Horn Sentence の評価 とその制御

大谷木 重夫

あらすじ

本論文ではHorn Sentenceの評価をHorn sentenceで制御する方法を提案する。方針としてはPrattのDynamic logicを若干修正したものを制御言語として設定し、それを再度Horn Sentenceに翻訳する方法をとる。表現能力を確めるためにHorn Sentenceのprover他2,3の例に当ってみた。

1.はじめに

宣言文と実行文の橋渡しをするものとしてHorn Sentenceが注目を集めている。ここでその意義について詳しく述べている余裕がないので省略するが、本研究の目的は次の二点にある。

(i) Hardwareの技術進歩とArchitecture上の変革を想定した場合、その評価方式に並列処理機能をもち込む必要がある。

(ii) 人工知能実現の核となる推論機能の強力化のためには豊かな推論制御構造をもち込む必要がある。

Horn Sentence自身はprimitiveな並列評価方法を表明しているが、問題をHorn Sentenceで宣言型で記述し、そのprimitiveな評価方法に基づいて評価したのでは有効な推論は実現し得ない。従って何等かの制御が必要であろう。そこで新たに制御用言語を設定しようという考えが生まれる。本論文ではDynamic logicを採用することとした。その理由は

(i) Horn sentenceの自然な評価からDynamic logicのプログラムが得られることが多い。

(ii) 並列性を明示することができる。

(iii) Debugging programが書ける。

の三点である。

ところでオリジナルな言語とそのControllerが全く別であるというのはしかしながら極めて不便なことである。即ち

(i) Controllerのcontrollerを書くのにどうするか?

(ii) Hardware側から見た場合、2つの異なる言語を同時に受け入れねばなら

ない。

ところで Horn sentence はそれ自体が primitive な並列処理機能を表現しているのであるからその機能を用いれば制御文を Horn sentence で書き、実行をその primitive な並列処理機能にまかせる方が、全体の evaluator を設計する上では簡明である。これが Dynamic logic のプログラムを更に Horn sentence に翻訳する理由である。

もちろんそれなら最初から制御迄含めて Horn sentence で書いた方が簡単ではないかという素朴な疑問も生ずるが、それはかなりわづらわしいものである。

近い将来

「プログラムを書く ＝ Controller を書く」

というテーゼが実現するのではないかというのが私個人の勝手な想定である。

2. Dynamic logic の model としての Horn logic world

ここではまず最も簡単な Dynamic logic として regular first order Dynamic logic を述べ、その model を Horn logic の上に作り上げる。

RG を regular program の全体、RGDL を論理式の全体としてそれを次のように定義する。

〔定義1〕

1) a : atomic function symbol $\Longrightarrow a \in RG$
2) $p \in RGDL \Longrightarrow p? \in RG$
3) $\alpha, \beta \in RG \Longrightarrow \alpha^*,\ \alpha \cup \beta,\ \alpha ; \beta \in RG$
4) p : atomic formula $\Longrightarrow p \in RGDL$
5) $p, q \in RGDL \wedge \alpha \in RG \Longrightarrow p \vee q,\ \exists x p,\ \neg p,\ \langle\alpha\rangle p \in RGDL$

次に RGDL の model 解釈を述べよう。

〔定義2〕

triple $\langle D, m, U \rangle$ が RGDL の model であるとは

1) $D \neq \phi$
2) $U \subset \{ I \mid I ; \{p_\alpha\} \rightarrow 2^D \}$

但し、$a_1, \cdots, a_n$; atomic function symbol

p_α : atomic predicate

とするとき

$m(p?) = \{ (I, I) \mid I \models p \}$

$m(\alpha ; \beta) = m(\alpha) \circ m(\beta)$

$m(\alpha \cup \beta) = m(\alpha) \cup m(\beta)$

$m(\alpha^*) = m(\alpha)^*$

3) p が atomic formula のとき

$I \models p \equiv_{fD} I(p) \equiv 1$ or D

4) $P, Q \in RGDL \wedge \alpha \in RG$ のとき

$I \models \neg p \equiv_{fD}$ $I \models p$ が成立しない

$I \models p \vee q \equiv_{fD} I \models p \vee I \models q$

$I \models p \wedge q \equiv_{fD} I \models p \wedge I \models q$

$I \models \exists x p \equiv_{fD} \exists d \in D\ p(d)$

$$I \models \langle \alpha \rangle P \equiv_{fD} \exists J \ (I, J) \in m(\alpha)$$

次に与えられたHorn sentenceからRGDLのmodelを構成する方法を述べよう。

Horn sentenceの一つの部分系として

$$\begin{cases} P_1(\alpha_{11} x) \wedge \cdots \wedge P_n(\alpha_{1n} x) \Longrightarrow P_1(\beta_{11} x) \wedge \cdots \wedge P_n(\beta_{1n} x) \\ \quad \vdots \\ P_1(\alpha_{m1} x) \wedge \cdots \wedge P_n(\alpha_{mn} x) \Longrightarrow P_1(\beta_{m1} x) \wedge \cdots \wedge P_n(\beta_{mn} x) \\ P_1(\gamma_1 x) \wedge \cdots \wedge P_n(\gamma_n x) \end{cases} \qquad \text{---} \ (\mathrm{I})$$

を仮定する。

これに対し実行の各ステップを識別するために

$$\begin{cases} P_1(s, \alpha_{11} x) \wedge \cdots \wedge P_n(s, \alpha_{1n} x) \Longrightarrow P_1(sa_1, \beta_{11} x) \wedge \cdots \wedge P_n(sa_1, \beta_{1n} x) \\ \quad \vdots \\ P_1(s, \alpha_{11} x) \wedge \cdots \wedge P_n(s, \alpha_{mn} x) \Longrightarrow P_1(sa_m, \beta_{m1} x) \wedge \cdots \wedge P_n(sa_m, \beta_{mn} x) \\ P_1(s_0, \gamma_1 x) \wedge \cdots \wedge P_1(s_0, \gamma_n x) \end{cases} \qquad \text{---} \ (\mathrm{II})$$

なるHorn sentenceを考える。

HをHorn sentence Iに対するHerbrandt worldとして $D = \cup H^n$ と定義しよう。また

$$\hat{S}_0(P_1)(x) \equiv_{fD} x \in \gamma_1(D)$$
$$\vdots$$
$$\hat{S}_0(P_n)(x) \equiv_{fD} x \in \gamma_n(D)$$

とし、

いま η が

$$\eta(P_1)(x) \equiv x \in \delta_1(D)$$
$$\vdots$$
$$\eta(P_n)(x) \equiv x \in \delta_n(D)$$

であるとき

$$\eta \, \hat{a}_j \, (P_1)(x) \equiv_{fD} x \in \beta_{j1} \{ \alpha^{-1}_{j1} \, \overline{\alpha_{j1} \delta_1}(D) \cap \cdots \cap \alpha^{-1}_{jn} \delta_n(D) \}$$
$$\vdots$$
$$\eta \, \hat{a}_j \, (P_n)(x) \equiv_{fD} x \in \beta_{jn} \{ \alpha^{-1}_{j1} \, \overline{\alpha_{j1} \delta_1}(D) \cap \cdots \cap \alpha^{-1}_{jn} \, \overline{\alpha_{jn} \delta_n}(D) \}$$

ここで $\overline{\alpha\beta}$ は α, β の pullback

と定める。

また a_i を atomic function でその意味を $m_s(a_i) = (s, s\hat{a}_i)$ とし.

更にIIの式中に現われるskolem関数は $\hat{S}_0$, $\hat{a}_j$ で通常と同じ解釈を受けるものとする。

$U = \hat{S}_0(\hat{a}_1 + \cdots + \hat{a}_m)^*$ なる解釈空間Uを設定すれば $\langle D, U, \{a_i, \alpha_i\}\{P_i\} \rangle$ はDLのmodelである。

例1. いま

$$P(x, I(x), e) \wedge S(a, y, z)$$
$$P(x, I(y), z) \wedge S(x, y, z) \wedge S(y, x, z)$$
$$\Longrightarrow S(z, x, y) \wedge P(x, I(y), z)$$

なるHorn sentenceをとる。

証明の各ステップを識別するために変数sを新たに導入して

$$P(s_0, x, I(x), e) \wedge S(s_0, a, y, z)$$
$$P(s, x, I(y), z) \wedge S(s, x, y, z) \wedge S(s, y, x, z)$$
$$\Longrightarrow S(sa, z, x, y) \wedge P(sa, xI(y), z)$$
と書き直そう。
$$\gamma_1(x, y, z) = (x, I(x), e)$$
$$\gamma_2(x, y, z) = (a, y, z)$$
$$\alpha_1(x, y, z) = (x, I(y), z)$$
$$\alpha_2(x, y, z) = (x, y, z)$$
$$\alpha_3(x, y, z) = (y, x, z)$$
$$\beta_2(x, y, z) = (z, x, y)$$
$$\beta_4(x, y, z) = (x, I(y), z)$$
とおくと、
$$P(s, \alpha_1(x, y, z)) \wedge S(s, \alpha_2(x, y, z)) \wedge S(s, \alpha_3(x, y, z))$$
$$\Longrightarrow P(sa, \beta_1(x, y, z)) \wedge S(sa, \beta_2(x, y, z))$$
$$P(s_0, \gamma_1(x, y, z)) \wedge S(s_0, \gamma_2(x, y, z))$$
となる。
$$\hat{s}_0(P)(x, y, z) \equiv (x, y, z) \in \gamma_1(D)$$
$$\equiv y = I(x) \wedge z = e$$
$$\hat{s}_0(S)(x, y, z) \equiv (x, y, z) \in \gamma_2(D)$$
$$\equiv x = a$$
$$\hat{s}_0\hat{a}_1(P)(x, y, z) \equiv (x, y, z) \in \beta_1(\alpha_1^{-1}\overline{\alpha_1\gamma_1}(D) \cap \alpha_2^{-1}\overline{\alpha_2\gamma_2}(D) \cap \alpha_3^{-1}\overline{\alpha_3\gamma_2}(D))$$
$$\overline{\alpha_1\gamma_1} = \gamma_1$$
$$(x, y, z) \in \alpha_1^{-1}\alpha_1\gamma_1(D) = \alpha_1^{-1}\gamma_1(D)$$
$$\Longleftrightarrow \alpha_1(x, y, z) \in \gamma_1(D)$$
$$\Longleftrightarrow (x, I(y), z) \in \gamma_1(D)$$
$$\Longleftrightarrow I(y) = I(x) \wedge z = e$$
$$\Longleftrightarrow y = x \wedge z = e$$
$$(x, y, z) \in \beta_1(D)$$
$$\Longleftrightarrow \exists y' \quad y = I(y')$$
従って
$$(x, y, z) \in \beta_1\alpha_1^{-1}, \overline{\alpha_1\gamma_1}(D)$$
$$\Longleftrightarrow y = I(y') \wedge y' = x \wedge z = e$$
$$\Longleftrightarrow y = I(x) \wedge z = e$$
$$\overline{\alpha_2\gamma_2} = \gamma_2$$
$$(x, y, z) \in \alpha_2^{-1}\overline{\alpha_2\gamma_2}(D) = \alpha_2^{-1}\gamma_2(D)$$
$$\Longleftrightarrow \alpha_2(x, y, z) \in \gamma_2(D)$$
$$\Longleftrightarrow (x, y, z) \in \gamma_2(D)$$
$$\Longleftrightarrow x = a$$
従って $(x, y, z) \in \beta_1\alpha_2^{-1}\overline{\alpha_2\gamma_2}(D)$
$$\Longleftrightarrow x = a, \quad y = I(y')$$
$\therefore$ $(x, y, z) \in \beta_1\alpha_2^{-1}\overline{\alpha_2\gamma_2}(D) \cap \beta_1\alpha_2^{-1}\overline{\alpha_2\gamma_2}(D)$
$$\Longleftrightarrow x = a \wedge y = I(a) \wedge z = e$$

$$\overline{\alpha_3 \gamma_2}\,(x, y, z) = (a, a, z)$$

$$(x, y, z) \in \alpha_3^{-1}\ \overline{\alpha_3 \gamma_2}\ (D)$$

$$\iff \alpha_3\,(x, y, z) \in \overline{\alpha_3 \gamma_2}\,(D)$$

$$\iff (y, x, z) \in (a, a, D)$$

$$\iff y = x = a$$

$$(x, y, z) \in \beta_1\ \alpha_3^{-1}\ \overline{\alpha_3 \gamma_2}\ (D)$$

$$\iff {}^{\exists}(x, y', z) \in \alpha_3^{-1}\ \overline{\alpha_3 \gamma_2}\ (D)$$

$$y = I(y')$$

$$\iff x = a,\ \ y = I(a)$$

$$\therefore\ \ \hat{S_0}\,\hat{a}\ (P)\,(x, y, z) \equiv x = a \wedge y = I(a) \wedge z = e$$

また $\hat{S_0}\,\hat{a}\ (S)\,(x, y, z) \equiv (x, y, z) \in \beta_2(\alpha_1^{-1}\overline{\alpha_1\gamma_1}(D) \cap \alpha_2^{-1}\alpha_2\gamma_2(D) \cap \alpha_3^{-1}\overline{\alpha_3\gamma_3}(D))$

であるから $\hat{S_0}\,\hat{a}\ (S)\,(x, y, z) \equiv x = e,\ y = a,\ z = a$

もとの Horn sentence の model で $P(\eta x)$ が成立することは

$$\hat{S_0} \models \langle a^* \rangle\ (\eta_* \Rightarrow P)$$

但し $\eta_*(x) \iff x \in \eta(D)$ とする。

また $S(\eta x)$ が成立することは

$$\hat{S_0} \models \langle a^* \rangle\ ({}^{\forall}x\,\eta_*(x) \Rightarrow {}^{\exists}y, z\ \ S(x, y, z))$$ と書ける。

3. Regular program の Horn sentence への翻訳

ここでは Dynamic logic の Regular program として書かれた文を Horn sentence へ翻訳する方法を述べる。

1) Regular program の生成する実行列が全て primitive operation からなる場合、与えられた正則式 Γ に対し右線型文法をとってきて双方の生成する系列全体が等しくなるようにできる。そこでその右線型文法を

$$A_i \longrightarrow a_{ij}\ A_j \mid a_i \qquad i, j = 1, \cdots, n$$ としよう。

この時

$$\begin{cases} Q(s_0, A) \wedge P_1(s_0, \gamma x) \wedge \cdots \wedge P_n(s_0, \gamma_n x) \\ Q(s, A_i) \Rightarrow \mathrm{Guard}\ (s, a_{ij}) \wedge Q(s a_{ij}, A_j) \\ Q(s, A_i) \Rightarrow \mathrm{Guard}\ (s, a_i) \wedge Q(s a_i, \Lambda) \\ \mathrm{Guard}(s, a_i) \wedge P_1(s, \alpha_{i1} x) \wedge \cdots \wedge P_n(s, \alpha_{in} x) \Rightarrow P_1(s a_i, \beta_{i1} x) \wedge \cdots \wedge P_n(s a_i, \beta_{in} x) \end{cases}$$

$$i, j = 1 \cdots, n \qquad\qquad --- \quad (3.1)$$

なる Horn sentence を作れば

$$\hat{S_0} \models \langle \Gamma \rangle (\eta_* \Rightarrow P) \equiv P(\eta x) \text{ が } (3.1) \text{ の最小モデルで成立}$$

が成り立つ。

例2. $\mathrm{Reader}\ (s, ax) \wedge \mathrm{Buffer}\ (s, y) \Rightarrow \mathrm{Reader}\ (sa_1, x) \wedge \mathrm{Buffer}\ (sa_1, ay)$

$\mathrm{Buffer}\ (s, xb) \Rightarrow \mathrm{Writer}\ (sa_2, b) \wedge \mathrm{Buffer}\ (s, x)$

$\mathrm{Reader}\ (s_0, \gamma) \wedge \mathrm{Buffer}\ (s_0, \Lambda) \wedge \mathrm{Writer}\ (s_0, \Lambda)$

なる Horn sentence が与えられているものとする。

$$(a_1; a_2)^*;\ (\mathrm{Reader}\ (x) \Rightarrow x = 1\ )? \qquad --- \quad (3.2)$$

なる DL のプログラムは読んで書いてという動作をスタックが空になるまで実行するものである。

$$A \longrightarrow a_1 B \qquad B \longrightarrow a_2 A$$

$$A \longrightarrow (\mathrm{Reader}\ (x) \Rightarrow x = \Lambda\ )?$$

とおくと A, B は正則 set を定めるが、A の定める正則 set と (3, 2) の定める正則 set は一致する。従って controll を含む Horn sentence は

$Q(s_0, A) \wedge Reader(s_0, \gamma) \wedge Buffer(s_0, \Lambda) \wedge Writer(s_0, \Lambda)$

$Q(s, A) \Rightarrow Guard(s, a_1) \wedge Q(sa_1, B)$

$Q(s, A) \Rightarrow Test(s, Reader(x) \Rightarrow x = \Lambda) \wedge Q(s, \Lambda)$

$Q(s, B) \Rightarrow Guard(s, a_2) \wedge Q(sa_2, A)$

$Guard(s, a_1) \wedge Reader(s, ax) \wedge Buffer(s, y)$

$\quad \Rightarrow Reader(sa_1, x) \wedge Buffer(sa_1, ay)$

$Guard(s, a_2) \wedge Buffer(s, xb) \Rightarrow Writer(sa_2, b) \wedge Buffer(s, x)$

但し Test 文は $s \models P(x) \equiv_{fD} Test(s, P(x))$

により def される constant predicate のようになる。

2) $\langle \alpha \rangle P?$ 型 sentence の取扱い

$\langle \alpha \rangle P?$ が或る正則式の一部として現われたとするとそれを定義する右線型文法の中では $A \rightarrow \langle \alpha \rangle P?;\ B$ として現われることになる。

α を定義する右線型文法を

$C_i \longrightarrow a_{ij}\ C_j \mid a_i$ とするとき predicate "Stack" を新たに設けて

$Q(s, A) \Rightarrow Stack(s, s) \wedge Q(s, C_0)$

$Q(s, C_i) \wedge Stack(s, t) \Rightarrow Guard(s, a_i) \wedge Stack(sa_i, t) \wedge Q(sa_{ij}, C_j)$

$Q(s, C_i) \wedge Stack(s, t) \Rightarrow Guard(s, a_i) \wedge Stack(sa_i, t) \wedge Q(sa_i, \Lambda)$

$Q(s, \Lambda) \wedge Stack(s, t) \wedge Test(s, P) \Rightarrow Q(t, B)$

なる制御文を付加すればよい。

3) $[\alpha] P?$ 型 sentence の取扱い

1), 2) で一応 Dynamic logic で書いたプログラムを Horn sentence へ翻訳する手続が述べられたわけであるが、ここで問題なのは $[\alpha^*] P?$ 型 sentence である。
$\neg \langle \alpha^* \rangle \neg P$ と解釈したのでは本来 $[\alpha^*] P$ が成立するときにプログラムがとまらない状態となる。

$$(P \Rightarrow [\alpha] P) \Rightarrow (P \Rightarrow [\alpha^*] P)$$

なる定理を用いて積極的に $[\alpha^*] P$ の成立を確めたい場合も生じよう。上記の式を状態 s を用いて書くと

$$\forall s (P \Rightarrow [\alpha] P) \Rightarrow \forall s (P \Rightarrow [\alpha^*] P)$$

である。$\forall s (P \Rightarrow [\alpha] P)$ は $P \Rightarrow [\alpha] P$ なる式の内包的意味である。プログラムの途中でこのような内包的意味を refer するためには新たな内包オペレータ ^ を導入しておく必要を生ずる。すると

$\hat{s} \models P \wedge (P \Rightarrow [\alpha] P)^{\wedge} \Rightarrow \hat{s} \models [\alpha^*] P$ であるから

$[\alpha^*] P? = P \wedge (P \Rightarrow [\alpha] P)^{\wedge}? + \langle \alpha \rangle \neg P?;\ false?;$

として実現できよう。
α が有限であれば $(P \Rightarrow [\alpha] P)^{\wedge}$ は時の状態に依存しない計算可能な述語である。

4. Context-free Dynamic Logic

Theorem prover の evaluator を書こうとすると Regular first order dynamic logic を導入する必要がある。

〔定義3〕

1) a : atomic function symbol $\Rightarrow$ $a \in$ CFPROG

2) $p \in$ CFDL $\Rightarrow$ $p? \in$ CFPROG

3) α : CFPROG上の mutual recursive な方程式の解 $\Rightarrow$ $\alpha \in$ CFPROG

4) p : atomic formula $\Rightarrow$ $p \in$ CFDL

5) $p, q \in$ CFDL , $\alpha \in$ CFPROG

$$\Rightarrow p \vee q,\ \exists x p,\ \neg p,\ \langle \alpha \rangle p \in \text{CFDL}$$

semantics その他については RGDL と同様である。

context-free dynamic logic を定理証明に応用することを考えよう。
いま次のような Horn sentence が与えられているものとしよう。

$$P_1(\alpha_{11} x) \wedge \cdots\cdots \wedge P_n(\alpha_{1n} x) \Rightarrow P_1(\gamma_1 x)$$
$$P_1(\beta_{11} x) \wedge \cdots\cdots \wedge P_n(\beta_{1n} x) \Rightarrow P_1(\varepsilon_1 x)$$
$$\vdots$$
$$P_1(\alpha_{n1} x) \wedge \cdots\cdots \wedge P_n(\alpha_{nn} x) \Rightarrow P_n(\gamma_n x)$$
$$P_1(\beta_{n1} x) \wedge \cdots\cdots \wedge P_n(\beta_{nn} x) \Rightarrow P_n(\varepsilon_n x)$$
$$P_1(\delta_1 x) \wedge \cdots\cdots \wedge P_n(\delta_n x)$$

またこのとき $P_1(\eta x)$を証明するものとしよう。
まず Top-down と Bottom-up を入れ換えるために

$$\left\{\begin{array}{l}
\tilde{P}_1(s_0, \eta x) \\
\tilde{P}_1(s, \gamma_1 x) \Rightarrow \tilde{P}_1(sa_1, \alpha_{11} x) \wedge \cdots\cdots \wedge \tilde{P}_n(sa_1, \alpha_{1n} x) \\
\tilde{P}_1(s, \varepsilon_1 x) \Rightarrow \tilde{P}_1(sb_1, \beta_{11} x) \wedge \cdots\cdots \wedge \tilde{P}_n(sb_1, \beta_{1n} x) \\
\qquad \vdots \\
\tilde{P}_n(s, \gamma_n x) \Rightarrow \tilde{P}_1(sa_n, \alpha_{n1} x) \wedge \cdots\cdots \wedge \tilde{P}_n(sa_n, \alpha_{nn} x) \\
\tilde{P}_n(s, \varepsilon_n x) \Rightarrow \tilde{P}_1(sb_n, \beta_{n1} x) \wedge \cdots\cdots \wedge \tilde{P}_n(sb_n, \beta_{nn} x)
\end{array}\right.$$

なる Clause の集合を考えて

$$\begin{aligned}
A_1 = {} & (\tilde{P}_1 \Rightarrow \delta_1 *)? \\
& + (\tilde{P}_1 \Rightarrow \gamma_1 *)? \ ; a_1 ; [\langle A_1 \rangle \text{true} \wedge \cdots\cdots \wedge \langle A_n \rangle \text{true}]? \\
& + (\tilde{P}_1 \Rightarrow \varepsilon_1 *)? \ ; b_1 ; [\langle A_1 \rangle \text{true} \wedge \cdots\cdots \wedge \langle A_n \rangle \text{true}]? \\
& \quad \vdots \\
A_n = {} & (\tilde{P}_n \Rightarrow \delta_n *)? \\
& + (\tilde{P}_n \Rightarrow \gamma_n *)? \ a_n ; [\langle A_1 \rangle \text{true} \wedge \cdots\cdots \wedge \langle A_n \rangle \text{true}]? \\
& + (\tilde{P}_n \Rightarrow \varepsilon_n *)? \ b_n ; [\langle A_1 \rangle \text{true} \wedge \cdots\cdots \wedge \langle A_n \rangle \text{true}]?
\end{aligned}$$

とおくと $s_0 A_1$ が prover となる。
$\wedge$ − part は sequencing することも出来てその場合は

$$[\langle A_1 \rangle \text{true} \wedge \cdots\cdots \wedge [A_n \rangle \text{true}]?$$

を $\quad \langle A_1 \rangle \text{true}? ; \cdots\cdots ; \langle A_n \rangle \text{true}?$

と入れ換えればよい。

謝辞：本研究の機会を与えてくださいました電子技術総合研究所、制御部長、佐藤孝平氏
並びに適切なご指導、ご討論を頂きました同論理システム研究室長、田村浩一郎氏
及び同研究室の皆様、又、推論機構研究室，横井俊夫 主任研究官，佐藤泰介技官
に感謝致します。

参考文献：D. Harel, "First-Order Dynamic Logic" LNCS 68 Springer 1979