

Chapter 4

LTL model checking

Course “Specification and Verification of Parallel Systems”

Juliane Päßler, Tobias John

Spring 2024

Overview

- Classification of properties
- Modeling program executions





Section

Dual Operators

Spring 2024



Section

Classification of properties

Spring 2024

Classification of properties

We can classify properties expressible in LTL. *Examples*:

Classification

invariant $\Box\varphi$

"liveness" $\Diamond\varphi$

obligation $\Box\varphi \vee \Diamond\psi$

recurrence $\Box\Diamond\varphi$

persistence $\Diamond\Box\varphi$

reactivity $\Box\Diamond\varphi \vee \Diamond\Box\psi$

fairness special form of other properties

- φ, ψ: non-temporal formulas



Safety (slightly simplified)

- important basic class of properties
- informally "nothing bad ever happens"

Definition (Safety/invariant)

- A **invariant** formula is of the form

$$\Box\varphi \quad \text{for some prop. formula } \varphi.$$

Safety formulae express *invariance* of some **state property** φ: that φ holds in every state of the computation.



Safety (slightly simplified)

- important basic class of properties
- informally "nothing bad ever happens"

Definition (Safety/invariant)

- A **invariant** formula is of the form

$$\Box\varphi \quad \text{for some prop. formula } \varphi.$$

- A **conditional safety** formula is of the form

$$\varphi \rightarrow \Box\psi \quad \text{for some prop. formulas } \varphi \text{ and } \psi.$$

Safety formulae express *invariance* of some **state property** φ: that φ holds in every state of the computation.



Liveness properties (simplified)

Definition (Liveness)

- A **liveness** formula is of the form

$$\Diamond\varphi$$

for some prop. formula φ.

Liveness formulae *guarantee* that some event φ eventually happens: that φ holds in at least one state of the computation.



Liveness properties (simplified)

Definition (Liveness)

- A **liveness** formula is of the form

$$\Diamond \varphi$$

  for some prop. formula φ.
- A **conditional liveness** formula is of the form

$$\varphi \rightarrow \Diamond \psi$$

  for prop. formulas φ and ψ.

Liveness formulae *guarantee* that some event φ eventually happens: that φ holds in at least one state of the computation.



Safety vs. Lifeness

- the negation of a safety formula is a liveness formula

Example (Mutex)

Mutual exclusion is a safety property. Let c_i denote that process P_i is executing in the critical section. Then

$$\Box\neg(c_1 \wedge c_2)$$

expresses that it should always be the case that not both P_1 and P_2 are executing in the critical section.

- negation of the formula:

$$\Diamond(c_1 \wedge c_2)$$

which expresses that eventually it *is* the case that both P_1 and P_2 are executing in the critical section.



Obligation

Definition (Obligation)

A **simple obligation** formula is of the form

$$\Box\varphi \vee \Diamond\psi$$

for propositional formulas φ and ψ.

- equivalently

$$\Diamond\varphi' \rightarrow \Diamond\psi \quad \text{with } \varphi' = \neg\varphi$$



Obligation (2)

Proposition

Every safety and liveness formula is also an obligation formula.

Proof.

It's a consequence of the following equivalences.

$$\Box\varphi \leftrightarrow \Box\varphi \vee \Diamond\bot$$
$$\Diamond\varphi \leftrightarrow \Box\bot \vee \Diamond\varphi$$

and the facts that $\models \neg\Box\bot$ and $\models \neg\Diamond\bot$. □



Recurrence

Definition (Recurrence)

A **recurrence** formula is of the form

$$\Box\Diamond\varphi$$

for some propositional formula φ.

- *infinitely many* positions satisfies φ.



Persistence

Definition (Persistence)

A **persistence** formula is of the form

$$\Diamond\Box\varphi$$

for some propositional formula φ.

- dual to “infinitely often”
- aka: stabilization



Recurrence and Persistence

Observation

Recurrence and persistence are duals.

$$\neg(\Box\Diamond\varphi) \leftrightarrow (\Diamond\Box\neg\varphi)$$
$$\neg(\Diamond\Box\varphi) \leftrightarrow (\Box\Diamond\neg\varphi)$$



Reactivity

Definition (Reactivity)

- A **simple reactivity** formula is of the form

$$\Box\Diamond\varphi \vee \Diamond\Box\psi$$

  for prop. formulas φ and ψ.
- A very general class of formulae are conjunctions of reactivity formulae.



Reactivity

Definition (Reactivity)

- A **simple reactivity** formula is of the form

$$\Box\Diamond\varphi \vee \Diamond\Box\psi$$

  for prop. formulas φ and ψ.
- A very general class of formulae are conjunctions of reactivity formulae.
- equivalent:

$$\Box\Diamond\psi' \rightarrow \Box\Diamond\varphi \quad \text{with } \psi' = \neg\psi$$



Weak Fairness

- at every time step: if an object is always enabled, it will be taken eventually

$$\Box(\Box enabled(\tau) \rightarrow \Diamond taken(\tau)),$$



Weak Fairness

- at every time step: if an object is always enabled, it will be taken eventually

$$\Box(\Box enabled(\tau) \rightarrow \Diamond taken(\tau)),$$

Proposition

Weak fairness can be specified as the following equivalent recurrence formula.

$$\Box\Diamond(enabled(\tau) \rightarrow taken(\tau))$$



Strong Fairness

- if an object is infinitely often enabled, it will be taken infinitely often

Proposition

Strong fairness can be specified as the following reactivity formula.

$$\Box\Diamond enabled(\tau) \rightarrow \Box\Diamond taken(\tau)$$





Section
GCD Example

Spring 2024

GCD code

- consider a program P
- input: two integers $a > 0, b > 0$
- local variables: x, y with $x = a$ and $y = b$
- output: integer g
- program contains nine *locations*

```
l1   while x != y do
l2     if x > y do
l3       x := x−y
l4     else if x < y do
l5       y := y−x
l6     fi
l7   od
l8   g := x
l9
```



GCD Example

Below is a computation π of our recurring GCD program.

P-computation

States are of the form $\langle l_n, x, y, g\rangle$.

$$
\begin{aligned}
\pi: \quad & \langle l_1, 21, 49, 0\rangle \to \langle l_2, 21, 49, 0\rangle \to \langle l_3, 21, 49, 0\rangle \to \\
& \langle l_1, 21, 28, 0\rangle \to \langle l_2, 21, 28, 0\rangle \to \langle l_3, 21, 28, 0\rangle \to \\
& \langle l_1, 21, 7, 0\rangle \to \langle l_2, 21, 7, 0\rangle \to \langle l_4, 21, 7, 0\rangle \to \langle l_5, 21, 7, 0\rangle \to \\
& \langle l_1, 14, 7, 0\rangle \to \langle l_2, 14, 7, 0\rangle \to \langle l_4, 14, 7, 0\rangle \to \langle l_5, 14, 7, 0\rangle \to \\
& \langle l_1, 7, 7, 0\rangle \to \langle l_8, 7, 7, 0\rangle \to \langle l_9, 7, 7, 7\rangle \to \cdots
\end{aligned}
$$



GCD Example

- $terminated$ denotes the formula $at(l_9)$.

Do the following properties hold for π? And why?

1. $\Box terminated$ (safety)
2. $at(l_1) \rightarrow terminated$
3. $at(l_9) \rightarrow terminated$
4. $at(l_8) \rightarrow \Diamond terminated$ (conditional liveness)
5. $\Diamond at(l_8) \rightarrow \Diamond terminated$ (obligation)
6. $\Box(\gcd(x, y) \doteq \gcd(a, b))$ (safety)
7. $\Diamond terminated$ (liveness)
8. $\Diamond\Box(y \doteq \gcd(a, b))$ (persistence)
9. $\Box\Diamond terminated$ (recurrence)



Summary

- Classification of properties
- Modeling program executions
- Next week: exercise sheet
- Next lecture:
  - discuss exercise
  - start with model checking algorithm



References I

Bibliography

