モデルチェッキング

国立情報学研究所

佐藤一郎
E-mail: ichiro@nii.ac.jp

形式的検証技術：方法論とツール

証明支援系　(Proof Checker/Logical Framework)
- 一階・高階述語論理・型理論
- 汎用 (例えば、数学・論理体系自身)
- (全体性質の) 半自動証明 (演繹) ＋対話的証明

状態探索系・モデル検査系　(Modal/Temporal Logic)
- 様相・時相論理 (離散・連続・確率)
- 状態遷移系に特化 (例：同期・分散アルゴリズム)
- 部分性質の全自動証明

時相論理 (Temporal Logic)

状態の遷移や時間の経過の観点より
　システムの性質を記述するための論理体系
線形時間時相論理
　LTL (Linear Time Temporal Logic)
分岐時間時相論理
　CTL (Computation Tree Logic)
　μ計算 (μ-calculus)
モデル検査
　時相論理で表現された性質を
　システムが満たすかどうかを検査すること

分岐時相論理

時相論理(CTL:Computation Tree Logic)

$$\Phi ::= P \mid \neg\Phi \mid \Phi \wedge \Phi \mid \Phi \vee \Phi \mid \Phi \supset \Phi$$
$$\mid AG\ \Phi \mid EG\ \Phi \mid AF\ \Phi \mid EF\ \Phi$$
$$\mid AX\ \Phi \mid EX\ \Phi \mid A[\Phi\ U\ \Phi] \mid E[\Phi\ U\ \Phi]$$

AX　p…along All paths, p holds in the neXt state
EX　p…there Exists a path where p holds in the neXt state
AG　p…along All paths p holds Globally
EG　p…there Exists a path where p holds Globally
AF　p…along All paths p holds at some state in the Future
EF　p…there Exists a path where p holds at some state in the Future
A[p U q]　…along All paths, p holds Until q holds
E[p U q]　…there Exists a path where p holds Until q holds

c.f. CTL*　A Φ and E Φ forms are permitted.

分岐時相論理の定義(AとEを除く)

$K, s \models p$

$K, s \models \neg p \quad \Leftrightarrow \quad K, s \models p$ ではない

$K, s \models p \wedge q \Leftrightarrow K, s \models p$ かつ $K, s \models q$

$K, s \models p \vee q \Leftrightarrow K, s \models p$ または $K, s \models q$

$K, s \models \mathrm{X}p \Leftrightarrow K, N(s) \models p$

$K, s \models \mathrm{G}p \Leftrightarrow$ 任意の $i \geq 0$ に対して $K, N^i(s) \models p$

$K, s \models \mathrm{F}p \Leftrightarrow$ ある $i \geq 0$ に対して $K, N^i(s) \models p$

$K, s \models p\mathrm{U}q \Leftrightarrow$ ある $i \geq 0$ に対して

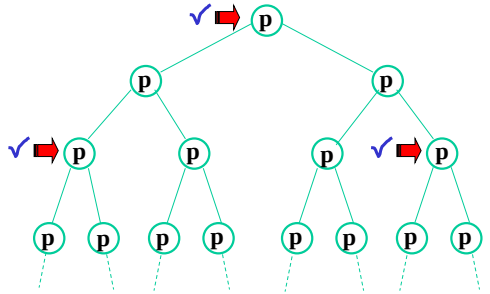
$K, N^i(s) \models p$ かつ $0 \leq j < i$ に対して $K, N^j(s) \models q$

ここで $N(s)$ とは状態 s に対して次の状態をsを与え、$N^i(s)$ とは N の i 回の適用を示す。

Ichiro Satoh

分岐時相論理の解釈

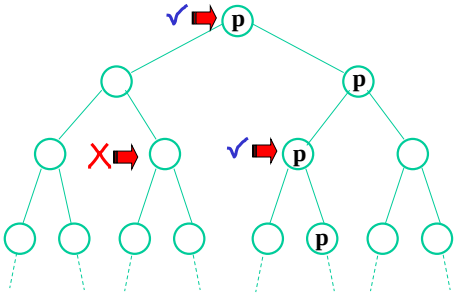
AG p



Ichiro Satoh

分岐時相論理の解釈

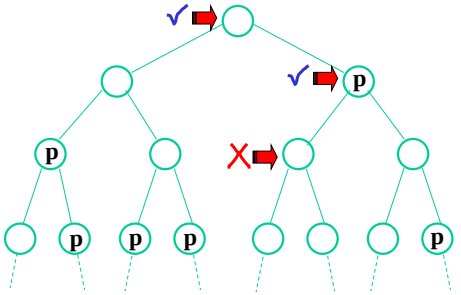
EG p



Ichiro Satoh

分岐時相論理の解釈

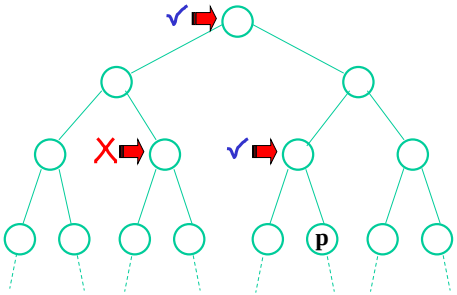
AF p



Ichiro Satoh

分岐時相論理の解釈

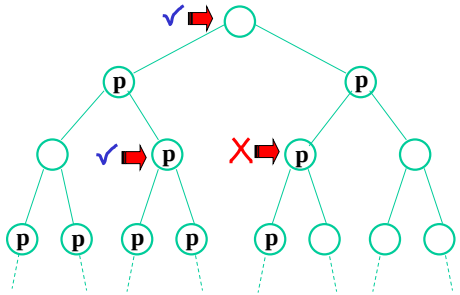
EF p



Ichiro Satoh

分岐時相論理の解釈

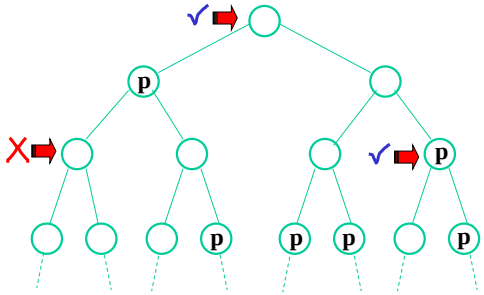
AX p



Ichiro Satoh

分岐時相論理の解釈

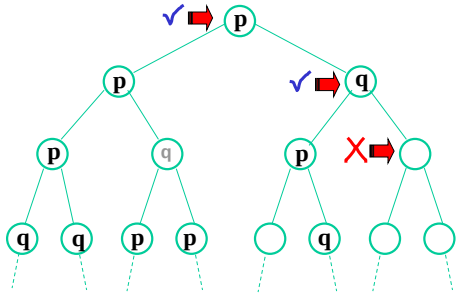
EX p



Ichiro Satoh

分岐時相論理の解釈

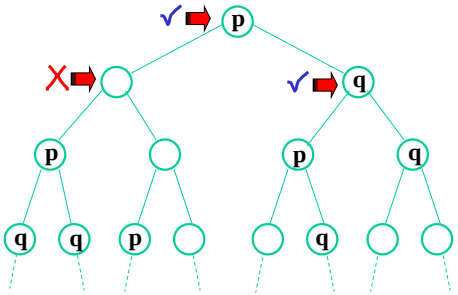
A[p U q]



Ichiro Satoh

分岐時相論理の解釈

E[p U q]



システム特性

Safety properties:

- ***Nothing "bad" ever happens***
- **Formalized using state invariants**
  - ***execution never reaches a "bad" state***

Liveness properties:

- ***Something "good" eventually happens***
- **Formalized using temporal logic**
  - ***special logic for describing sequences***

検証例

排他制御に要求される性質

初期状態
state
e.g. NCS1 ∧ NCS2

静的性質(Safety Property)
AG(state)
e.g. AG(¬ (CS1 ∧CS2))

動的性質(Liveness Property)
AG(requested ⊃ AF acknowledged)
e.g. AG((TRY1 ⊃ AF(CS1)) ∧ AG(TRY2 ⊃ AF(CS2))

検証例

An upwards travelling elevator at the second floor does not changes its direction when it has passengers waiting to go to the fifth floor

AG((floor=2 ∧ direction=up ∧ button5pressed)
⊃ A[direction=up U floor=5])

排他制御

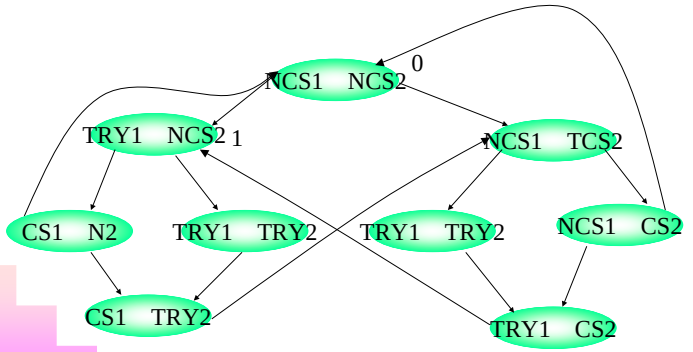
```
class Semaphore {
  private int count = 0;
  public Semaphore() {}
  public synchronized void P() {
    while (count == 0) {
      try {
        wait();
      } catch (InterruptedException ex) {}
    }
    count = count-1;
  }
  public synchronized void V() {
    count = count+1;
    notifyAll();
  }
}
```

Ichiro Satoh

モデル

NCS: noncritical region
TRY: trying region
CS: critical region

相互排除制御のモデル



Ichiro Satoh

Validate the model

"Execute" the model to test it

- simulate executions of the system
- check satisfaction of safety properties along simulated executions

Exhaustive analysis

- generate *reachability graph* to verify safety and liveness properties

Generate counterexamples to illustrate failures

Ichiro Satoh

CTLによるモデルチェッキング

CTL は、時系列上のイベントの順番を記述する時相論理体系である．
初期状態からの状態遷移系列を無限木の形で表し、この無限木に
どのようなpath があるかを指定する

CTLによるモデルチェッキングの利点

計算コストが小さい　O(|p|(|M|+|R|)

ただし、|p|は判定する式pの長さ、|M|は状態数、|R|は状態遷移の数

Ichiro Satoh

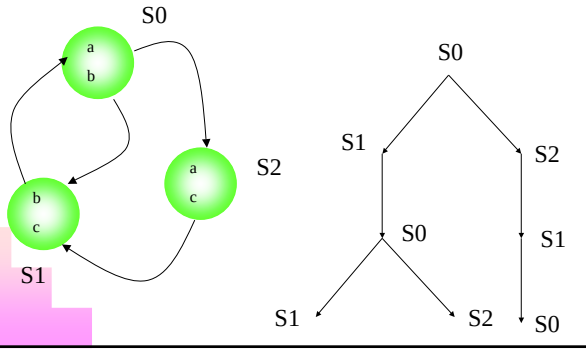
CTLによるモデルチェッキング

Used in studying behaviors of reactive systems
Typically involves three steps:
- Create a finite state model (FSM) of the system design
- Specify critical correctness properties
- Validate the model w/r to the specifications

Ichiro Satoh

モデルチェッキングの方法

モデル上の（マーカの）到達関係を調べる方法
木構造に変換して経路を調べる方法



Ichiro Satoh

線形時相論理

ϕ ::= P | $\neg P$ | $\phi \wedge \phi$ | $\phi \vee \phi$ | $\Box\phi$ | $\Diamond\phi$ | $\bigcirc\phi$

M |= $P \Leftrightarrow P \in L(s_0)$
M |= $\neg P \Leftrightarrow P \in L(s_0)$
M |= $\phi_1 \wedge \phi_2 \Leftrightarrow$ M |= ϕ_1 and M |= ϕ_2
M |= $\phi_1 \vee \phi_2 \Leftrightarrow$ M |= ϕ_1 or M |= ϕ_2
M |= $\bigcirc\phi \Leftrightarrow M^1 \models \phi$
M |= $\Box\phi \Leftrightarrow M^i \models \phi$ for any $i \geq 0$
M |= $\Diamond\phi \Leftrightarrow M^i \models \phi$ for some $i \geq 0$

M |= ϕ --- M は ϕ を充足する。

Ichiro Satoh

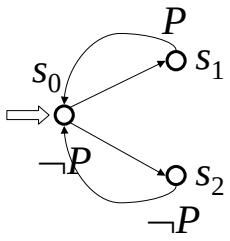
性質

M|= $\Box\phi$ iff M |= $\phi \wedge \bigcirc\Box\phi$
M|= $\Diamond\phi$ iff M |= $\phi \vee \bigcirc\Diamond\phi$

Ichiro Satoh

線形時相論理の例

例:



$s_1,s_0,\ldots \models P$

$s_1,s_0,\ldots \models \Diamond P$

$s_0,s_1,s_0,\ldots \models \Diamond P$

…

$s_0,s_2,s_0,s_2,s_0,s_1,s_0,\ldots \models \Diamond P$

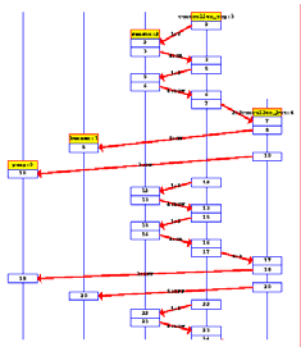
…

$s_0,s_1,s_0,s_1,s_0,s_2,s_0,\ldots \models \Box\Diamond P$

SPIN

SPIN automatically generates sequence diagrams to represent executions
- random
- guided
- interactive

Automates tracing between system view
- sequence diagram
- Promela description
- state diagram
- textual execution traces



時相論理の限界

公平性
- 弱公平性
- 強公平性

並列性
- (インターリーブ) 並行実行
- 並列実行

ハイブリットシステム
- 実時間性や確率の導入
- アナログ系との融合

公平性

CTLでは時相演算子(X,U,F,G)は量化演算子(E,A)ともに用いることから、「pが無限回成り立つ経路が存在する」を扱えない。

E F∞ p (文法上許されない)

CTL*やPLTL(命題線形時相論理)では論理式:

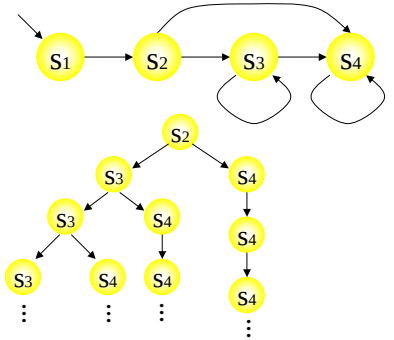
G F p または □◇p

によりpが無限回成り立つことを表現可能

分岐時相論理

CTL式の真偽値

状態グラフ (state graph) 上の状態とそこから始まるパスを考える

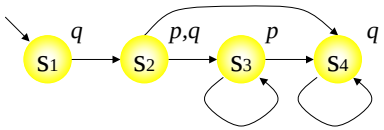


分岐時相論理の解釈

CTL式の真偽値

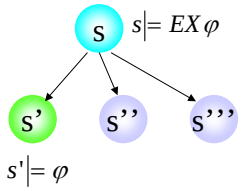
CTL式 が状態sで成立する(がsで真): $s \models \varphi$

原子命題, ∨, ∧, ¬ (not), → :その状態だ けで真偽値が決まる



分岐時相論理の解釈

CTL式の真偽値 $s \models EX\varphi \Leftrightarrow s$ のある次の状態s'で

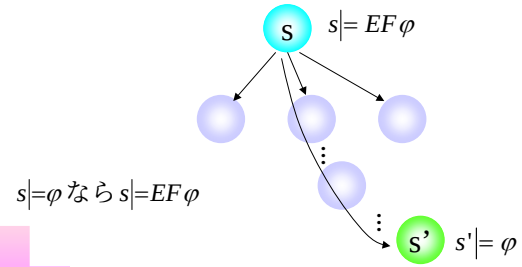


分岐時相論理の解釈

CTL式の真偽値

$s \models EF\varphi \Leftrightarrow s$から始まるあるパス中に

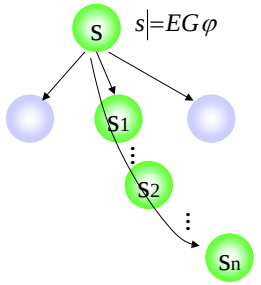
$s' \models \varphi$ なるs'が存在



分岐時相論理の解釈

CTL式の真偽値

$s \models EG\varphi$ から始まるあるパス中では，
そのパス中の すべての状態sにおいて $s' \models \varphi$

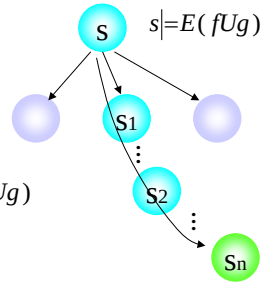


$s \models \varphi$
$s_1 \models \varphi$
$s_2 \models \varphi$
…
$s_n \models \varphi$

分岐時相論理の解釈

CTL式の真偽値

$s \models E(fUg)$ から始まるあるパス中では，gが
成り立つまでfが成り立ちつづける



$s \models E(fUg)$

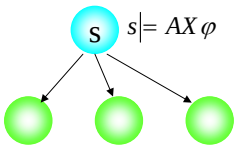
$s \models g$ なら $s \models E(fUg)$

$s \models f$
$s_1 \models f$
$s_2 \models f$
…
$s_n \models g$

分岐時相論理の解釈

CTL式の真偽値

$s \models AX\varphi$ のすべての次の状態s' で $s' \models \varphi$



$s \models AX\varphi$

時相論理

時相論理とは
状態の遷移や時間の経過の観点よりシステムの性質を記述するための論理体系

代表的形式

- 線形時間時相論理
  LTL (Linear Time Temporal Logic)
- 分岐時間時相論理
  CTL (Computation Tree Logic)
  μ計算 (μ-calculus)
- モデル検査
  時相論理で表現された性質を
  システムが満たすかどうかを検査すること

様相論理

可能性:	～は必然である	～は可能である
時間:	いつでも～である	～であるときがある
場所:	どこでも～である	あるところで～である
確実性:	～でなければならない	～かもしれない
信念:	～を知っている	～を信じている

$\Box P$ $\Diamond P$

様相論理

- 様相論理
- 必然性と可能性の論理
- 時相論理
- 知識と信念の論理
- 義務論理
- ダイナミック論理
- 内包論理

論理体系

古典論理 (classical logic)
- 命題論理
- 1階述語論理 … 導出原理, Prolog

非古典論理 (non-classical logic)
- 様相論理
- 高階論理
- 内包論理
- 多ソート論理
- 多値論理, fuzzy論理
- 非単調論理
- etc.

様相論理式

命題変数
P, *Q*, *R*, …

α, β：論理式

$\neg\alpha$	…	「αでない」
$\alpha \wedge \beta$	…	「αかつβ」
$\alpha \vee \beta$	…	「αまたはβ」
$\alpha \Rightarrow \beta$	…	「αならばβ」
$\alpha \Leftrightarrow \beta$	…	「αとβは同値」
$\Box\alpha$	…	「αは必然である」
$\Diamond\alpha$	…	「αは可能である」

$\Box\alpha = \neg\Diamond\neg\alpha$
$\Diamond\alpha = \neg\Box\neg\alpha$

可能世界意味論

□α　◇α

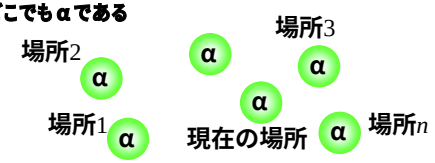
現在の場所・時刻とは別の場所・時刻における
αの真理値を考慮する

場所・時刻 … 可能世界 (possible world)
可能世界ごとに論理式を解釈

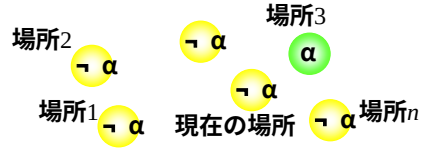
Ichiro Satoh

可能世界意味論

□α … どこでもαである



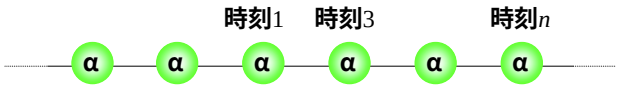
◇α … どこかではαである



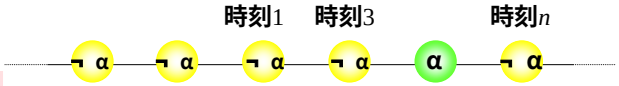
Ichiro Satoh

可能世界意味論

□α … いつでもαである



◇α … いつかはαである



Ichiro Satoh