, ALC, SHOIQ, ...
- Different choice for applying temporal operators: concepts, TBox axioms, ABox assertions
- Additional constraints on concepts and roles: rigid concepts, rigid roles
- interpretation domains: expanding, constant
- …..
Different temporal extensions

- **Explicit** notion of time or **implicit** time
- **Interval-based** notion of time or **point-based time**
  - External representation of time or internal representation
- **Linear time** or **branching time**

- Different choice for applying temporal operators: concepts, TBox axioms, ABox assertions
- Additional constraints on concepts and roles: rigid concepts, rigid roles
- Interpretation domains: expanding, constant
- ……
Reasoning about actions

- Representation and Reasoning about Actions

- Situation Calculus  [Mcc63]

- John McCarthy
  - father of AI, 1956
  - Winner of Turing Award, 1971

*John McCarthy (1927-2011)*
Action Formalisms

Based on propositional logics:
- Based on PDL \([\text{GL95}]\)
- Based on LTL \([\text{CGV02}]\)

Based on DL?

Based on first- or higher-order logics:
- Situation Calculus
- Fluent Calculus
- .......

Gap?
# Action Formalisms

<table>
<thead>
<tr>
<th>Based on propositional logics</th>
<th>Based on DL ?</th>
<th>Gap ?</th>
<th>Action formalism based on DL</th>
</tr>
</thead>
<tbody>
<tr>
<td>based on PDL</td>
<td></td>
<td></td>
<td>based on DL [GL95]</td>
</tr>
<tr>
<td>based on LTL</td>
<td></td>
<td></td>
<td>based on LTL [CGV02]</td>
</tr>
<tr>
<td>Based on first- or higher-order logics</td>
<td>Situation Calculus</td>
<td></td>
<td>Situation Calculus</td>
</tr>
<tr>
<td></td>
<td>Fluent Calculus</td>
<td></td>
<td>Fluent Calculus</td>
</tr>
<tr>
<td></td>
<td>........</td>
<td></td>
<td>........</td>
</tr>
</tbody>
</table>

[BLM’05]
DL-Based Action Formalisms

- Background knowledge: RBox, TBox
- States: ABoxes
- Action: $\alpha = (\text{pre}, \text{occ}, \text{post})$
  - $\text{pre}$: ABox assertions
  - $\text{occ}$: primitive literals
  - $\text{post}$: set of conditional post-conditions, $\varphi/\psi$

- Update ABox after the execution of actions.
Extension of the DL-based action formalism

Basic idea: construct more powerful formalism, action theory + description logic + dynamic logic

• Background knowledge: RBox, TBox
• Atomic actions: come from Baader et al.’s formalism
  \[ \alpha \equiv (\text{pre}, \text{occ}, \text{post}) \]
• Complex actions:
  \[ \pi, \pi' ::= \alpha | \phi? | \pi \cup \pi' | \pi ; \pi | \pi^* \]
• Formulas:
  \[ \phi, \psi ::= C(p) | R(p,q) | <\pi>\phi | [\pi]\phi | \neg \phi | \phi \lor \psi | \phi \land \psi \]

• Dynamic description logic DDL(X@)
  - X: DLs ranging from ALCO to ALC\text{H}OIQ
  - X@: extension of X with the @ constructor.
(1) **Complex actions can be constructed**

- **TBox:**
  
  \[
  \text{Customer} \equiv \text{Person} \sqcap \exists \text{holds.CreditCard} \\
  \text{VIPcustomer} \equiv \text{Customer} \sqcap \geq 10 \text{ boughr.}(\text{Book} \sqcup \text{CD})
  \]

- **Atomic Actions:**
  
  \[
  \text{buybook}(a,b) \equiv ( \{\text{Customer}(a), \text{Book}(b)\}, \emptyset; \\
  \{\neg \text{Instore}(b)/\text{Instore}(b), \text{Instore}(b)/\text{bought}(a,b)\} )
  \]
  
  \[
  \text{order}(b) \equiv ( \{\text{((Book} \sqcup \text{CD})(b))\}, \emptyset; \\
  \{\neg \text{Instore}(b)/\text{Instore}(b)\} )
  \]

- **Complex Action:**
  
  \[
  \text{VIPbuybook}(a,b) \equiv \text{VIPcustomer}(a)?; \\
  ( (\text{Instore}(b)?; \text{buybook}(a,b)) \cup \\
  (\neg \text{Instore}(b)?; \text{order}(b); \text{buybook}(a,b)) )
  \]
(2) Properties on (complex) actions can be described directly

- **necessary conditions** for the execution of (complex) actions
  - `<VIPbuybook(a,b)>true → (VIPcustomer(a)∧Book(b))`
  - `<VIPbuybook(a,b)>true → Instore(b)`

- **results** on the execution of actions
  - `[VIPbuybook(a,b)]bought(a,b)`
  - `[buybook(a,b)]bought(a,b)`
Features of DDL(\(X\@\)) (3/3)

(3) Reasoning problems on actions be reduced to the satisfiability problem of formulas

- Executability of actions
- Projection problem
- Consistency/realizability of actions
  - whether a given action makes sense w.r.t. the knowledge base
    buybook\((a1,b)\); buybook\((a2,b)\)

- Satisfiability problem
  - a Tableau decision algorithm is provided.
  - the complexity upper-bound is
    - EXPSpace if \(X\in\{ALCO, ALCHO, ALCOQ, ALCHOQ\}\),
    - N2EXPTime if \(X\in\{ALCOI, ALCHOI, ALCOIQ, ALCHOIQ\}\).
Temporal extension of DDL($X@$)

To investigate temporal properties of actions.

**Approach:**
- the ongoing of time is embodied as the execution of atomic actions (time units)
- two temporal assertions are introduced:

\[
\phi, \psi ::= C(p) \mid R(p,q) \mid <\pi>\phi \mid [\pi]\phi \mid \neg\phi \mid \phi \lor \psi \mid \mathbf{E}(\phi \mathbf{U}^\pi \psi) \mid \mathbf{A}(\phi \mathbf{U}^\pi \psi)
\]

\(\mathbf{E}(\phi \mathbf{U}^\pi \psi)\): there exists some path of \(\pi\) such that “\(\phi\) until \(\psi\)” holds.

\(\mathbf{A}(\phi \mathbf{U}^\pi \psi)\): “\(\phi\) until \(\psi\)” holds in any path of \(\pi\) .
Temporal extension of DDL(X@)

To investigate temporal properties of actions.

Approach:
– the ongoing of time is embodied as the execution of atomic actions (time units)
– two temporal assertions are introduced:
\[ \phi, \psi ::= C(p) \mid R(p,q) \mid <\pi>\phi \mid [\pi]\phi \mid \neg \phi \mid \phi \lor \psi \mid E(\phi U^n \psi) \mid A(\phi U^n \psi) \]

\[ E(\phi U^n \psi) : \text{there exists some path of } \pi \text{ such that } "\phi \text{ until } \psi" \text{ holds.} \]
\[ A(\phi U^n \psi) : "\phi \text{ until } \psi" \text{ holds in any path of } \pi . \]

\[ \text{EX } \phi = \text{def } \lor \alpha \in N_A <\alpha>\phi \]
\[ E(\phi U \psi) = \text{def } E(\phi U^{a_1 \cup \ldots \cup a_n} \psi) \]
\[ A(\phi U \psi) = \text{def } A(\phi U^{a_1 \cup \ldots \cup a_n} \psi) \]
Temporal extension of DDL(X@)

To investigate temporal properties of actions.

Approach:
- the ongoing of time is embodied as the execution of atomic actions (time units)
- two temporal assertions are introduced:

\[ \phi, \psi ::= C(p) | R(p,q) | <\pi>\phi | [\pi]\phi | \neg\phi | \phi \vee \psi | E(\phi U^\pi \psi) | A(\phi U^\pi \psi) \]

E(\phi U^\pi \psi) : there exists some path of \( \pi \) such that "\( \phi \) until \( \psi \)" holds.
A(\phi U^\pi \psi) : “\( \phi \) until \( \psi \)” holds in any path of \( \pi \).

\[
\begin{align*}
\text{EX } \phi &= \underset{\text{def}}{\vee} \alpha \in N_a <\alpha>\phi \\
E(\phi U \psi) &= \underset{\text{def}}{E(\phi U^{(a_1 \cup \ldots \cup a_n)^*} \psi)} \\
A(\phi U \psi) &= \underset{\text{def}}{A(\phi U^{(a_1 \cup \ldots \cup a_n)^*} \psi)}
\end{align*}
\]

\[
\begin{align*}
\text{EF } \phi &= \underset{\text{def}}{E(\text{true} U \phi)} \\
\text{AF } \phi &= \underset{\text{def}}{A(\text{true} U \phi)} \\
\text{EG } \phi &= \underset{\text{def}}{\neg A(\text{true} U \phi)} \\
\text{AG } \phi &= \underset{\text{def}}{\neg E(\text{true} U \phi)} \\
\text{AX } \phi &= \underset{\text{def}}{\neg E(\text{true} U \phi)}
\end{align*}
\]
Description example of TDDL(X@)

- **liveness property**: good things will eventually happen.
  \[
  EF((\exists \text{bought}\neg.\text{Customer})(b)) \\
  E(\text{Instore}(b) \ U^{\text{VIPbuybook}(a,b)} \neg \text{Instore}(b))
  \]

- **safety property**: bad things will never happen.
  \[
  AG \neg(\geq 2 \text{bought}\neg.\text{Customer})(b) \\
  AG ( \text{Instore}(b) \lor (\exists \text{bought}\neg.\text{Customer})(b) )
  \]

- Reduced to satisfiability problem of formulas.
- A Tableau decision algorithm is provided.
Limitation of DDL(\(X@\))/TDDL(\(X@\))

- **TBox:**
  - only concept definitions, no GCIs
  - acyclic

- **RBox:**
  - on transitive property

- **Atomic action:**
  - no defined concept name occurring in the effect set \(\text{post}\).

**Why?**
- difficulty of ABox updating.
Difficulty of ABox updating

Example.

• TBox:
  \( \text{Trans}(R), \ A \sqsubseteq \exists R.A, \ A \sqcap B \sqsubseteq \bot, \ B \sqsubseteq \forall R.B \)

• ABox:
  \( A(a) \)

• Update or new information:
  \( (\exists R.B)(a) \)
## Some results on ABox update

<table>
<thead>
<tr>
<th>Assumptions</th>
<th>DLs</th>
<th>Approach</th>
<th>References</th>
</tr>
</thead>
<tbody>
<tr>
<td>Acyclic TBox; no defined concept names occurring in $U$</td>
<td>ALC~ALC QIO</td>
<td>PMA semantics &amp; only primitive concept names are counted when measuring distance.</td>
<td>LLMW06, LLMW11</td>
</tr>
<tr>
<td>DL-Lite$_F$</td>
<td>PMA semantics.</td>
<td></td>
<td>GLPR06, GLPR07</td>
</tr>
<tr>
<td>DL-Lite$_{R}^{pr}$</td>
<td>Both revision and update. Based on $fcl_T(A)$</td>
<td></td>
<td>KZ11, KZC13</td>
</tr>
<tr>
<td>DL-Lite$_{FR}$</td>
<td>Based on $cl_T(A)$.</td>
<td></td>
<td>CKNZ10</td>
</tr>
</tbody>
</table>
Thank you!

[Image of a head with a question mark]