LOGICS FOR PROBABILISTIC PROGRAMMING*

Extended Abstract

by
John H. Reif
Aiken Computation Lab.
Harvard University

1. Introduction

This paper introduces a logic for probabilistic programming[+] PROB-DL (for probabilistic dynamic logic; see Section 2 for a formal definition). This logic has "dynamic" modal operators in which programs appear, as in Pratt's [1976] dynamic logic DL. However the programs of PROB-DL contain constructs for probabilistic branching and looping whereas DL is restricted to nondeterministic programs. The formula $\{a\}_{\sigma}p$ of PROB-DL denotes "with measure $\geq\sigma$, formula p holds after executing program a."

Previously, the mathematical semantics of probabilistic nondeterministic programs have been considered by Saheb-Djahrami [1978] and Kozen [1979]. Ramshaw [1979] has provided some Hoar-like axioms for probabilistic programming and applied these in the proofs of various probabilistic programs. Kozen [1979] poses as an open problem to *determine a logic for probabilistic programming with a complete and consistent axiomatization.*

In Section 3, we show that PROB-DL has a complete and consistent axiomatization, (using techniques derived from Parikh's [1978] completeness proof for the propositional dynamic logic). As a corollary, we have a one-exponential algorithm for determining satisfiability in PROB-DL (our algorithm uses an iterative model construction technique similar to that of Pratt [1978 and 1979]).

Section 4 presents a probabilistic quantified boolean logic (PROB-QBF) which also has applications to probabilistic programming. The quantified propositional variables of PROB-QBF are given random truth assignments with certain fixed measures. PROB-QBF also has a "measure-testing" operation for determining the probability that a given formula is satisfied.

[+]Probabilistic programs are considered here to be programs which with fixed input execute by taking probabilistic choices at certain branch points.

We also introduce a related new machine, the μ-TM, which has fixed measures associated with possible next-moves and has "measure-testing" operations for determining the probability of acceptance of subsequent computations. The time and space complexity classes of μ-TMs (with unrestricted "measure tests") are shown to be similar to the time and space complexity classes (respectively) of the alternating Turing Machines of [Chandra, Kozen, Stockmeyer, 1978]. However, restricting the "measure tests" of μ-TMs results in interesting new "polynomial time hierarchies" for μ-TMs apparently unrelated to the "polynomial time hierarchy" of Stockmeyer [1977] for alternating machines. Our "polynomial time hierarchies" for μ-TMs generize the complexity class R of Gill [1974], Miller [1975] and Adleman [1978] and the complexity class #P of Valient [1977] (which allow only one "measure test").

2. The Definition of PROB-DL

We define here only a propositional version of our probabilistic program logic (though in Section 5 we describe how to extend the logic to first order). Also, we restrict the measure domain to $D = \{\text{rationals } \sigma \mid 0 \leq \sigma \leq 1\}$ (the logic's definition can be extended to the case where D is a Banach space as in Kozen [1979]).

2A. Syntax of PROB-DL

Let Σ_0 be the set of *atomic programs*, consisting of symbols A, B,... with distinguished elements *null, no-op* $\in \Sigma_0$. Let Φ_0 be a set of *atomic formulas*, consisting of propositional variables P, Q,... with distinguished elements *true, false* with fixed truth assignments 1, 0, respectively.

We define *probabilistic programs* $\Sigma = a,b,...$ and *formulas* $\Phi = p,q,...$ mutually recursively. From probabilistic programs $a,b \in \Sigma$, formulas $p,q \in \Phi$ and $\Phi \in D$ we may form probabilstic programs:

(i) all atomic programs Σ_0
(ii) a;b ("*sequencing*: execute a followed by b")
(iii) p? ("*test*: does p hold in the current state?")
(iv) σ:a|b ("*probabilistic choice*: execute a with measure σ and execute b with measure $1-\sigma$")

*This research was supported in part by the National Sciences Foundation Grant Number MCS-7921024.

Preprint of paper appearing in 12th Annual ACM Symposium on Theory of Computing, Los Angeles, CA, April 1980, pp. 8-13.

(v) $(\sigma:a)^*$ ("*probabilistic looping*: execute $a;(\sigma:a)^*$ with measure σ and execute *no-op* with measure $1-\sigma$").

We may construct the formulas:

(vi) $\neg p$ ("*logical negation*")
(vii) $p \vee q$ ("*logical disjunction*")
(viii) $\{a\}_\sigma p$ ("*measure test*: with measure $\geq\sigma$, p holds after a"). We require $\sigma > 0$.

Let Σ, Φ be the minimal sets satisfying (i)-(v) and (vi)-(viii), respectively. (Also, we informally extend to logic to allow the usual logical connectives $\wedge$, $\rightarrow$, $\leftrightarrow$).

2B. Semantics of PROB-DL

A *model* of PROB-DL is a structure $\mathcal{M} = (S^{\mathcal{M}}, \mu^{\mathcal{M}}, \models^{\mathcal{M}})$ where:

(i) $S^{\mathcal{M}}$ is a set of *states*
(ii) $\mu^{\mathcal{M}}$: $(\Sigma_0 \times S^{\mathcal{M}} \times S^{\mathcal{M}}) \rightarrow D$ is a partial function providing the measure $\mu(A,s,s')$ of *atomic actions* $(A,s,s') \in \Sigma_0 \times S^{\mathcal{M}} \times S^{\mathcal{M}}$
(iii) $\models^{\mathcal{M}} \subseteq S^{\mathcal{M}} \times \Phi_0$ gives all states at which propositional variables hold.

We require that for all states $s,s' \in S^{\mathcal{M}}$

$\mu(\underline{null},s,s') = 0$
$\mu(\underline{no\text{-}op},s,s') = 1$ if $s = s'$ and 0 else

We also require for each atomic program $A \in \Sigma_0$ that

$$1 \geq \sum_{s,s' \in S^{\mathcal{M}}} \mu^{\mathcal{M}}(A,s,s')$$

We now extend $\mu^{\mathcal{M}}$ to define the measure of all elements of $\Sigma \times S^{\mathcal{M}} \times S^{\mathcal{M}}$. We shall extend $\models^{\mathcal{M}}$ to a subset of $S^{\mathcal{M}} \times \Phi$ defining all states at which formulas hold. We shall drop the superscript $\mathcal{M}$ for a fixed model $\mathcal{M}$ (and occasionally we will write $\mathcal{M},s \models p$ to denote $s \models^{\mathcal{M}} p$).

Consider probabilistic program $a \in \Phi$ and states $s,s' \in S$.

E1 If $a = b_1;b_2$ ("sequencing") then
$$\mu(a,s,s') = \sum_{s'' \in S} \mu(b_1,s,s'') \cdot \mu(b_2,s'',s')$$

E2 If $a = p?$ ("test") then
$$\mu(a,s,s') = 1 \quad \text{if } (s = s' \text{ and } s \models p)$$
$$= 0 \quad \text{else}$$

E3 If $a = (\sigma:b_1|b_2)$ ("probabilistic choice") then
$$\mu(a,s,s') = \sigma \cdot \mu(b_1,s,s') + (1-\sigma)\mu(b_2,s,s')$$

E4 If $a = (\sigma:b)^*$ ("probabilistic looping") then
$$\mu(a,s,s') = \sigma \cdot \mu((b;a),s,s'),s,s') + 1 - \sigma$$

Each $(a,s,s') \in \Phi \times S \times S$ with $\mu(a,s,s') > 0$ is an *action* and has *measure* $\mu(a,s,s')$. We require that $\mu(a,s,s') \leq 1$.

To extend $\models$ we let

E5 $s \models \neg p$ iff $s \not\models p$

E6 $s \models p \vee q$ iff $s \models p$ or $s \models q$

E7 $s \models \{a\}_\sigma p$ iff $\sigma \leq \sum_{s' \in S} \mu(a,s,s')$

Furthermore, the extended μ and $\models$ are required to be the unique minimal fixed points of above equations EQ = (E1,...,E7).

Formula $p \in \Phi$ is *satisfiable* iff $\mathcal{M},s \models p$ for some model $\mathcal{M}$ and state $s \in S^{\mathcal{M}}$. Let p be *valid* if $\neg p$ is not satisfiable. A structure $\mathcal{M} = (S^{\mathcal{M}}, \mu^{\mathcal{M}}, \models^{\mathcal{M}})$ is a *nonstandard model* if $\mathcal{M}$ satisfies all the restrictions for a model except E4 is weakened to

E4' If $a = (a:b)^*$ then
$$\mu(a,s,s') \geq \sigma\mu((b;a),s,s') + 1 - \sigma$$

We now establish an analogue to the Small Model Theorem of [Fischer and Ladner, 1979].

THEOREM 1. *If a formula p_0 of* PROB-DL *is satisfiable, then there is a model of size $O(2^{|p_0|^k})$ which satisfies p_0, for some constant $k > 0$.*

Proof (Sketch). Suppose $\mathcal{M}$ is a (possibly infinite) model satisfying p_0 at state s_0. We shall require a finite set of formulas $c\ell(p_0) \in \Phi$ ("the closure of p_0") defined as in Fischer and Ladner [1979] by introducing new Q-variables associated with subformulas of p_0, so that $|c\ell(p_0)|$ is bounded by a polynomial in $|p_0|$. We show by a structural induction that the measures of actions (a,s,s') of $\mathcal{M}$ can be restricted to rationals of *low precision*:

elements of $\{X/Y \mid 0 < X,Y < 2^{|p_0|^k}\}$

Let $\overline{\mathcal{M}} = (S^{\overline{\mathcal{M}}}, \mu^{\overline{\mathcal{M}}}, \models^{\overline{\mathcal{M}}})$ where

(i) $S^{\overline{\mathcal{M}}} = \{\bar{s} \mid s \in S^{\mathcal{M}}\}$ with
$\bar{s} = \{s' \in S^{\mathcal{M}} \mid \forall q \in c\ell(p_0)\ \ \mathcal{M},s \models q$ iff $\mathcal{M},s' \models q\}$

(ii) $\forall a \in \Sigma,\ s,s' \in S^{\mathcal{M}}$
$$\mu^{\overline{\mathcal{M}}}(a,\bar{s},\bar{s'}) = \sum_{s'' \in S^{\mathcal{M}}} \mu^{\mathcal{M}}(a,s,s'')$$

(iii) $\overline{\mathcal{M}},\bar{s} \models p$ iff $\mathcal{M},s \models p$.

We can now show that p_0 is satisfied at $\bar{s}_0$ in $\overline{\mathcal{M}}$. □

3. Complete and Consistent Axioms for Probabilistic Programming

We provide an axiomization of PROB-DL below (we assume in this section PROB-DL contains no tests of form $p?$).

A1 All tautoligies

A2 $\{a\}_\sigma(p \rightarrow q) \rightarrow (\{a\}_\sigma p \rightarrow \{a\}_\sigma q)$

A3 $\{(\sigma_1:a|b)\}_{\sigma_2} p \leftrightarrow \bigvee_{\sigma_3,\sigma_4:\ \sigma_2 \geq \sigma_1\cdot\sigma_3 + (1-\sigma_1)\cdot\sigma_4} (\{a\}_{\sigma_3} p \wedge \{b\}_{\sigma_4} p)$

(σ_3,σ_4 are restricted to the same order of precision as σ_1,σ_2)

A4 $\{a;b\}_\sigma p \leftrightarrow \{a\}_1(\{b\}_\sigma p)$

A5 $\{(\sigma_1:a)^*\}_{\sigma_2} p \leftrightarrow \{\sigma_1:(a;(\sigma_1:a)^*) \mid \underline{\text{no-op}}\}_{\sigma_2} p$

A6 $p \wedge \{(\sigma:a)^*\}_1 (p \rightarrow \{a\}_1 p) \rightarrow \{(\sigma:a)^*\}_1 p$

For each formula $p \in \Phi$, let $\vdash p$ if p can be derived from the axioms A1-A6 by *rules*:

R1 (modus ponens) if $\vdash p$ and $\vdash p \rightarrow q$ then $\vdash q$

R2 (generization) if $\vdash p$ then $\vdash \{a\}_\sigma p$.

We now sketch a proof of

THEOREM 2. *Formula* p_0 *is valid iff* $\vdash p_0$.

There are three proofs [Segerberg, 1977], [Pratt, 1978], [Parikh, 1978] of the completeness of various axiomizations of the propositional dynamic logic of [Fischer and Ladner, 1978]; the below proof for PROB-DL is an extension of Parikh's completeness proof.

Let a *pseudomodel* be a (possibly infinite) digraph $G = (V,E,L,s_0)$ with

(i) *vertices* V; each $s \in V$ consists of a set of formulas and is considered a *state*
(ii) *edges* $E \subseteq V \times V$
(iii) *edge labeling* L; each edge label has the form $(\sigma:a)$ where $\sigma \in D$ and a is a *semiatomic program* (a is atomic or $a = (\sigma:b)^*$)
(iv) a *fixed root* $s_0 \in V$

Let $s \xrightarrow{\sigma:a} s'$ denote there exists an edge $(s,s') \in E$ with label $(\sigma:a)$ (intuitively, "action (a,s,s') has measure σ").

Let the *describing formula* of G be $p_G = p_{s_0}$ where for each $s \in V$,

$$p_s = \left(\bigwedge_{q \in s} q \right) \wedge \bigwedge_{s \xrightarrow{\sigma:a} s'} \{a\}_\sigma p_{s'} .$$

Let pseudomodel $G = (V,E,L,s_0)$ be *inconsistent* if there exists some state $s \in V$ and formulas $p, \neg p \in s$. Note that if $s \xrightarrow{\sigma:a} s'$ and $\vdash \neg p_{s'}$, then by R2 $\vdash \{a\} \neg p_{s'}$, and we can show $\vdash \neg p_s$.

LEMMA 1. *If* G *is inconsistent, then* $\vdash \neg p_G$.

Let pseudomodel G' be *derived* from pseudomodel $G = (V,E,L,s_0)$ by choosing a state $s \in V$ and applying one of the following modifications:

M1 add an axiom to s

M2 if $p, (p \rightarrow q) \in s$ and if $q \notin s$ then add q to s (i.e., apply modus ponens)

M3 if $\{a\}_\sigma p \in s$ where a is semiatomic and $s \xrightarrow{\sigma:a} s'$ and $p \notin s'$ then add p to s'

M4 if $\{a\}_\sigma p \in s$ where a is semiatomic and $\sigma' > 0$ (where σ' is $\sigma - \sum_{\forall s'',\sigma'' \text{ with } s \xrightarrow{\sigma'':a} s'' \text{ and } p \in s''} \sigma''$) then add a new state $s' = \{p\}$ and edge (s,s') with label $(\sigma':a)$ (so $s \xrightarrow{\sigma':a} s'$ in G')

LEMMA 2. *If* G' *is derived from* G *and* $\vdash \neg p_G$, *then* $\vdash \neg p_G$.

Recall that a *nonstandard model* $\mathcal{M}$ is similar to a model as defined in Section 2B, except we do not require equation E4 of Section 2B to strictly hold.

LEMMA 3. *Given a formula* $p_0 \in \Phi$, *either* $\vdash \neg p_0$ *or there is a nonstandard model* $\mathcal{M}$ *of* p_0 *which satisfies axiom A6.*

Proof (Sketch). We define an initial pseudomodel G_0 consisting of a single state s_0 and no edges. Let $\mathcal{C}_0 = \{G_0\}$ and for $\ell = 1,2,\ldots$ let $\mathcal{C}_\ell$ be the set of pseudomodels derived by modifications M1-M4 from the pseudomodels of $\mathcal{C}_{\ell-1}$. By König's lemma, either

CASE 1 $\exists \ell_0 \geq 0$ such that all $G \in \mathcal{C}_{\ell_0}$ are inconsistent so by Lemma 2, $\vdash \neg p_G$, or

CASE 2 Else there is no such ℓ_0. Let G_* be the "limit pseudomodel" of $\mathcal{C}_0, \mathcal{C}_1, \ldots$

We now construct the required nonstandard model $\mathcal{M} = (S^{\mathcal{M}}, \mu^{\mathcal{M}}, \models^{\mathcal{M}})$ where

(i) $S^{\mathcal{M}}$ = the vertices of G_*
(ii) for each $s \in S^{\mathcal{M}}$, let $\mathcal{M},s \models p$ iff $p \in s$
(iii) for each propositional variable $A \in \Phi_0$, let

$$\mu^{\mathcal{M}}(A,s,s') = \begin{cases} \sigma & \text{if } s \xrightarrow{\sigma:A} s' \text{ in } G_* \\ 0 & \text{else.} \end{cases}$$

We extend $\mu^{\mathcal{M}}$ to $\Phi \times S^{\mathcal{M}} \times S^{\mathcal{M}}$ by equations E1, E2, E3 of Section 2B, but in the case $a = (\sigma:b)^*$ we have instead of E4 the equation

E4" $\mu(a,s,s') = \text{MIN}(\sigma\mu((a;b),s,s') + 1 - \sigma, \sigma')$

(where σ' is defined by the label $s \xrightarrow{\sigma':a} s'$ if (s,s') is an edge of G_* and else $\sigma' = 0$.)

Given the (possible infinite) nonstandard model $\mathcal{M}$ for formula p_0 of Lemma 3, we now apply Theorem 1 to construct a finite nonstandard model $\bar{\mathcal{M}} = (S^{\bar{\mathcal{M}}}, \mu^{\bar{\mathcal{M}}}, \models^{\bar{\mathcal{M}}})$. Note that $\bar{\mathcal{M}}$ satisfies p_0 at $\bar{s}_0$ but in general is nonstandard (may not satisfy E4 of Section 2B). To construct a *standard model* for p_0 from $\bar{\mathcal{M}}$ we iteratively apply equations: EQ" = (E1,E2,E3,E4",E5,E6,E7) (recomputing the measure of the actions of non-atomic programs) until convergence. Since μ and $\models$ monotonicly decrease on each iteration, we are assured of eventual convergence to minimal fixed points of EQ. Thus we have established Theorem 2.

As a corollary to Theorem 2, we have

COROLLARY 1. *There is a* $O(2^{|p_0|^k})$ *algorithm for testing satisfiability of any formula* p_0 *of* PROB-DL, *for some constant* $k \geq 0$.

Proof (Sketch). We use an iterative construction as in Pratt [1979].

[1] Initially, let the state set S consist of all sets of formulas $s \subseteq cl(p_0)$
   (a) s is not inconsistent (i.e., if $q \in s$ then $\neg q \notin s$)
   (b) s is closed under rules R1, R2 of Section 3
   (c) s is "closed" under the axioms A1-A6 of Section 3 (i.e., if $\{a;b\}_\sigma\, p \in s$ then $\{a\}_1\, \{b\}_\sigma\, p \in s$ as in A4, etc.)

[2] For each atomic program A of p_0 and each $s,s' \in S$ let $\mu(A,s,s')$ be the minimum $\sigma \in D$ such that

$$\{A\}_\sigma, p \in s \text{ and } \sigma' \geq \sigma \text{ implies } p \in s'$$

[3] Determine the measure of non-atomic actions by iteratively computing a minimal fixed point of equations $EQ' = (E1,E2,E3,E4',E5,E6,E7)$.

[4] If there is a state $s \in S$ for which we may apply modifications M3,M4,M5 then delete s and go to step [3].

[5] If p_0 is contained in any state $s \in S$ then return "p_0 is satisfiable," else return "p_0 is unsatisfiable."

By Theorem 1, the measures are rationals with small precision, so only $O(2^{|p_0|^k})$ iterations suffice for convergence in step [3]. □

4. A Probabilistic Quantified Boolean Logic with Measure Tests

We define here a logic PROB-QBF (for probabilistic boolean logic) which has some interesting applications to the theory of probabilistic programming. (Also, in the next section we use the constructs of PROB-QBF to augment the logic of PROB-DL defined in Section 2).

The formulae of PROB-QBF are formed from:

(i) free instances of *propositional variables:* $X,Y,\ldots$
(ii) the usual *logical connectives:* $\neg,\vee,\wedge,\rightarrow,\leftrightarrow$
(iii) *random quantification:* a formula p may have the form:

$$(\underline{\text{CHOOSE}}\ X \in \underline{\text{RANDOM}}(\sigma{:}1|0)\ \ \underline{\text{IN}}\ q)$$

where the bound variable X is set to 0 with measure σ and X is set to 1 with measure $(1-\sigma)$ in a formula q where X occurs free.
(iv) *measure tests:* formula p may have the form $(\text{PROB}(q) \geq \sigma)$; p holds with measure 1 for those instances where q holds with measure $\geq\sigma$, and else p holds with measure 0.

The semantics of a formula p of PROB-QBF can be formally specified by recursively defining the measure of subformulae of p. A formula p is *satisfiable* if p holds with measure 1 for some truth assignment of the free (unbound) variable of p. (Note a logic equivalent to PROB-QBF could be defined by allowing measure tests of the form $(\#q \geq m)$, where $\#(q)$ is the number satisfying instances of formula q for given truth assignments of the bound variables and m is a nonnegative integer.)

To characterize the complexity of PROB-QBF we introduce a new computational model the μ-TM (for measuring Turing Machine).

The μ-TM is essentially a nondeterministic Turing Machine with

(i) *state set* Q containing distinguished *initial state* $q_I \in Q$ and accepting state $q_A \in Q$. In addition, Q contains disjoint measure *testing states* Q_T and *complementing states* Q_N.
(ii) *Tape alphabet* Δ, *input alphabet* $\Sigma \subseteq \Delta$ *blank* and *endmarker symbols* $b,\# \in \Delta - \Sigma$. We have $t > 1$ tapes $1,2,\ldots,t$ with distinguished *input tape* 1, *work tapes* $2,\ldots,t$ and distinguished *measuring* tape 2.
(iii) The *measure of the next moves* of the μ-TM are defined by the function:

$$\mu: (Q \times \Delta^t \times Q \times \Delta^t \times \{\underline{\text{left}},\underline{\text{right}},\underline{\text{same}}\}^t) \longrightarrow D$$

where D are the rationals on the interval $[0,1]$. Thus, $\mu(q,d,q',d',h)$ gives the probability of moving directions $h = (h_1,\ldots h_t) \in \{\underline{\text{left}},\underline{\text{right}},\underline{\text{same}}\}^t$ from state $q \in Q$ after scanning symbols $d = (d_1,\ldots,d_t) \in \Delta^t$ on tapes $1,\ldots,t$ and then taking new state $q' \in Q$ with the tape heads scanning symbols $d' = (d'_1,\ldots,d'_t) \in \Delta^t$ respectively.

We require that for each state $q \in Q$ and $d \in \Delta^t$

$$1 \geq \sum_{\substack{q' \in Q \\ d' \in \Delta^t \\ h \in \{\underline{\text{left}},\underline{\text{right}},\underline{\text{same}}\}^t}} \mu(q,d,q',d',h) \quad .$$

Let a *configuration* C be a sequence containing the current nonblank tape contents, with current state superimposed to designate the positions of the scan heads. Given on *input string* $w \in \Sigma^n$, the *initial configuration* C_I contains initial state q_I with input tape contents designated by $\#q_I w\#$ and all the work tapes blank. We extend μ so that $\mu(C,C')$ gives the measure of a move (one step) from configuration C to configuration C'. The contents of the measuring tape will always be assumed to encode in binary a positive rational.

Let a *computation sequence* be a sequence of moves with nonzero measure from the initial configuration from C_I. Let the *time cost* (on input w) be the length of the maximum computation sequence, if it exists, and let the *space cost* be the maximum number of nonblank cells of any tape on any computation sequence.

We define *acceptance* of the μ-TM by a labeling L of each configuration C; if C contains state q then

(a) let $L(C) = 1$ if $q = q_A$ is *accepting*, else
(b) if $q \in Q_T$ let $L(C) = 1$ if $\mu_C \leq$ the contents of the measuring tape
$= 0$ else
(c) if $q \in Q_N$ let $L(C) = 1 - \mu_C$
(d) else let $L(C) = \mu_C$ where

$$\mu_C = \sum_{\forall C'} \mu(C,C') \cdot L(C'))$$

(Intuitively, μ_C is a weighted sum of the measure of subsequent configurations. In (a) all *accepting* configurations have measure 1, in (b) we compare μ_C with the contents of the measuring tape, in (c) we *complement* the measure of μ_C, and else (d) we take μ_C as the measure of configuration C.) Let L* be a minimal such above labeling; a simple inductive argument shows there is a unique minimal labeling L*. Let the μ-TM *accept* $w \in \Sigma^n$ iff $L^*(C_I) = 1$.

In the following let A-TM denote the alternating Turing machine of [Chandra, Kozen, Stockmeyer, 1978], let N-TM denote a nondeterministic Turing machine, and let D-TM denote a deterministic Turing machine. Let λ-TIME(T(n)) be the class of languages accepted by λ-TMs within time limit T(n), and let λ-SPACE(S(n)) be the class of languages accepted by λ-TMs within space limit S(n), for $\lambda \in \{\mu, A, N, D\}$.

Also, let λ-TIME(poly) = $\bigcup_{k\geq 0} \lambda$-TIME($n^k$) and let λ-SPACE(poly) be $\bigcup_{k\geq 0} \lambda$-SPACE($n^k$). [Chandra, Kozen, and Stockmeyer, 1978] show that A-TIME(poly) is poly-time equivalent to satisfiablity of the quantified boolean logic QBF. Similar techniques can be used to show:

LEMMA 4. μ-TIME(poly) *is poly-time equivalent to* PROB-QBF *satisfiability.*

Note that any A-TM may be considered a μ-TM with the measure of moves restricted to either 0 or 1 and with measure tests restricted so that the measuring tape is always empty (denoting a test if the measure of subsequent configurations are >0). Thus,

$$\text{A-SPACE}(S(n)) \subseteq \mu\text{-SPACE}(S(n))$$

and

$$\text{A-TIME}(T(n)) \subseteq \mu\text{-TIME}(T(n))$$

The labeling L* defining acceptance of a μ-TM with space bound $S(n) \geq \log n$ (time bound $T(n) \geq n$) may easily be shown to be computable by a D-TM in time $c^{S(n)}$ for some $c > 1$ (in space $T(n)^2$, respectively). Thus from lower bound known results of [Chandra, Kozen, and Stockmeyer, 1978] we have

THEOREM 3. *For* $S(n) \geq \log n$, A-SPACE(S(n)) = μ-SPACE(S(n)) = $\bigcup_{c\geq 1}$D-TIME($c^{S(n)}$)

THEOREM 4. *For* $T(n) \geq n$, A-TIME(T(n)) $\subseteq \mu$-TIME(T(n)) $\subseteq$ D-SPACE(T(n)) *and* N-SPACE(T(n)) $\subseteq$ A-TIME($T(n)^2$) $\subseteq \mu$-TIME($T(n)^2$)

Thus

COROLLARY 2. PROB-QBF *satisfiability is poly-time complete in* D-SPACE(poly).

Let a μ^k-TM be a μ-TM restricted to k measure tests on any computation sequence. Note that μ^1-TIME(poly) contains the complexity class R considered by Gill [1974], Miller [1975] and Adleman [1978]. Also the #P class of enumeration problems [Valiant,1977] is contained in μ^1-TIME(poly).

The μ^k-TIME(poly) form an interesting new "polynomial time hierarchy" μ^0-time(poly) $\subseteq$ μ^1-TIME(poly) $\subseteq \ldots$ D-SPACE(poly) apparently unrelated to the "polynomial time hierarchy" of Stockmeyer [1977].

The proof of Lemma 4 is extended to show

THEOREM 5. *For each* $k \geq 0$, μ^k-TIME(poly) *is poly-time equivalent to the satisfiability for* PROB-QBF *formulas with no more than* k *nested "measure tests."*

We may also define a μ_k-TM to be a μ-TM restricted so that on any computation sequence there are no more than k alternations of "measure tests" followed by "measure complementations." These μ_k-TMs have another interesting "polynomial time hierarchy" μ_0-TIME(poly) $\subseteq \mu_1$-TIME(poly) $\subseteq \ldots$ D-SPACE(poly).

5. Further Work

The probabilistic logics introduced in this paper have for simplicity been restricted to boolean variables and measure domains over the rationals on the interval [0,1]. I described here only the most fundamental complexity and completeness results for such propositional probabilistic logics. Some further extensions of this work are:

A The measure domain D may be generalized to a *Banach space* as in Kozen [1979]. For example, the measure domain can be self-referencial as in a Scott-Strachy mathematical semantics.

B The constructs of PROB-DL and μ-QBF may be combined and extended to a *first order dynamic logic of probabilistic programming*, as informally described below:

(i) First-Order Terms: We assume for each $k > 0$, k-addic function signs $f_1^{(k)}, f_2^{(k)}, \ldots$ and also o-addic constant symbols $C_1, C_2, \ldots$ As usual, each model will contain a universe U with the constant symbols interpreted to be elements of U and each k-addic function sign interpreted to be a mapping: $U^k \rightarrow U$. Variables, constant signs and *terms* are formed by recursively composing functions signs to terms.

(ii) We allow *programs* to be constructed from atomic programs by the sequencing, test, probabilistic choice and looping constructs of PROB-DL, and

(iii) *formulas* are constructed from the usual logical connectives, as well as the model "measure tests" of PROB-DL and a generalization of the μ-QBF random quantification:

The formula

$$(\text{CHOOSE } X \in \text{RANDOM}(\sigma_1 : t_1, \sigma_2 : t_2, \ldots, \sigma_r : t_r) \text{ in } q)$$

binds variable X to the interpretation of terms $t_1, t_2, \ldots, t_r$ with measure $\sigma_1, \sigma_2, \ldots, \sigma_r$ respectively.

This first order logic seems sufficiently powerful to be useful in proofs of probabilistic programs such as in [Ramshaw, 1979].

C The *multiprocess logic* of [Peterson and Reif, 1980] may be extended to allow for probabilistic strategies in multiprocess games.

Bibliography

Adleman, L. Two theorems on random polynomial time. Proc. 19th Symp. on Foundations of Computer Science, Ann Arbor, Oct. 1978, 75-83.

Chandra, A.K., D.C. Kozen, and L.J. Stockmeyer. "Alternation," IBM Research Report RC 7489, Yorktown Heights, N.Y., Jan. 1978.

Chandra, A.K., and L.J. Stockmeyer. "Alternation," Proc. 17th Annual Symp. on Foundations of Computer Science, 1976, pp. 98-108.

Fischer, M.J. and Lander, R.E. Propositional Dynamic Logic of Regular Programs, Journal of Computer and System Sciences 18, 194-211 (1979).

Gill, J. Computational complexity of probabilistic Turing machines. Proc. 6th ACM Symp. on Theory of Computing, May 1974, 91-95.

Harel, D., and V.R. Pratt. Nondeterminism in logics of programs, in "Fifth ACM Symp. on Principles of Programming Languages, 1978."

Kozen, D. Semantics of probabilistic programs. Proc. of the 20th Annual IEEE Symp. on Foundations of Comp. Sci., 1979, 101-114.

Kripke, S.A. Sementical analysis of modal logic I: Normal modal propositional calculi. Z. Math. Logik Grundlagen, Math. 9, (1963), 67-96.

Miller, G. Riemann's hypothesis and tests for primality. Proc. 7th ACM Symp. on Theory of Computing, May 1975, 234-239.

Parikh, R. A completeness result for PDL. Symposium on Mathematical Foundations of Computer Science, Zakopane, Warsaw, Sept. 1978.

Pratt, V.R. Semantical considerations on Floyd-Hoare Logic. Proc. 17th Ann. IEEE Symp. on Foundations of Comp. Sci., Oct. 1976, 109-121.

Pratt, V.R. A practical decision method for propositional dynamic logic. Proc. 10th Ann. ACM Symp. on Theory of Computing, San Diego, Calif., May 1978, 326-337.

Reif, J.H. and Peterson, G. "A Dynamic Logic of Multiprocessing with Incomplete Information," 7th Symp. Principles of Programming Languages, Los Vegas, Nevada, Jan. 1980.

Ramshaw, L.H. Formalizing the Analysis of Algorithms. Ph.D. Thesis, Computer Science, Stanford University, June 1979.

Saheb-Djahrami, N. Probabilistic LCF. Fifth ALP, 1978, 442-451.

Segerberg, K. A completeness theorem in the modal logic of programs. Preliminary report. Notices of the AMS, 24, 6, A-552, Oct. 1977.

Stockmeyer, L.J. The polynomial-time hierarchy. Theoretical Computer Science, 3, 1977, 1-22.

Valiant, L.G. The complexity of enumeration and reliability problems. SIAM, J. on Computing (to appear). Also Technical Report of Comp. Science Dept., Edinburgh Univ., Edinburgh, Scotland.