

## モデル検査(2)

---

### 時相論理

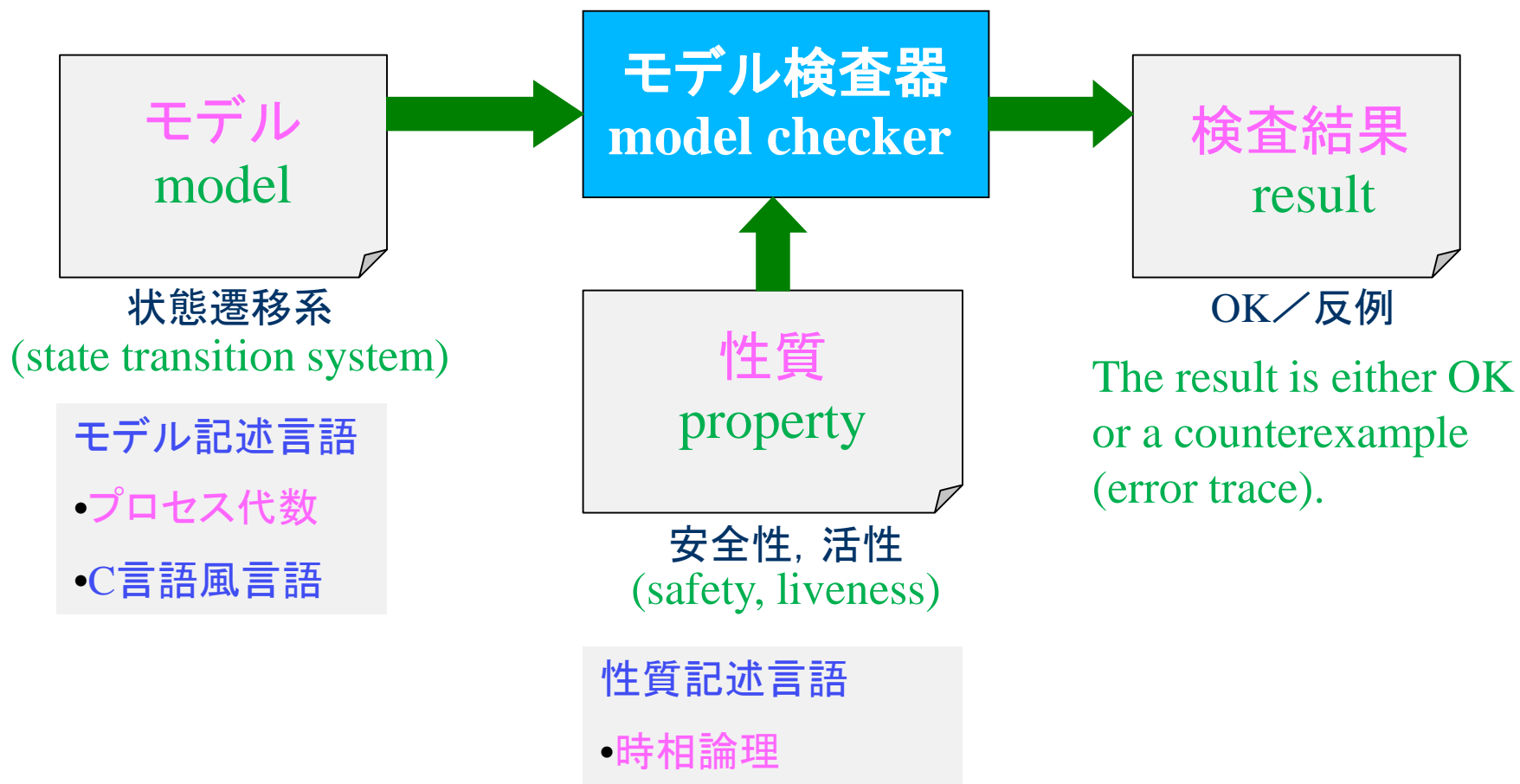
## Model Checking (2) Temporal Logic

- |                |  |
|----------------|--|
| 1. 時相論理の概要     | 1. Overview of temporal logic                |
| 2. 状態遷移系の形式化   | 2. Formalization of state transition systems |
| 3. 線形時相論理 LTL  | 3. Linear Temporal Logic: LTL                |
| 4. 計算木時相論理 CTL | 4. Computational Tree Logic: CTL             |

### ■Reference

Model Checking, E.M. Clarke, Jr. et al, MIT Press (1999)

# モデル検査器の概要 (Overview of model checker)



# 1. 時相論理の概要 (Overview of temporal logic)

---

## 逐次プログラムの検証

- 古典論理による入出力関係の表現で十分.

To verify sequential programs, it is enough to represent the systems by input/output relationships in classical logic.

## リアクティブシステム, 並行システムの検証

- 通常は非停止. 内部の状態遷移の表現が重要.

To verify reactive, concurrent systems, which are often non-terminating, it is important to represent the internal state transitions of the systems in non-classical logic.

---

# 時相論理の概要 (Overview of temporal logic)

---

## 時相論理

Temporal logic allows you to

- リアクティブシステム, 並行システムの性質を記述できる.  
describe properties of reactive, concurrent systems,
  - 時間にかかわる表現が可能.  
describe properties related to time,
  - 時相演算子で状態遷移の性質を表現.  
describe properties of state transition using temporal operators.
  - 線形時相論理 LTL  
LTL: Linear Temporal Logic
  - 計算木時相論理 CTL  
CTL: Computational Tree Logic
-

## 2. 状態遷移系の形式化

(Formalization of state transition systems)

---

$V = \{v_1, \dots, v_n\}$ : 並行システムにおけるシステム変数の集合  
(the set of variables in a concurrent system)

$D$ : 各変数がとる値の有限集合 (領域)  
(domain: a finite set of values taken by the variables)

$s = (a_1, \dots, a_n)$ : システム変数のとる値 (状態) のベクトル  
(state: a vector of values taken by the variables)

$AP$ : 状態から真偽値が定まる原始命題の集合

例: 「 $v_1 = 3$ 」は原始命題

(the set of atomic propositions: Their truth values are determined in each state. For example, “ $v_1=3$ ” is an atomic proposition whose value is true in the states where the value of  $v_1$  is 3.)

---

# クリプキ構造 (Kripke structure)

クリプキ構造  $M = (S, S_0, R, L)$  : 状態遷移系の一種

State transition systems are formally modeled by the Kripke structure defined as 4-tuple.

1.  $S$ : 状態の有限集合. ( $S$  is a finite set of states)
2.  $S_0 \subseteq S$ : 初期状態の集合. ( $S_0 \subseteq S$  is a set of initial states)
3.  $R \subseteq S \times S$ : 状態遷移関係. ( $R \subseteq S \times S$  is a state transition relation)
4.  $L: S \rightarrow 2^{AP}$ : 各状態に「その状態で真である原始命題の集合」をラベル付けする関数.

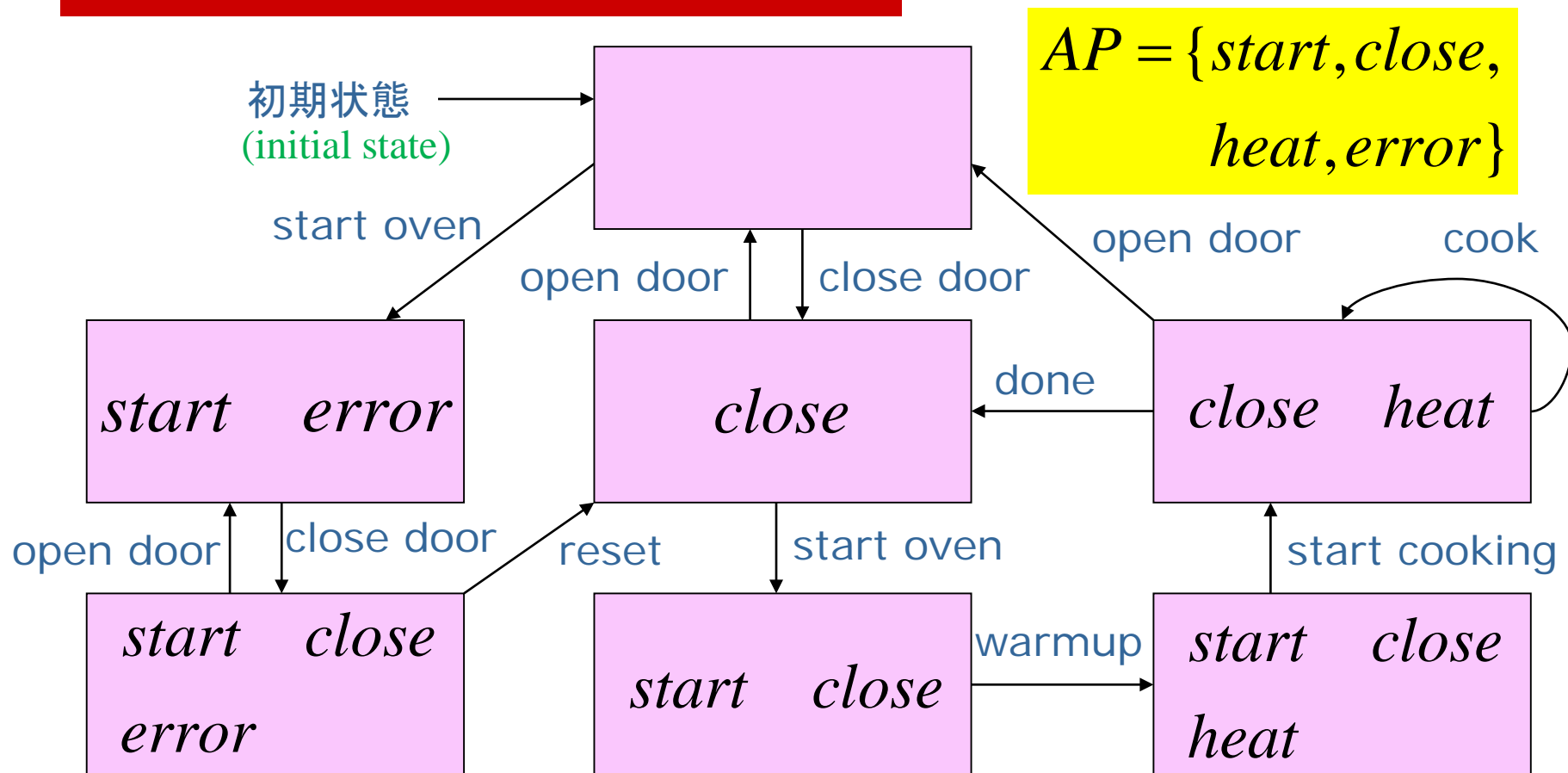
ベキ集合  
( $AP$ のすべての部分集合の集合)

powerset  
(the set of all subsets of  $AP$ )

( $L: S \rightarrow 2^{AP}$  is a labeling function, which assigns each state in  $S$  with a label, i.e. a subset of  $AP$  that are true in that state)

# クリプキ構造の例 (オーブンレンジ)

(An example of Kripke structure: an oven)

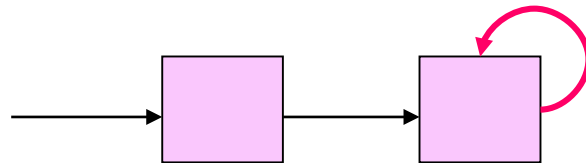


# パス (Path)

どの状態  $s$  においても必ず遷移先の状態  $s'$  があるものとする。

この条件が成り立たないときは、 $s$  から  $s$  自身に遷移可能なように  $R$  を修正。

We assume every state  $s$  has a state  $s'$  to which transition from  $s$  is possible. Otherwise, we will fix  $R$  so that transition from  $s$  to  $s$  itself is possible.



状態  $s$  からのパス (path): 状態の無限列  $\pi = (s_0 s_1 s_2 \dots)$ .

ただし、 $s_0 = s$  ,  $(s_i, s_{i+1}) \in R$  (for all  $i \geq 0$ ).

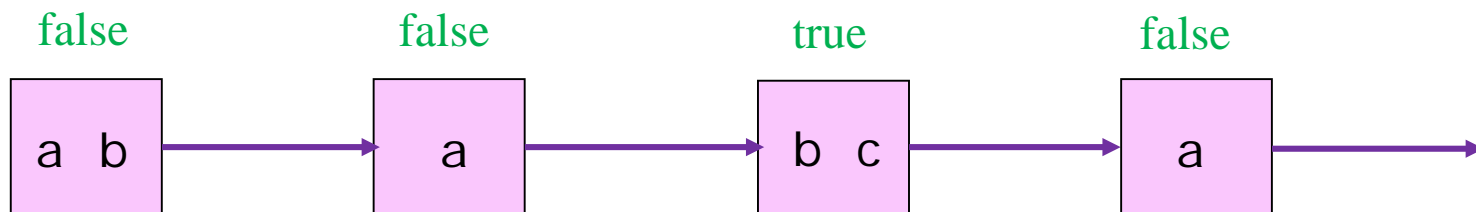
A path from a state  $s$  is an infinite sequence of states  $\pi = (s_0 s_1 s_2 \dots)$ .  
where  $s_0 = s$  and  $(s_i, s_{i+1}) \in R$  for all  $i$ .



### 3. 線形時相論理 LTL (Linear Temporal Logic: LTL)

---

Is  $c$  true in this path?



このパスでは、いつか必ず  $c$  が成り立つ

LTLでは、この性質を  $Fc$  で表す

In this path,  $c$  will be eventually true.

In LTL, this property is expressed as  $Fc$ .

---

# LTLの構文論 (Syntax of LTL)

LTL式の構文を時相演算子 **X, F, G, U** を用いて帰納的に定義.

The syntax for LTL formulas are defined inductively by using temporal operators **X, F, G, and U** as follows.

## LTL式 (LTL formulas)

- $p \in AP$  が原始命題ならば,  $p$  は LTL 式である.

An atomic proposition  $p$  is an LTL formula.

- $f, g$  が LTL 式ならば,

$\neg f, f \vee g, f \wedge g, f \rightarrow g$

もそれぞれ LTL 式である.

- $f, g$  が LTL 式ならば,

$Xf, Ff, Gf, f U g$

もそれぞれ LTL 式である.

These eight formulas are LTL formulas, if  $f$  and  $g$  are LTL formulas.

# LTLの直観的な意味(1/2)

(Intuitive meaning of LTL)

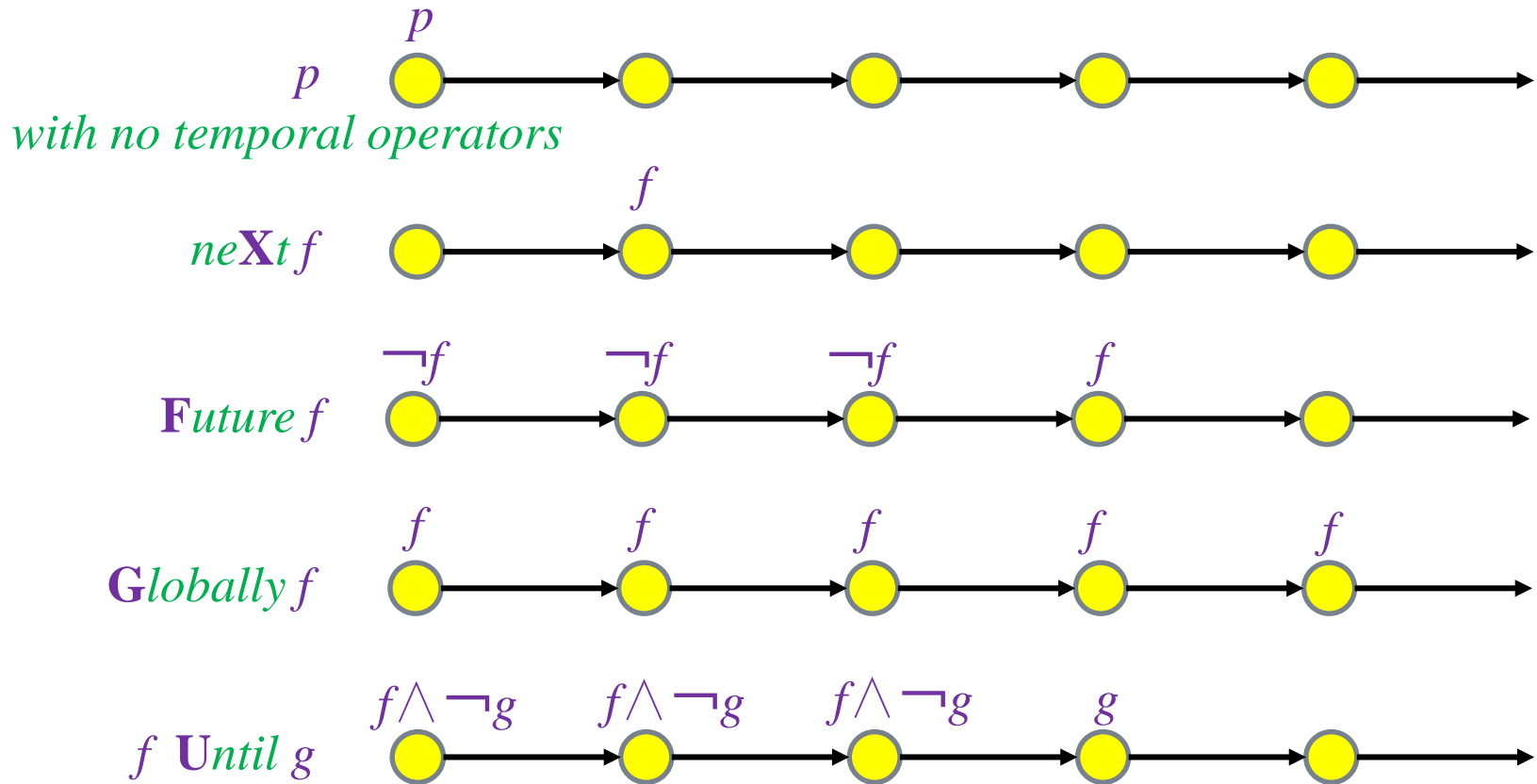
LTL 式の真偽は, 状態遷移系上のパスごとに定まる.

The truth of an LTL formula is determined for each path.

<b>Xf</b>	neXt time	パス上の次の状態で $f$ が成り立つ $f$ is true in the <i>neXt</i> state on the path.
<b>Ff</b>	Finally, Future, eventually	パス上のある状態でいつかは $f$ が成り立つ $f$ will be <i>Finally</i> true in some state on the path.
<b>Gf</b>	Globally, always	パス上のすべての状態でいつも $f$ が成り立つ $f$ is true in all the states ( <i>Globally</i> ) on the path.
<b>fUg</b>	Until	パス上のある状態で $g$ が成り立ち, かつ その直前までのすべての状態で $f$ が成り立つ $g$ is true in some state $s$ on the path, and $f$ is true in all the states before $s$ ( <i>Until</i> $g$ is true).

# LTLの直観的な意味(2/2)

(Intuitive meaning of LTL)



# LTLモデル検査器 (LTL Model Checker)

---

SPIN など

## LTLモデル検査器

モデルおよびLTL式  $f$  が入力されると、そのモデルの初期状態から始まるすべてのパスについて、性質  $f$  が成り立つかどうかを検査する。

$f$  が成り立たないようなパスがあれば、それを反例として出力する。

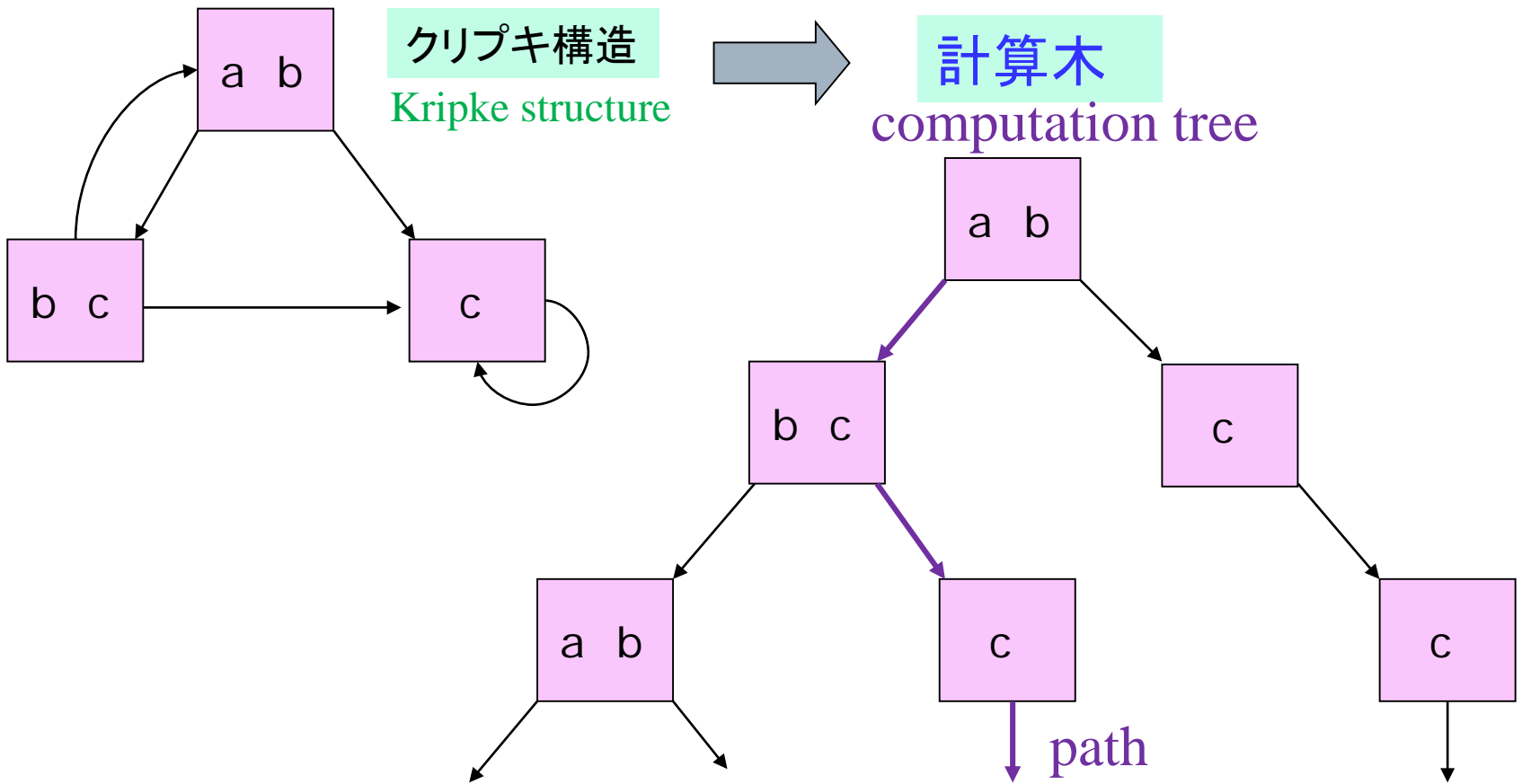
When an LTL formula  $f$  is input to an LTL model checker such as SPIN, it checks all the paths  $\pi$  starting from the initial states to see if  $\pi \models f$ .

If it finds a path with  $\pi \models \neg f$ , it will output that path.

---

# 4. 計算木時相論理 CTL

(Computation Tree Logic: CTL)



# CTLの構文論 (Syntax of CTL)

CTL では, LTL の時相演算子に加えて, **パス限量子 E と A** を用いる.

(CTL uses the path quantifiers **E** and **A** in addition to LTL temporal operators)

## CTL式 (CTL formulas)

- $p \in AP$  が原始命題ならば,  $p$  は CTL 式である.

An atomic proposition  $p$  is a CTL formula.

- $f, g$  が CTL 式ならば,

$$\neg f, f \vee g, f \wedge g, f \rightarrow g$$

もそれぞれ CTL 式である.

- $f, g$  が CTL 式ならば,

$$\mathbf{E}x f, \mathbf{E}f, \mathbf{E}g f, \mathbf{E}(f \mathbf{U} g), \mathbf{A}x f, \mathbf{A}f, \mathbf{A}g f, \mathbf{A}(f \mathbf{U} g)$$

もそれぞれ CTL 式である.

These 12 formulas are CTL formulas, if  $f$  and  $g$  are CTL formulas.

# CTLの直観的な意味: E, A

(Intuitive meaning of CTL: E, A)

CTL 式の真偽は, 着目する状態から始まる計算木ごとに定まる.

The truth of a CTL formula is determined for each computation tree starting from a state.

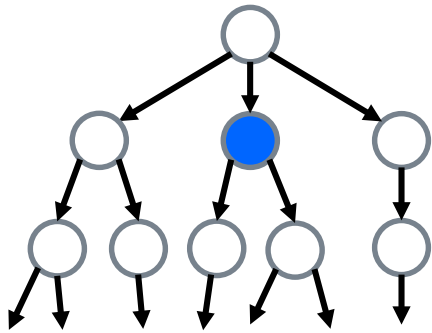
<b>Ef</b>	there <b>E</b> xists a path	この状態から始まるあるパスにおいて $f$ が成り立つ.  There exists a path in the computation tree for which $f$ is true.
<b>Af</b>	for <b>A</b> ll paths	この状態から始まるすべてのパスにおいて $f$ が成り立つ.  $f$ is true for all paths in the computation tree.



# CTLの直観的な意味：Eの使用例

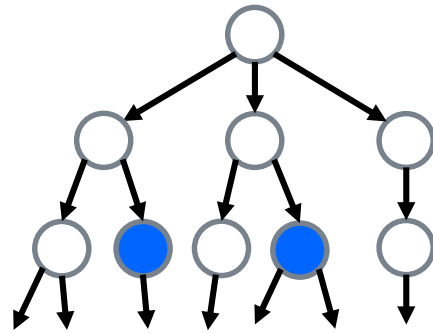
(Intuitive meaning of CTL: Sample use of E)

---



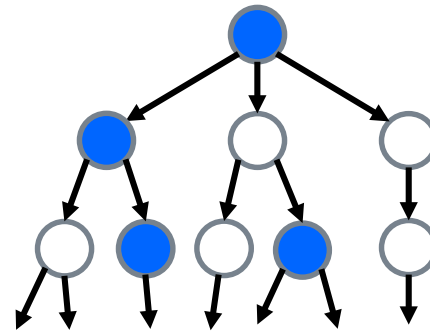
**EX** blue

There **exists** a path such that the **next** state is blue.



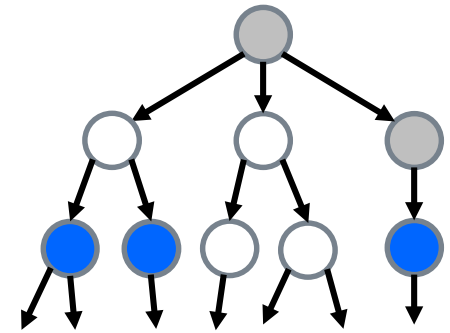
**EF** blue

There **exists** a path that will, in **future**, get blue.



**EG** blue

There **exists** a path that is **globally** blue.



**E**(gray **U** blue)

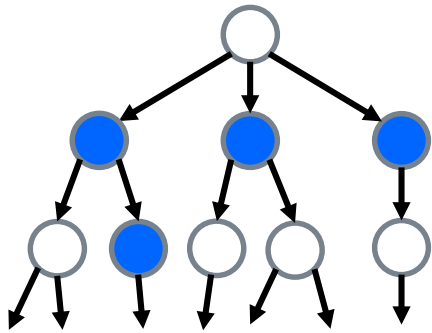
There **exists** a path that is gray **until** blue.

---

# CTLの直観的な意味: Aの使用例

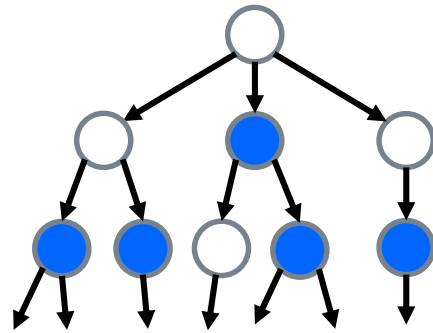
(Intuitive meaning of CTL: Sample use of A)

---



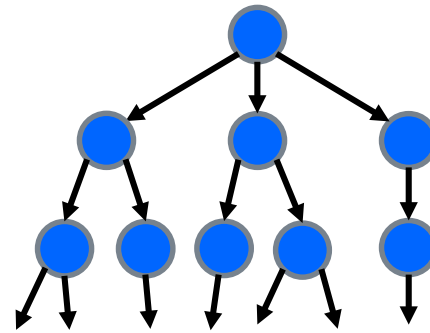
$AX$  blue

For all paths, the next state is blue.



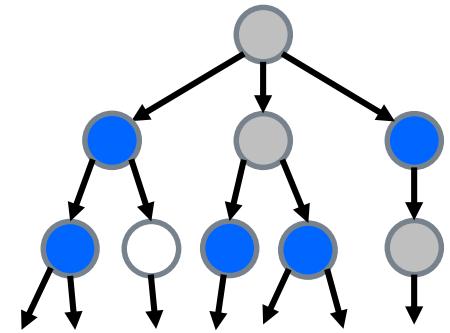
$AF$  blue

For all paths, it will finally become blue.



$AG$  blue

For all paths, it is globally blue.



$A(\text{gray } U \text{ blue})$

For all paths, it is gray until blue.

---

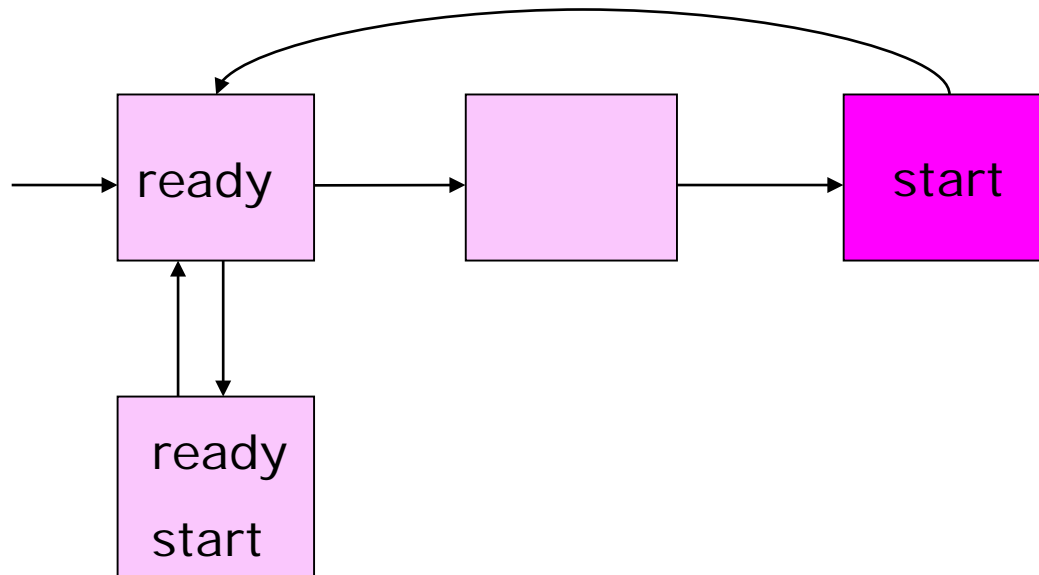
## CTL式の例(1/4) (Example of CTL formula)

---

**EF**(*start* ∧ ¬*ready*)

*start* は真だが *ready* が偽である状態に至る経路が存在する。

There exists a path where in future we will see that *start* is true but *ready* is not.



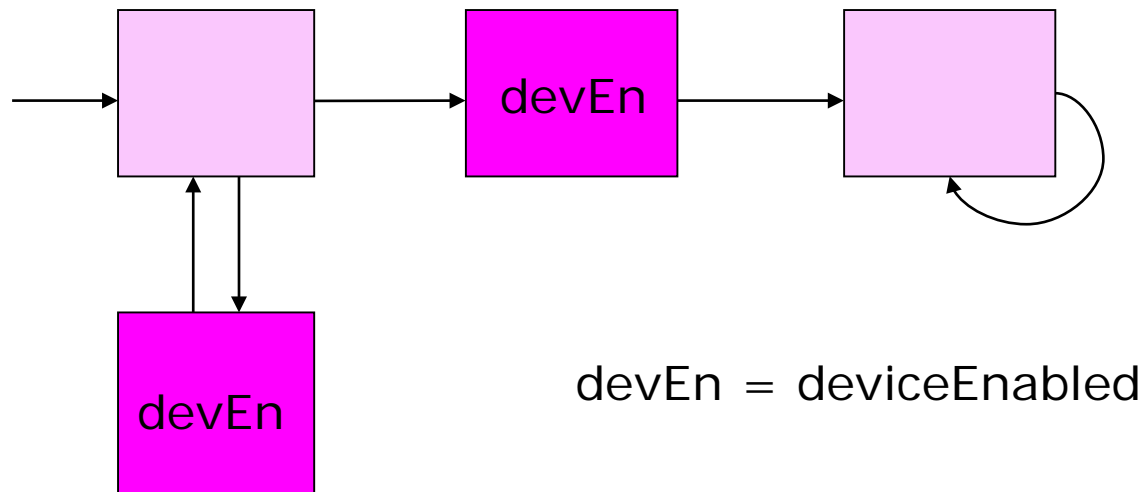
## CTL式の例(2/4) (Example of CTL formula)

---

### **AF** *deviceEnabled*

どの経路においても、いつか *deviceEnabled* が真となる。

For all paths, we will see that *deviceEnabled* will be finally true.



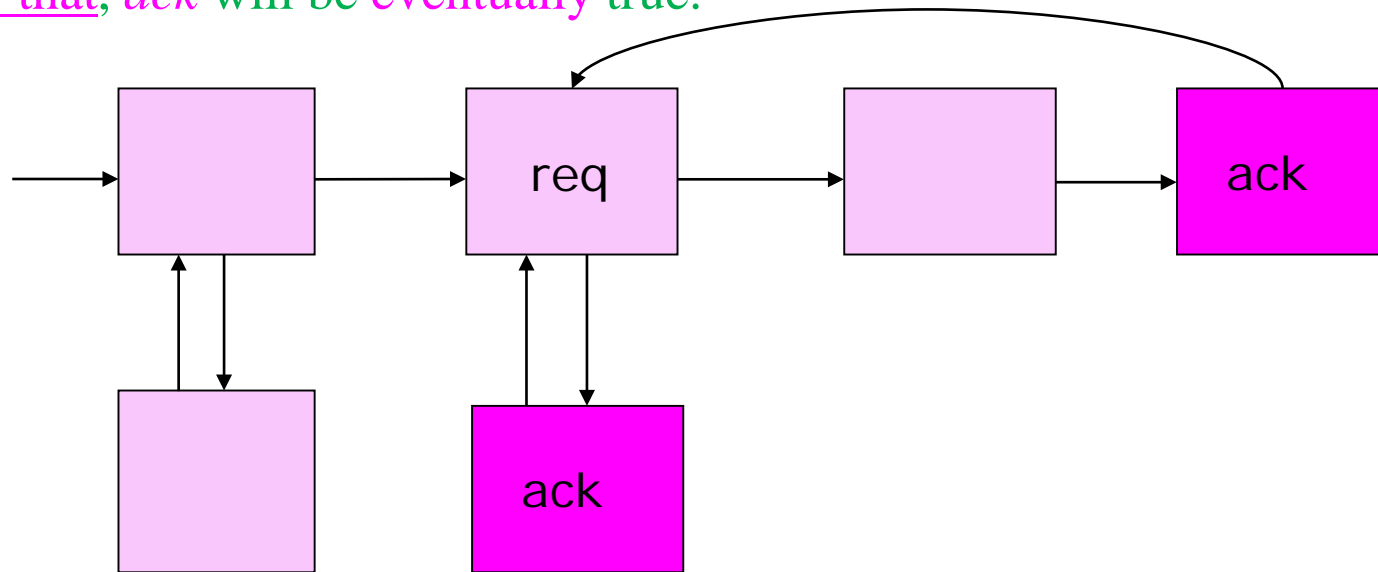
## CTL式の例(3/4) (Example of CTL formula)

---

**AG(req → AF ack)**

どの経路のどの状態においても, *req* が真ならば, いつかは *ack* が真となる.

In any path, it is always true that if *req* is true at that time, then for all paths after that, *ack* will be eventually true.



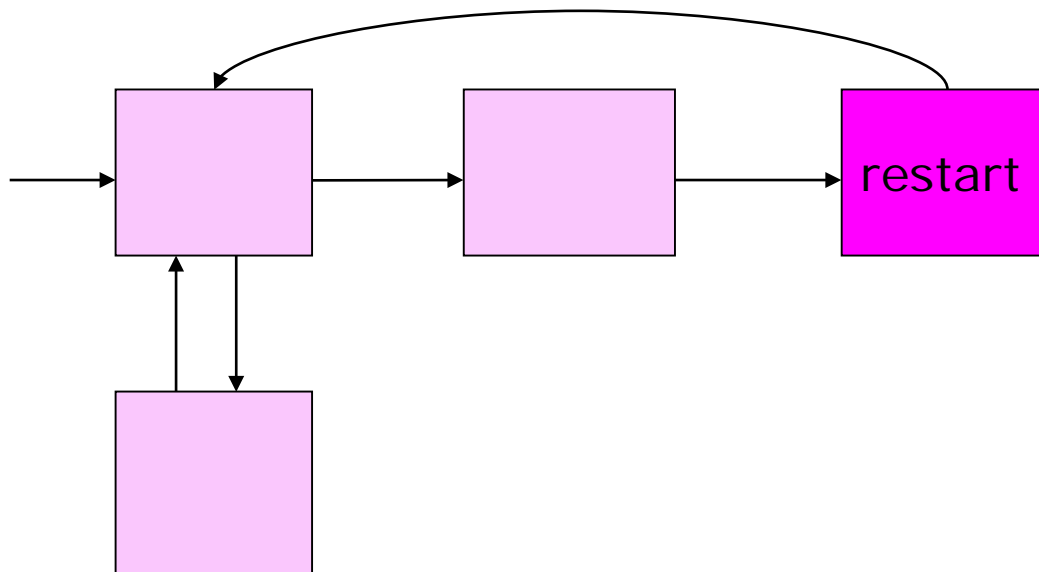
## CTL式の例(4/4) (Example of CTL formula)

---

### **AG(EF restart)**

どの状態からも、(経路を適切に選べば) *restart* 状態に到達可能である。

From any state, there is a path to reach a restart state.



# 演習問題4

## Exercise 4

---

つぎの等式を証明せよ.

(1)  $\mathbf{F}f = \mathbf{true} \mathbf{U} f$

(2)  $\neg \mathbf{G}f = \mathbf{F} \neg f$

(3)  $\neg \mathbf{A}f = \mathbf{E} \neg f$

(4)  $\neg \mathbf{X}f = \mathbf{X} \neg f$

Prove the following identical equations.

参考

$$\neg(\forall x)P = (\exists x)\neg P$$

$$\neg(\exists x)P = (\forall x)\neg P$$

---

---

以下のスライドは, 参考までに, LTLとCTLの形式的な意味論について  
やや詳しく書いたものです

## 参考

---



# LTLの形式的意味論(1/3) 表記法

## (Formal semantics of LTL: Notation)

---

パス  $\pi = (s_0 s_1 \dots)$

A path is represented by an infinite sequence  $\pi = (s_0 s_1 s_2 \dots)$  of states.

$\pi^i = (s_i s_{i+1} \dots)$ : 第  $i$  番目以降の部分パス

A subpath of  $\pi$  starting from  $s_i$ , i.e.  $(s_i s_{i+1} \dots)$ , is denoted by  $\pi^i$ .

$\pi \models f$ : LTL 式  $f$  がパス  $\pi$  において真 ( $\pi$  は  $f$  を満たす)

The LTL formula  $f$  is true for the path  $\pi$  ( $\pi$  fulfills  $f$ ).

$\pi \not\models f$ : 上式の否定

The negation of the above formula

---

# LTLの形式的意味論(2/3) 時相演算子を含まないとき

(Formal semantics of LTL: When no temporal operators involved)

LTL 式の真偽値の帰納的な定義

( $p \in AP$  は原始命題,  $f, g$  は LTL 式)

$$\pi \models p \iff p \in L(s_0)$$

$$\pi \models \neg f \iff \pi \not\models f$$

$$\pi \models f \vee g \iff \pi \models f \text{ or } \pi \models g$$

$$\pi \models f \wedge g \iff \pi \models f \text{ and } \pi \models g$$

$$\pi \models f \rightarrow g \iff \pi \models \neg f \vee g$$

The inductive definition of the truth for LTL formulas, where  $p$  is an atomic proposition and  $f, g$  are LTL formulas.

$s_0$ に付けられた  
ラベル(原始命題の集合)

the label (a subset of  $AP$ ) attached to  $s_0$

$f$  が時相演算子を含まないとき,

$f$  の真偽値はパス  $\pi$  の最初の状態  $s_0$  における真偽値となる.

If  $f$  involves no temporal operators,  
the truth of  $f$  is its truth in the initial state  $s_0$  of the path  $\pi$ .

# LTLの形式的意味論(3/3) 時相演算子を含むとき

(Formal semantics of LTL: When temporal operators involved)

---

$$\pi = (s_0 s_1 s_2 \dots)$$

$$\pi^i = (s_i s_{i+1} \dots)$$

$$\pi \models \mathbf{X}f \quad \Leftrightarrow \pi^1 \models f$$

$$\pi \models \mathbf{F}f \quad \Leftrightarrow \text{there exists a } k \geq 0 \text{ such that } \pi^k \models f$$

$$\pi \models \mathbf{G}f \quad \Leftrightarrow \text{for all } i \geq 0, \pi^i \models f$$

$$\pi \models f \mathbf{U}g \quad \Leftrightarrow \text{there exists a } k \geq 0 \text{ such that } \pi^k \models g \text{ and} \\ \text{for all } 0 \leq j < k, \pi^j \models f$$

# CTLの形式的意味論(1/3) 表記法

## (Formal semantics of CTL: Notation)

---

$s \models f$  : CTL 式  $f$  が, 状態  $s$  から始まる計算木において真  
( $s$  は  $f$  を満たす)

The CTL formula  $f$  is true for the computation tree starting from the state  $s$  ( $s$  fulfills  $f$ ).

$s \not\models f$  : 上式の否定 (The negation of the above formula)

---

# CTLの形式的意味論(2/3) 時相演算子を含まないとき

(Formal semantics of CTL: When no temporal operators involved)

CTL 式の真偽値の帰納的な定義  
( $p \in AP$  は原始命題,  $f, g$  は CTL 式)

$$\begin{aligned} s \models p &\iff p \in L(s) \\ s \models \neg f &\iff s \not\models f \\ s \models f \vee g &\iff s \models f \text{ or } s \models g \\ s \models f \wedge g &\iff s \models f \text{ and } s \models g \\ s \models f \rightarrow g &\iff s \models \neg f \vee g \end{aligned}$$

The inductive definition of the truth for CTL formulas, where  $p$  is an atomic proposition and  $f, g$  are CTL formulas.

$s$  に付けられた  
ラベル(原始命題の集合)

the label (a set of atomic propositions) attached to  $s$

$f$  が時相演算子を含まないとき,  
 $f$  の真偽値は状態  $s$  (計算木の根)における真偽値となる.

If  $f$  involves no temporal operators,  
the truth of  $f$  is its truth in the state  $s$ , the root of the computation tree.

## CTLの形式的意味論(3/3) 時相演算子を含むとき

(Formal semantics of CTL: When temporal operators involved)

---

$s \models \mathbf{E}f \iff$  there exists a path  $\pi$  starting from  $s$  such that  $\pi \models f$

$s \models \mathbf{A}f \iff$  for every path  $\pi$  starting from  $s$ ,  $\pi \models f$

---