# モデル検査入門

蓮尾 一郎

京都大学数理解析研究所
http://www.kurims.kyoto-u.ac.jp/~ichiro

## 1 導入：モデル検査とは？

### 1.1 形式検証

ソフトウェア，システムの**正しさ**は重要！ ロケットが落ちたり，携帯電話が回収されたり...
ソフトウェア，システムの**品質保証**のために—

- **テスト，シミュレーション：** いろいろな入力（テストケース）を入力してみて，うまく行くか見てみる
  - 産業界で標準的
  - 比較的低コスト（？）で，たくさんのバグが見つかる．が，
  - テストをパスしたからといって，もうバグが残っていない確証は**全くない**．
- **形式検証，形式手法** (formal verification, formal methods)：バグがないことを，**数学的に証明**

ありがちな批判：

> 形式検証は「おもちゃ」みたいなもので，（ひまな）研究者だけのものだ

が，最近はそうでもない！ [3]
形式検証にはさらに主に 2 つの方法が：

- **定理証明**：数学的な証明を書き下す．ペンと紙でやってもいいが，大変なので（この手の証明は場合分けがとても多くて，しかも大抵面白くない），**定理証明器**（theorem prover）や**証明支援器**（proof assistant）と呼ばれるソフトウェアを用いる（PVS，Isabell/HOL，Coq など）．
  - 長所：複雑なシステムや，パラメータが無限通りありうるシステムも検証できる．
  - 短所：人的コスト！ スペシャリスト（数学者や論理学者，高時給）を数日　数ヶ月拘束する．
- **モデル検査**：検証したいシステムを表す Kripke **構造** $M$ と，検証したい性質を表す**様相論理式** $\varphi$ を作って，$M$ **が** $\varphi$ **を満たすか**（$M \models \varphi$）をソフトウェア（**モデル検査器**，model checker）を用いて調べる．
  - 長所：多くの場合全自動！ $M$ と $\varphi$ を作ったら，モデル検査器に入力してボタンを押すだけ．また，$M$ が $\varphi$ を満たさない場合，**エラー軌跡**が出力され，バグを解消する手がかりになる．
  - 短所：モデル検査は基本的に「しらみつぶし」なので，
    - 無限の $M$ （たとえば，パラメータが任意の自然数）に対しては適用できない．
    - 有限でも複雑な $M$ や $\varphi$ に対しては，**状態爆発**（state explosion）を起こす．

**注意 1.1 (計算量)** 計算機科学においては，計算の複雑さを「入力が大きくなったとき，それに応じて計算時間がどれくらい増えるか」で量る（**計算量**）．$P$, $NP$, 指数時間とか．指数時間計算量の問題は「計算できない」（「計算量爆発」）．たとえば [5] を見よ．

すなわち，

| モデル検査 | | 定理証明 |
|---|---|---|
| コスト低 | $\longleftrightarrow$ | コスト高 |
| 完全度低 | $\longleftrightarrow$ | 完全度高 |

よって，システムの開発においては次のような手順が一般的：

- 設計の早い段階においては，単純化された（＝サイズの小さい）設計図に対してモデル検査を適用し，バグを発見する．
- 後になってより細かくなった設計図・ソフトウェアに対して定理証明を適用し，バグがないことを最終的に確かめる．

といっても，定理証明が行われるのはそのコストに見合う場合のみ（NASA のソフトウェアや，暗号プロトコルの検証など）．

この講義では，モデル検査の基礎の基礎について学ぶ[*1]．すなわち，

- みたすべき性質（**仕様** specification）を表現する**様相論理式**（modal logic formula）とは何か？
- システムを表現する Kripke **モデル**（Kripke model）とは何か？
- 両者の間の充足関係を，どのように自動的に判定するか？

### 1.2 モデル検査器

たくさんある．たとえば，SPIN model checker，mCRL2，PRISM，Uppaal など．

## 2 命題論理

モデル検査に用いる様相論理（**LTL**，**CTL**，**CTL**$^*$）について学ぶ前に，まずもっと基本的な様相論理 **K** について学ぼう．さらにそれには，まず命題論理から．

### 2.1 構文論（syntax）

集合 AP を一つ定めておく．その元 $p \in \mathsf{AP}$ を**原子論理式**（atomic formula）とよぶ．

**定義 2.1 (命題論理式)** 次の規則で生成されるものを**命題論理式**と呼ぶ．

$$
\begin{aligned}
p \in \mathsf{AP} &\implies p \text{ は命題論理式} \\
\varphi \text{ は命題論理式} &\implies \neg\varphi \text{ は命題論理式} \\
\varphi, \psi \text{ は命題論理式} &\implies \varphi \wedge \psi,\ \varphi \vee \psi,\ \varphi \to \psi \text{ は命題論理式}
\end{aligned}
$$

記号 $\neg, \wedge, \vee, \to$ を**論理演算子**と呼ぶ．

[*1] モデル検査の**適用**は簡単で面白くないが，モデル検査器を**作る**ための理論はおもしろい！

それぞれの演算子の直感的な意味は次の通り．

| | |
|---|---|
| $\neg\varphi$ | 否定　$\varphi$ でない |
| $\varphi\wedge\psi$ | 連言　$\varphi$ かつ $\psi$ |
| $\varphi\vee\psi$ | 選言　$\varphi$ または $\psi$ |
| $\varphi\to\psi$ | 含意　$\varphi$ ならば $\psi$ |

**注意 2.2** が！これはあくまで「直感的な意味」であって，今のところ命題論理式は「ただの記号列」でしかない．また，たとえば $\varphi\wedge\psi\wedge\xi$ という命題論理式は存在しないことに注意．$(\varphi\wedge\psi)\wedge\xi$ と $\varphi\wedge(\psi\wedge\xi)$ は両方とも命題論理式だが，記号列として違う以上この 2 つは異なる命題論理式であり，区別しなければいけない．

## 2.2　意味論（semantics）

これから命題論理式の「意味」を，**数学的に**定義していく[*2]．

命題論理式 $\varphi$ が真かどうかは，それぞれの原子論理式 $p\in\mathsf{AP}$ が真かによって決まる．AP の部分集合 $V$ を一つ定めて，それを「真な原子論理式の集合」としよう．

**注意 2.3** AP の部分集合 $V\subseteq\mathsf{AP}$ を定めることと，元が二つの集合への関数

$$\mathsf{AP}\longrightarrow\{\top,\bot\}$$

を定めることは同じである．（$\top$ は「真」，$\bot$ は「偽」と思ってみよう）

**定義 2.4** 原子論理式の集合 AP とその部分集合 $V\subseteq\mathsf{AP}$ が定まっているとき，次のように関係 $V\models\varphi$ を定める．$V\models\varphi$ が成り立つとき，$\varphi$ が $V$ **のもとで真である**という．

$$\begin{aligned}
V\models p\quad(p\in\mathsf{AP}) &\overset{\text{def}}{\iff} p\in V\\
V\models\neg\varphi &\overset{\text{def}}{\iff} V\models\varphi\text{ が成り立たない}\\
V\models\varphi\wedge\psi &\overset{\text{def}}{\iff} V\models\varphi\text{ と }V\models\psi\text{ が両方成り立つ}\\
V\models\varphi\vee\psi &\overset{\text{def}}{\iff} V\models\varphi\text{ か }V\models\psi\text{ の少なくとも一方が成り立つ}\\
V\models\varphi\to\psi &\overset{\text{def}}{\iff} V\models\varphi\text{ が成り立っていれば必ず }V\models\psi\text{ も成り立つ}
\end{aligned}$$

**注意 2.5** よって $V\models\varphi$ が成り立たないなら（すなわち $\varphi$ が偽なら），$\psi$ が何であっても $\varphi\to\psi$ は真．

**注意 2.6** 記号 $\iff$ や $\Longrightarrow$ と，$\to$ を区別せよ！$\iff$ や $\Longrightarrow$ は「メタ」な含意関係を表すのに対して，$\to$ は様相論理の体系の中の記号に過ぎない．

注意 2.2 に出てきた $(\varphi\wedge\psi)\wedge\xi$ と $\varphi\wedge(\psi\wedge\xi)$ は異なる命題論理式として区別すべきだが，意味が等しいことは直感的に明らか．この「意味が等しい」ことは次のように正確に定義される．

**定義 2.7** 原子論理式の集合 AP が定まっているとき，二つの命題論理式 $\varphi$ と $\psi$ が**論理的に同値**であるとは，次のようなことをいう：任意の $V\subseteq\mathsf{AP}$ について，$V$ のもとでの真偽が一致する，すなわち

$$V\models\varphi\quad\iff\quad V\models\psi\ .$$

[*2] 構文論 syntax と 意味論 semantics を対比せよ．

# 3 様相論理 K

モデル検査で用いる様相論理は **LTL**，**CTL**，**CTL**$^*$ などだが，それらの雛形たる基本的な様相論理 **K** について学ぼう．

様相論理 **K** は，命題論理の論理演算子 $\neg, \wedge, \vee, \to$ にさらに演算子 $\Box$ を加えた論理である．様相論理式 $\Box\varphi$ の直感的な意味は，「必ず $\varphi$」「$\varphi$ しなければならない」「$\varphi$ であることを知っている」「$\varphi$ が証明可能」などなど．

様相論理（modal logic）の歴史自体は古く，アリストテレスによって研究が始められた．長い歴史を持つ研究を現代的に整理したのは C.I. Lewis と C.H. Langford である．これらの研究は主に哲学的興味から行われて，たとえばこの授業で用いる様相論理の意味論は哲学者 S. Kripke （など）による．

## 3.1 構文論（syntax）

命題論理の場合と同様，原子論理式の集合 AP を定めておいて，その上で考える．

**定義 3.1 (K-論理式)** 次の規則で生成されるものを **K-論理式**（あるいは単に**論理式**）と呼ぶ．

$$
\begin{aligned}
p \in \mathsf{AP} \quad &\Longrightarrow \quad p \text{ は } \mathbf{K}\text{-論理式} \\
\varphi \text{ は } \mathbf{K}\text{-論理式} \quad &\Longrightarrow \quad \neg\varphi \text{ は } \mathbf{K}\text{-論理式} \\
\varphi, \psi \text{ は } \mathbf{K}\text{-論理式} \quad &\Longrightarrow \quad \varphi \wedge \psi,\ \varphi \vee \psi,\ \varphi \to \psi \text{ は } \mathbf{K}\text{-論理式} \\
\varphi \text{ は } \mathbf{K}\text{-論理式} \quad &\Longrightarrow \quad \Box\varphi \text{ は } \mathbf{K}\text{-論理式} \qquad \longleftarrow \text{ new!!}
\end{aligned}
$$

記号 $\neg, \wedge, \vee, \to, \Box$ を**論理演算子**と呼ぶ．

$\Box\varphi$ の直感的な意味は（上で触れたように）いろいろあるが，ここではとりあえず「必ず $\varphi$」と読むことにしよう．

## 3.2 Kripke 意味論（Kripke semantics）

様相論理の Kripke **意味論**の特徴は，様相論理式 $\varphi$ の真偽を

　　たくさんある**可能世界**のそれぞれで

判定することである．

**定義 3.2 (Kripke フレーム)** Kripke **フレーム**（Kripke frame）とは，次のような 2 つ組 $F = (S, R)$ のことを言う．

- $S$ は空でない集合．$S$ の各元 $s \in S$ を**状態**（state）や**可能世界**（possible world）と呼ぶ．
- $R$ は $S$ 上の二項関係．$R$ を**到達可能関係**（accessibility relation）と呼ぶ．

**定義 3.3 (Kripke モデル)** 原子論理式の集合 AP が定まっているとき，その上の Kripke **モデル**（Kripke model）とは，次のような 3 つ組 $M = (S, R, V)$ のことを言う，

- $(S, R)$ は Kripke フレーム．

- $V$ は次の型の関数：

$$V \ : \ S \times \mathsf{AP} \quad \longrightarrow \quad \{\top, \bot\}$$

$V$ を**付値関数**（valuation function）と呼ぶ．

**注意 3.4**
- 二項関係 $R$ とは，正確に言うと $S \times S$ の部分集合のこと．すなわち

$$R \quad \subseteq \quad S \times S \ .$$

$R$ を使って，「状態 $s_1$ から状態 $s_2$ に**到達可能**か？」を表現しているのである．つまり，$(s_1, s_2) \in R$ であることを「$s_1$ から $s_2$ へ到達可能」と理解する．「$s_1$ から $s_2$ へ矢印がある」と読んでもいい．$(s_1, s_2) \in R$ であることを $s_1 R s_2$ とも書く．
- 付値関数は，各原子論理式 $p \in \mathsf{AP}$ と各可能世界 $s \in S$ について，

$$V(s, p) \quad = \quad \top \text{ または } \bot \ ,$$

すなわち「$p$ は $s$ で真か？偽か？」を判定する． 注意 2.3 も参照せよ．

Kripke モデル $M = (S, R, V)$ は，命題論理に対する $V$ を拡張して「可能世界がたくさんあるように」したものである．これに基づいて，定義 2.4 と同様に，**K**-論理式の真偽を定めていこう．

**定義 3.5** 原子論理式の集合 $\mathsf{AP}$ と Kripke モデル $M = (S, R, V)$ が定まっているとき，次のように関係 $M, s \models \varphi$ を定める．$M, s \models \varphi$ が成り立つとき，$\varphi$ **が** Kripke **モデル** $M$ **の状態** $s$ **で真である**という．

$$\begin{aligned}
M, s \models p \quad (p \in \mathsf{AP}) \quad &\overset{\text{def}}{\Longleftrightarrow} \quad V(s, p) = \top \\
M, s \models \neg\varphi \quad &\overset{\text{def}}{\Longleftrightarrow} \quad M, s \models \varphi \text{ が成り立たない} \\
M, s \models \varphi \wedge \psi \quad &\overset{\text{def}}{\Longleftrightarrow} \quad M, s \models \varphi \text{ と } M, s \models \psi \text{ が両方成り立つ} \\
M, s \models \varphi \vee \psi \quad &\overset{\text{def}}{\Longleftrightarrow} \quad M, s \models \varphi \text{ か } M, s \models \psi \text{ の少なくとも一方が成り立つ} \\
M, s \models \varphi \to \psi \quad &\overset{\text{def}}{\Longleftrightarrow} \quad M, s \models \varphi \text{ が成り立っていれば必ず } M, s \models \psi \text{ も成り立つ} \\
M, s \models \Box\varphi \quad &\overset{\text{def}}{\Longleftrightarrow} \quad sRs' \text{ であるようなすべての } s' \text{ について，} M, s' \models \varphi \text{ が成り立つ}
\end{aligned}$$

すなわち，$\Box\varphi$ が状態 $s$ で真であるのは，$s$ の「次の」状態すべてで $\varphi$ が真であるときである．

**定義 3.6** **K**-論理式 $\varphi$ が Kripke フレーム $F = (S, R)$ で**恒真**（valid）であるとは，

かってな付値関数 $V : S \times \mathsf{AP} \to \{\top, \bot\}$ と，かってな状態 $s \in S$ に対して，$(S, R, V), s \models \varphi$ となること

をいう．これを $F \models \varphi$ と書き表す．

**例 3.7** $S = \{a, b, c\}$，$R = \{(a, a), (b, b), (c, c), (a, b), (b, c)\}$ において，論理式 $\Box p \to p$ は恒真．

## 3.3 Kripke フレームの構造と公理型の対応理論

さまざまな，次のようなタイプの結果が知られている：

Kripke フレーム $F = (S, R)$ で論理式 $\varphi$ が恒真 $\iff$ Kripke フレームが条件 (...) をみたす．

このような，「フレームの構造を指し示す論理式」として代表的なものを書くと，

| | |
|---|---|
| (**T**) | $\Box\varphi \to \varphi$ |
| (**4**) | $\Box\varphi \to \Box\Box\varphi$ |
| (**D**) | $\Box\varphi \to \Diamond\varphi$ |
| (**B**) | $\varphi \to \Box\Diamond\varphi$ |
| (**5**) | $\Diamond\varphi \to \Box\Diamond\varphi$ |

ただしここで，$\Diamond\varphi$ は $\neg\Box\neg\varphi$ の略記．直感的には「必ず $\varphi$ でない，というわけではない」，つまり「$\varphi$ かもしれない」と読める．

**注意 3.8** 上であげた「論理式」**T**，**4**，**D**，**B**，**5** のそれぞれは，正確にいうと論理式でなく「論理式の族」である．たとえば **T** は $\Box\varphi \to \varphi$ の型をした論理式全体を指し示し，現れる $\varphi$ は何であってもよい．**T** の指し示す論理式の族には $\Box p \to p$ や $\Box(p \wedge \neg q) \to (p \wedge \neg q)$ などが含まれている．

正確を期すため，今後このような「論理式の族」を**公理型**（axiom scheme）と呼ぼう．

**定義 3.9** Kripke フレーム $F$ においてある公理型が**恒真**であるとは，その公理型に属するすべての論理式が $F$ で恒真であることをいう．たとえば，$F$ で **T** が恒真であるとは，すべての論理式 $\varphi$ について

$$F \models \Box\varphi \to \varphi$$

となることである．

次のような意味で，Kripke フレームの構造を公理系を使って特徴付けることができる．

**定理 3.10** かってな Kripke フレーム $F = (S, R)$ について，次が成り立つ．

1. **T** が $F$ で恒真 $\iff$ $R$ は反射的（reflexive）．
2. **4** が $F$ で恒真 $\iff$ $R$ は推移的（transitive）．
3. **D** が $F$ で恒真 $\iff$ $R$ は継続的（serial）．
4. **B** が $F$ で恒真 $\iff$ $R$ は対称的（symmetric）．
5. **5** が $F$ で恒真 $\iff$ $R$ はユークリッド的（Euclidean）．

ただし，

**定義 3.11** 集合 $S$ 上の 2 項関係 $R$ について，

1. $R$ が**反射的**とは，すべての $s \in S$ について $sRs$ となることをいう．
2. $R$ が**推移的**とは，すべての $s, s', s'' \in S$ について

   $$sRs' \quad \text{かつ} \quad s'Rs'' \qquad \Longrightarrow \qquad sRs''$$

   となることをいう．
3. $R$ が**継続的**とは，すべての $s \in S$ について $sRs'$ となるような $s' \in S$ が存在することをいう．
4. $R$ が**対称的**とは，すべての $s, s' \in S$ について

   $$sRs' \qquad \Longrightarrow \qquad s'Rs$$

   となることをいう．

5. $R$ が**ユークリッド的**とは，すべての $s, s', s'' \in S$ について

$$sRs' \quad \text{かつ} \quad sRs'' \qquad \Longrightarrow \qquad s'Rs''$$

となることをいう．

6. 反射的かつ推移的な $R$ を **preorder** という．
7. 反射的，推移的かつ反対称的な $R$ を**順序**（order）という．ただし，$R$ が反対称的であるとは

$$sRs' \quad \text{かつ} \quad s'Rs \qquad \Longrightarrow \qquad s = s'$$

となることをいう．

8. 反射的，推移的かつ対称的な $R$ を**同値関係**（equivalence relation）という．

**問題 3.12** 論理式 $\Box\Diamond p \to \Diamond\Box p$ が偽であるような Kripke モデル（とその状態）を見つけよ．

**問題 3.13** 次の論理式それぞれに対して，それが偽であるような Kripke モデル（とその状態）を見つけよ．

$$\Box\Diamond p \to \Diamond\Box p \ , \quad \Diamond p \to \Box p \ , \quad \Box\Box p \to \Box p \ , \quad \Diamond\Box p \to \Box\Diamond p \ .$$

**例 3.14** $\Box\varphi$ を「$\varphi$ であることを知っている」と理解するような様相論理を**知識論理**（epistemic logic）という．知識論理と「信念の論理」（$\Box\varphi$ を「$\varphi$ だと信じている」と読む）を区別するのは，公理型 **T** が恒真であるかどうか．両者に共通して恒真である（と普通思われている）公理系は，

$$\begin{array}{ll} (\mathbf{4}) & \Box\varphi \to \Box\Box\varphi \\ (\mathbf{5}^\circ) & \neg\Box\varphi \to \Box\neg\Box\varphi \end{array}$$

この場合，**4** を「**正の内省** positive introspection」，**5°** を「**負の内省** negative introspection」と呼ぶ．

**例 3.15** Muddy children puzzle （時間があれば）

## 4　システムの Kripke モデルによる表現

様相演算子 $\Box$ をどう読むかによって，Kripke フレームの到達可能関係 $R$ の意味づけも変わる（たとえば知識論理では，$sRs'$ は「$s$ と $s'$ が同じに見える」と読む）．

モデル検査では，Kripke モデルの

- 状態 $s \in S$ をシステムの**状態**，
- 到達可能関係 $R$ をシステムの**計算過程**

と解釈することで，（検証したい）システムを Kripke モデルとして表現する．

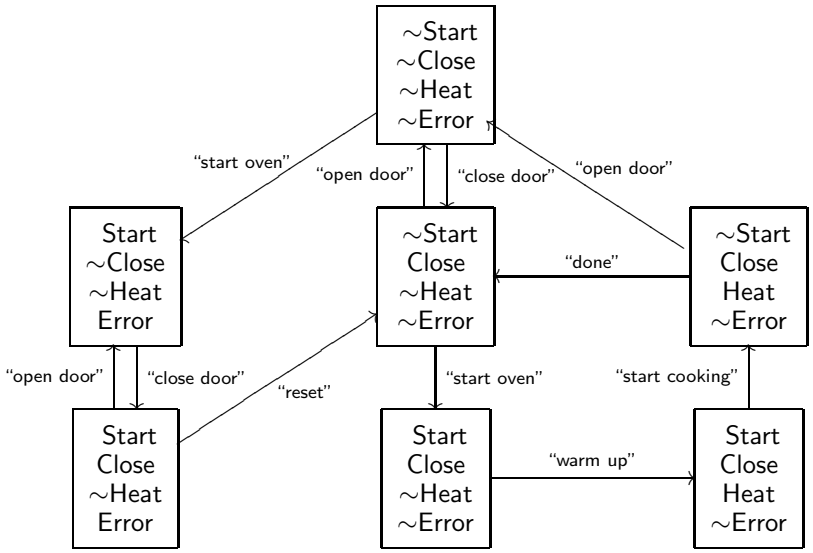
**注意 4.1** 検証したいシステムは，多くの場合

- **リアクティブ**（reactive）：外部からの入力に対して反応するが，システム自体は停止しない．「入力をもらって，出力して停止」というのとは違う．
- **並行システム**（concurrent）：システム自体が多くの**サブシステム**（または**コンポーネント** component）からなっており，それらが並行して実行している．

$\lambda$ 計算の項は入力から出力への関数と思うことができるが，以上のような性質を持つシステムには「関数としての表現」はなじまず，Kripke モデルの方が表現としてより適当．以上の 2 つの性質は Kripke モデルの以下の性質として現れる．

- Kripke モデルにはループ
$$s_0 R s_1,\ s_1 R s_2,\ \ldots,\ s_n R s_0$$
があってよい．すなわち，計算過程が停止するとはかぎらない．
- 一つの状態 $s$ から，2 つの異なる状態 $s', s''$ に計算が進み得る（$sRs'$，$sRs''$）．この性質を**非決定性**（non-determinism）とよぶ．
非決定性と並行システムの関係について，たとえば 2 つのサブシステム $\mathcal{S}_1$，$\mathcal{S}_2$ が同時に（独立に）実行しているシステム $\mathcal{S}_1 \parallel \mathcal{S}_2$ を考えよう．各状態において
次に $\mathcal{S}_1$ が実行するか？　または　次に $\mathcal{S}_2$ が実行するか？
によって（少なくとも）2 つの計算過程が可能である．こうして非決定性が自然に現れる．

**例 4.2 (Microwave)** [2, Fig. 4.3] からの例を少し改変したもの．ここで AP = {Start, Close, Heat, Error}.



# 5 様相論理 LTL，CTL，CTL*

それでは前節に引き続いて，モデル検査でシステムの満たすべき性質（**仕様**）を記述するために使う様相論理 **LTL**，**CTL**，**CTL*** について学ぼう．このうちもっとも豊かなものが **CTL*** で，**LTL** と **CTL** は **CTL*** の一部分である．**CTL** は compuational tree logic（計算木論理），**LTL** は linear temporal logic（線形時相論理）の略．

## 5.1 CTL* の syntax

**K** では**様相演算子**は □ の 1 種類だったが[*3]，**CTL*** にはもっとたくさん様相演算子がある．

| | | 直観的な意味 |
|---|---|---|
| **パス量化子**（path quantifiers） | $\mathbf{E}\varphi$ | あるパスが存在して，there Exists |
| | $\mathbf{A}\varphi$ | すべてのパスにおいて，for All |
| **時相演算子**（temporal operators） | $\mathbf{X}\varphi$ | （パスの上の）次の状態で，neXt time |
| | $\mathbf{F}\varphi$ | いつか将来，in the Futhre |
| | $\mathbf{G}\varphi$ | これからずっと，Globally |
| | $\varphi\mathbf{U}\psi$ | いつか将来 $\psi$ がなりたって，そのときまではずっと $\varphi$，Until |
| | $\varphi\mathbf{R}\psi$ | $\varphi$ が成り立たないうちはずっと $\psi$, Release |

2 つの種類の様相演算子に対応して，**CTL*** には 2 つの種類の論理式がある．

**定義 5.1 (CTL* の状態論理式，パス論理式)** 次の規則で生成されるものを **CTL*** の**状態論理式，パス論理式**と呼ぶ．

$$
\begin{aligned}
p \in \mathsf{AP} &\implies p \text{ は状態論理式} \\
\varphi, \psi \text{ は状態論理式} &\implies \neg\varphi, \varphi \vee \psi \text{ は状態論理式} \\
\varphi \text{ はパス論理式} &\implies \mathbf{E}\varphi, \mathbf{A}\varphi \text{ は状態論理式} \\
\varphi \text{ は状態論理式} &\implies \varphi \text{ はパス論理式} \\
\varphi, \psi \text{ はパス論理式} &\implies \neg\varphi, \varphi \vee \psi, \mathbf{X}\varphi, \mathbf{F}\varphi, \mathbf{G}\varphi, \varphi\mathbf{U}\psi, \varphi\mathbf{R}\psi \text{ はパス論理式}
\end{aligned}
$$

**注意 5.2** 上の定義は，**相互帰納的**（mutually inductive）な定義になっている．

## 5.2 CTL* の semantics

**CTL*** の論理式の真偽は **K** と同様に Kripke モデルによって定まるが，特に

- 状態論理式は Kripke モデルの**各状態においてその真偽が定まり**（$M, s \models \varphi$），
- 対してパス論理式は Kripke モデルの**各パスにおいて**その真偽が定まる（$M, \pi \models \varphi$），

ただしここで，

**定義 5.3 (パス)** Kripke フレーム $F = (S, R)$ における（**無限**）**パス**とは，状態がなす無限の列

$$\pi = s_0, s_1, \ldots$$

[*3] ◇ も数えれば 2 種類だが，この授業では ◇ は ¬□¬ の略記として導入したのであった．

であって $s_i \to s_{i+1}$ であるもの，すなわち

$$(s_i, s_{i+1}) \in R \ , \qquad \text{for } i = 0, 1, \ldots$$

となるもののことをいう．

パス $\pi = s_0, s_1, \ldots$ に対して，その $s_i$ から始まる**末尾**（suffix）を $\pi^i$ と書くことにする．すなわち，$\pi^i$ は（無限）パスで

$$\pi^i = s_i, s_{i+1}, \ldots \ .$$

**定義 5.4** Kripke モデル $M = (S, R, V)$ が定まっているとき，

- 各状態 $s \in S$ と各状態論理式 $\varphi$ に対しての関係 $M, s \models \varphi$ と，
- 各パス $\pi$ と各パス論理式 $\varphi$ に対しての関係 $M, \pi \models \varphi$

を図 1 のように，論理式の構成に関して帰納的に定める．

**図 1 CTL$^*$ の semantics**

$$\begin{aligned}
M, s \models p \quad (p \in \mathsf{AP}) &\stackrel{\text{def}}{\iff} V(s, p) = \top \\
M, s \models \neg\varphi &\stackrel{\text{def}}{\iff} M, s \models \varphi \text{ が成り立たない} \\
M, s \models \varphi \vee \psi &\stackrel{\text{def}}{\iff} M, s \models \varphi \text{ か } M, s \models \psi \text{ の少なくとも一方が成り立つ} \\
M, s \models \mathbf{E}\varphi &\stackrel{\text{def}}{\iff} s \text{ から始まるパス } \pi \text{ で，} M, \pi \models \varphi \text{ となるようなものがある} \\
M, s \models \mathbf{A}\varphi &\stackrel{\text{def}}{\iff} s \text{ から始まるすべてのパス } \pi \text{ について，} M, \pi \models \varphi \text{ がなりたつ} \\
M, \pi \models \varphi \quad (\varphi \text{ は状態論理式}) &\stackrel{\text{def}}{\iff} \pi \text{ の最初の状態 } s \text{ で，} M, s \models \varphi \text{ がなりたつ} \\
M, \pi \models \neg\varphi &\stackrel{\text{def}}{\iff} M, \pi \models \varphi \text{ が成り立たない} \\
M, \pi \models \varphi \vee \psi &\stackrel{\text{def}}{\iff} M, \pi \models \varphi \text{ か } M, \pi \models \psi \text{ の少なくとも一方が成り立つ} \\
M, \pi \models \mathbf{X}\varphi &\stackrel{\text{def}}{\iff} M, \pi^1 \models \varphi \\
M, \pi \models \mathbf{F}\varphi &\stackrel{\text{def}}{\iff} \text{ある } k \geq 0 \text{ が存在して，} M, \pi^k \models \varphi \\
M, \pi \models \mathbf{G}\varphi &\stackrel{\text{def}}{\iff} \text{すべての } k \geq 0 \text{ について，} M, \pi^k \models \varphi \\
M, \pi \models \varphi\mathbf{U}\psi &\stackrel{\text{def}}{\iff} \text{次のような } k \geq 0 \text{ が存在する：} \\
&\qquad M, \pi^k \models \psi \text{ であり，かつ，すべての } i \in [0, k-1] \text{ に対して } M, \pi^i \models \varphi \\
M, \pi \models \varphi\mathbf{R}\psi &\stackrel{\text{def}}{\iff} \text{各 } k \geq 0 \text{ にたいして，} \\
&\qquad M, \pi^k \models \psi \text{ がなりたつか，あるいはある } j \in [0, k-1] \text{ に対して } M, \pi^j \models \varphi
\end{aligned}$$

**定義 5.5** 二つの状態論理式 $\varphi$ と $\psi$ が**論理的に同値**であるとは，次のようなことをいう：すべての Kripke モデル $M$ とそのすべての状態 $s$ に対して，

$$M, s \models \varphi \quad \Longleftrightarrow \quad M, s \models \psi \ .$$

二つのパス論理式 $\varphi$ と $\psi$ が**論理的に同値**であるとは，次のようなことをいう：すべての Kripke モデル $M$ とそのすべてのパス $\pi$ に対して，

$$M, \pi \models \varphi \quad \Longleftrightarrow \quad M, \pi \models \psi \ .$$

パス・状態論理式が論理的に同値であることを，$\varphi \cong \psi$ と書く．

**例 5.6**

$$
\begin{aligned}
\varphi \mathbf{R} \psi &\cong \neg(\neg\varphi \mathbf{U} \neg\psi) \\
\mathbf{F}\varphi &\cong \mathsf{True}\mathbf{U}\varphi \qquad \text{ただし True は，勝手な } p \in \mathsf{AP} \text{ を決めたときの } p \vee \neg p \text{ の略記．} \\
\mathbf{G}\varphi &\cong \neg\mathbf{F}\neg\varphi \\
\mathbf{A}\varphi &\cong \neg\mathbf{E}\neg\varphi
\end{aligned}
$$

## 5.3 CTL と LTL

**CTL** と **LTL** はともに $\mathbf{CTL}^*$ の部分論理である．

**CTL** においては，時相演算子 $\mathbf{X}, \mathbf{F}, \mathbf{G}, \mathbf{U}, \mathbf{R}$ が現れたらそのすぐ前（外側）にパス演算子 $\mathbf{A}, \mathbf{E}$ がついていなくてはならない．正確には以下の通り．（定義 5.1 と比較しよう）

**定義 5.7 (CTL の状態論理式，パス論理式)** 次の規則で生成されるものを **CTL** の**状態論理式，パス論理式**と呼ぶ．

$$
\begin{aligned}
p \in \mathsf{AP} &\implies p \text{ は状態論理式} \\
\varphi, \psi \text{ は状態論理式} &\implies \neg\varphi, \varphi \vee \psi \text{ は状態論理式} \\
\varphi \text{ はパス論理式} &\implies \mathbf{E}\varphi, \mathbf{A}\varphi \text{ は状態論理式} \\
\varphi, \psi \text{ は状態論理式} &\implies \mathbf{X}\varphi, \mathbf{F}\varphi, \mathbf{G}\varphi, \varphi\mathbf{U}\psi, \varphi\mathbf{R}\psi \text{ はパス論理式}
\end{aligned}
$$

**命題 5.8** 以上の定義は，次の定義と同値．**CTL** の状態論理式は次のように生成される：

$$
\begin{aligned}
p \in \mathsf{AP} &\implies p \text{ は状態論理式} \\
\varphi, \psi \text{ は状態論理式} &\implies \neg\varphi, \varphi \vee \psi \text{ は状態論理式} \\
\varphi, \psi \text{ は状態論理式} &\implies \mathbf{AX}\varphi, \mathbf{EX}\varphi, \mathbf{AF}\varphi, \mathbf{EF}\varphi, \mathbf{AG}\varphi, \mathbf{EG}\varphi, \\
&\qquad \mathbf{A}(\varphi\mathbf{U}\psi), \mathbf{E}(\varphi\mathbf{U}\psi), \mathbf{A}(\varphi\mathbf{R}\psi), \mathbf{E}(\varphi\mathbf{R}\psi) \text{ は状態論理式}
\end{aligned}
$$

すなわち，**CTL** は様相演算子を 10 種類

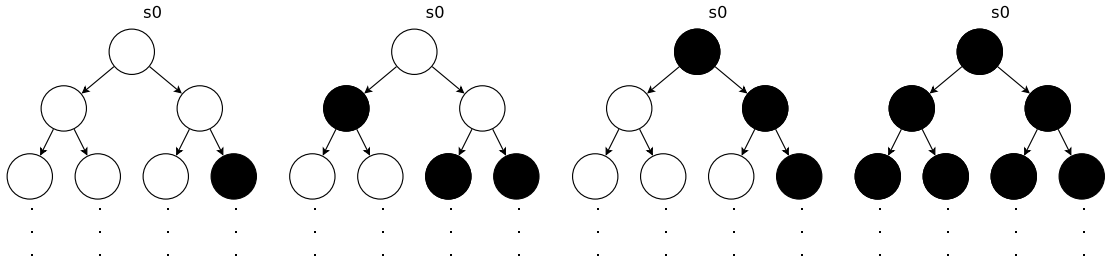
$$\mathbf{AX},\ \mathbf{EX},\ \ldots,\ \mathbf{A}(\_\,\mathbf{R}\,\_),\ \mathbf{E}(\_\,\mathbf{R}\,\_)$$

持った様相論理だと思うことができる．よって状態論理式とパス論理式の区別は **CTL** においてはあまり意味がない．以下，**CTL** に関しては状態論理式のことを**論理式**と呼ぶことにしよう．

**補題 5.9** **CTL** の 10 種類の様相演算子は，すべて $\mathbf{EX}, \mathbf{EG}, \mathbf{EU}$ の 3 種類を用いて書ける．というのは，すべての **CTL** 状態論理式 $\varphi, \psi$ に対して，次の論理的同値が成り立つ．

$$
\begin{aligned}
\mathbf{AX}\varphi &\cong \neg\mathbf{EX}\neg\varphi \\
\mathbf{EF}\varphi &\cong \mathbf{E}(\mathsf{True}\mathbf{U}\varphi) \qquad (\mathsf{True} \text{ は例 5.6 と同じ}) \\
\mathbf{AG}\varphi &\cong \neg\mathbf{EF}\neg\varphi \\
\mathbf{AF}\varphi &\cong \neg\mathbf{EG}\neg\varphi \\
\mathbf{A}(\varphi\mathbf{U}\psi) &\cong \neg\mathbf{E}(\neg\psi\mathbf{U}(\neg\varphi \wedge \neg\psi)) \wedge \neg\mathbf{EG}\neg\psi \\
\mathbf{A}(\varphi\mathbf{R}\psi) &\cong \neg\mathbf{E}(\neg\varphi\mathbf{U}\neg\psi) \\
\mathbf{E}(\varphi\mathbf{R}\psi) &\cong \neg\mathbf{A}(\neg\varphi\mathbf{U}\neg\psi)
\end{aligned}
$$

図 2 **CTL** の各様相演算子の解釈




$M, s_0 \models \mathbf{EF}\varphi$　　$M, s_0 \models \mathbf{AF}\varphi$　　$M, s_0 \models \mathbf{EG}\varphi$　　$M, s_0 \models \mathbf{AG}\varphi$

**例 5.10** $\mathbf{EF}\varphi, \mathbf{AF}\varphi, \mathbf{EG}\varphi, \mathbf{AG}\varphi$ の semantics を理解しよう．図 2 を見る．

**例 5.11** システム検証に使われる **CTL** 論理式の代表的なものをあげてみる：

- $\mathbf{EF}(\mathsf{Start} \wedge \neg \mathsf{Ready})$ : Start は成り立つけれど Ready がなりたたないような，状態に到達しうる（こういうのは避けたい）．
- $\mathbf{AG}(\mathsf{Req} \rightarrow \mathbf{AF}\mathsf{Ack})$ : リクエストが起こったら，そのうち必ず受け取り確認（acknowledgment）がなされる．
- $\mathbf{AG}(\mathbf{AF}\mathsf{DeviceEnabled})$ : どの時点においても，それからしばらく待っていたらデバイスが使用可能になる．違う言い方をすると，すべてのパスにおいて DeviceEnabled が**無限回**真である．
- $\mathbf{AG}(\mathbf{EF}\mathsf{Restart})$ : どの状態からでも再起動が可能．

**LTL**（linear temporal logic）は「パスの分岐構造」を表現することができない，制限された論理である．**LTL** には（**CTL** の場合と逆に）「パス論理式しかない」と思うことができる．

**定義 5.12 (LTL のパス論理式，状態論理式)** 次の規則で生成されるものを **LTL** の**パス論理式**と呼ぶ．

$$p \in \mathsf{AP} \quad \Longrightarrow \quad p \text{ はパス論理式}$$
$$\varphi, \psi \text{ はパス論理式} \quad \Longrightarrow \quad \neg\varphi, \varphi \vee \psi, \mathbf{X}\varphi, \mathbf{F}\varphi, \mathbf{G}\varphi, \varphi\mathbf{U}\psi, \varphi\mathbf{R}\psi \text{ はパス論理式}$$

**LTL** の**状態論理式**は，パス論理式 $\varphi$ を用いて $\mathbf{A}\varphi$ とかけるもののこと．

# 6 CTL モデル検査

**モデル検査**の「問題」は次のように記述できる．

**入力：** Kripke モデル $M = (S, R, V)$ と，状態論理式 $\varphi$
**出力：** $\varphi$ をみたす状態 $s$ の集合，すなわち $\{s \in S \mid M, s \models \varphi\}$

この問題を解く**アルゴリズム**を見つけて，実装して（モデル検査器），自動で解けるようにしたい！ この節では，状態論理式が **CTL** のそれである場合のアルゴリズムを解説する．

モデル検査の問題をどのように応用するかというと，

- 検証したいシステムを Kripke モデル $M$ として表現する．（例 4.2 のように）
- 検証したい性質を状態論理式 $\varphi$ として書き下す．（例：$\mathbf{AG}(\mathsf{Start} \rightarrow \mathbf{AF}\mathsf{Heat})$）

- モデル検査問題を解いて，状態の集合 $\{s \mid M, s \models \varphi\}$ を求める．この集合の中に**初期状態**（システムの実行開始時の状態）が入っていればOK!

**注意 6.1** 問題が与えられたとき，それを解くアルゴリズムが常に存在するとは限らない！（計算可能性，長谷川先生の授業参照）．また仮にアルゴリズムが存在したとしても，計算量が大きい（たとえば指数時間計算量，注意 1.1 を参照）ものは「使い物にならない」．

**仮定 6.2**（モデル検査のための）**アルゴリズム**で扱える入力はすべて有限．（無限のデータが納まる記憶媒体はないし，その処理には無限時間かかる！）なので，この節（§6）以降は，

- 原始論理式の集合 AP
- Kripke モデル $M = (S, R, V)$（定義 3.3）の状態集合 $S$

の両方が有限集合であると仮定する．

命題 5.8 と補題 5.9 から，チェックしたい **CTL** の（状態）論理式 $\varphi$ は次のように帰納的に定義されているとしてよい：

$$p \quad (p \in \mathsf{AP}) \ , \quad \neg\varphi \ , \quad \varphi \vee \psi \ , \quad \mathbf{EX}\varphi \ , \quad \mathbf{E}(\varphi \mathbf{U} \psi) \ , \quad \mathbf{EG}\varphi \ .$$

論理式 $\varphi$ の**構成について帰納的に**，集合

$$S_\varphi := \{s \in S \mid M, s \models \varphi\}$$

を求めていこう．

- $\varphi = p \in \mathsf{AP}$ のとき，$S_p$ は $V$（Kripke モデルの付値関数）を見ればわかる．すなわち，
$$S_p = \{s \in S \mid V(s, p) = \top\} \ .$$
- $\varphi = \psi \vee \chi$ のとき[*4]． 集合の和 $S_{\psi\vee\chi} = S_\psi \cup S_\chi$ をとる．
- $\varphi = \neg\psi$ のとき． 補集合 $S_{\neg\psi} = S \setminus S_\psi$ をとる．
- $\varphi = \mathbf{EX}\psi$ のとき．一つ先の状態で $\psi$ が成り立っている状態を集めてくる．すなわち，
$$S_{\mathbf{EX}\psi} = \{s \in S \mid (s, s') \in R \text{ かつ } s' \in S_\psi \text{ なる } s' \text{ が存在する}\}$$
- $\varphi = \mathbf{E}(\psi \mathbf{U} \chi)$ のとき．$\chi$ がなりたつ状態（$s \in S_\chi$）それぞれからスタートして，$R$ をさかのぼっていく．さかのぼっていくうち $\psi$ が成り立っているうちは，その状態を $S_{\mathbf{E}(\psi\mathbf{U}\chi)}$ に加えていく．正確には図 3 を見よ．
- $\varphi = \mathbf{EG}\psi$ のとき．ちょっと複雑なので，次節（§6.1）でゆっくり．

**注意 6.3** アルゴリズム *CheckEU* は**疑似コード**で書かれている．たとえば，集合演算（集合の差 $\setminus$ や集合の和 $\cup$ など）をサポートするプログラミング言語はほとんどないので，実際プログラムする際はそれらをどう実装するか考えなければならない．

**問題 6.4** アルゴリズム *CheckEU* が必ず停止することを証明せよ．また，アルゴリズムの正しさ（正しい $S_{\mathbf{E}(\psi\mathbf{U}\chi)}$ が求まること）を証明せよ．

[*4] 「論理式 $\varphi$ の**構成について帰納的に**」というのは，「小さい $\varphi$ から順番に」というような意味．なのでここでは，$S_\psi$ と $S_\chi$ の計算はもう終わっている．

**図 3** $S_\psi$ と $S_\chi$ を用いて $S_{\mathbf{E}(\psi\mathbf{U}\chi)}$ を計算するアルゴリズム

**procedure** *CheckEU*$(S_\psi, S_\chi)$
  $T := S_\chi$;
  $S_{\mathbf{E}(\psi\mathbf{U}\chi)} := S_\chi$;
  **while** $T \neq \emptyset$ do
    **choose** $s \in T$;
    $T := T \setminus \{s\}$;
    **for all** $t$ **such that** $R(t, s)$ **do**
      **if** $t \notin S_{\mathbf{E}(\psi\mathbf{U}\chi)}$ **and** $t \in S_\psi$ **then**
        $S_{\mathbf{E}(\psi\mathbf{U}\chi)} := S_{\mathbf{E}(\psi\mathbf{U}\chi)} \cup \{t\}$;
        $T := T \cup \{t\}$;
      **end if**;
    **end for all**;
  **end while**;
**end procedure**;

## 6.1 $S_{\mathbf{EG}\psi}$ を計算するアルゴリズム

おおざっぱなアイデアは次のとおり.

> $M, s \models \mathbf{EG}\psi$ のとき，パス $\pi = s, s_1, s_2, \ldots$ で $s_i \models \psi$ $(\forall i \in [1, \infty))$ なるものが存在する．が，$M = (S, R, V)$ の状態集合 $S$ は有限（仮定 6.2）だから，無限パス $\pi$ は最終的に「同じところを行ったり来たりしなければならない」

以下，このアイデアを数学的に正確にした上で，$S_{\mathbf{EG}\psi}$ を計算するアルゴリズムを考えよう.

**注意 6.5** Kripke フレーム（定義 3.2）は，より一般には**有向グラフ**と呼ばれる（点の集合とその間の線からなる構造で，しかも各々の線には向きがついている）．有向・無向グラフを扱うさまざまなアルゴリズムの研究は**グラフ理論**と呼ばれて盛んなので，その成果を使っていこう.

**定義 6.6** Kripke フレーム（= 有向グラフ）$F = (S, R)$ の**強連結成分**（strongly connected component）とは，$F$ の部分グラフ[*5] $C = (S', R')$ で次の条件を満たすもののことをいう.

- $C$ は強連結（strongly connected）．すなわち，かってな頂点 $s, s' \in S'$ をとったとき，$s$ から $s'$ への $C$ の中での有限パスが存在する：

  $$\exists s_0, s_1, \ldots, s_m \in S' \quad \text{such that} \quad s = s_0, \quad (s_0, s_1) \in R', \quad \ldots, \quad (s_{m-1}, s_m) \in R', \quad s_m = s'.$$

  この有限パスは長さ 0 でもいいことに注意.
- $C$ は極大（maximal）．すなわち，$C \subseteq C'$ でかつ $C'$ が強連結ならば，$C' = C$.（「$C$ に何か点を足すと，強連結でなくなる」）

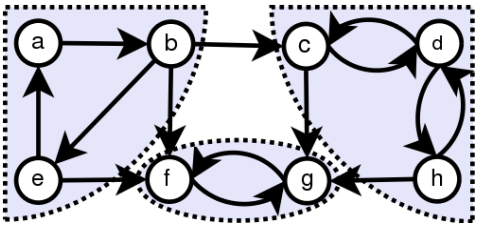
[*5] $(S, R)$ の部分グラフとは，$S' \subseteq S$ かつ $R' \subseteq R$ なるグラフ $(S', R')$ のこと．すなわち，「頂点と矢印をいくつか選んできて作ったグラフ」.

強連結成分 $C$ が**非自明**（non-trivial）とは，次のようで**ない**ことをいう：

$S$ が 1 点集合，かつ，$R = \emptyset$

（自明な強連結成分のなかでは，行ったり来たりできない！）

**図 4** グラフと強連結成分，出典：`http://en.wikipedia.org/wiki/Strongly_connected_components`




上でのべたアイデアの中の「同じところを行ったり来たりする」というのは，「ある強連結成分にとどまる」と言い換えることができる（次の補題）．少し記号を導入しよう：到達可能関係 $R$ の $S_\psi$ への制限を $R_\psi$ と書く．すなわち，

$$R_\psi \ := \ R \cap (S_\psi \times S_\psi) \ .$$

**補題 6.7** $M, s \models \mathbf{EG}\psi$ であることと，次の 2 つの条件がみたされることは同値．

1. $s \in S_\psi$.
2. 有向グラフ $(S_\psi, R_\psi)$ の中に非自明な強連結成分 $C$ があって，かつ $s$ から $C$ の中の状態 $t$ に至る $(S_\psi, R_\psi)$ の中の有限パスが存在する．

*Proof.*（sketch）[$\Longrightarrow$] $M, s \models \mathbf{EG}\psi$ の定義から，$s$ から始まる $S_\psi$ の中の無限パス $\pi$ が存在する．$\pi$ の中に現れる状態は有限種類しかないから（仮定 6.2），かならず「$\pi$ に無限回現れる状態」がある．そうでない（すなわち有限回しか現れない）状態たちを考えると，それらが現れるのは $\pi$ のうち初めの有限の長さの部分だけであるから，$\pi$ は次のように分解できる：

$$\pi = \pi_0 \cdot \pi_1 \ , \quad \text{ただし}$$

- $\pi_0$ は（当然）有限長，
- $\pi_1$ は次を満たす：状態 $s$ が $\pi_1$ に（一度でも）現れたら，$s$ は $\pi_1$ で無限回現れる．

$\pi_1$ に現れる状態の集合を $S'$ とすると，$\pi_1$ の定義から $S'$ は明らかに強連結で非自明[*6]．それを極大なものに拡大して $C$ とすると（問 6.9），$C$ は非自明な強連結成分．さらに $\pi_0$ が $s$ から $C$ へ至る有限パスを与える．

[$\Longleftarrow$] 存在が仮定された $s$ から $t$ への有限パスを $\pi_0$ と書く．$C$ は非自明な強連結成分だから，$t$ から $t$ への有限パスがかならず存在する．これを $\pi_1$ と書こう．すると，$\pi_0$ のあと $\pi_1$ を無限回繰り返す無限パス

$$\pi := \pi_0 \cdot \pi_1^\omega$$

は $s$ から始まる無限パスで，$S_\psi$ の中にある． □

補題 6.7 から，$S_{\mathbf{EG}\psi}$ を計算するアルゴリズム *CheckEG* が導ける（図 5）．この節（§6）はじめの，他の様相演算子に対するアルゴリズムと合わせて，**CTL** のモデル検査のアルゴリズムができた．

[*6] 正確に言うと，「$S'$ と，$R$ の $S'$ への制限とが定める部分グラフは強連結で非自明」．

**注意 6.8** ここで述べた，**CTL** モデル検査のためのアルゴリズムの計算量は，$\mathcal{O}(|\varphi| \cdot (|S| + |R|))$，すなわち線形時間．

**図 5** $S_\psi$ を用いて $S_{\mathbf{EG}\psi}$ を計算するアルゴリズム

**procedure** *CheckEG*$(S_\psi)$
  $\mathsf{SCC} := \{S_\psi$ の非自明強連結成分 $\}$;　　//たとえば Tarjan のアルゴリズム [1] を使う
  $T := \bigcup_{C \in \mathsf{SCC}} \{s \mid s \in C\}$;
  $S_{\mathbf{EG}\psi} := T$;
  **while** $T \neq \emptyset$ do
    **choose** $s \in T$;
    $T := T \setminus \{s\}$;
    **for all** $t \in S_\psi$ **such that** $R(t, s)$ **do**
      **if** $t \notin S_{\mathbf{E}(\psi \mathbf{U} \chi)}$ **then**
        $S_{\mathbf{E}(\psi \mathbf{U} \chi)} := S_{\mathbf{E}(\psi \mathbf{U} \chi)} \cup \{t\}$;
        $T := T \cup \{t\}$;
      **end if**;
    **end for all**;
  **end while**;
**end procedure**;

**問題 6.9**　1. 空でなく強連結な任意の部分グラフ $C$ は，強連結成分に拡張できることを示せ．つまり，強連結成分 $C'$ で $C \subseteq C'$ なるものが存在する．

2. Kripke フレーム $F = (S, R)$ の状態集合 $S$ 上の 2 項関係 $\cong$ を，「同じ強連結成分に属する」という関係として定める．すなわち，

$$s \cong s' \quad \overset{\text{def}}{\Longleftrightarrow} \quad \text{ある（非自明でなくてもよい）強連結成分 } C \text{ が存在して，} s, s' \in C.$$

この $\cong$ が同値関係（定義 3.11）であることを証明せよ．

**例 6.10** 例 4.2 の Kripke モデルを $M = (S, R, V)$，いちばん上にある状態を $s_0$ と書くとき，

$$M, s_0 \models \mathbf{AG}(\mathsf{Start} \to \mathbf{AF}\mathsf{Heat})$$

がなりたつかどうか判定しよう．

1. まず，チェックしたい論理式を $\mathbf{EX}, \mathbf{EG}, \mathbf{EU}$ のみを使った式に「書き換える」（補題 5.9）．
2. そののち上のアルゴリズムを適用．

# 7　公平性（fairness）

なぜ例 6.10 が失敗したのだろうか？

「ドアをずっと開けたり閉めたり」とかいうパスがある $\Longrightarrow$ このようなパスは排除したい！

このように「可能なパスを制限する」ことを**公平性**の制約（fairness constraint）と呼ぶ．許されるパスを**公平な**（fair）パスと呼ぶ．

**注意 7.1** なぜ「公平性」とよぶか？ たとえば：


tokenA ばかりがずっと成り立つ，つまり A が優先されるような実行パスは「公平でない」！

公平性の制約を数学的に次のように定義しよう．

**定義 7.2** 1. $M = (S, R, V)$ を Kripke モデルとするとき，$M$ 上の**公平性制約**（fairness constraint）とは，状態の集合の集まり[*7]

$$\mathsf{FC} \subseteq \mathcal{P}S$$

のことを言う．

2. $M$ の（無限）パス $\pi = s_0, s_1, \ldots$ に対して，状態の集合 $\inf(\pi)$ を

$$\inf(\pi) := \{s \in S \mid \text{無限にたくさんの } i \geq 0 \text{ に対して}, s = s_i\}$$

と定める．つまり，$\inf(\pi)$ は $\pi$ に無限回現れる状態の集合．

3. $M$ の（無限）パス $\pi = s_0, s_1, \ldots$ が公平性制約 FC について**公平**（fair）であるとは，次がなりたつことをいう：

$$\text{すべての } P \in \mathsf{FC} \text{ に対して}, \qquad \inf(\pi) \cap P \neq \emptyset \ .$$

**例 7.3** 公平性制約

$$\mathsf{FC} = \big\{\, \{s, s'\}, \{s''\} \,\big\}$$

は次のように読める：

「状態 $s$ を無限回通るか，状態 $s'$ を無限回通る」$(*)$，かつ
「状態 $s''$ を無限回通る」ような
パスのみを考える．

さらに，条件 $(*)$ は，

状態の集合 $\{s, s'\}$ を無限回通る，つまり $\big\{\, i \geq 0 \mid s_i \in \{s, s'\} \,\big\}$ は無限集合

とも言い換えられる．（なぜ？）

**CTL$^*$，CTL，LTL** の論理式の semantics を，公平性を勘案したものに修正しよう．

**定義 7.4 (公平性制約のもとでの様相論理式の解釈)** Kripke モデル $M = (S, R, V)$ と公平性制約 FC を定めたとき，状態・パス論理式 $\varphi$ が $M$ と FC のもとで**真である**という関係

$$M, s \models_{\mathsf{FC}} \varphi \quad \text{または} \quad M, \pi \models_{\mathsf{FC}} \varphi$$

[*7] つまり，「状態の集合の集合」

を次のように定める．

$$
\begin{aligned}
M, s \models_{\mathsf{FC}} p \quad (p \in \mathsf{AP}) \quad &\overset{\text{def}}{\iff} \quad V(s,p) = \top \text{ で，かつ } s \text{ から始まる } \textbf{fair} \text{ なパス } \pi \text{ が存在する．} \\
M, s \models_{\mathsf{FC}} \mathbf{E}\varphi \quad &\overset{\text{def}}{\iff} \quad s \text{ から始まる } \textbf{fair} \text{ なパス } \pi \text{ で，} M, \pi \models_{\mathsf{FC}} \varphi \text{ となるようなものがある} \\
M, s \models_{\mathsf{FC}} \mathbf{A}\varphi \quad &\overset{\text{def}}{\iff} \quad s \text{ から始まるすべての } \textbf{fair} \text{ なパス } \pi \text{ について，} M, \pi \models_{\mathsf{FC}} \varphi \text{ がなりたつ}
\end{aligned}
$$

$\varphi$ が $p, \mathbf{E}\varphi, \mathbf{A}\varphi$（状態論理式）以外の場合は図 1 と同じ．

次に，公平性制約のもとでの **CTL** モデル検査のアルゴリズムを考えよう．すなわち，

**入力：** Kripke モデル $M = (S, R, V)$，公平性制約 $\mathsf{FC}$，それと **CTL** の状態論理式 $\varphi$
**出力：** $\mathsf{FC}$ のもとで $\varphi$ をみたす状態 $s$ の集合，すなわち $\{s \in S \mid M, s \models_{\mathsf{FC}} \varphi\}$

**CTL** の場合と同様に，論理式の構成について帰納的に集合

$$S_\varphi := \{s \in S \mid M, s \models_{\mathsf{FC}} \varphi\}$$

を求めていく．まず最初に，$S_\psi$ から $S_{\mathbf{EG}\psi}$ を求めるアルゴリズムを考えよう．

アイデアは §6.1 で述べた強連結成分を使うアルゴリズムと一緒．しかし今の場合は，強連結成分（最終的にそのなかを行ったり来たりする部分）の中に，公平性制約 $\mathsf{FC}$ が「無限回訪れることを要請している」状態がきちんと含まれていなければならない．

**定義 7.5** $M = (S, R, V)$ を Kripke モデル，$\mathsf{FC}$ をその上の公平性制約とする．Kripke フレーム $(S, R)$ の強連結成分 $C$ が**公平**（fair）であるとは，各 $P_i \in \mathsf{FC}$ について $C \cap P_i \neq \emptyset$ がなりたつことを言う．

**補題 7.6** $M, s \models_{\mathsf{FC}} \mathbf{EG}\psi$ であることと，次の 2 つの条件がみたされることは同値．

1. $s \in S_\psi$，すなわち $M, s \models_{\mathsf{FC}} \psi$.
2. $(S_\psi, R_\psi)$ の中に非自明で fair な強連結成分 $C$ があって，かつ $s$ から $C$ の中の状態 $t$ に至る $(S_\psi, R_\psi)$ の中の有限パスが存在する．

*Proof.* 補題 6.7 と同様． □

よって $S_\psi$ から $S_{\mathbf{EG}\psi}$ を求めるアルゴリズムは，*CheckEG*（図 5）において 2 行目を

$$\mathsf{SCC} := \{S_\psi \text{ の非自明で fair な強連結成分 }\}$$

に変更したものである．

その他の形の論理式（$\varphi = \psi \vee \chi, \neg\psi, \mathbf{EX}\psi, \mathbf{E}(\psi \mathbf{U} \chi)$）について $S_\varphi$ を計算しよう．その際論理式 $\mathbf{EG}\mathsf{True}$ が重要：この論理式を以下 $\mathsf{Fair}$ と略記する．

$$\mathsf{Fair} := \mathbf{EG}\mathsf{True}, \qquad \text{ただし } \mathsf{True} \text{ は例 5.6 と同じ}$$

すると次がなりたつ．

**補題 7.7** 1. $M, s \models_{\mathsf{FC}} \mathsf{Fair}$ であることと，$s$ から始まる fair なパス $\pi$ が存在することは同値．（$M, s \models_{\mathsf{FC}} \mathsf{True}$ はすべての状態 $s$ でなりたつことに注意）
2. $p \in \mathsf{AP}$ に対して，$M, s \models_{\mathsf{FC}} p$ であることと $M, s \models p \wedge \mathsf{Fair}$ であることは同値．
3. $M, s \models_{\mathsf{FC}} \mathbf{EX}\psi$ であることと $M, s \models \mathbf{EX}(\psi \wedge \mathsf{Fair})$ であることは同値．

4. $M, s \models_{\mathsf{FC}} \mathbf{E}(\psi \mathbf{U} \chi)$ であることと $M, s \models \mathbf{E}(\psi \mathbf{U}(\chi \wedge \mathsf{Fair}))$ であることは同値.

*Proof.* 定義より明らか（チェック！）．3，4 では，$\psi, \chi$ は状態論理式であることに注意（**CTL** だから）□

公平性制約なしの $M, s \models \varphi$ をチェックするアルゴリズムはもう知っているから（§6），以上から公平性制約付きのモデル検査を行うアルゴリズムをえた.

**例 7.8** 例 6.10 のモデル検査を，「ユーザが正しく操作する」という気持ちの次の公平性制約のもと，行ってみよう.

$$\mathsf{FC} = \left\{ s \mid s \models \mathsf{Start} \wedge \mathsf{Close} \wedge \neg \mathsf{Error} \right\}$$

# 8 その他のトピック

- **LTL** モデル検査（タブロー法による）
- **CTL**$^*$ モデル検査（**LTL** と **CTL** を組み合わせる）
- 記号的モデル検査（symbolic model checking）

# 9 参考書

前半の命題・様相論理については，日本語なら [4] が標準的な教科書（おすすめ！）．中盤からのモデル検査については，[2] に拠った.

# 参考文献


[1] Alfred V. Aho, John E. Hopcroft, and Jeffrey D. Ullman. *The Design and Analysis of Computer Algorithms*. Addison-Wesley, 1974.

[2] E.M. Clarke, O. Grumberg, and D.A. Peled. *Model Checking*. MIT Press, 1999.

[3] Rance Cleaveland. THERE AND BACK AGAIN: Lessons learned on the way to the market. Invited talk at ETAPS 2007, 2007.

[4] 小野 寛晰. **情報科学における論理**. 日本評論社, 1994.

[5] 渡辺 治. 計算量理論（入門編）講義ノート. `http://www.is.titech.ac.jp/~watanabe/class/comp/part1.pdf`, 2002. Revised in 2005.

## 10 notes for myself

- muddy children puzzle:
  - common knowledge: at least one dirty child
  - in a round: all the children are asked, their answers heard by all children.
  - prove: a child who sees k dirty children says "Yes I know" at in round k, but no earlier.

# 11　レポート課題

**問題 11.1** かってな原子論理式 $p, q, r$ について，次の命題論理式が恒真であることを示せ．

1. $p \to (q \to p)$
2. $(p \to (q \to r)) \to ((p \to q) \to (p \to r))$

**問題 11.2** 問 3.13 に答えよ．

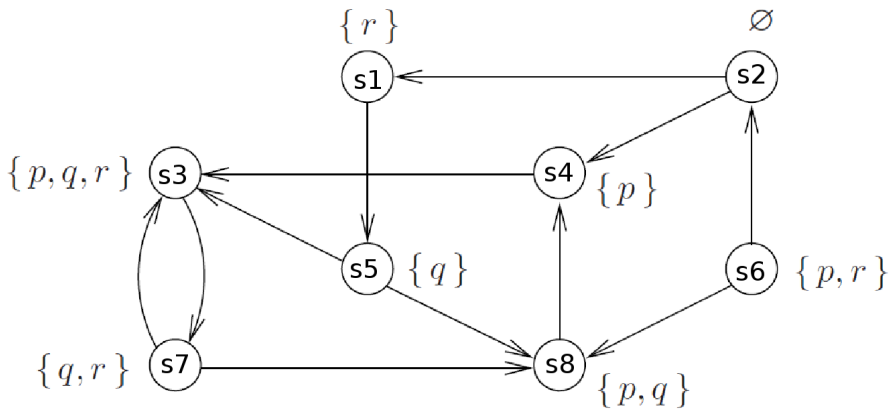
**問題 11.3** 定理 3.10 の項目 3–5 を証明せよ．

**問題 11.4 CTL** 論理式の論理的同値性

$$\mathbf{A}(\varphi \mathbf{R} \psi) \cong \neg \mathbf{E}(\neg\varphi \mathbf{U} \neg\psi)$$

を証明せよ．

**問題 11.5** $\mathsf{AP} = \{p, q, r\}$ とする．Kripke モデル




において，状態の集合

$$\{s \mid s \models \mathbf{EF}((p \leftrightarrow r) \wedge \neg(p \leftrightarrow q))\}$$

を求めよ．ただし，$\varphi \leftrightarrow \psi$ は $(\varphi \to \psi) \wedge (\psi \to \varphi)$ の略（「$\varphi$ と $\psi$ の真偽が一致する」という気持ち）．上の図において各状態に付記されているのは，その状態において成り立つ原子論理式．たとえば，$V(s_4, p) = \top, V(s_4, q) = V(s_4, r) = \bot$ である．