Goal specification in presence of non-deterministic actions

Chitta Baral and **Jicheng Zhao** [1]

Abstract.
One important aspect in directing cognitive robots or agents is to formally specify what is expected of them. This is often referred to as goal specification. For agents whose actions have deterministic consequences various goal specification languages have been proposed. The situation is different, and less studied, when the actions may have non-deterministic consequences. For example, a simple goal of achieving p has many nuances such as making sure that p is achieved, trying ones best to achieve p, preferring guaranteed achievement of p over possible achievement, and so on. Similarly, there are many nuances in expressing the goal to try to achieve p, and if it fails then to achieve q. We develop an extension of the branching time temporal logic CTL*, which we call π-CTL*, and show how the above mentioned goals can be expressed using it, and why they can not be expressed in CTL*. We compare our approach to an alternative approach proposed in the literature.

1 Introduction and Motivation

In recent years it has been proposed [1, 7, 8, 2] that temporal logic be used to specify goals (of cognitive robots or agents) that go beyond only putting conditions on the final state. Most of these papers – except [8], only consider the case when actions are deterministic. In presence of non-deterministic actions, as first mentioned in [3], specifying goals has new challenges. For example, consider the following domain.

Example 1 *There are five different states: s_1, s_2, s_3, s_4, and s_5. The proposition p is only true in state s_4. The other states are distinguishable based on fluents which we do not elaborate here. Suppose the only possible actions and their consequences are the following: The agent can go from state s_1 to s_5 by action a_6. In state s_1, if the agent performs the action a_1, it may go to s_2 or s_3. In s_2, the action a_2 will take the agent to s_4 while the action a_5 may take it to either s_4 or s_5. If the agent is in state s_2, the action a_7 will take it to either state s_2 or state s_4. If the agent is in state s_3, the action a_4 will take it to state s_5. The action a_3 will take the agent from s_3 to either s_4 or s_5. Besides, in each state, there is always an action nop that keeps the agent in the same state. The transition graph of this domain is in Figure 1. In it we do not show the transitions due to the nop action.*

Consider an agent whose goal is to try its best to reach a state where p is true. The agents and its controllers are aware that some of the available actions have non-deterministic effects. Thus they are looking for policies – mapping from states to actions – instead of simple plans consisting of action sequences. Moreover, due to non-determinism they are worried about how to specify their goal so that the goal is not so strict that it is unachievable and still conveys the meaning of 'trying its best'. To appreciate the various nuances in expressing this goal let us consider several policies and compare them with respect to the goal of trying ones best to reach p. In the following we will represent a policy by a set of pairs of states and actions of the form (s, a) which will intuitively mean that in state s, action a should be executed. It is assumed that if no action is specified for a state then the action nop takes place.

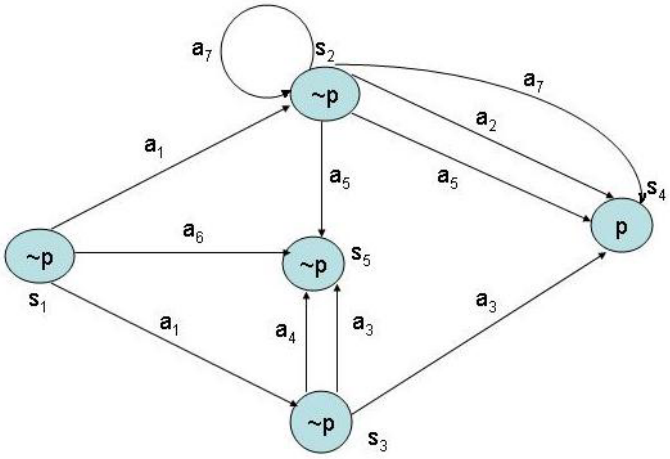

Figure 1. Transition between the locations

1. Policy $\pi_1 = \{(s_1, a_1), (s_2, a_2), (s_3, a_3)\}$
2. Policy $\pi_2 = \{(s_1, a_1), (s_2, a_5), (s_3, a_3)\}$
3. Policy $\pi_3 = \{(s_1, a_6)\}$
4. Policy $\pi_4 = \{(s_1, a_1), (s_2, a_2), (s_3, a_4)\}$
5. Policy $\pi_5 = \{(s_1, a_1), (s_2, a_5), (s_3, a_4)\}$
6. Policy $\pi_6 = \{(s_1, a_1), (s_2, a_7), (s_3, a_3)\}$
7. Policy $\pi_7 = \{(s_1, a_1), (s_2, a_7), (s_3, a_4)\}$

To specify the goal of 'trying ones best to reach p' let us try to use existing temporal logic formalisms to specify this goal. Since the use of policies lead to multiple trajectories, we can not directly use the specification $\Diamond p$ from linear temporal logic with future operators (LTL) [5, 4]. Thus we next try to express this goal in the branching time temporal logic CTL*, where we have the operators A (meaning 'for all paths') and E (meaning 'there exists a path') at our disposal.

[1] Department of Computer Science and Engineering, Arizona State University, AZ 85281, USA email: {chitta,jicheng}@asu.edu

Now suppose the initial state of our agent is s_1. From s_1 there is a path to s_4. Thus the CTL* goal $\mathsf{E}\Diamond p$ will be true with respect to s_1 and the transition function regardless of what policy one chooses including π_3. Definitely, π_3 is not a policy that is trying its best to get to p. Thus the specification $\mathsf{E}\Diamond p$ is incorrect. Now let us consider the goal $\mathsf{A}\Diamond p$. This goal is too strong as even if we consider our initial state as s_2 from which there is a policy that guarantees reaching p, the goal $\mathsf{A}\Diamond p$ will not be true with respect to s_2 and the transition function. This is because the semantics of E and A are tied to the overall transition relation, and not tied to a given policy. One way to overcome this is to either tie the semantics of E and A to the policy under consideration or introduce new operators (say, E_π and A_π) that tie the paths to the policy under consideration. In this paper we chose the second path, as to express certain goals it becomes necessary to have both versions (E, A, E_π and A_π) of the branching time operators. In this case the intuitive meaning of the operator A_π is 'for all paths that correspond to the policy under consideration' and the operator E_π is 'there exists a path that corresponds to the policy under consideration'.

Now let us discuss some of the nuances in expressing the goal 'try ones best to reach p'. Earlier we dismissed $\mathsf{E}\Diamond p$ and $\mathsf{A}\Diamond p$ as inappropriate. Let us now consider the goals $\mathsf{E}_\pi\Diamond p$ and $\mathsf{A}_\pi\Diamond p$. Both of these goals are satisfied if our initial state is s_2 and our policy includes (s_2, a_2). On the other hand if our initial state is s_3, we do not have a policy that will make $\mathsf{A}_\pi\Diamond p$ true, but the policy that includes (s_3, a_3) will make $\mathsf{E}_\pi\Diamond p$ true.

Going back to having s_2 as the initial state, it is preferable to have the policy include (s_2, a_7) instead of (s_2, a_5) in the absence of a_2. That is because while following (s_2, a_7) even though there is no guarantee that p will be reached, at any point during the execution there is always hope (a possible path sanctioned by the policy under consideration) that one might reach p in the future. This is not the case with respect to the policy (s_2, a_5), as once s_5 is reached there is no way to get to p. How does on express a goal that encodes this preference?

Now let us consider s_1 as our initial state. The policies π_1, π_2, π_4, π_5, π_6 and π_7 each satisfy the goal $\mathsf{E}_\pi\Diamond p$, while the policy π_3 does not. (Also no policy satisfies the goal $\mathsf{A}_\pi\Diamond p$.) Now let us compare π_1, π_2, π_4, π_5, π_6 and π_7. We claim that π_1 is preferable to the others with respect to the intuitive goal of trying ones best to achieve p. Among π_1 π_2, and π_6, the first one is preferable as once one gets to s_2 the π_1 guarantees that p will be reached while such is not the case with π_2 and π_6. π_6 is preferable than π_2 as once one gets to s_2, the π_6 guarantees that there is always hope of reach p in the future. Among π_4, π_5, and π_7, we have similar result for the same reason. Between π_1 and π_4, the former is preferable as if s_3 is reached during the execution then with respect to π_1 there is still hope that p may be reached, but such is not the case with respect to π_4. Between π_2 and π_5, the former is preferable for the same reason. The goal representation questions that arise are: How does one specify a goal with respect to which π_1, π_2 and π_4 are acceptable but π_5 is not? π_1 is acceptable, and π_2, π_4 and π_5 are not?

Our research in this paper on specifying goals in presence of non-deterministic actions is motivated by the work in [3] where some of the above nuances are mentioned. But there, a specialized language is proposed, and although it can capture some of the nuances it is not clear if it can capture all of them. Moreover, as we will elaborate a bit in Section 4, the approach and definitions in [3] have several other shortcomings. In this paper we try to stay within the framework of temporal logic and figure out a way to express non-deterministic goals using temporal logic such that it is easy to combine temporally expressed goals from earlier work [1, 7, 2] with the kind of goals that we discuss in this paper.

The rest of the paper is organized as follows. In Section 2 we introduce the notions that are necessary to express goals such as 'try your bests to achieve p' in presence of non-deterministic effects and give the syntax and semantics of the language π-CTL*. In Section 3 we show how various goals can be encoded in π-CTL*. In Section 4 we relate our proposal with some of the constructs in [3] and finally we conclude in Section 5.

2 Syntax and semantics of π-CTL*: goal specification in presence of non-deterministic actions and policies

In this section we show how to expand on the notions in the previous section so as to be able to specify goals such as the ones mentioned in Section 1. To start with in presence of non-deterministic actions, we need to expand the notion of a plan from a simple sequence of actions to a *policy which is a mapping from states to actions*. This is necessary because in presence of non-deterministic actions an agent can not be sure during planning time what state it would be in after executing an action or a sequence of actions. Thus often there may not exist a conformational plan consisting of sequence of actions, while there may exist a policy that will achieve a goal.

In the non-deterministic domain, we not only have to consider if there exists a path (even if that path is not a path that the agent will possibly follow by adhering to its policy), it is more important to consider if the path that exists is *a path consistent with respect to the policy that is followed*. To express the later we introduce the operators E_π (and A_π) which means that there exists a path (and for all paths respectively) consistent with respect to the policy π.

We now formally define the syntax and semantics of this extended branching time logic, which we will refer to as π-CTL*.

2.1 Syntax of π-CTL*

The syntax of state and path formulas in π-CTL* is as follows. Let $\langle p\rangle$ denote an atomic proposition, $\langle sf\rangle$ denote state formulas, and $\langle pf\rangle$ denote path formulas.

$$\langle sf\rangle ::= \langle p\rangle \mid \langle sf\rangle \wedge \langle sf\rangle \mid \langle sf\rangle \vee \langle sf\rangle \mid \neg\langle sf\rangle \mid \mathsf{E}\langle pf\rangle \mid \mathsf{A}\langle pf\rangle \mid \mathsf{E}_\pi\langle pf\rangle \mid \mathsf{A}_\pi\langle pf\rangle$$

$$\langle pf\rangle ::= \langle sf\rangle \mid \langle pf\rangle \vee \langle pf\rangle \mid \neg\langle pf\rangle \mid \langle pf\rangle \wedge \langle pf\rangle \mid \langle pf\rangle \,\mathsf{U}\, \langle pf\rangle \mid \bigcirc\langle pf\rangle \mid \Diamond\langle pf\rangle \mid \Box\langle pf\rangle$$

The new symbols A_π and E_π are the branching time operators meaning 'for all paths that agree with the policy that is being executed' and 'there exists a path that agrees with the policy that is being executed' respectively. Since we now allow actions to be non-deterministic the transition function Φ is now a mapping from states and actions to a *set of states*. Now we need to consider two transition relations R and R_π, the first used for defining paths for A and E and the second used in defining paths for A_π and E_π.

Recall that $R(s, s')$ means that the state of the world can change from s to s' in one step. This could be due to an agent's action, or an exogenous action. Thus $R(s, s')$ is true if there exists an action a such that $s' \in \Phi(a, s)$. $R_\pi(s, s')$ on the other hand means that the state of the world can change from s to s' in one step by following the agent's policy π. Thus $R_\pi(s, s')$ is true if $s' \in \Phi(\pi(s), s)$.

Definition 1 Path starting from s consistent with respect to the given policy Let s be a state, Φ be the transition between states due to actions, and π be a policy.

A sequence of states $\sigma = s_0, s_1, \ldots$ such that $s_0 = s$ is said to be a path starting from s consistent with respect to π, if σ is a path in R_π. I.e., $s_{i+1} \in \Phi(\pi(s_i), s_i)$.

We refer to such paths as π-paths. □

In Section 4 finite trajectories are considered. For that purpose we will abuse notation and by finite π-paths we will refer to finite sequences $s_0, s_1, \ldots, s_n$ such that for $0 \leq i < n$, $s_{i+1} \in \Phi(\pi(s_i), s_i)$.

2.2 Formal semantics of π-CTL*:

Recall that in CTL* truth of state formulas is defined with respect to a pair (s_j, R), where s_j is a state, R is the transition relation.

Definition 2 (Truth of state formulas in π-CTL*) The truth of state formulas is defined with respect to the triplet (s_j, R, R_π), where s_j and R are as before, and R_π is the transition relation with respect to the policy π. In the following p denotes a propositional formula sf_is are state formulas and pf_is are path formulas.

- $(s_j, R, R_\pi) \models p$ iff p is true in s_j.
- $(s_j, R, R_\pi) \models \neg sf$ iff $(s_j, R, R_\pi) \not\models sf$.
- $(s_j, R, R_\pi) \models sf_1 \wedge sf_2$ iff $(s_j, R, R_\pi) \models sf_1$ and $(s_j, R, R_\pi) \models sf_2$.
- $(s_j, R, R_\pi) \models sf_1 \vee sf_2$ iff $(s_j, R, R_\pi) \models sf_1$ or $(s_j, R, R_\pi) \models sf_2$.
- $(s_j, R, R_\pi) \models \mathsf{E}\, pf$ iff there exists a path σ in R starting from s_j such that $(s_j, R, R_\pi, \sigma) \models pf$.
- $(s_j, R, R_\pi) \models \mathsf{A}\, pf$ iff for all paths σ in R starting from s_j we have that $(s_j, R, R_\pi, \sigma) \models pf$.
- $(s_j, R, R_\pi) \models \mathsf{E}_\pi\, pf$ iff there exists a path σ in R_π starting from s_j such that $(s_j, R, R_\pi, \sigma) \models pf$.
- $(s_j, R, R_\pi) \models \mathsf{A}_\pi\, pf$ iff for all paths σ in R_π starting from s_j we have that $(s_j, R, R_\pi, \sigma) \models pf$. □

Definition 3 (Truth of path formulas in π-CTL*) The truth of path formulas are now defined with respect to the quadruple (s_j, R, R_π, σ), where s_j, R, and R_π are as before and σ given by the sequence of states $s_0, s_1, \ldots$, is a path.

- $(s_j, R, R_\pi, \sigma) \models sf$ iff $(s_j, R, R_\pi) \models sf$.
- $(s_j, R, R_\pi, \sigma) \models \neg pf$ iff $(s_j, R, R_\pi, \sigma) \not\models pf$
- $(s_j, R, R_\pi, \sigma) \models pf_1 \wedge pf_2$ iff $(s_j, R, R_\pi, \sigma) \models pf_1$ and $(s_j, R, R_\pi, \sigma) \models pf_2$.
- $(s_j, R, R_\pi, \sigma) \models pf_1 \vee pf_2$ iff $(s_j, R, R_\pi, \sigma) \models pf_1$ or $(s_j, R, R_\pi, \sigma) \models pf_2$.
- $(s_j, R, R_\pi, \sigma) \models \bigcirc pf$ iff $(s_{j+1}, R, R_\pi, \sigma) \models pf$.
- $(s_j, R, R_\pi, \sigma) \models \Box pf$ iff $(s_k, R, R_\pi, \sigma) \models pf$, for all $k \geq j$.
- $(s_j, R, R_\pi, \sigma) \models \Diamond pf$ iff $(s_k, R, R_\pi, \sigma) \models pf$, for some $k \geq j$.
- $(s_j, R, R_\pi, \sigma) \models pf_1\ \mathsf{U}\ pf_2$ iff there exists $k \geq j$ such that $(s_k, R, R_\pi, \sigma) \models pf_2$, and for all i, $j \leq i < k$, $(s_i, R, R_\pi, \sigma) \models pf_1$. □

2.3 Plans/policies for π-CTL* goals:

Now we need to define when a mapping π from states to actions is a policy with respect to a π-CTL* goal G, an initial state s_0, and a transition function Φ from states and actions to sets of states.

Definition 4 (Policy for a goal from an initial state) Given an initial state s_0, a policy π, a transition function Φ, and a goal G we say π is a policy for G from s_0, denoted by $(s_0, \pi) \models G$, iff $(s_0, R, R_\pi) \models G$. □

From the above definition it is clear that *goals must be state formulas*. This is in contrast to the case where actions are deterministic and instead of policies one only needs to deal with plans consisting of action sequences. There it is preferable to have goals as path formulas.

3 Goal representation with π-CTL*

In this section we show how various kinds of goals which can not be appropriately expressed in LTL or CTL*, can be expressed using π-CTL*. We categorize our goals to three classes: reachability goals; maintainability goals; and bi-composed goals.

3.1 Reachability goals

In considering whether a goal can be reached or not, following are some observations when actions have non-deterministic effects.

1. Given the initial state and a policy, a state s is not reachable in any π-path. By following such a policy, it is impossible for the agent to get into a state where p can be reached by following the policy.
2. Given the initial state and a policy, a state s may be reachable in some π-paths and not in others. By following such a policy, at some stage, it is possible that the agent gets into a state from where it can guaranteedly reach s. It is also possible that the agent gets into a state from where he can never reach s by following the policy.
3. Given the initial state and a policy, a state s can be reached in any π-path. The agent can guarantee to reach s in any future state by following such a policy.

We now illustrate how various kinds of reachability goals can be specified in π-CTL*. In this we consider the domain in Example 1 and the subsequent discussion.

1. The goal "from the initial state there is a possibility that p can be reached" is expressed by the specification $E_\pi \Diamond p$. With respect to the initial state s_1, the policies π_1, π_2, π_4, π_5, π_6, and π_7 each satisfy the goal $\mathsf{E}_\pi \Diamond p$, while the policy π_3 does not.
2. The goal "from the initial state p must be reached" is expressed by the specification $A_\pi \Diamond p$. With respect to the initial state s_1, this goal is not satisfied as there is no policy which can make this true. But with respect to the state s_2 the policy $\{(s_2, a_2)\}$ satisfies this goal.
3. The goal "all along the trajectory there is always a possible path to p" is expressed as $A_\pi \Box (E_\pi \Diamond p)$. With respect to the initial state s_1, this goal is not satisfied as there is no policy which can make this true. But with respect to the state s_2 the policy $\{(s_2, a_7)\}$ satisfies this goal. So does the policy $\{(s_2, a_2)\}$.
4. The goal "from the initial state, if it is possible to reach p, the agent should possibly reach p" is expressed as $E \Diamond p \rightarrow E_\pi \Diamond p$. Policies that satisfy this goal are: π_1, π_2, π_4, π_5, π_6, and π_7. A policy that does not satisfy this goal is π_3.
   Intuitively, this goal says that if it is possible to reach p, then the agent can possibly reach it. Otherwise, we consider the agent to have achieved this goal even if it does nothing. The usefulness of such goals will be shown in the next section where the agent can pursue an alternative goal when its initial goal is not achievable.

5. The goal "If from any state that is reachable by following the policy it is possible to reach p, then the agent should possibly reach p from that state" is expressed as $A_\pi\Box(E\Diamond p \rightarrow E_\pi\Diamond p)$. Policies π_1, π_2, and π_6 satisfy this goal while policies π_4, π_5, and π_7 do not satisfy this goal.

3.2 Maintainability goals

In this paper we consider maintainability as the opposite of reachability. For example, in the deterministic domain, given a plan, we say the propositional formula p is maintained iff $\neg p$ cannot be reached. It is also the case in the non-deterministic domain. For example, if we require that p must be maintained in any trajectory starting from a state by following the policy, then the policy is satisfied iff there is no trajectory such that $\neg p$ is reached. Formally, it is $\neg E_\pi\Diamond\neg p$, which is equivalent to $A_\pi\Box p$. As a consequence, in formulating the goals about maintainability, we can indeed translate them into the goals of checking whether a state can be reached or not, thus the various notions of reachability from the previous section have corresponding notions of maintainability.

3.3 Bi-composed goals: combining two sub-goals

In this section we consider goals constructed by the composition of two sub-goals. Even if we only consider reachability goals, because of the various nuances of individual reachability goals, their composition leads to many more nuances. Intuitively, the dynamics of the various possibilities comes from the following aspects:

1. Initially a formula can be reached by the policy; however, during the execution of the actions in the policy, when the agents come to some state, it may realize that from that point on the formula it intended to reach can never be reached.
2. Initially a formula can be reached by the policy, but the formula is not guaranteed by the policy. During the execution of the actions in the policy, when the agent gets to some state, it may realize that the formula can be guaranteedly reached even taking the non-deterministic property of the domain into account.

In general, different (bi-composed) goals can be constructed by different answers to the following questions: Do we have to reach the first goal? When we give up the first goal? When the first goal is reached, do we still need to reach the second goal? When the first goal cannot be reached, do we need to reach the second goal? In the processing of reaching the second goal, do we need to keep an eye on the first goal? When we start to keep an eye on the first goal? In what condition, we pause the second goal and resume the first goal? When the whole formula is considered as satisfied? We now specify – using π-CTL* – some bi-composed goals made up of reachability goals, corresponding to various answers to the above mentioned questions.

1. We require the policy must reach p, and must reach q after reaching p. We start to consider q only after reaching p, we do not care whether q is reached or not in the process of reaching p. The π-CTL* representation of the goal is $A_\pi\Diamond(p \wedge A_\pi\Diamond q)$.
2. In a state if it is possible to reach p, try to reach p until it is impossible to do so. From the state that p can never be reached, try to reach q until it is impossible to do so. The π-CTL* representation of this goal is $A_\pi\Box((E\Diamond p \rightarrow E_\pi\Diamond p) \wedge ((\neg E\Diamond p \wedge E\Diamond q) \rightarrow E_\pi\Diamond q))$.
3. If there is a trajectory that makes it possible to reach p, try to reach it. If you are in a state that p can never be reached, you must reach q from that state. The π-CTL* representation of the goal is $A_\pi\Box((E\Diamond p \rightarrow E_\pi\Diamond p) \wedge (\neg(E\Diamond p) \rightarrow A_\pi\Diamond q))$.

4 Relation with Dal Lago et al.'s formulation

In this section we consider some of the constructs from [3] and show how they can be expressed using π-CTL*.
Their goal language is defined as follows where p denotes propositional formulas and g denotes extended goals.

$$
\begin{array}{rcl}
p & ::= & \top \mid \bot \mid \neg p \mid p \vee p \mid p \wedge p \\
g & ::= & p \mid g\ \mathbf{And}\ g \mid g\ \mathbf{Then}\ g \mid g\ \mathbf{Fail}\ g \mid\ \mathbf{Repeat}\ g \mid \\
 & & \mathbf{DoReach}\ p \mid \mathbf{TryReach}\ p \mid \\
 & & \mathbf{DoMaint}\ p \mid \mathbf{TryMaint}\ p
\end{array}
$$

Their definition of a policy satisfying a goal from an initial state is based on defining two states $\mathcal{S}_g(s)$ and $\mathcal{F}_g(s)$. Intuitively, $\mathcal{S}_g(s)$ is the set of finite paths consistent with respect to π that leads to success in the achievement of g from s. Similarly, $\mathcal{F}_g(s)$ is the set of finite paths consistent with respect to π that leads to failure in the achievement of g from s. In their definition, a policy π satisfies a goal g – denoted by $(s,\pi) \models_{dlpt} g$ – from an initial state s if $\mathcal{F}_g(s) = \emptyset$.

They define the sets $\mathcal{S}_g(s)$ and $\mathcal{F}_g(s)$ for each of the different kinds of goals. We give them below. In these definitions $\sigma \leq \sigma'$ means σ is a prefix of σ' (we also say that σ' is an extension of σ) , $first(\sigma)$ and $last(\sigma)$ denote the first and last state in σ, min is defined with respect to $\leq$, and semicolon denotes concatenation and $\sigma;\sigma'$ is only defined when $last(\sigma) = first(\sigma')$.
Using the above results we now show how some of their constructs can be expressed in π-CTL*. Note that we use $(s,\pi) \models_{dlpt} g$ to denote a policy π satisfies a goal g from an initial state s in [3].

Proposition 1 Let s_0 be an initial state and π be a policy.

1. $(s_0,\pi) \models_{dlpt} p$ iff $(s_0,\pi) \models p$.
2. $(s_0,\pi) \models_{dlpt} \mathbf{TryReach}\ p$ iff $(s_0,\pi) \models \mathsf{A}_\pi\Box\mathsf{E}_\pi\Diamond p$.
3. $(s_0,\pi) \models_{dlpt} \mathbf{DoReach}\ p$ iff $(s_0,\pi) \models \mathsf{A}_\pi\Diamond p$.
4. $(s_0,\pi) \models_{dlpt} \mathbf{TryMaint}\ p$ iff $(s_0,\pi) \models \mathsf{A}_\pi\Box p$.
5. $(s_0,\pi) \models_{dlpt} \mathbf{DoMaint}\ p$ iff $(s_0,\pi) \models \mathsf{A}_\pi\Box p$.
6. $(s_0,\pi) \models_{dlpt} g_1\ \mathbf{And}\ g_2$ iff $(s_0,\pi) \models g_1 \wedge g_2$. □

4.1 Discussion on the comparisons

So far we have given some formal results about the formulation in [3] and about the relation between some of the constructs their constructs and our language of π-CTL*. In this subsection we make some further comparisons and observations.

1. $\mathbf{TryReach}\ p$: The intended meaning of this goal is that an agent policy that satisfies this goal must do its best to reach p. That means if the agent follows its policy then at every state reached while following this policy there is a path (which is possible by following the agent's policy but not necessarily guaranteed by the agent's policy) from that state to a state where p is true. From Proposition 1 this can be expressed in π-CTL* by $\mathsf{A}_\pi\Box\mathsf{E}_\pi\Diamond p$.

   Now suppose the intention is that once p is reached p must remain true after that, then in that case the specification would be $\mathsf{A}_\pi\Box\mathsf{E}_\pi\Diamond\Box p$. Similarly, if the intention is that the existence of the path be only true at the initial state s_0 then we can remove the $\mathsf{A}_\pi\Box$ in the beginning and either $\mathsf{E}_\pi\Diamond p$ or $\mathsf{E}_\pi\Diamond\Box p$ would suffice.

It is not clear how these alternatives and the other ones mentioned earlier in the paper (with respect to reachability) can be expressed using the language in [3].

2. $\mathbf{DoReach}\ p$: The intended meaning of this goal is that an agent policy that satisfies this goal must take the world to a state where p is true. That means if the agent follows its policy then no matter what path it takes (due to the non-deterministic effect of actions), all those paths lead to a state where p is true. From Proposition 1 this can be expressed in π-CTL* by $\mathsf{A}_\pi \Diamond p$.

   Now suppose the intention is that once p is reached p must remain true after that, then in that case the specification would be $\mathsf{A}_\pi \Diamond \Box p$. It is not clear how these alternatives can be expressed using the language in [3].
3. $\mathbf{DoMaint}\ p$: The intended meaning of this goal is that an agent policy that satisfies this goal must take paths where p is true in all states of the path. That means if the agent follows its policy then no matter what path it takes (due to the non-deterministic effect of actions), p is true in all the states of those paths. From Proposition 1 this can be expressed in π-CTL* by $\mathsf{A}_\pi \Box p$.
4. $\mathbf{TryMaint}\ p$: From Proposition 1 the exact representation of this, based on the characterization in [3], in π-CTL* by $\mathsf{A}_\pi \Box p$. This is same as the representation of $\mathbf{DoMaint}\ p$. Thus, according to [3] there is no distinction between $\mathbf{TryMaint}\ p$ and $\mathbf{DoMaint}\ p$, which is somewhat unintuitive.

   It seems to us there is a more intuitive (but different) characterization of the notion of trying to maintain p, according to which if the agent follows its policy then at every state reached while following this policy there is a path (which is possible by following the agent's policy but not necessarily guaranteed by the agent's policy) from that state where p is true all through the path. This can be expressed in π-CTL* by $\mathsf{A}_\pi \Box \mathsf{E}_\pi \Box p$.

   Now suppose the intention is that the existence of the path be only true at the initial state s_0 then we can remove the $\mathsf{A}_\pi \Box$ in the beginning and $\mathsf{E}_\pi \Box p$ would suffice.
5. $g_1\ \mathbf{Then}\ g_2$: There are several different intuitive meaning of this based on whether or not g_2 is forced to be false before g_1 is achieved. Besides, with respect to [3], the goal $p_1\ \mathbf{Then}\ p_2$ is satisfied when p_1 and p_2 are both true in the initial state. This seems counterintuitive to us. We will discuss this in detail in the full version of the paper.
6. $g_1\ \mathbf{Fail}\ g_2$: The intended meaning of this goal is to achieve g_1 and when it becomes clear that g_1 is not achievable then g_2 is achieved. Earlier we discussed various nuances of expressing this. Only one of those can correspond to the semantics of [3].
7. It seems that the notion E_π, and other temporal notions such as $\bigcirc$ and $\Diamond$ are not expressible using the formalism in [3].
8. The formalism in [3] does not allow nesting of many of the operators. There are no such restrictions in π-CTL*.

There are many other issues of concern with respect to the formulation in [3]. For example, intuitively, for all policy $policy$, $(s, R, R_{policy}) \models \mathbf{DoMaint}\, p$ iff $(s, R, R_{policy}) \models \mathbf{TryMaint}\, p$ where p is a proposition formula. However, $(s, R, R_{policy}) \models \mathbf{DoMaint}\ p\ \mathbf{Fail}\ q$ cannot imply that $(s, R, R_{policy}) \models \mathbf{TryMaint}\ p\ \mathbf{Fail} q$. Nevertheless, the paper [3] was the first one (to the best of our knowledge) that talked about the issues that crop up when expressing goals in a non-deterministic domain, and thus is a pioneer in that respect. We hope we have given an alternative appropriate framework to address the issues raised there.

5 Conclusion and future work

In this paper we considered representing goals with temporal aspects in presence of non-deterministic actions. We analyzed why temporal logics such as LTL and CTL* were thought to be not adequate to express certain kind of goals. By our analysis we discovered the source of confusion: the notion of $path$ that is tied to the branching time operators A and E. We showed that by introducing additional branching time operators A_π and E_π where the path is tied to the policy being executed we can express the goals that were thought inexpressible using temporal logic in [3].

We believe our approach is preferable to the approach in [3] where a specialized language is introduced. This is because by using our language we can still represent the goals that were traditionally represented (when actions were assumed to be deterministic) using temporal logic, and also combine such goals with the kind of goals discussed in this paper. Use of a specialized language, as in [3], makes this difficult if not impossible.

In terms of future work we need to consider some further generalizations. For example, in certain cases we may need to distinguish between paths solely due to actions that can be executed by an agent, and paths solely due to exogenous actions. Or we may need to consider paths that interleave agent's actions and exogenous actions in a particular way such as alternating them. Each of these may necessitate use of additional transition relations (such as R_a corresponding to agent's actions, and R_e corresponding to exogenous actions) and additional branching time operators (such as E_e, A_e, E_a, and A_a).

We also plan to further elaborate on the π-CTL* specifications of the various constructs of [3]. Finally, we also plan to consider alternative notions of maintainability [6] and explore its relation with the formulations in this paper.

6 Acknowledgement

This work was partially supported by NSF grant number 0070463, NASA grant number NCC2-1232 and a grant from ARDA under the second phase of its AQUAINT program. The authors acknowledge encouragement and feedback from Matt Barry of NASA JPL and Vladimir Lifschitz. The authors also acknowledge the valuable comments of the referees of this paper and its earlier versions.

REFERENCES

[1] F. Bacchus and F. Kabanza, 'Planning for temporally extended goals', *Annals of Math and AI*, **22**, 5–27, (1998).
[2] C. Baral, V. Kreinovich, and R. Trejo, 'Computational complexity of planning with temporal goals', in *IJCAI 2001*, (2001).
[3] U. Dal Lago, M. Pistore, and P. Traverso, 'Planning with a language for extended goals', in *AAAI'02*, pp. 447–454, (2002).
[4] E. A. Emerson, 'Temporal and modal logic', in *Handbook of theoretical computer science: volume B*, ed., J. van Leeuwen, 995–1072, MIT Press, (1990).
[5] Z. Manna and A. Pnueli, *The temporal logic of reactive and concurrent systems: specification*, Springer Verlag, 1992.
[6] M. Nakamura, C. Baral, and M. Bjareland, 'Maintainability: a weaker stabilizability like notion for high level control', in *Proc. of AAAI'00*, pp. 62–67, (2000).
[7] R. Niyogi and S. Sarkar, 'Logical specification of goals', in *Proc. of 3rd international conference on Information Technology*, pp. 77–82, (2000).
[8] M. Pistore and P. Traverso, 'Planning as model checking for extended goals in non-deterministic domains', in *IJCAI'01*, (2001).