Проверка на модели (model checking)

Общее введение

- Проверить многопоточный алгоритм/протокол — очень сложно.
- Перебрать, двоичные деревья — комбинаторный взрыв. Нужен более содержательный подход.
- Проверка на моделях — тоже по сути перебор, только очень хорошо оптимизированный.
- Первоначальная идея: Clarke E., Emerson E., Sistla A., Automatic verification of finite-state con- current systems using temporal logic specifications 1986.

Линейная темпоральная логика: операторы

- Вариант моделей Крипке для модальной логики (интуиция из интуиционистской логики подходит).
- Миры выстроены в линейном порядке.
- В моделях Крипке для ИИВ (неявная) зависимость от миров была у импликации и отрицания. В ЛТЛ зависиость явная, и выражается с помощью следующих дополнительных связок:
  1. $G(\alpha)$ или $\Box\alpha$: утверждение α выполнено в любой момент (начиная с текущего).
  2. $P(\alpha)$ или $\bigcirc\alpha$: утверждение α выполнено в следующий момент.
  3. $E(\alpha)$ или $\Diamond\alpha$: утверждение α неизбежно выполнено в будущем, в какой-то момент (начиная с текущего).
  4. $U(\alpha, \beta)$ или $\alpha\mathcal{U}\beta$: утверждение α истинно, пока β не станет истинным, после чего α может быть любым.

Примеры выражений

Постановка задачи

Определение

Системой переходов назовём граф состояний, в котором каждое состояние отражает содержимое памяти компьютера, а переходы соответствуют инструкциям (операциям), выполняемым компьютером

Хотим научиться проверять, выполнено ли φ при всех возможных вариантах выполнения программы, то есть при всех возможных путях в системе переходов TS:

$$TS \models \varphi$$

Можем понимать систему переходов как модель для логического исчисления.

Автоматы Бюхи

- Автоматы Бюхи отличаются от конечного автомата правилом проверки допустимости строки: бесконечная строка α допускается конечным автоматом Бюхи, если в процессе применения автомата к ней допускающее состояние будет посещено бесконечное количество раз.
- Рассмотрим автомат с двумя состояниями (0,1) и полным графом переходов. Строка $(ab)^\omega$ будет принята, строка $a(b^\omega)$ — нет.
- Можно рассматривать недетерминированные автоматы Бюхи.
- Обобщённые недетерменированные автоматы Бюхи — такие, где есть список допускающих состояний, каждое из которых должно быть посещено бесконечное количество раз.

Представление моделей ЛТЛ как множества слов

Рассмотрим следующую грамматику для формул:
$\varphi ::= \top \mid a \mid \varphi \,\&\, \varphi \mid \neg\varphi \mid \bigcirc \varphi \mid \varphi \mathcal{U} \varphi$
Тогда:

$$W(\varphi) = \{\sigma \in (\mathcal{P}(a))^{\omega} \mid \sigma \models \varphi\}$$

На строке $\sigma = S_0 S_1 S_2 \ldots$ (каждый $S_i \subseteq \mathcal{P}(a)$) истинность задаётся так:

$\sigma \models \top$	всегда
$\sigma \models a$	$a \in S_0$
$\sigma \models \varphi_1 \,\&\, \varphi_2$	$\sigma \models \varphi_1$ и $\sigma \models \varphi_2$
$\sigma \models \neg\varphi$	$\sigma \not\models \varphi$
$\sigma \models \bigcirc\varphi$	$\sigma[1\ldots] \models \varphi$
$\sigma \models \varphi_1 \mathcal{U} \varphi_2$	существует $j.\sigma[j\ldots] \models \varphi_2$ и $\sigma[i\ldots] \models \varphi_1, i < j$

Разрешимость задачи $TS \models \varphi$ в ЛТЛ

Теорема

Если система переходов конечна и не содержит терминальных состояний, то существует алгоритм, разрешающий данную задачу для любого φ. В случае отрицательного ответа алгоритм строит контрпример.

Доказательство.

Построим недетерменированный автомат Бюхи по системе переходов для формулы $\neg\varphi$ (принимает строку тогда и только тогда, когда она удовлетворяет формуле), затем по автомату построим удовлетворяющую строку, если она существует. $\square$

Пример реализации: SPIN

- Один из первых инструментов (1991 год).
- Используется специальный язык для описания алгоритмов/протоколов (Promela).
- Язык позволяет формализовать параллельные вычисления.
- Программа может быть вычислена в разных окружениях (например, случайное исполнение).
- Также, к программе могут быть добавлены условия, которые либо будут доказаны — либо будет найден контрпример (контрпример будет предъявлен).

Пример программы на Promela: каковы возможные значения n?

```
byte n = 0, finish = 0;
active [2] proctype P() {
    byte register, counter = 0;
    do :: counter = 10 -> break
       :: else ->
            register = n;
            register++;
            n = register;
            counter++
    od;
    finish++
}
active proctype WaitForFinish() {
    finish == 2;
    printf(“n = %d\n”, n)
}
```