Expressive Goal Specification Languages in Presence of Non-deterministic Actions

By: Chitta Baral and Jicheng Zhao
Dept. of Computer Science and Engineering
Arizona State University, USA

(presented by Pedro Cabalar)
Outline

- Introduction
- LTL and CTL$^*$ overview
- The proposal: $\pi$-CTL$^*$
- Related work
- Conclusions
Outline

☞ Introduction

❑ LTL and CTL* overview

❑ The proposal: $\pi$-CTL*

❑ Related work

❑ Conclusions
Introduction: deterministic domains

- Deterministic domain: plan = sequence of actions that leads to a sequence of states from initial state;
Introduction: deterministic domains

- Deterministic domain: plan = sequence of actions what leads to a sequence of states from initial state;
- Interest: to specify properties of the agent’s plan;
Introduction: deterministic domains

• Deterministic domain: plan = sequence of actions what leads to a sequence of states from initial state;
• Interest: to specify properties of the agent’s plan;
• E.g.: LTL not only specifies properties of the “final state” but also the intermediate states in the sequence;
Introduction: deterministic domains

- Deterministic domain: plan = sequence of actions what leads to a sequence of states from initial state;
- Interest: to specify properties of the agent’s plan;
- E.g.: LTL not only specifies properties of the “final state” but also the intermediate states in the sequence;
Example: “move through states that avoid \( p \)”
Introduction: deterministic domains

- Deterministic domain: plan = sequence of actions what leads to a sequence of states from initial state;
- Interest: to specify properties of the agent’s plan;
- E.g.: LTL not only specifies properties of the “final state” but also the intermediate states in the sequence; Example: “move through states that avoid $p$”
- CTL$^*$ also specifies the properties of the states leading from states in the main path.
Introduction: deterministic domains

- Deterministic domain: plan = sequence of actions what leads to a sequence of states from initial state;
- Interest: to specify properties of the agent’s plan;
- E.g.: LTL not only specifies properties of the “final state” but also the intermediate states in the sequence;
  Example: “move through states that avoid $p$”
- CTL* also specifies the properties of the states leading from states in the main path. Example: “move through states from which you could reach $p$ (with another action)”
Introduction: non-deterministic domains

• Now we cannot associate a sequence of actions to a sequence of states. The action sequence either results in multiple sequences, or is not well defined;
**Introduction: non-deterministic domains**

- Now we cannot associate a sequence of actions to a sequence of states. The action sequence either results in multiple sequences, or is not well defined;

- Instead, we adopt the definition of a policy $\pi$ to be a mapping from each state to an action. Example:

$$\pi_1 = \{(s_1, a_1), (s_2, a_2), (s_3, a_3), (s_4, nop)\}$$
Introduction: non-deterministic domains

- Now we cannot associate a sequence of actions to a sequence of states. The action sequence either results in multiple sequences, or is not well defined;

- Instead, we adopt the definition of a policy $\pi$ to be a mapping from each state to an action. Example:

  $\pi_1 = \{(s_1, a_1), (s_2, a_2), (s_3, a_3), (s_4, nop)\}$

- Under this setting, specifying goals has new challenges: [Dal Lago, Pistore & Traverso 02] talk about “extended goals.”
An example

“Try your best to reach $p$”
An example

Policy $\pi_1$
An example

Policy $\pi_2$ clearly worse than $\pi_1$!
An example

Policy $\pi_3$

worse than $\pi_1$! but $\pi_2$?
An example

Policy $\pi_4$ worse than $\pi_2$ and $\pi_3$
An example

Policy $\pi_5$

worse than $\pi_3$
An example

Policy $\pi_6 \text{ worse than } \pi_4, \pi_5$
An example

Policy $\pi_7$
Really bad!
Outline

✔ Introduction

❑ LTL and CTL* overview

❑ The proposal: $\pi$-CTL*

❑ Related work

❑ Conclusions
Outline

✔ Introduction

☞ LTL and CTL* overview

❑ The proposal: $\pi$-CTL*

❑ Related work

❑ Conclusions
Linear Temporal Logic LTL

- Linear time: sequence of states
- Operators:
  \( \Box p = \) always \( p \)
  \( \Diamond p = \) eventually \( p \)
  \( \bigcirc p = \) next \( p \)
  \( p \cup q = p \) true until \( q \)
Branching Temporal logic CTL*  

- Branching time  
- New operators for paths
Branching Temporal logic CTL

- Branching time
- New operators for paths
Branching Temporal logic CTL*

- Branching time
- New operators for paths
Branching Temporal logic CTL*

- Branching time
- New operators for paths
Branching Temporal logic CTL

- Branching time
- New operators for paths
Branching Temporal logic CTL*

\[ A\phi = \text{for any path, } \phi \text{ holds} \]
\[ E\phi = \text{for some path, } \phi \text{ holds} \]
Branching Temporal logic CTL

Examples:

\(A\Diamond p\) = all paths reach \(p\)
\(E\Box p\) = in some path, always \(p\)
Branching Temporal logic CTL*  

Syntax:
\[ \langle p \rangle = \text{propositional formula}; \]
\[ \langle sf \rangle = \text{“state” formula}; \]
\[ \langle pf \rangle = \text{“path” formula} \]
Branching Temporal logic CTL*  

Syntax:
\[ \langle p \rangle = \text{propositional formula}; \]
\[ \langle sf \rangle = \text{“state” formula}; \]
\[ \langle pf \rangle = \text{“path” formula} \]

\[ \langle sf \rangle ::= \langle p \rangle | \langle sf \rangle \land \langle sf \rangle | \langle sf \rangle \lor \langle sf \rangle | \neg \langle sf \rangle | E \langle pf \rangle | A \langle pf \rangle \]
Branching Temporal logic CTL*

Syntax:

$\langle p \rangle = \text{propositional formula};$

$\langle sf \rangle = \text{“state” formula};$

$\langle pf \rangle = \text{“path” formula}$

\[
\langle sf \rangle ::= \langle p \rangle \mid \langle sf \rangle \land \langle sf \rangle \mid \langle sf \rangle \lor \langle sf \rangle \mid \neg \langle sf \rangle \mid \text{E} \langle pf \rangle \mid \text{A} \langle pf \rangle
\]

\[
\langle pf \rangle ::= \langle sf \rangle \mid \langle pf \rangle \lor \langle pf \rangle \mid \neg \langle pf \rangle \mid \langle pf \rangle \land \langle pf \rangle \mid \langle pf \rangle \text{ U } \langle pf \rangle \mid \text{ O } \langle pf \rangle \mid \text{ ♦ } \langle pf \rangle \mid \text{ □ } \langle pf \rangle
\]
Branching Temporal logic CTL*  

Semantics (state formulas). We use a pair \((s_j, R)\) with \(s_j\) a state and \(R\) the transition relation:

- \((s_j, R) \models p\) iff \(p\) is true in \(s_j\).

- \(\neg, \land, \lor\) as usual.

- \((s_j, R) \models E pf\) iff there exists a path \(\sigma\) in \(R\) starting from \(s_j\) such that \((s_j, R, \sigma) \models pf\).

- \((s_j, R) \models A pf\) iff for all paths \(\sigma\) in \(R\) starting from \(s_j\) we have that \((s_j, R, \sigma) \models pf\).
Branching Temporal logic CTL*  

Semantics (path formulas). We use triplet \((s, R, \sigma)\) where \(\sigma\) given by the sequence of states \(s_0, s_1, \ldots\), is a path.

- \((s_j, R, \sigma) \models s f \text{ iff } (s_j, R) \models s f.\)
- \(\neg, \land, \lor\) as usual.
- \((s_j, R, \sigma) \models \bigcirc f \text{ iff } (s_{j+1}, R, \sigma) \models f\)
- \((s_j, R, \sigma) \models \Box f \text{ iff } (s_k, R, \sigma) \models f, \text{ for all } k \geq j.\)
- \((s_j, R, \sigma) \models \Diamond f \text{ iff } (s_k, R, \sigma) \models f, \text{ for some } k \geq j.\)
- \((s_j, R, \sigma) \models f_1 \lor f_2 \text{ iff there exists } k \geq j \text{ such that } (s_k, R, \sigma) \models f_2 \text{ and for all } i, j \leq i < k, (s_i, R, \sigma) \models f_1.\)
Branching Temporal logic CTL*

- A sequence of actions \( a_1, \ldots, a_n \) is a **plan** with respect to the initial state \( s_0 \) and a CTL* goal \( G \) if 
  \[ (s_0, R, \sigma) \models G, \]
  where \( \sigma \) is the trajectory corresponding to \( s_0 \) and \( a_1, \ldots, a_n \).
Outline

✓ Introduction

✓ LTL and CTL* overview

❑ The proposal: $\pi$-CTL*

❑ Related work

❑ Conclusions
Outline

✔ Introduction

✔ LTL and CTL* overview

☞ The proposal: $\pi$-CTL*

❑ Related work

❑ Conclusions
When using CTL* for goal specification . . .

- Goals $E \Diamond p$ and $A \Diamond p$ in $s_1$: policies are not distinguishable in them!
When using CTL* for goal specification . . .

- Goals $E \diamond p$ and $A \diamond p$ in $s_1$: policies are not distinguishable in them!
- Besides $A$ and $E$ in capturing all paths in the plan, we may also need to consider those paths by following the policy.
The extension of CTL*: \( \pi\text{-CTL}^* \)

- One natural extension of CTL* is to group the set of paths from the initial state that all correspond to the same policy:
The extension of CTL*: π-CTL*

- One natural extension of CTL* is to group the set of paths from the initial state that all correspond to the same policy:
  - $A_\pi pf$: ‘for all paths that agree with the policy $\pi$, $pf$ holds’;
  - $E_\pi pf$: ‘there exists a path that agrees with the policy $\pi$ for which $pf$ holds’.
The extension of CTL*: $\pi$-CTL$	extsuperscript{*}$

- One natural extension of CTL$^*$ is to group the set of paths from the initial state that all correspond to the same policy:
  - $A_\pi pf$: ‘for all paths that agree with the policy $\pi$, $pf$ holds’;
  - $E_\pi pf$: ‘there exists a path that agrees with the policy $\pi$ for which $pf$ holds’.

- **Semantics**: we add the policy accessibility relation $R_\pi$
  \[(s_j, R, R_\pi) \models sf \quad (s_j, R, R_\pi, \sigma) \models pf\]
The extension of CTL*: $\pi$-CTL*

- Given an initial state $s_0$, a policy $\pi$, a transition function $\Phi$, and a goal $G$ we say $\pi$ is a policy for $G$ from $s_0$, iff $(s_0, R, R_{\pi}) \models G$. 
The extension of CTL*: $\pi$-CTL$	extsuperscript{*}$

- Given an initial state $s_0$, a policy $\pi$, a transition function $\Phi$, and a goal $G$ we say $\pi$ is a policy for $G$ from $s_0$, iff $(s_0, R, R_\pi) \models G$.

- Note that goals must be state formulas rather than path formulas (in contrast to deterministic domains).
The extension of CTL*: \( \pi\text{-CTL}^* \)

- Given an initial state \( s_0 \), a policy \( \pi \), a transition function \( \Phi \), and a goal \( G \) we say \( \pi \) is a policy for \( G \) from \( s_0 \), iff \((s_0, R, R_\pi) \models G\).

- Note that goals must be state formulas rather than path formulas (in contrast to deterministic domains).

- Some examples of Goal Specification with \( \pi\text{-CTL}^* \) . . .
The weakest reachability goal “from the initial state there is a possibility that $p$ can be reached” is expressed by $E_{\pi} \Diamond p$. 
The weakest reachability goal “from the initial state there is a possibility that $p$ can be reached” is expressed by $E_{\pi} \diamond p$. From $s_1$, all policies but $\pi_7$ satisfy the goal.
A stronger goal “from the initial state \( p \) must be reached” is expressed as \( A_{\pi} \diamond p \). For \( s_1 \), no policy makes it true.
A stronger goal “from the initial state $p$ must be reached” is expressed as $A_π □ p$. For $s_1$, no policy makes it true. But, for instance, for $s_2$ the policy $\{(s_2, a_2)\}$ satisfies the goal.
“All along the trajectory there is always a possible path to $p$ by following the policy” is expressed as $A_{\pi} \Box (E_{\pi} \Diamond p)$. For $s_1$, no policy.
“All along the trajectory there is always a possible path to $p$ by following the policy” is expressed as $A_{\pi \Box} (E_{\pi \Diamond} p)$. For $s_1$, no policy. For $s_2$, policies $\{(s_2, a_2)\}$ and $\{(s_2, a_7)\}$ satisfy this goal.
However, policy \( \{(s_2, a_5)\} \) does not, (we could go to \( s_5 \) from where \( p \) can not be reached).
More examples

- $A_\pi(E \diamond p) = “$All along the trajectory there is always a possible path to $p$, but this path is not necessary abide the policy the agent taken”.$
More examples

- $A_\pi(E \diamond p) = \text{“All along the trajectory there is always a possible path to } p, \text{ but this path is not necessary abide the policy the agent taken”}.$

- $A(E_{\pi} \diamond p) = \text{“For any state that is reachable from the initial state, there is always a path to } p \text{ by following the policy.”}$
More examples

• $A_{\pi}(E\lozenge p) = \text{“All along the trajectory there is always a possible path to } p\text{, but this path is not necessary abide the policy the agent taken”}$. 

• $A(E_{\pi}\lozenge p) = \text{“For any state that is reachable from the initial state, there is always a path to } p\text{ by following the policy.”}$

• $E\lozenge p \rightarrow E_{\pi}\lozenge p = \text{“from the initial state, if it is possible to reach } p, \text{ the agent should possibly reach } p\text{”. Useful to allow the agent to pursue an alternative goal when it realizes that its initial goal is no longer achievable.”}$
More examples

• $A_\pi(E \Diamond p) = \text{“All along the trajectory there is always a possible path to } p, \text{ but this path is not necessary abide the policy the agent taken”}.$

• $A(E_\pi \Diamond p) = \text{“For any state that is reachable from the initial state, there is always a path to } p \text{ by following the policy.”}$

• $E \Diamond p \rightarrow E_\pi \Diamond p = \text{“from the initial state, if it is possible to reach } p, \text{ the agent should possibly reach } p\text{”}. \text{ Useful to allow the agent to pursue an alternative goal when it realizes that its initial goal is no longer achievable.}$

• $A_\pi \Box (E \Diamond p \rightarrow E_\pi \Diamond p) = \text{idem, but now from any state in the trajectory (not only initial one).}$
Outline

☑ Introduction

☑ LTL and CTL* overview

☑ The proposal: $\pi$-CTL*

❑ Related work

❑ Conclusions
Outline

✓ Introduction

✓ LTL and CTL* overview

✓ The proposal: $\pi$-CTL*

☞ Related work

❑ Conclusions
## Related work

<table>
<thead>
<tr>
<th>Dal Lago et al.’s formulation</th>
<th>$\pi$-CTL*</th>
</tr>
</thead>
<tbody>
<tr>
<td>$p$</td>
<td>$p$</td>
</tr>
<tr>
<td>TryReach $p$</td>
<td>$A_\pi \square E_\pi \Diamond p$</td>
</tr>
<tr>
<td>DoReach $p$</td>
<td>$A_\pi \Diamond p$</td>
</tr>
<tr>
<td>TryMaint $p$</td>
<td>$A_\pi \square p$</td>
</tr>
<tr>
<td>DoMaint $p$</td>
<td>$A_\pi \square p$</td>
</tr>
<tr>
<td>$g_1 \text{ And } g_2$</td>
<td>$g_1 \land g_2$</td>
</tr>
</tbody>
</table>
Related work

• Translation of other constructs still under study:
  \[ g_1 \text{ Then } g_2 \]
  \[ g_1 \text{ Fail } g_2 \text{ (particularly interesting)} \]
  \textbf{Repeat } g_2

• Some features difficult (or impossible) to represent in Dal Lago et al’s approach: \( E_{\pi}, \bigcirc, \bigdiamond, \) nesting operators.
Outline

✔ Introduction

✔ LTL and CTL* overview

✔ The proposal: \( \pi \)-CTL*

✔ Related work

❑ Conclusions
Outline

✔ Introduction
✔ LTL and CTL* overview
✔ The proposal: $\pi$-CTL*
✔ Related work
☞ Conclusions
Conclusions

• We extended CTL* to handle non-deterministic actions;

• Compared to Dal Lago et. al.’ formulation, our language does not define specialized syntax and semantics;

• More extensions are needed, some of them currently under study:
  – The comparison of different policies;
  – Using contexts: the agent can take different actions w.r.t. different contexts in a state.