A Temporal Logic for Markov Chains

Wojciech Jamroga
Clausthal University of Technology, Germany
wjamroga@in.tu-clausthal.de

ABSTRACT

Most models of agents and multi-agent systems include information about possible states of the system (that defines relations between states and their external characteristics), and information about relationships between states. *Qualitative* models of this kind assign no numerical measures to these relationships. At the same time, *quantitative models* assume that the relationships are measurable, and provide numerical information about the degrees of relations. In this paper, we explore the analogies between some qualitative and quantitative models of agents/processes, especially those between transition systems and Markovian models.

Typical analysis of Markovian models of processes refers only to the expected utility that can be obtained by the process. On the other hand, modal logic offers a systematic method of describing phenomena by combining various modal operators. Here, we try to exploit linguistic features, offered by propositional modal logic, for analysis of Markov chains and Markov decision processes. To this end, we propose *Markov temporal logic* – a multi-valued logic that extends the branching time logic CTL*.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent Systems*; I.2.4 [**Artificial Intelligence**]: Knowledge Representation Formalisms and Methods—*Modal logic*

General Terms

Theory

Keywords

Temporal logic, Markov chains, Markov decision processes

1. INTRODUCTION

There are many different models of agents and multi-agent systems; however, most of them follow a similar pattern. First of all, they include information about possible situations (states of the system) that defines relations between states and their external characteristics (essentially, "facts of life" that are true in these states). Second, they provide information about relationships between states (e.g, possible transitions between states).

Models that share this structure can be, roughly speaking, divided into two classes. *Qualitative models* provide no numerical measures for these relationships. They are widely used as basic models of computational systems, in semantics of programming languages (including agent-oriented languages), and in specification and verification of systems. Qualitative models seem especially suited for domains in which quantitative information cannot be reliably obtained nor assumed. They are also used to model situations in which the goal of an agent (or the whole system) is not to maximize a measurable output, but rather to achieve a state that matches certain characteristics (specified e.g. by a logical formula).

Quantitative models assume that relationships are measurable, and provide numerical information about the degrees of relations. For the relations between states, the degrees are usually given as probabilities. For the "qualities" of particular states, one often talks about *rewards* or *utilities*. Quantitative representations are used in stochastic modeling (Markov chains), decision theory and reinforcement learning (Markov decision processes), game theory (strategic and extensive game forms) etc. In this paper, we explore analogies between transition systems and Markovian models in order to provide a more expressive language for reasoning about, and specification of agents in stochastic environments.

Analysis of quantitative process models is usually based on the notion of expected reward. Still, other features of Markov chains and Markov decision processes can be also interesting. We propose to use the methodology of propositional modal logic in order to study quantitative properties of systems and processes. *Markov temporal logic* for Markov chains, introduced in Section 4, is our first step in this direction. We also briefly consider two extensions of the logic: for Markov decision processes (where a single decision maker is present) and for multi-agent Markov decision processes (in which many agents can play simultaneously).

1.1 Related Work

Related work includes research on multi-valued logics, especially fuzzy logics [23, 11], probabilistic logics [21, 22], and multi-valued modal logics [9, 7]. Of the latter, [17] is particularly relevant, as it defines a multi-valued version of CTL*, with propositions and accessibility relations taking values from a finite quasi-Boolean algebra. Still, the approach of [17] is too abstract to give an account of quanti-

Cite as: A Temporal Logic for Markov Chains, Wojciech Jamroga, *Proc. of 7th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2008)*, Padgham, Parkes, Müller and Parsons (eds.),May,12-16.,2008,Estoril,Portugal,pp.697-704.

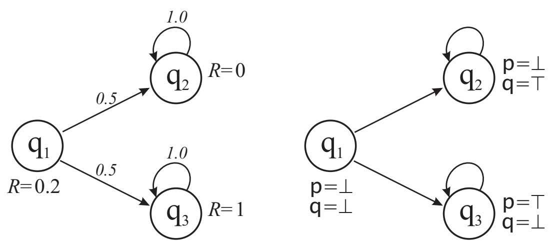

Figure 1: (A) Markov chain. (B) Unlabeled transition system

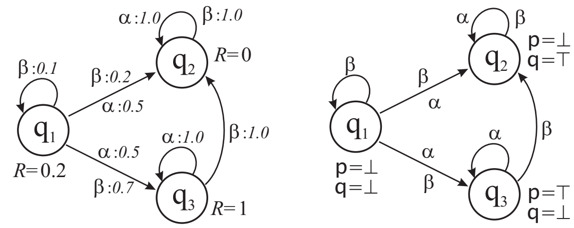

Figure 2: (A) Markov decision process. (B) Labeled transition system

tative analysis of processes (e.g., by operators that compute the expected and/or average truth value along a given path).

Logics of probability [12] are also related to the phenomena we study here. Important examples of such logics are two probabilistic variants of CTL: PCTL [13] for real time, and pCTL* [2] for discrete time; both allow to express probability bounds for a specified behavior. However, logics of probability do not use the machinery of multi-valued logics. More importantly, like probabilistic logics, they focus on the probabilities of events (e.g., behaviors), and it is often hard to attribute an intuitive meaning to combinations (or patterns) of different probability values. In contrast, we will argue in Section 2.3 that combining utilities has a very natural commonsense interpretation.

Our work comes very close to [6], where the "Discounted CTL" (DCTL) is proposed. In fact, our Markov temporal logic directly extends the ideas behind DCTL; a more detailed comparison is presented in Section 6. The variant of multi-valued CTL from [18], where the domain of truth values can be any c-semiring (rather than simply the interval $[0, 1]$ of real numbers), is also relevant. While it does not address quantitative analysis of processes directly, the choice of c-semirings makes such analysis possible (at least in principle). It may be interesting to consider a similar generalization of our framework in the future.

2. LOOKING FOR ANALOGIES

We begin with drawing some analogies between the quantitative and qualitative approaches to computational systems. In particular, we are interested in similarities between Markovian models of processes and transition systems.

2.1 Quantitative vs. Qualitative Models

The simplest Markovian models are *Markov chains* [20, 16, 10], discrete-time stochastic processes in which the next state of the system depends only on the current state and possibly the current action(s), but it does not directly depend on the past states of the system. A formal definition is given in Section 3.2. An example Markov chain is depicted in Figure 1, together with an unlabeled transition system. It is easy to see the similarities. First, states in the Markov chain are assigned real reward values R, and states in the transition system are assigned valuations of atomic propositions $\mathsf{p}, \mathsf{q}, \ldots$. Moreover, both kinds of structures include a set of states and a (single) binary transition relation on states; however, in the Markov chain, tuples of the relation are annotated with transition probabilities.

Markov decision processes [4, 3] can be seen as an extension of Markov chains, where several actions are available in each state. We observe that Markov decision processes are very much like labeled transition systems. In both cases, the action-transition structure can be modeled by a number of binary relations on states (one relation per action), although the elements of relations in Markov decision processes are annotated with probability values (cf. Figure 2). We also observe the similarity between multi-agent Markov decision processes from [5] and concurrent game structures from [1].

2.2 Quantitative vs. Qualitative Descriptions

The tradition of decision theory and reinforcement learning puts forward the quantitative notion of expected utility which represents the average of "what we can get" for all possible executions of the process. At the same time, logical approaches are usually concerned with "limit properties" like the existence of an execution that displays a specific temporal pattern. Logical frameworks are not very well suited to coping with models that involve probabilities: the existence of a particular kind of execution may be of little interest if this kind of execution is unlikely to happen. It does not mean, however, that these "limit properties" are irrelevant: in some cases we do want to e.g. make sure that there is no path violating an important security property. The point we are trying to make in this paper is that *both* kinds of properties are interesting and worth using to describe processes.

One of the nicer features of temporal logics – especially branching-time logics like CTL and CTL* – is that they offer a systematic approach in which properties of particular paths (executions) are distinguished from the properties of sets of paths (e.g., the set of all executions of a process). The first kind of properties is facilitated by *temporal operators* like "always" ($\Box$), "eventually" ($\Diamond$), "next" ($\bigcirc$) etc. The second kind is based on *path quantifiers* like "for all paths" (A) and "there is a path" (E). Both kinds of operators can be combined: e.g., $\mathsf{E}\Box\mathsf{safe}$ says "there is a path such that the system is always in a safe state". The same approach can be employed within the quantitative framework. For instance, besides the expected value of cumulative future reward, we can ask of the maximal (or minimal) cumulative reward. Or, we might be concerned with the expected value of minimal guaranteed reward etc. We propose a precise semantics for such combinations (and a semantics of interplay between qualitative and quantitative properties) in Section 4.

2.3 Logical operators as Minimizers and Maximizers

Note that – when truth values represent utility of an agent – temporal operators "sometime" and "always" have a very natural interpretation. "Sometime p" ($\Diamond p$) can be rephrased as "p is achievable in the future". Thus, under the assumption that agents want to obtain as much utility as possible, it is natural to view the operator as maximizing the utility value along a given temporal path. Similarly, "always p" ($\Box p$) can be rephrased as "p is guaranteed from now on". In other words, $\Box p$ asks for the minimal value of p on the path. On a more general level, every universal quantifier is essentially a minimizer of truth values, while existential quantifiers can be seen as maximizers. Thus, $\mathsf{A}\gamma$ ("for all paths γ") minimizes the utility specified by γ across all paths that can occur, etc. Also, conjunction and disjunction can be seen as a minimizer and a maximizer: $\varphi \vee \psi$ reads easily as "the utility that can be achieved through φ or ψ", while $\varphi \wedge \psi$ reads as "utility guaranteed by both φ and ψ".

Of course, the idea of defining semantics of conjunction and disjunction through functions min and max, respectively, is not new: the same semantic approach is used e.g. in fuzzy logic [23, 11]. Also, interpreting quantifiers as outcome maximization/minimization operators, can be traced back to the game semantics of classical logic [14, 19].

3. BASIC MODELS: MARKOV CHAINS AND MARKOV DECISION PROCESSES

Markov chains have been proposed to represent and study properties of processes in which transitions can be described in terms of probabilities. Markov chains are often used for generation of semi-random sequences of words, symbols or events (algorithms generating spam messages are a good example here). For these applications, states of a system (chain) play mostly technical role, as we are mainly after the events being generated. However, Markov chains can be also used to model and analyze existing processes (especially as parts of *Markov decision processes*, perhaps the most popular models of reinforcement learning). In that case, we are usually interested in properties of the states: either qualitative (i.e., some facts being true or false in different states of the process) or quantitative (representing utilities or rewards that the process is expected to yield in particular states). Even more importantly, we are interested in how these (qualitative or quantitative) properties accumulate as the system progresses in time.

3.1 Domain

A *domain* $D = \langle U, \top, \bot, \overline{} \rangle$ consists of: (1) a set $U \subseteq \mathbb{R}$ of *utility values* (or simply *utilities*); (2) special values $\top, \bot$ standing for the logical truth and falsity, respectively; $\hat{U} = U \cup \{\top, \bot\}$ will be called the *extended utility set*; and, finally, (3) a complement function $\overline{} : \hat{U} \to \hat{U}$. A domain should satisfy the following conditions:

1. $U \subseteq \mathbb{R}$;
2. The operations of addition and multiplication have their typical properties on $\hat{U}$, and $\hat{U}$ is closed under averaging, i.e., for every probability distribution P over $\hat{U}$ (discrete or continuous), $\sum_{u \in \hat{U}} u\ P(u) \in \hat{U}$;
3. U is closed under complement: if $u \in U$ then $\overline{u} \in U$;
4. Complement reverts the classical truth values: $\overline{\top} = \bot$ and $\overline{\bot} = \top$;
5. $\top \geq 0$;

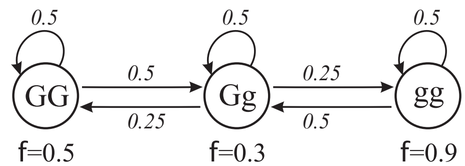

Figure 3: Markov chain for the gene model

6. $\bot \leq u$ and $\top \geq u$ for all $u \in \hat{U}$;[1]
7. The complement is quasi-boolean wrt max, min, i.e., for every $u_1, u_2, u \in \hat{U}$: $\overline{\max(u_1, u_2)} = \min(\overline{u_1}, \overline{u_2})$, $\overline{\min(u_1, u_2)} = \max(\overline{u_1}, \overline{u_2})$, $\overline{u_1} \leq \overline{u_2}$ iff $u_2 \leq u_1$, and $\overline{\overline{u}} = u$.

In the rest of the paper, we will assume that $U = [0, 1], \top = 1, \bot = 0, \overline{u} = 1 - u$. This closely resembles the setting in [6]. Admittedly, using 0 and 1 to represent "false" and "true" has a long tradition in logic; there is also a tradition of using values between 0 and 1 in multi-valued logics.

3.2 Markov Chains

Typically, a Markov chain is a directed graph with probabilistic transition relation. In our definition, we include also a device for assigning states with utilities and/or propositional values. This is done through *utility fluents* which generalize atomic propositions from modal logic, in the sense that they can also take real numbers as their values.

Definition 1 (Markov chain). *A* Markov chain *over domain $D = \langle U, \top, \bot, \overline{} \rangle$, and a set of utility fluents Π is a tuple $M = \langle St, \tau, \pi \rangle$, where:*

- *St is a set of* states *(we will assume that the set is finite and nonempty throughout the rest of the paper);*
- *$\tau : St \times St \to [0, 1]$ is a* stochastic transition relation *that assigns each pair of states q_1, q_2 with a probability $\tau(q_1, q_2)$ that, if the system is in q_1, it will change its state to q_2 in the next moment. For every $q_1 \in St$, $\tau(q_1, \cdot)$ is assumed to be a probability distribution, i.e. $\sum_{q \in St} \tau(q_1, q) = 1$.*
  By abuse of notation, we will sometimes write $\tau(q)$ to denote the set of states accessible in one step from q, i.e. $\{q' \mid \tau(q, q') > 0\}$.
- *$\pi : \Pi \times St \to \hat{U}$ is a* valuation of utility fluents.

Example 1 (Gene model). *Consider the following extension of the "gene model" from [10]. A trait in animals of a particular species is governed by a pair of genes, each of whom may be of type G or g. Very often the GG and Gg types are indistinguishable in appearance; we say that type G dominates type g. Thus, an individual can have the* dominant *combination GG,* recessive *combination gg, or* hybrid *combination Gg (which is genetically the same as gG).*

Mating of two animals produces an offspring that inherits one gene of the pair from each parent, and the basic assumption of genetics is that these genes are selected at random, independently of each other. Suppose that we breed animals by starting with an individual of known genetic character

[1] Note that this implies that $\max(u, \top) = \top$, $\min(u, \top) = u$, $\min(u, \bot) = \bot$, and $\max(u, \bot) = u$ for all $u \in \hat{U}$.

and mate it with a hybrid. We assume that there is at least one offspring. Then, at each round, a random offspring is chosen and mated with a hybrid, and so on. Suppose also that a statistical study of survival produced the following fitness function for individuals of the species (in relation to genotype): $f(GG) = 0.5$, $f(Gg) = 0.3$, and $f(gg) = 0.9$ – i.e., the individuals with recessive genes are the fittest, and hybrids are the least fit of all. Furthermore, utility fluent f *is used to represent fitness values. A Markov chain that models the process is shown in Figure 3.*

A *run* in Markov chain M is an infinite sequence of states $q_0q_1\dots$ such that each q_{i+1} can follow q_i with a non-zero probability, i.e., for every $i = 0, 1, \dots$ we have $\tau(q_i, q_{i+1}) > 0$. We denote the set of all runs in M by $\mathcal{R}_M$. The set of runs starting from state q is denoted by $\mathcal{R}_M(q)$.[2] Let $\lambda = q_0q_1...$ be a run and $i \in \mathbb{N}_0$. Then: $\lambda[i] = q_i$ denotes the ith position in λ; $\lambda[i..j] = q_i \dots q_j$ denotes the subpath of λ from position i to j; and $\lambda[i..\infty] = q_iq_{i+1}\dots$ denotes the infinite subpath of λ from position i on.

Finite prefixes of runs are called *histories*. $\mathcal{H}_M = \{h \mid h = \lambda[0..i]$ for some $\lambda \in \mathcal{R}_M, i\}$ denotes the set of all histories in M. $\mathcal{H}_M(q)$ is the set of histories starting from q; $\mathcal{H}^k_M(q)$ restricts the set further to the histories of length k. Note that each history h can be uniquely identified with the set of runs that "complete" it. By a slight abuse of notation, we will also use h to denote the set, and $\mathcal{H}_M(q)$ to denote all such subsets of $\mathcal{R}_M(q)$. Finally, by $\lambda(h)$ we denote an arbitrary infinite continuation of h (e.g., the run from h which is minimal wrt to alphabetical ordering of runs).

3.3 Markov Decision Processes

Markov decision processes extend Markov chains with an explicit action structure: transitions are now connected to actions that generate them.

Definition 2 (Markov decision process). *A* Markov decision process *over domain $D = \langle U, \top, \bot, ^{-}\rangle$, and a set of utility fluents Π is a tuple $\mathcal{M} = \langle St, Act, \tau, \pi\rangle$, where: St, π are like in a Markov chain, Act is a nonempty finite set of actions, and $\tau : St \times Act \times St \to [0, 1]$ is a stochastic transition relation; $\tau(q_1, \alpha, q_2)$ defines the probability that, if the system is in q_1 and the agent executes α, the next state will be q_2. For every $q \in St, \alpha \in Act$, we assume that either (1) $\tau(q, \alpha, q') = 0$ for all q' (i.e., α is not enabled in q), or (2) $\tau(q, \alpha, \cdot)$ is a probability distribution.*

Additionally, we define $act(q) = \{\alpha \in Act \mid \exists q'.\ \tau(q, \alpha, q') > 0\}$ as the set of enabled actions in q.

A policy is a conditional plan that specifies future actions of the decision-making agent. Policies can be stochastic as well, thus allowing for randomness in the agent's play.

Definition 3. *A* policy *(or* strategy*) in a Markov decision process $\mathcal{M} = \langle St, Act, \tau, \pi\rangle$ is a function $s : States \times Act \to [0, 1]$ that assigns each state q with a probability distribution over the enabled actions $act(q)$. That is, $s(q, \alpha) \in [0, 1]$ for all $q \in St, \alpha \in act(q)$, and $\sum_{\alpha\in act(q)} s(q, \alpha) = 1$. Values of $s(q, \alpha)$ for $\alpha \notin act(q)$ are irrelevant.*

Policy s is pure *iff for each state q it specifies a single action α (i.e., $s(q, \alpha) = 1$, and $s(q, \alpha') = 0$ for all the other α'). By abuse of notation, we will sometimes write $s(q) = \alpha$ instead of $s(q, \alpha) = 1$ for pure policies.*

The set of all policies in M is denoted by Σ_M. The set of all pure policies in M is denoted by σ_M.

[2]If the model is clear from the context, the subscripts will be omitted.

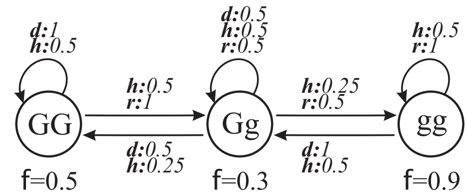

Figure 4: Markov decision process that allows for various mating policies

Note that, if the agent's policy is fixed, a Markov decision process reduces to a Markov chain.

Definition 4. *Policy $s : States \times Act \to [0, 1]$ instantiates* MDP *$\mathcal{M} = \langle St, Act, \tau, \pi\rangle$ to a Markov chain $\mathcal{M} \dagger s = \langle St', \tau', \pi'\rangle$ with $St' = St$, $\pi' = \pi$, and $\tau'(q, q') = \sum_{\alpha\in act(q)} s(q, \alpha)\ \tau(q, \alpha, q')$.*

Example 2 (Gene model ctd.). *An extension of the "gene model" Markov chain from Example 1 is shown in Figure 4. Now, it is possible to mate the offspring with an animal that has dominant genes (action d), recessive genes (action r), or hybrid genes (action h). Note that the pure policy $s(GG) = s(Gg) = s(gg) = h$ instantiates the* MDP *to the Markov chain from Figure 3.*

4. MTL$_0$: A LOGIC OF MARKOV CHAINS

In this section we present our first take on Markov Temporal Logic (MTL), a logic that allows for flexible reasoning about outcomes of agents acting in stochastic environments. The core of the logic is called MTL$_0$, and addresses outcomes of Markov chains. Intuitively, MTL$_0$ is a quantitative analogue of the branching-time logic CTL* [8]; we will formalize (and prove) this claim later, in Section 4.4.

Operators of MTL$_0$ include path quantifiers $\mathsf{E}, \mathsf{A}, \mathsf{M}$ for the maximal, minimal, and average outcome of a set of temporal paths, respectively, and temporal operators $\Diamond, \Box, \mathsf{m}$ for the maximal, minimal, and average outcome along a given path.[3] Propositional operators follow the same pattern. Besides $\vee, \wedge$ for maximization and minimization of outcomes obtained from different utility channels or related to different goals, we use (after [6]) the "weighted average" operator $\oplus$ which will prove useful when we formulate e.g. fixpoint properties of temporal operators with discount. Additionally, we introduce a "defuzzification" operator $\preccurlyeq$; $\varphi_1 \preccurlyeq \varphi_2$ yields "true" if the outcome of φ_1 is less or equal to φ_2, and "false" otherwise. This provides a neat two-valued interface to the logic. Among other advantages, it allows to define the classical computational problems of validity, satisfiability and model checking for MTL.

[3]The temporal operators will allow to discount future outcomes with a discount factor c. Also, we will introduce the "until" operator $\mathcal{U}$, which is more general than $\Diamond$.

4.1 Syntax of MTL$_0$

The syntax of MTL$_0$ (parameterized by a set of utility fluents Π) is defined as follows:

$$\begin{array}{lll}\varphi & ::= & p \mid \neg\varphi \mid \varphi\wedge\varphi \mid \varphi\oplus_c\varphi \mid \varphi\preccurlyeq\varphi \mid \mathsf{E}\gamma \mid \mathsf{M}\gamma,\\ \gamma & ::= & \varphi \mid \neg\gamma \mid \gamma\wedge\gamma \mid \bigcirc_c\gamma \mid \Box_c\gamma \mid \gamma\,\mathcal{U}_c\,\gamma \mid \mathsf{m}_c\gamma.\end{array}$$

where $p \in \Pi$ is a utility fluent, and $c \in (0,1]$ is a discount factor. We will use the symbol $\mathcal{L}state(\Pi)$ to denote the set of "state formulae" φ, and $\mathcal{L}path(\Pi)$ to denote the set of "path formulae" γ.

Additionally, we define the Boolean constants T, F (standing for "true" and "false"), disjunction, and the "sometime" temporal operator $\Diamond$ as below. Except for T, all of them are just standard definitions that can be found in any textbook on temporal logic. We will show in Section 4.2 that their semantics corresponds to our intuition also in this setting.

- $\mathrm{T} \equiv p \preccurlyeq p$,
- $\mathrm{F} \equiv \neg\mathrm{T}$,
- $\varphi_1 \vee \varphi_2 \equiv \neg(\neg\varphi_1 \wedge \neg\varphi_2)$,
- $\mathsf{A}\gamma \equiv \neg\mathsf{E}\neg\gamma$,
- $\gamma_1 \vee \gamma_2 \equiv \neg(\neg\gamma_1 \wedge \neg\gamma_2)$,
- $\Diamond_c\gamma \equiv \mathrm{T}\,\mathcal{U}_c\,\gamma$,
- $\varphi_1 \cong \varphi_2 \equiv (\varphi_1 \preccurlyeq \varphi_2) \wedge (\varphi_2 \preccurlyeq \varphi_1)$.

We may also use the following shorthands for discount-free versions of temporal operators: $\bigcirc \equiv \bigcirc_1$, $\Diamond \equiv \Diamond_1$, $\Box \equiv \Box_1$, $\mathcal{U} \equiv \mathcal{U}_1$.

EXAMPLE 3. *The following* MTL$_0$ *formulae define some interesting characteristics of the breeding process from Example 1:* $\mathsf{Mm}_{0.9}\mathsf{f}$ *(expected average fitness with time discount* 0.9*),* $\mathsf{Am}_{0.9}\mathsf{f}$ *(guaranteed average fitness with the same discount factor),* $\mathsf{M}\Box\mathsf{f}$ *(expected minimal undiscounted fitness), and* $\mathsf{A}\Diamond\mathsf{f}$ *(guaranteed maximal fitness).*

4.2 Semantics of MTL$_0$

The main idea behind MTL$_0$ is to treat formulae in a sufficiently general way, so that they can represent both quantitative utilities and qualitative truth values referring to something which is *completely* true or false, like a task that has been completely achieved. Besides advantages in terms of modeling, this allows to freely mix qualitative and quantitative properties, which (hopefully) makes the resulting semantics elegant and powerful. Thus, we are going to treat complex formulae as fluents, just like the atomic utility fluents from Π, through a valuation function that assigns formulae with extended utility values from $\hat{U}$.

Let $M = \langle St, \tau, \pi\rangle$ be a Markov chain over domain $D = \langle U, \top, \bot, ^{-}\rangle$ and a set of utility fluents Π. The truth value of formulae in M is determined by the valuation function $[\cdot] : (St \times \mathcal{L}state(\Pi)) \cup (\mathcal{R} \times \mathcal{L}path(\Pi)) \to \hat{U}$, defined below. We will omit M in $[\cdot]_{M,q}, [\cdot]_{M,\lambda}$ when the model is clear from the context.

- $[p]_q = \pi(p, q)$, for $p \in \Pi$;
- $[\neg\varphi]_q = \overline{[\varphi]_q}$;
- $[\varphi_1 \wedge \varphi_2]_q = \min([\varphi_1]_q, [\varphi_2]_q)$;
- $[\varphi_1 \oplus_c \varphi_2]_q = (1-c)\cdot[\varphi_1]_q + c\cdot[\varphi_2]_q$;
- $[\varphi_1 \preccurlyeq \varphi_2]_q = \top$ if $[\varphi_1]_q \le [\varphi_2]_q$ and $\bot$ otherwise;
- $[\varphi]_{M,\lambda} = [\varphi]_{M,\lambda[0]}$.
- $[\neg\gamma]_\lambda = \overline{[\gamma]_\lambda}$;
- $[\gamma_1 \wedge \gamma_2]_\lambda = \min([\gamma_1]_\lambda, [\gamma_2]_\lambda)$;
- $[\bigcirc_c\gamma]_\lambda = c\cdot[\gamma]_{\lambda[1..\infty]}$;
- $[\Box_c\gamma]_{M,\lambda} = \inf_{i=0,1,\dots}\{c^i[\gamma]_{M,\lambda[i..\infty]}\}$;
- $[\gamma_1\,\mathcal{U}_c\,\gamma_2]_\lambda =$ $\sup_{i=0,1,\dots}\{\min(\ \min_{0\le j<i}\{c^j[\gamma_1]_{\lambda[j..\infty]}\},\ c^i[\gamma_2]_{\lambda[i..\infty]})\}$;
- The Markovian temporal operator m_c produces the average discounted reward along the given run:

$$[\mathsf{m}_c\gamma]_\lambda = \begin{cases} (1-c)\sum_{i=0}^{\infty} c^i[\gamma]_{\lambda[i\ldots\infty]} & \text{if } c<1\\ \lim_{i\to\infty}\frac{1}{i+1}\sum_{j=0}^{i}[\gamma]_{\lambda[i\ldots\infty]} & \text{if } c=1\end{cases}$$

- $[\mathsf{E}\gamma]_q = \sup\{[\gamma]_\lambda \mid \lambda \in \mathcal{R}(q)\}$;
- The Markovian path quantifier $\mathsf{M}\gamma$ produces the expected truth value γ across all the possible runs (from now on). Given M, q, we first define the probability space $\langle\mathcal{R}(q), \mathcal{H}(q), pr\rangle$ induced by the next-state transition probabilities τ (cf. also [6, 16]). In this space, elementary outcomes are runs from $\mathcal{R}(q)$, events are sets of runs that share the same finite prefix (i.e., ones from $\mathcal{H}(q)$), and the probability measure $pr : \mathcal{H}(q) \to [0,1]$ is defined as $pr(q_0 \ldots q_1) = \tau(q_0, q_1)\cdot\ldots\cdot\tau(q_{i-1}, q_i)$. Then, we use the valuation of γ as the random variable; the truth value of $\mathsf{M}\gamma$ is defined as its expected value:

$$[\mathsf{M}\gamma]_q = \lim_{k\to\infty}\sum_{h\in\mathcal{H}^k(q)}[\gamma]_{\lambda(h)}\tau(h[0],h[1])\cdot\ldots\cdot\tau(h[k-1],h[k]).$$

EXAMPLE 4. *The valuations of the* MTL$_0$ *formulae from Example 3 for the breeding process from Figure 3 are as follows.* $[\mathsf{Mm}_{0.9}\mathsf{f}]_{GG} = 0.484$*,* $[\mathsf{Mm}_{0.9}\mathsf{f}]_{Gg} = 0.480$*, and* $[\mathsf{Mm}_{0.9}\mathsf{f}]_{gg} = 0.554$*; i.e., the expected average fitness with time discount* 0.9 *is* 0.484, 0.480, 0.554 *if we start with dominant, hybrid, and recessive genes, respectively. Moreover,* $[\mathsf{Am}_{0.9}\mathsf{f}]_{GG} = 0.32$*,* $[\mathsf{Am}_{0.9}\mathsf{f}]_{Gg} = 0.3$*, and* $[\mathsf{Am}_{0.9}\mathsf{f}]_{gg} = 0.36$*: the guaranteed average fitness (with discount) is* 0.32*,* 0.3*, and* 0.36*, respectively. Finally, the expected minimal undiscounted fitness* $[\mathsf{M}\Box\mathsf{f}]_q = 0.3$ *for all states* q*, and the guaranteed maximal fitness* $[\mathsf{A}\Diamond\mathsf{f}]_q = 0.3$ *for all states* q*.*

PROPOSITION 1. *We note that the derived operators have the following semantic characteristics:*[4]

1. $[\mathrm{T}]_{M,q} = \top$ *for every* M, q*;*
2. $[\mathrm{F}]_{M,q} = \bot$ *for every* M, q*;*
3. $[\varphi_1 \vee \varphi_2]_{M,q} = \max([\varphi_1]_{M,q}, [\varphi_2]_{M,q})$*;*
4. $[\gamma_1 \vee \gamma_2]_{M,\lambda} = \max([\gamma_1]_{M,\lambda}, [\gamma_2]_{M,\lambda})$*;*
5. $[\mathsf{A}\gamma]_{M,q} = \inf\{[\gamma]_{M,\lambda} \mid \lambda \in \mathcal{R}(q)\}$*;*
6. $[\Diamond_c\gamma]_{M,\lambda} = \sup_{i=0,1,\dots}\{c^i[\gamma]_{M,\lambda[i..\infty]}\}$*;*
7. $[\varphi_1 \cong \varphi_2]_{M,q} = \top$ *if* $[\varphi_1]_{M,q} = [\varphi_2]_{M,q}$*, and* $\bot$ *otherwise.*

The undiscounted versions of temporal operators "always" and "sometime" have the usual relationship, but it does not transfer to the discounted case. Moreover, discounted "always" is trivial for many domains.

[4]Proofs of propositions (omitted here due to lack of space) can be found in the technical report [15].

PROPOSITION 2. *1.* $[\Box\gamma]_{M,\lambda} = [\neg\Diamond\neg\gamma]_{M,\lambda}$,

2. $[\Box_c\gamma]_{M,\lambda} = 0$ *if* $c < 1$ *and* $\hat{U} \subseteq \mathbb{R}_+ \cup \{0\}$.

4.3 Levels of Truth

Since every domain must include a distinguished value for the classical (complete) truth, validity of formulae can be defined in a straightforward way.

DEFINITION 5 (LEVELS OF VALIDITY). *Let M be a Markov chain, q a state in M, and φ a formula of* MTL_0. *Then:*

- φ *is* true in M, q *(written $M, q \models \varphi$) iff* $[\varphi]_{M,q} = \top$.
- φ *is* valid in M *(written $M \models \varphi$) iff it is true in every state of M.*
- φ *is* valid for Markov chains *(written $\models \varphi$) iff it is valid in every Markov chain M.*

EXAMPLE 5. *Let M be the Markov chain from Figure 3 with additional utility fluents* $\mathsf{0.3}$, $\mathsf{0.32}$ *and* $\mathsf{0.36}$ *such that* $\pi(\mathsf{0.3}, q) = 0.3$, $\pi(\mathsf{0.32}, q) = 0.32$, *and* $\pi(\mathsf{0.36}, q) = 0.36$ *for all* $q \in St$. *Then, we have that* $M, GG \models \mathsf{Am}_{0.9}\mathsf{f} \cong \mathsf{0.32}$, $M, Gg \models \mathsf{Am}_{0.9}\mathsf{f} \cong \mathsf{0.3}$, *and* $M, gg \models \mathsf{Am}_{0.9}\mathsf{f} \cong \mathsf{0.36}$. *Moreover, the following formula is valid in M:* $M \models \mathsf{0.3} \preccurlyeq \mathsf{Am}_{0.9}\mathsf{f} \wedge \mathsf{Am}_{0.9}\mathsf{f} \preccurlyeq \mathsf{0.36}$.

Note that T is valid for Markov chains, while F is true in no M, q. Other examples of validities are: $\mathsf{A}\Box\gamma \cong \mathsf{A}\neg\Diamond\neg\gamma$, $\mathsf{E}\Box\gamma \cong \mathsf{E}\neg\Diamond\neg\gamma$ etc. (cf. Proposition 2.1).

Definition 5 enables the traditional view of MTL_0 that identifies "the logic" with the set of valid formulae of that logic. Moreover, it allows to define the typical decision problems for MTL_0 in a natural way:

- Given a formula φ, the *validity problem* asks if $\models \varphi$;
- Given a formula φ, the *satisfiability problem* asks if there are M, q such that $M, q \models \varphi$;
- Given a model M, state q and formula φ, the *model checking problem* asks if $M, q \models \varphi$. Alternatively, the output of model checking can be defined as the value of $[\varphi]_{M,q}$.

An important corollary of Proposition 1.7 is that the notion of equivalence defined by $\cong$ is strong enough to make equivalent formulae interchangeable on all levels of validity.

COROLLARY 3. *If* $M, q \models \varphi_1 \cong \varphi_2$, *and* ψ' *is obtained from* ψ *through replacing an occurrence of* φ_1 *by* φ_2, *then* $M, q \models \psi$ *iff* $M, q \models \psi'$.

4.4 Transition Systems as Markov Chains. Correspondence between MTL$_0$ and CTL*

Markov chains can be seen as generalizations of transition systems, where quantitative information is added via non-classical values of atomic statements and probabilities of transitions. As action labels are absent in Markov chains, these in fact generalize *unlabeled* transition systems (UTS). In this section, we redefine UTS as a proper subclass of Markov chains, in which all the fluents can assume only classical truth values.

DEFINITION 6. *Let M be a Markov chain. Formula φ is* propositional *in M iff it can take only the values of* $\top, \bot$, *i.e.,* $[\varphi]_{M,q} \in \{\top, \bot\}$ *for all* $q \in St$.

Propositions have a simple characterization for Markov chains.

PROPOSITION 4. *Let M be a Markov chain and φ a formula of* MTL_0. *Then φ is propositional in M iff formula* $(\varphi \cong \mathrm{F}) \vee (\varphi \cong \mathrm{T})$ *is valid in M.*

An *unlabeled transition system* can be defined as a Markov chain with only propositional fluents. This way, we obtain the class of models that are used for qualitative branching-time logics, i.e. CTL and CTL*. Of course, when interpreting formulae of CTL*, one must also ignore the probabilities that are present in Markov chains. The next two propositions show that MTL_0 strictly generalizes CTL*.

PROPOSITION 5. *Let M be a transition system, and φ a formula of* CTL*. *Then,* $M, q \models_{\text{MTL}_0} \varphi$ *iff* $M, q \models_{\text{CTL}^*} \varphi$.

PROPOSITION 6. *There is a transition system M with states q, q' which cannot be distinguished by any* CTL* *formula, and can be distinguished by a formula of* MTL_0.

4.5 State-Based MTL$_0$

"CTL without star" (or "vanilla CTL") is the most often used variant of computation tree logic, mainly due to the complexity of its model checking problem and the fact that its semantics can be defined entirely in relation to states. "Vanilla" CTL can be seen as a syntactic restriction of CTL*, in which every temporal modality is preceded by exactly one path quantifier. In this section, we consider a similar syntactic restriction on MTL_0; we call it state-based MTL_0.

DEFINITION 7. State-based MTL_0 *(sMTL_0 in short) is given by the following grammar (where $p \in \Pi$ stands for utility fluents, and $c \in (0, 1]$ for discount factors):*

$$\varphi ::= p \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \oplus_c \varphi \mid \varphi \preccurlyeq \varphi \mid \mathsf{E}\gamma \mid \mathsf{A}\gamma \mid \mathsf{M}\gamma,$$
$$\gamma ::= \bigcirc_c \varphi \mid \Box_c\varphi \mid \varphi\,\mathcal{U}_c\,\varphi \mid \mathsf{m}_c\varphi.$$

Lemma 7 shows that $\mathsf{M}\bigcirc_c \varphi$ implements the discounted expected value of φ in the next moment. Proposition 8 presents fixpoint characterizations for most modalities of sMTL_0. The results from [6] suggest that $\mathsf{M}\Box_c$ and $\mathsf{M}\,\mathcal{U}_c$ do not have fixpoint characterizations, but this remains to be formally proven.

LEMMA 7. *Let φ be a formula of* sMTL_0. *Then,* $[\mathsf{M}\bigcirc_c \varphi]_q = c\sum_{q' \in \tau(q)} [\varphi]_{q'} \tau(q, q')$.

PROPOSITION 8. *The following formulae of* sMTL_0 *are valid:*

- $\mathsf{E}\varphi_1\,\mathcal{U}_c\,\varphi_2 \cong \varphi_2 \vee \varphi_1 \wedge \mathsf{E}\bigcirc_c \mathsf{E}\varphi_1\,\mathcal{U}_c\,\varphi_2$;
- $\mathsf{A}\varphi_1\,\mathcal{U}_c\,\varphi_2 \cong \varphi_2 \vee \varphi_1 \wedge \mathsf{A}\bigcirc_c \mathsf{A}\varphi_1\,\mathcal{U}_c\,\varphi_2$;
- $\mathsf{E}\Box_c\varphi \cong \varphi \wedge \mathsf{E}\bigcirc_c \mathsf{E}\Box_c\varphi$;
- $\mathsf{A}\Box_c\varphi \cong \varphi \wedge \mathsf{A}\bigcirc_c \mathsf{A}\Box_c\varphi$;
- $\mathsf{Em}_c\varphi \cong \varphi \oplus_c \mathsf{E}\bigcirc \mathsf{Em}_c\varphi$;
- $\mathsf{Am}_c\varphi \cong \varphi \oplus_c \mathsf{A}\bigcirc \mathsf{Am}_c\varphi$;
- $\mathsf{Mm}_c\varphi \cong \varphi \oplus_c \mathsf{M}\bigcirc \mathsf{Mm}_c\varphi$.

The above characterizations enable computing the truth values of most sMTL_0 formulae by solving sets of simple equations.

EXAMPLE 6. *The valuations of formula* $\mathsf{Am}_{0.9}\mathsf{f}$ *for states* GG, Gg, gg *of the "gene model" Markov chain can be derived from the following equations:*

$$
\begin{aligned}
[\mathsf{Am}_{0.9}\mathsf{f}]_{GG} &= 0.1 \cdot 0.5 + 0.9 \min([\mathsf{Am}_{0.9}\mathsf{f}]_{GG}, [\mathsf{Am}_{0.9}\mathsf{f}]_{Gg}),\\
[\mathsf{Am}_{0.9}\mathsf{f}]_{Gg} &= 0.1 \cdot 0.3 + 0.9 \min([\mathsf{Am}_{0.9}\mathsf{f}]_{GG}, [\mathsf{Am}_{0.9}\mathsf{f}]_{Gg},\\
&\qquad [\mathsf{Am}_{0.9}\mathsf{f}]_{gg}),\\
[\mathsf{Am}_{0.9}\mathsf{f}]_{gg} &= 0.1 \cdot 0.9 + 0.9 \min([\mathsf{Am}_{0.9}\mathsf{f}]_{Gg}, [\mathsf{Am}_{0.9}\mathsf{f}]_{gg}).
\end{aligned}
$$

5. MTL$_1$: A LOGIC OF MARKOV DECISION PROCESSES

The main aim of this paper is to offer a systematic study of temporal operators for Markov chains; the study was presented in the previous section. This section briefly shows how MTL$_0$ can be extended to strategic reasoning about Markov decision processes. We propose to use an explicit strategic quantifier $\langle\langle a\rangle\rangle$, similar to the *cooperation modality* from alternating-time temporal logic ATL. The intuitive meaning of $\langle\langle a\rangle\rangle\,\varphi$ is "the most that the decision maker can make out of φ". Note that there is always only one agent behind an MDP, so putting his name (e.g., "a") inside the operator is superfluous – but it will make the framework easier to extend to the multi-agent case in the future.

5.1 Syntax and Semantics of MTL$_1$

The syntax of MTL$_1$ is given by the following grammar:

$$
\begin{aligned}
\vartheta &::= p \mid \neg\vartheta \mid \vartheta \wedge \vartheta \mid \vartheta \oplus_c \vartheta \mid \vartheta \preccurlyeq \vartheta \mid \langle\langle a\rangle\rangle\,\varphi,\\
\varphi &::= \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \oplus_c \varphi \mid \mathsf{E}\gamma \mid \mathsf{M}\gamma,\\
\gamma &::= \vartheta \mid \neg\gamma \mid \gamma \wedge \gamma \mid \bigcirc_c \gamma \mid \Box_c \gamma \mid \gamma\, \mathcal{U}_c\, \gamma \mid \mathsf{m}_c\gamma.
\end{aligned}
$$

Note that a is just a fixed symbol and not a parameter of the strategic operator.

Let $\mathcal{M} = \langle St, Act, \tau, \pi\rangle$ be a Markov decision process over domain $D = \langle U, \top, \bot, \overline{\ \cdot\ }\rangle$ and a set of utility fluents Π. The truth value of formulae in M is determined by the valuation function $[\cdot]$ that extends the valuation of MTL$_0$ formulae from Section 4.2 as follows:

- $[p]_{\mathcal{M},q} = \pi(p, q)$, for $p \in \Pi$;
- $[\neg\vartheta]_{\mathcal{M},q} = \overline{[\vartheta]_{\mathcal{M},q}}$;
- $[\vartheta_1 \wedge \vartheta_2]_{\mathcal{M},q} = \min([\vartheta_1]_q, [\vartheta_2]_{\mathcal{M},q})$;
- $[\vartheta_1 \oplus_c \vartheta_2]_{\mathcal{M},q} = (1-c)\cdot[\vartheta_1]_{\mathcal{M},q} + c\cdot[\vartheta_2]_{\mathcal{M},q}$;
- $[\vartheta_1 \preccurlyeq \vartheta_2]_{\mathcal{M},q} = \top$ if $[\vartheta_1]_{\mathcal{M},q} \le [\vartheta_2]_{\mathcal{M},q}$ and $\bot$ otherwise;
- $[\langle\langle a\rangle\rangle\,\varphi]_{\mathcal{M},q} = \sup\{[\varphi]_{M\dagger s,q} \mid s \in \Sigma_M\}$;
- $[\vartheta]_{\mathcal{M}\dagger s,\lambda} = [\vartheta]_{\mathcal{M},\lambda[0]}$.

We use the same definitions of derived Boolean and temporal operators as in Section 4.1. Additionally, we define $\vartheta_1 \cong \vartheta_2 \equiv \vartheta_1 \preccurlyeq \vartheta_2 \wedge \vartheta_2 \preccurlyeq \vartheta_1$, and $[\![a]\!]\varphi \equiv \neg\langle\langle a\rangle\rangle\neg\varphi$. The following proposition shows that $[\![a]\!]\varphi$ implements the outcome of the worst possible policy with respect to φ.

PROPOSITION 9. $[[\![a]\!]\varphi]_{\mathcal{M},q} = \inf_{s\in\Sigma_{\mathcal{M}}}\{[\varphi]_{\mathcal{M}\dagger s,q}\}$.

EXAMPLE 7. *Let* $\mathcal{M}$ *be the "gene model"* MDP *from Figure 4. Then, we have e.g.* $[\langle\langle a\rangle\rangle\,\mathsf{Mm}_{0.9}\mathsf{f}]_{GG} = 0.762$, $[\langle\langle a\rangle\rangle\,\mathsf{Mm}_{0.9}\mathsf{f}]_{Gg} = 0.791$, *and* $[\langle\langle a\rangle\rangle\,\mathsf{Mm}_{0.9}\mathsf{f}]_{gg} = 0.9$. *Indeed, using only individuals with recessive genes for mating is the best policy when we want to maximize the expected average fitness discounted with* 0.9.

On the other hand, mating with hybrids proves best if we want to minimize *the expected average fitness (with discount* 0.9*) from state GG on; for states Gg and gg, mating with dominant genes gives the worst expectancy:* $[[\![a]\!]\mathsf{Mm}_{0.9}\mathsf{f}]_{GG} = 0.484$, $[[\![a]\!]\mathsf{Mm}_{0.9}\mathsf{f}]_{Gg} = 0.464$, *and* $[[\![a]\!]\mathsf{Mm}_{0.9}\mathsf{f}]_{gg} = 0.507$.

We observe that various levels of satisfaction and validity of MTL$_1$ formulae (and thus also the typical computational problems) can be defined analogously to Section 4.3.

The semantic definition od $\langle\langle a\rangle\rangle$ refers to the set of all stochastic policies Σ, which suggests that looking for the best policy can be quite a complex task. Is it possible to restrict the search to pure policies only? Unfortunately, it turns out that it is not the case in general. However, we conjecture that an analogous property should hold for the "state-based" fragment of MTL$_1$.

PROPOSITION 10. *Let* $\vartheta \equiv \langle\langle a\rangle\rangle\,\varphi$ *be a formula of* MTL$_1$. *Then, equation* $[\langle\langle a\rangle\rangle\,\varphi]_{\mathcal{M},q} = \sup_{s\in\sigma_{\mathcal{M}}}\{[\varphi]_{\mathcal{M}\dagger s,q}\}$ *does* not *hold. It does not even hold for* labeled transition systems*, i.e., Markov decision processes where all the utility fluents take only classical truth values* $\top, \bot$.

CONJECTURE 11. *Let* $\vartheta \equiv \langle\langle a\rangle\rangle\,\varphi$ *be a formula of* MTL$_1$ *in which every temporal operator is immediately preceded by exactly one path quantifier, and every path quantifier is immediately preceded by exactly one strategic operator. Then:* $[\langle\langle a\rangle\rangle\,\varphi]_{\mathcal{M},q} = \sup_{s\in\sigma_{\mathcal{M}}}\{[\varphi]_{\mathcal{M}\dagger s,q}\}$.

5.2 Beyond MDP: the Multi-Agent Case

In the more general case, a system can include multiple agents/processes, interacting with each other. Here, we only briefly discuss how Markov temporal logic can be extended to handle such interaction.

On the language level, we propose to extend the strategic operator $\langle\langle a\rangle\rangle$ to a family of operators $\langle\langle A\rangle\rangle$, parameterized with groups of agents A. Intuitively $\langle\langle A\rangle\rangle\,\varphi$ refers to how much agents A can "make out of" φ by following their best joint policy. This would yield a language similar to the alternating-time temporal logic ATL* from [1], albeit with strategic operators separated from path quantifiers.

On the semantic level, multi-agent Markov decision processes [5] can be used as models. The semantics $\langle\langle A\rangle\rangle\,\varphi$ should be of course based on the maximal value of φ with respect to A's joint strategies. However, it is not entirely clear how *the other agents'* actions should be fixed in order to instantiate the MMDP to a Markov chain. One option is to assume that the opponents play a strategy that minimizes φ best. This way, operator $\langle\langle A\rangle\rangle$ would correspond to the maxmin of the two-player game where A is the (collective) maximizer, and the rest of agents fills in the role of the (collective) minimizer. Still, such a semantics would entail a very strong assumption, namely that the opponents of A must also play only *memoryless* strategies.

6. COMPARISON TO DCTL

Markov temporal logic (MTL), proposed in this paper, is in many respects similar to the "Discounted CTL" (DCTL) by de Alfaro and colleagues [6]. This section lists some differences between both logics.

1. In DCTL, the set of truth values is $[0, 1]$. We keep the choice more open: it can be any continuous subset of $\mathbb{R} \cup \{-\infty, +\infty\}$.

2. MTL has more general syntax than DCTL: MTL_0 extends CTL^* and MTL_1 extends the single-agent fragment of ATL^*, while de Alfaro et al.'s DCTL extends only the "vanilla" CTL.
3. E, A are true *path* quantifiers in our framework, in the sense that they refer to "limit properties" of paths. For aggregation of utilities via expected value, we propose a separate path operator M. In contrast, [6] propose a semantics in which both E, A are based on the expected reward. In consequence, neither universal nor existential quantification on paths is expressible in DCTL for models with quantitative transition relations. One peculiar consequence of such approach is that the DCTL's $\mathsf{E}\gamma$ yields the same truth value as $\mathsf{A}\gamma$ for all Markov chains, which is not the case in our framework. Another consequence is that the semantics of path quantifiers in [6] is different for qualitative and quantitative models, which is not the case in our semantics.
4. MTL includes the operator $\preccurlyeq$, which can serve both as a kind of crisp material implication on fuzzy operands, and as a "defuzzification" operator that maps quantitative characteristics to qualitative descriptions.
5. The last feature allows us to define the notions of satisfaction and validity. Thus, standard problems like satisfiability and validity are properly defined in our framework.
6. MTL includes the full "until" operator $\mathcal{U}$, while DCTL includes only "sometime" ($\Diamond$).
7. We propose only the "path semantics" for MTL. We believe it is more appropriate to introduce fixpoint operators rather than to define two different semantics of the same formulae.
8. In contrast to [6], we do not try to capture strategic properties of the decision-making agent with temporal path quantifiers. Instead, we propose to use an explicit strategic quantifier $\langle\langle a \rangle\rangle$.

In essence: we attempt at a more *systematic* exploration of linguistic features that are offered by propositional modal logic for analysis of Markovian models of agents.

7. CONCLUSIONS

Two kinds of models are used in multi-agent systems to represent and reason about behavior of agents/processes: quantitative and qualitative ones. In this paper, we suggest that both traditions are complementary rather than competitive. In fact, we believe that an integration of both approaches may bring a really powerful framework for dealing with multi-agent systems. To this, end, we propose *Markov temporal logic* MTL which can be seen as an extension of "Discounted CTL" from [6]. We show that the simplest version of MTL (for Markov chains) strictly extends the branching-time logic CTL^*, and we discuss some fixpoint properties for a "state-based" subset of the logic. Finally, we discuss how the basic logic can be extended to address strategic abilities of agents in Markov decision processes, in a way similar to ATL^*.

8. REFERENCES

[1] R. Alur, T. A. Henzinger, and O. Kupferman. Alternating-time Temporal Logic. *Journal of the ACM*, 49:672–713, 2002.

[2] A. Aziz, V. Singhal, R. K. Brayton, and A. L. Sangiovanni-Vincentelli. It usually works: The temporal logic of stochastic systems. In *Proceedings of CAV*, volume 939 of *LNCS*, pages 155–165, 1995.

[3] R. Bellman. *Dynamic Programming*. Princeton University Press, 1957.

[4] R. Bellman. A Markovian decision process. *Journal of Mathematics and Mechanics*, 6:679–684, 1957.

[5] C. Boutilier. Sequential optimality and coordination in multiagent systems. In *Proceedings of IJCAI*, pages 478–485, 1999.

[6] L. de Alfaro, M. Faella, T. Henzinger, R. Majumdar, and M. Stoelinga. Model checking discounted temporal properties. *Theoretical Computer Science*, 345:139–170, 2005.

[7] S. Easterbrook and M. Chechik. A framework for multi-valued reasoning over inconsistent viewpoints. In *International Conference on Software Engineering*, pages 411–420, 2001.

[8] E. A. Emerson. Temporal and modal logic. In *Handbook of Theoretical Computer Science*, volume B, pages 995–1072. Elsevier Science Publishers, 1990.

[9] P. Godefroid, M. Huth, and R. Jagadeesan. Abstraction-based model checking using modal transition systems. In *Proceedings of CONCUR*, volume 2154 of *LNCS*, pages 426–440, 2001.

[10] C. M. Grinstead and J. L. Snell. *Introduction to Probability*. Amer Mathematical Society, 1997.

[11] P. Hajek. *Metamathematics of Fuzzy Logic*. Kluwer, 1998.

[12] J. Y. Halpern. A logical approach to reasoning about uncertainty: a tutorial. In *Discourse, Interaction, and Communication*, pages 141–155. Kluwer, 1998.

[13] H. Hansson and B. Jonsson. A logic for reasoning about time and reliability. *Formal Aspects of Computing*, 6(5):512–535, 1994.

[14] J. Hintikka. *Logic, Language Games and Information*. Clarendon Press : Oxford, 1973.

[15] W. Jamroga. Markov temporal logic. Technical Report IfI-07-11, Clausthal University of Technology, 2007.

[16] J. G. Kemeny, L. J. Snell, and A. W. Knapp. *Denumerable Markov Chains*. Van Nostrand, 1966.

[17] B. Konikowska and W. Penczek. Model checking for multi-valued CTL*. In M. Fitting and E. Orlowska, editors, *Multi-Valued Logics*. 1998.

[18] A. Lluch-Lafuente and U. Montanari. Quantitative μ-calculus and CTL based on constraint semirings. *Electr. Notes Theor. Comput. Sci.*, 112:37–59, 2005.

[19] K. Lorenz and P. Lorenzen. *Dialogische Logik*. Darmstadt, 1978.

[20] A. Markov. Rasprostranenie zakona bol'shih chisel na velichiny, zavisyaschie drug ot druga. *Izvestiya Fiziko-matematicheskogo obschestva pri Kazanskom universitete*, 2(15):135–156, 1906.

[21] N. J. Nilsson. Probabilistic logic. *Artificial Intelligence*, 28(1):71–87, 1986.

[22] E. Ruspini, J. Lowrance, and T. Strat. Understanding evidential reasoning. *Artificial Intelligence*, 6(3):401–424, 1992.

[23] L. Zadeh. Fuzzy sets. *Information and Control*, 8(3):338–353, 1965.