

## ソフトウェアモデル論(2012年度) 第14回・2013/01/11

桑原 寛明  
情報理工学部 情報システム学科

### CTL(計算木論理)

(復習)

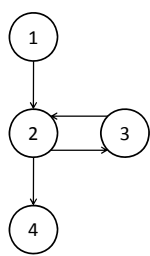
- 離散、点、未来、分岐
- 計算木に対する論理
  - 計算木は、ある状態を開始状態とするすべての経路を1つにまとめた木構造
  - 開始状態が木構造の根
- 経路の選択
  - すべての経路において、ある経路において
- タイミングの選択
  - 次、将来のいつか、今後ずっと、ある時点まで

ソフトウェアモデル論(2013/01/11)

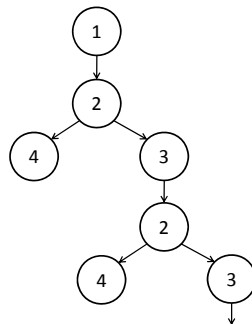
2

### 計算木

(復習)



状態遷移



計算木

ソフトウェアモデル論(2013/01/11)

3

### CTLの論理式

(復習)

1. 命題変数は状態式
2.  $P, Q$  が状態式ならば  $\neg P, P \wedge Q, P \vee Q, P \rightarrow Q$  は状態式
3.  $P$  が経路式ならば  $A P, E P$  は状態式
4.  $P, Q$  が状態式ならば  $X P, F P, G P, P U Q$  は経路式
5. 以上が状態式と経路式のすべてであり、状態式がCTLの論理式のすべて

ソフトウェアモデル論(2013/01/11)

4

### CTLの意味論

(復習)

- $M, s \models p$ 
  - $p \in L(s)$
- $M, s \models \neg P$ 
  - $M, s \models P$  でない
- $M, s \models P \wedge Q$ 
  - $M, s \models P$  かつ  $M, s \models Q$
- $M, s \models P \vee Q$ 
  - $M, s \models P$  または  $M, s \models Q$
- $M, s \models P \rightarrow Q$ 
  - $M, s \models P$  ならば  $M, s \models Q$

ソフトウェアモデル論(2013/01/11)

5

### CTLの意味論

(復習)

- $M, s \models AX P$ 
  - $\sigma(0)=s$  なるすべての経路  $\sigma$  において  $M, \sigma(1) \models P$
- $M, s \models EX P$ 
  - $\sigma(0)=s$  かつ  $M, \sigma(1) \models P$  なる経路  $\sigma$  が存在する
- $M, s \models AF P$ 
  - $\sigma(0)=s$  なるすべての経路  $\sigma$  においてある  $i \geq 0$  が存在して  $M, \sigma(i) \models P$
- $M, s \models EF P$ 
  - $\sigma(0)=s$  かつ  $M, \sigma(i) \models P$  なる  $i \geq 0$  が存在する経路  $\sigma$  が存在する

ソフトウェアモデル論(2013/01/11)

6

### CTLの意味論 (復習)

- $M, s \models \mathbf{AG} P$ 
  - $\sigma(0)=s$  なるすべての経路  $\sigma$  において任意の  $i \geq 0$  に対して  $M, \sigma(i) \models P$
- $M, s \models \mathbf{EG} P$ 
  - $\sigma(0)=s$  かつ任意の  $i \geq 0$  に対して  $M, \sigma(i) \models P$  なる経路  $\sigma$  が存在する
- $M, s \models \mathbf{A}[P \mathbf{U} Q]$ 
  - $\sigma(0)=s$  なるすべての経路  $\sigma$  においてある  $j \geq 0$  が存在して  $M, \sigma(j) \models Q$  かつ  $0 \leq i < j$  に対して  $M, \sigma(i) \models P$
- $M, s \models \mathbf{E}[P \mathbf{U} Q]$ 
  - $\sigma(0)=s$  かつ、ある  $j \geq 0$  が存在して  $M, \sigma(j) \models Q$  かつ  $0 \leq i < j$  に対して  $M, \sigma(i) \models P$  なる経路  $\sigma$  が存在する

ソフトウェアモデル論(2013/01/11) 7

### CTLの意味論 (復習)

ソフトウェアモデル論(2013/01/11) 8

### 経路演算子、時相演算子 (復習)

- $\neg \mathbf{AX} P = \mathbf{EX} \neg P$
- $\neg \mathbf{AF} P = \mathbf{EG} \neg P$
- $\neg \mathbf{AG} P = \mathbf{EF} \neg P$
- $\mathbf{EX}, \mathbf{EG}, \mathbf{EU}$  があれば他の演算子は表現可能
  - $\mathbf{EF} P = \mathbf{E}[T \mathbf{U} P]$   $T$  は恒真(任意の状態で真)
  - $\mathbf{A}[P \mathbf{U} Q] = \neg(\mathbf{EG} \neg Q \vee \mathbf{E}[\neg Q \mathbf{U} (\neg P \wedge \neg Q)])$
  - $\mathbf{EX}, \mathbf{EG}, \mathbf{EU}$  の組み合わせに限らない

ソフトウェアモデル論(2013/01/11) 9

### 例 (復習)

- $\text{COPY}, s_0 \models \mathbf{EF}(p \wedge q)$ 
  - $p, q$  がともに成り立つ状態に到達できる経路がある
- $\text{COPY}, s_0 \models \neg \mathbf{AF}(p \wedge q)$ 
  - すべての経路で  $p, q$  がともに成り立つ状態に到達できるわけではない

ソフトウェアモデル論(2013/01/11) 10

### LTL(線形時相論理)

- 離散、点、未来、線形
- ある状態を開始状態とする1つの経路に着目する論理
- 詳細は省略

ソフトウェアモデル論(2013/01/11) 11

### モデル検査アルゴリズム

- Kripke構造  $M$ 、状態  $s$ 、CTL式  $P$
- CTL式の演算子は  $\neg, \vee, \mathbf{EX}, \mathbf{EG}, \mathbf{EU}$
- $\text{Check}(M, P)$ 
  - $M$  の各状態に対して  $P$  の各部分式の中で成り立つものを求める
- 最終的に  $s$  で成り立つ式の中に  $P$  が含まれていれば  $M, s \models P$

ソフトウェアモデル論(2013/01/11) 12

モデル検査アルゴリズム

```

case:  $P \in PV$ 
  for all  $s \in S$  do
    if  $P \in L(s)$  then  $label(s) := label(s) \cup \{P\}$ 
case:  $P \equiv \neg Q$ 
  Check( $M, Q$ )
  for all  $s \in S$  do
    if  $Q \notin label(s)$  then  $label(s) := label(s) \cup \{\neg Q\}$ 
case:  $P \equiv Q_1 \vee Q_2$ 
  Check( $M, Q_1$ )
  Check( $M, Q_2$ )
  for all  $s \in S$  do
    if  $Q_1 \in label(s)$  or  $Q_2 \in label(s)$  then  $label(s) := label(s) \cup \{Q_1 \vee Q_2\}$ 
    
```

ソフトウェアモデル論(2013/01/11)

13

モデル検査アルゴリズム

- EX Q の場合
  - 一つ遷移すると Q が成り立つ状態に到達できる状態では EX Q が成り立つ

```

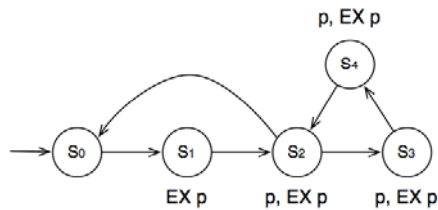
case:  $P \equiv EX Q$ 
  Check( $M, Q$ )
  for all  $s \in \{s \mid Q \in label(s)\}$  do
    for all  $s'$  such that  $R(s', s)$  do
       $label(s') := label(s') \cup \{EX Q\}$ 
    
```

ソフトウェアモデル論(2013/01/11)

14

モデル検査アルゴリズム

- EX Q の場合 (続き)



ソフトウェアモデル論(2013/01/11)

15

モデル検査アルゴリズム

- EG Q の場合
  - 常に Q が成り立っている経路を見つける
- Q が成り立つ状態のみに着目
- 強連結成分
  - 任意の2状態間に経路が存在
  - 極大
- 強連結成分中のいずれかの状態に到達可能

```

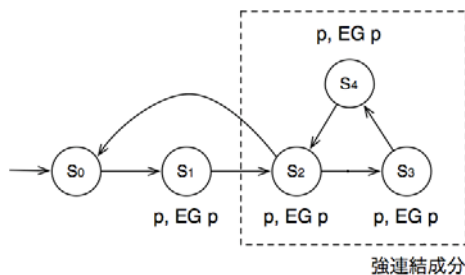
Check( $M, P$ )
 $S' := \{s \mid P \in label(s)\}$ 
 $SCC := \{C \mid C \text{ は } S' \text{ の強連結成分}\}$ 
 $T := \bigcup_{C \in SCC} \{s \mid s \in C\}$ 
for all  $s \in T$  do  $label(s) := label(s) \cup \{EG P\}$ 
while  $T \neq \emptyset$  do
  choose  $s \in T$ 
   $T := T - \{s\}$ 
  for all  $s'$  such that  $R(s', s)$  do
    if  $EG P \notin label(s')$  then
       $label(s') := label(s') \cup \{EG P\}$ 
   $T := T \cup \{s'\}$ 
    
```

ソフトウェアモデル論(2013/01/11)

16

モデル検査アルゴリズム

- EG Q の場合 (続き)



ソフトウェアモデル論(2013/01/11)

17

モデル検査アルゴリズム

- $E[Q_1 \cup Q_2]$  の場合
  - $Q_2$  が成り立っている状態から遷移をさかのぼって  $Q_1$  が成り立っている間  $E[Q_1 \cup Q_2]$  が成り立つ

```

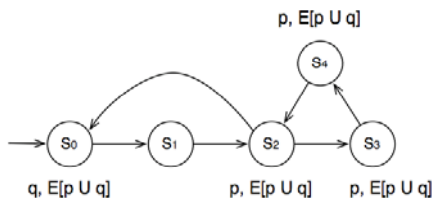
Check( $M, P$ )
Check( $M, Q$ )
 $T := \{s \mid Q \in label(s)\}$ 
for all  $s \in T$  do  $label(s) := label(s) \cup \{E[PUQ]\}$ 
while  $T \neq \emptyset$  do
  choose  $s \in T$ 
   $T := T - \{s\}$ 
  for all  $s'$  such that  $R(s', s)$  do
    if  $E[PUQ] \notin label(s')$  and  $P \in label(s')$  then
       $label(s') := label(s') \cup \{E[PUQ]\}$ 
   $T := T \cup \{s'\}$ 
    
```

ソフトウェアモデル論(2013/01/11)

18

モデル検査アルゴリズム

- $E[Q_1 \cup Q_2]$  の場合(続き)

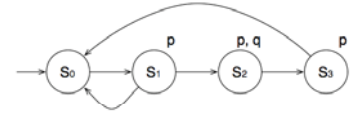


ソフトウェアモデル論(2013/01/11)

19

例

- COPY,  $s_0 \models E[TU \neg(\neg p \vee \neg q)]$   
 -  $EF(p \wedge q)$  か?



- $label(s_0) = \{T, \neg p, \neg q, \neg p \vee \neg q, E[TU \neg(\neg p \vee \neg q)]\}$
- $label(s_1) = \{T, \neg q, \neg p \vee \neg q, E[TU \neg(\neg p \vee \neg q)]\}$
- $label(s_2) = \{T, \neg(\neg p \vee \neg q), E[TU \neg(\neg p \vee \neg q)]\}$
- $label(s_3) = \{T, \neg q, \neg p \vee \neg q, E[TU \neg(\neg p \vee \neg q)]\}$

ソフトウェアモデル論(2013/01/11)

20