Теория вычислительных процессов и структур

Лекция №14. Верификация программ

Содержание лекции

Поиск ошибок в программах I

Существует четыре основных подхода:

1. Имитационное моделирование (тестирование прототипа программы)
2. Тестирование (всей программы или отдельных ее частей)
3. Дедуктивный анализ
4. Верификация модели программы (или проверка модели или model checking)

Стадии верификации программ I

1. Стадия моделирования (построение модели программы)
   - Данная стадия может выполняться при компиляции программы (например).
   - На данной стадии часто абстрагируются от неважных деталей.
   - При этом важно учесть в модели важные элементы программы
2. Спецификация
   - Описание требований, которым должна удовлетворять программа (или компонент программы).
   - Например, программа должна всегда завершаться с некоторым результатом (достичь заключительного состояния).
   - Трудно (невозможно?) сформулировать исчерпывающие требования к программе.
3. Верификация
   - Применение символьных алгоритмов (символьные вычисления).
   - Использование специальных структур данных для компактного описания модели и спецификации.
   - Применение методов редукции для сокращения количества состояний.

Суть model checking I

- Проверка соответствия модели системы частичной (полной?) спецификации.
- Частичная спецификация (формула темпоральной логики)
- Ответ на вопрос: Выполняется ли темпоральная формула в рамках данной модели?

Суть model checking II

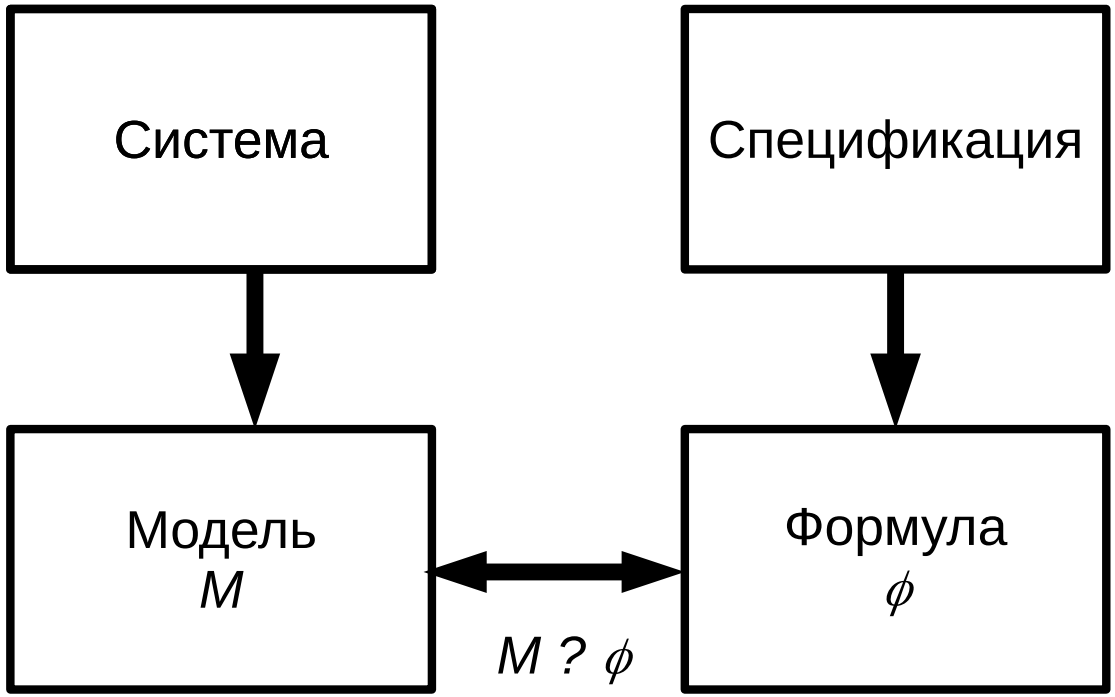


Рис.: Model checking

Модель Крипке I

Пусть задано множество AP — множество атомарных предикатов. Модель Крипке над AP есть четверка $M = (S, S_0, R, L)$ где:

- S — конечное множество состояний,
- S_0 — множество начальных состояний,
- $R \subset S \times S$ — отношение связности переходов из состояния в состояние $\forall s \in S \exists s' : (s, s') \in R$,
- $L : S \to 2^{AP}$ — функция разметки (состояний), сопоставляет состоянию подмножество переходов (запись 2^X обозначает множество всех подмножеств).

Последовательность $\pi = s_0 s_1 \ldots$ — путь в модели Коипке из состояния $s = s_0$. Для всех s_i выполняется $R(s_i, s_{i+1})$.

Модель Крипке I

К модели Крипке могут быть сведены многие представления программ:

- Представление состояний и переходов логической формулой;
- Булевы (логические) схемы;
- Последовательные и параллельные программ.

Пример модели Крипке I

Пример программы взаимного исключения

```
P0::
  L0:   while true do
  NC0:    wait (turn = 0);
  CR0:    turn = 1;
        end

P1::
  L1:   while true do
  NC1:    wait (turn = 1);
  CR1:    tunr = 0;
        end
```

Рис.: Пример программы

Пример модели Крипке I

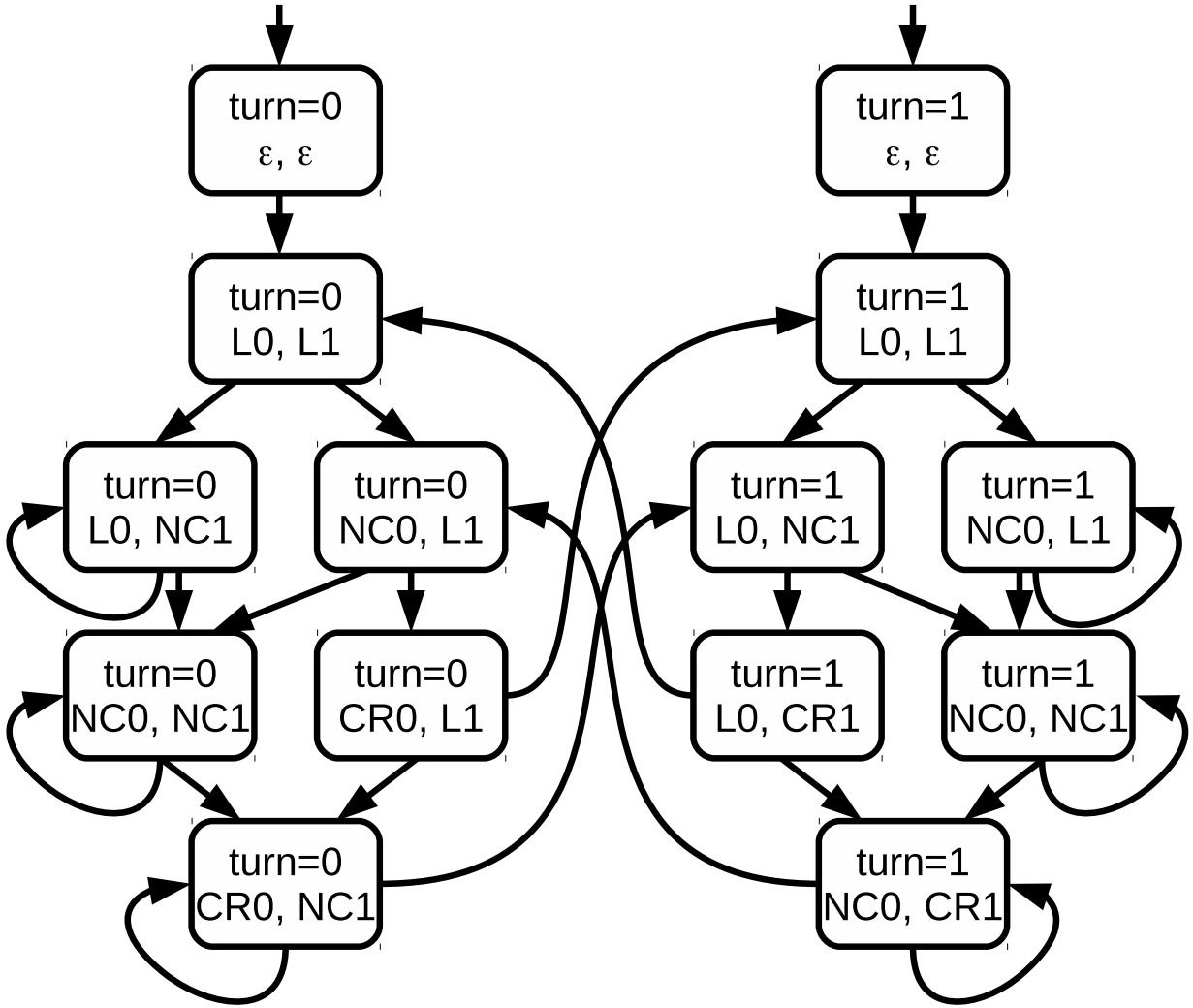


Рис.: Достижимые состояния модели Крипке

Пример модели Крипке I

Состояния модели Крипке помечаются атомарными предикатами, которые истинны в данном состоянии.
Рассмотрим атомарные предикаты

- $turn = 1$ — обозначим буквой t
- $pc_1 = NC_1$ — обозначим буквой p (pc — program counter)

Разметка состояний

- в состоянии s_0 $(turn = 0, pc_0 = \epsilon, pc_1 = \epsilon) : L(s_0) = \emptyset$,
- в состоянии s_1 $(turn = 1, pc_0 = \epsilon, pc_1 = \epsilon) : L(s_1) = \{t\}$,
- в состоянии s_2 $(turn = 0, pc_0 = L_0, pc_1 = NC_1) : L(s_2) = \{p\}$,
- в состоянии s_3 $(turn = 1, pc_0 = CR_0, pc_1 = NC_1) : L(s_3) = \{t, p\}$.

Пример модели Крипке I

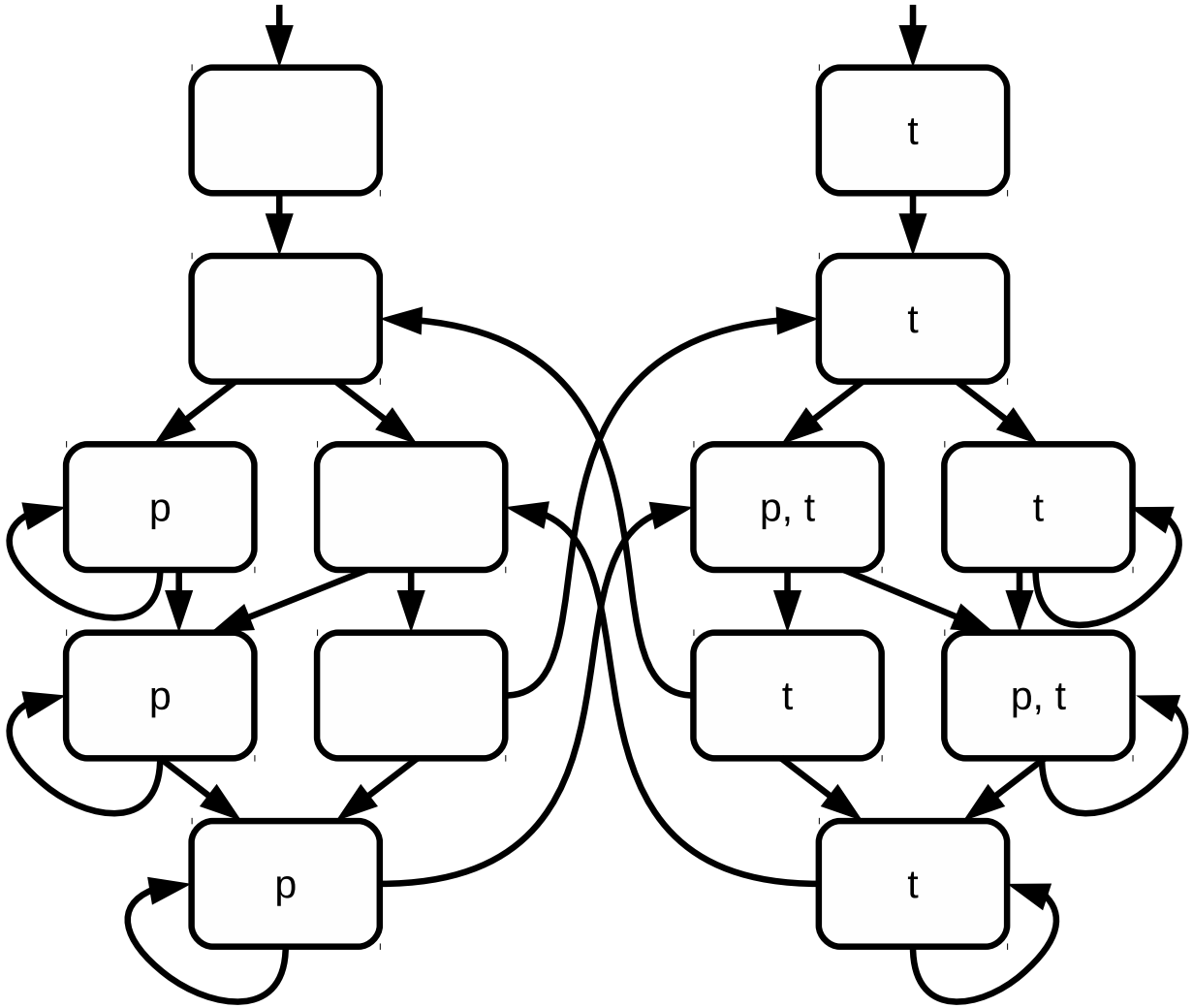


Рис.: модель Крипке

Темпоральная логика I

Формулы классической логики статичны — истинность не зависит от времени.
Состояние технической системы изменяется со временем.
Темпоральная логика — истинность формул зависит от момента времени, в который вычисляются значения формул
Например:

- Любое отправленное сообщение рано или поздно будет получено
- Пока ключ зажигания не вставлен, машина не поедет
- Всегда верно, что если производитель остановился, потребитель рано или поздно запустится.

Виды темпоральных логик I

Виды темпоральных логик:

- Линейная темпоральная логика: Linear Time Logic (LTL);
- Темпоральная логика деревьев вычислений: Computation Tree Logic (CTL, CTL*);

LTL : (не)выполнение свойств на *всех* путях в дереве вычислений.

CTL, CTL*: (не)выполнение свойств на *всех* или *на каком-либо* пути в дереве вычислений.

Кратко рассмотрим CTL*, но прежде покажем, что понимается под деревом вычислений.

Дерево вычислений I

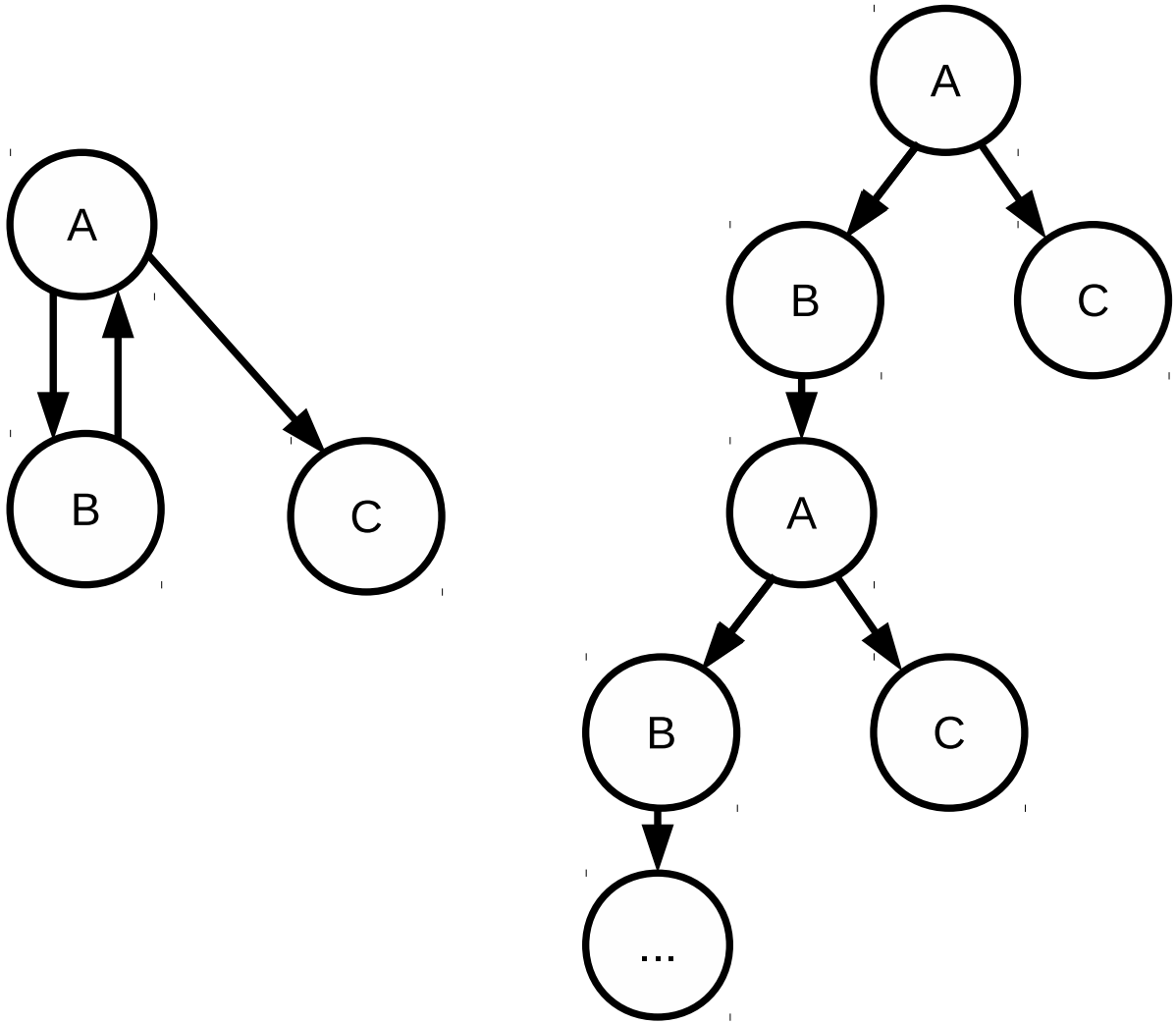


Рис.: слева — модель Крипке, справа — дерево вычислений

Темпоральная логика I

Синтаксис: правила составления темпоральных формул

Семантика: правила интерпретации темпоральных формул

Кванторы пути:

- A — “справедливо для всех путей”
- E — “справедливо для некоторого пути”

Кванторы времени:

- X — (neXt) в следующий момент времени
- G — (Globally) всегда
- F — (Future) рано или поздно, когда-нибудь в будущем
- U — (Until) выполнять утверждение слева, пока не выполнится утверждение справа
- R — (Repeat) пока не выполнится утверждение слева, выполнять утверждение справа

Синтаксис CTL* I

Пусть AP — множество атомарных высказываний
Синтаксис формул состояния:

- Если $p \in AP$, то p — формула состояния
- Если f и g — формула состояния, то $\neg f$, $f \vee g$, $f \wedge g$ — формула состояния,
- Если f — формула пути, то Af и Ef — формулы состояния.

Синтаксис формул пути:

- Если f — формула состояния, то f — формула пути,
- Если f и g — формулы пути, то $\neg f$, $f \vee g$, $f \wedge g$, Xf, Gf, Ff, fUg, fRg — формулы пути.

Семантика CTL* I

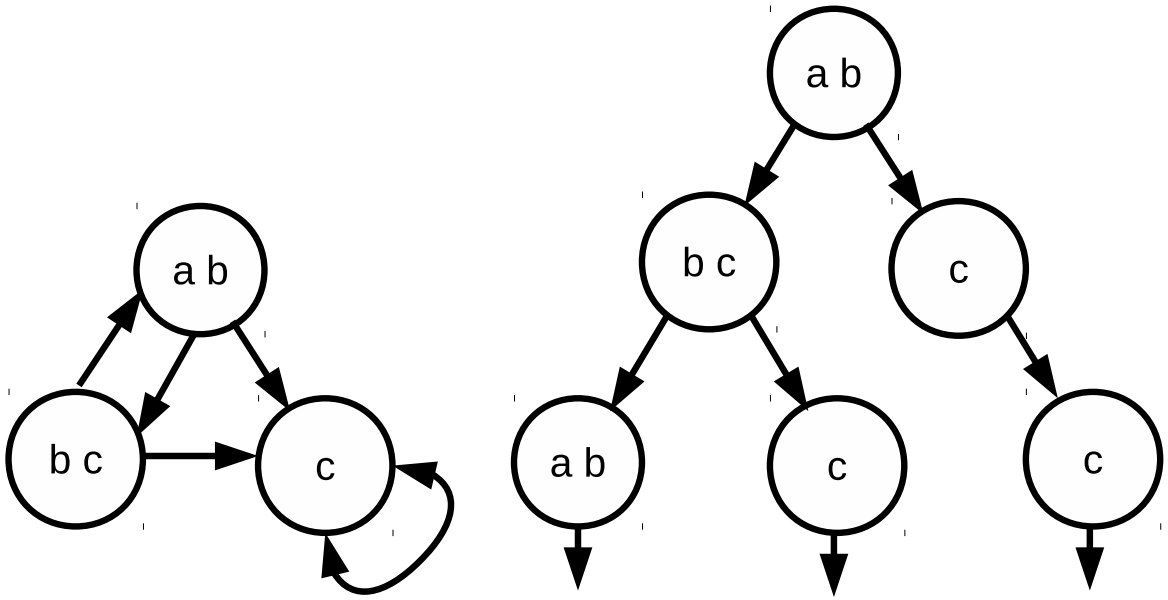
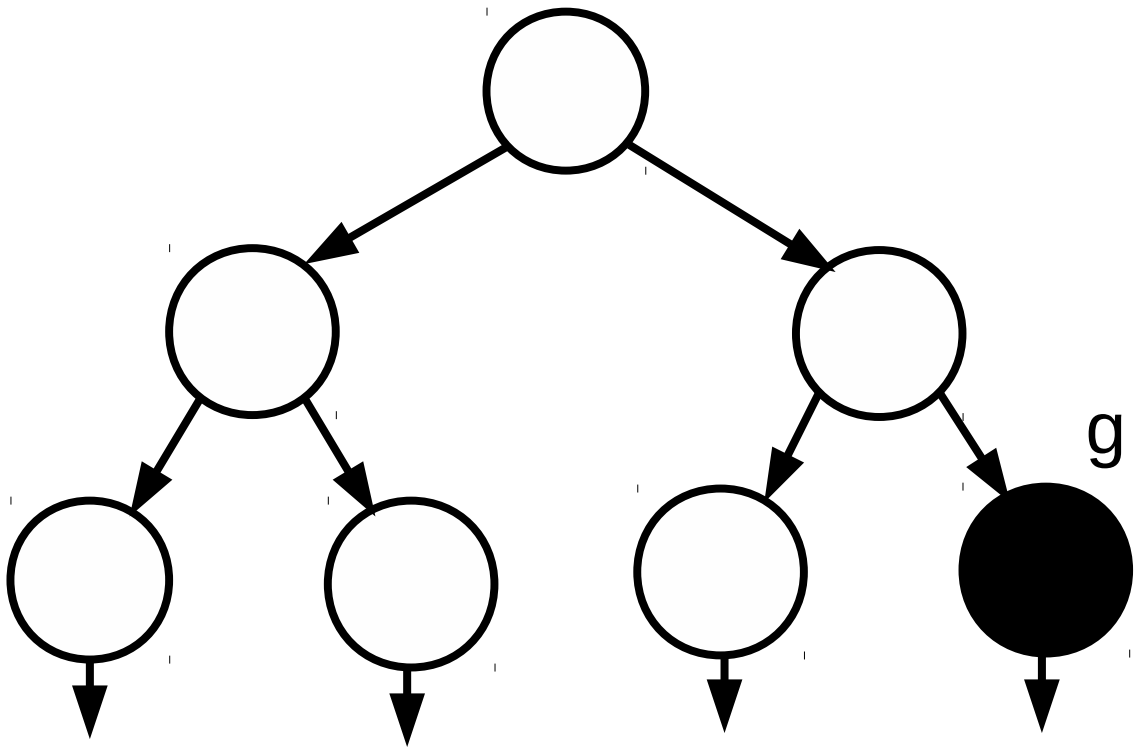


Рис.: слева — модель Крипке, справа — дерево вычислений

Обозначение $M, s \models f$ формула f выполняется на модели M с начальной вершиной s.
Отношение *models* определяется естественным образом индукцией по строению формулы.

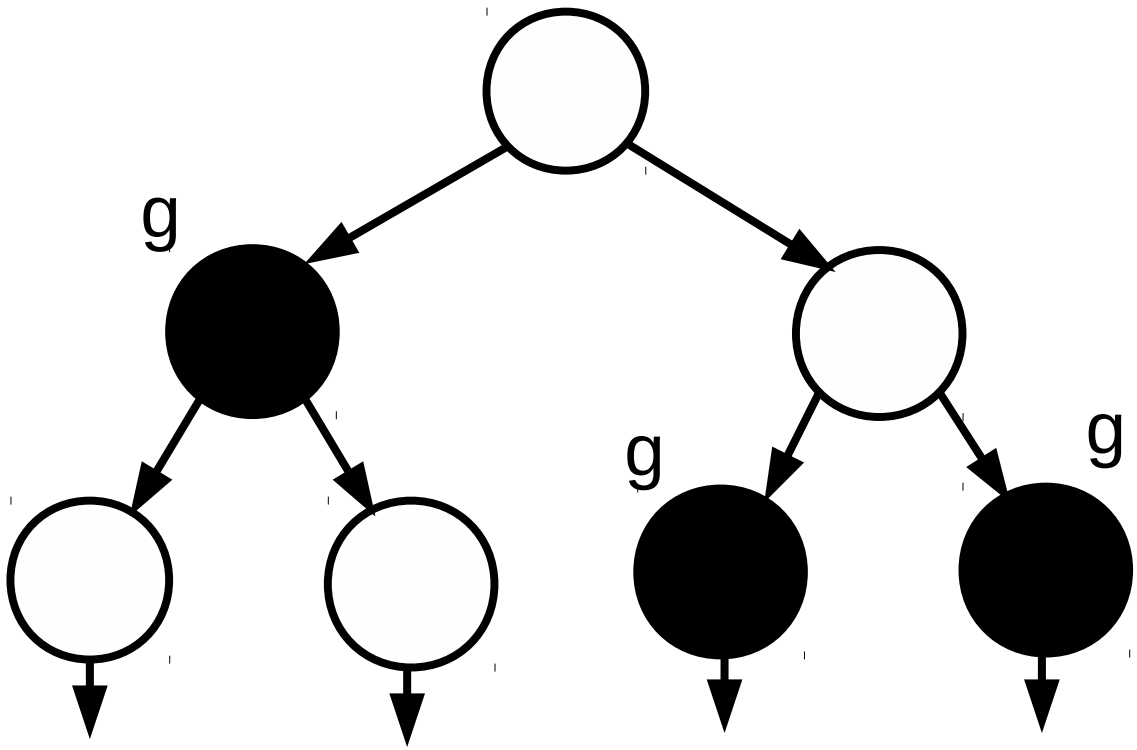
Примеры формул I

$$M, s_0 \models EFg$$



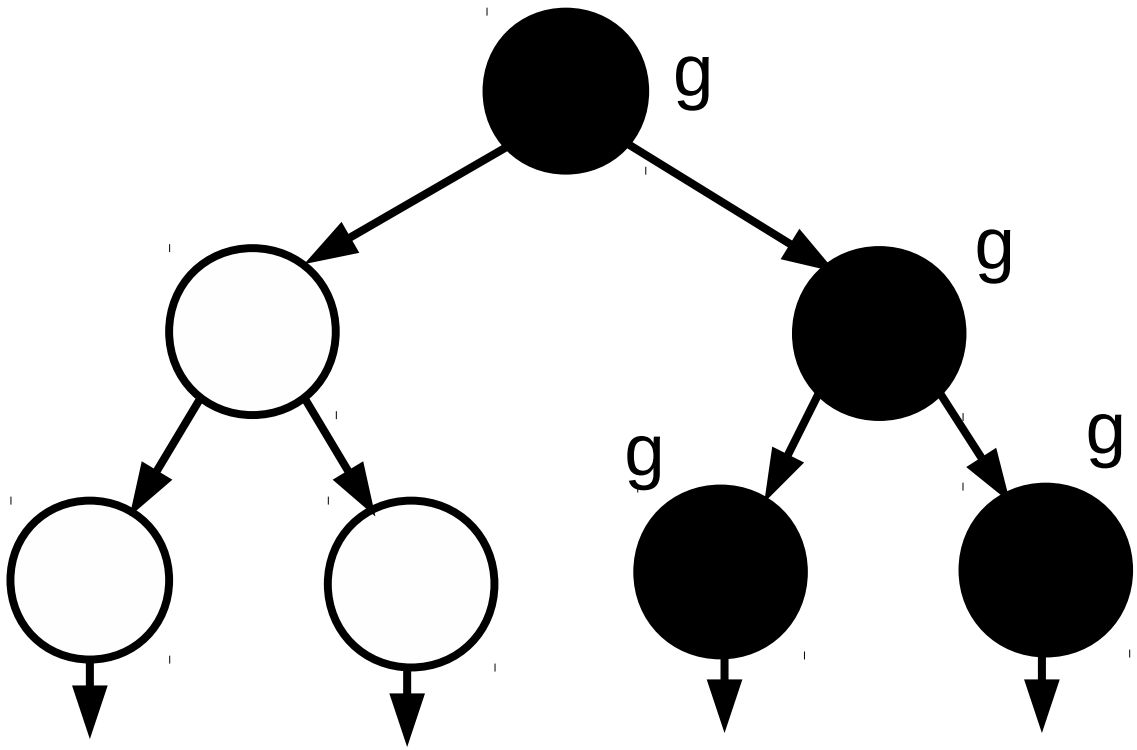
Примеры формул I

$$M, s_0 \models AFg$$



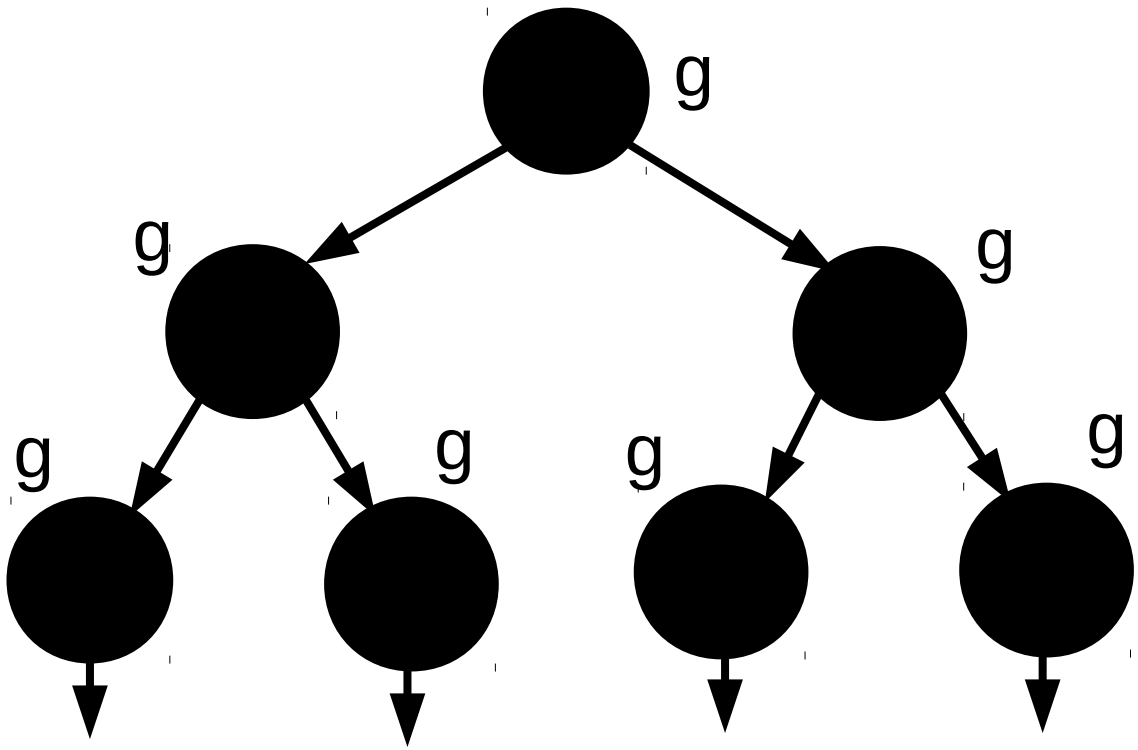
Примеры формул I

$$M, s_0 \models EGg$$



Примеры формул I

$$M, s_0 \models AGg$$



Логика CTL I

Сужение CTL*, допускающая только конструкции вида:

- $\neg f$
- $f \vee g$
- EXf
- EGf
- $E[fUg]$

Задача верификации I

Дано:

- Модель Крипке $M = (S, R, L)$
- Формула темпоральной логики f

Требуется определить:

- Множество $\{s \in S \mid M, s \models f\}$.