7. Temporal Logics

Temporal Logic

- An extension of propositional logic.
- A more direct way of describing dynamic behaviour.
- Operators to support linguistic ways of expressing temporal relationships: always, sometimes, until, ...

Example:

“The cat always returns to room 2.”

\( \forall t_1: T \bullet \exists t_2: T \bullet t_2 \geq t_1 \land \text{pos(cat, } t_2) = 2 \)

where \( \text{pos(cat, } t) \) represents the position of the cat at time \( t \).

First-order logic (Z-style)

Describing Dynamic Behaviour

Example:

“The cat always returns to room 2.”

\( \forall t_1: T \bullet \forall t_2: T \bullet \forall n: N \bullet \)

\((\text{req}(n, t_1) \land t_1 \leq t_2 \land \text{pos}(t_2) \neq n \land \)

\((\exists t_{\text{trav}}: T \bullet t_1 \leq t_{\text{trav}} \leq t_2 \land \text{pos}(t_{\text{trav}}) = n) \Rightarrow \)

\((\exists t_{\text{serv}}: T \bullet t_1 \leq t_{\text{serv}} \leq t_2 \land \text{serv}(n, t_{\text{serv}}))\)

Temporal Logic

Temporal Operators:

\( \mathbf{G} p \) “It will always be the case that \( p \).”

\( \quad \) “\( p \) will always be true.”

\( \mathbf{F} p \) “It will sometimes be the case that \( p \).”

\( \quad \) “\( p \) will eventually occur.”

\( \mathbf{X} p \) “\( p \) will be true in the next state.”

\( \quad \) “\( p \) will occur tomorrow (in the next step).”

Another example:

“The elevator never traverses a floor for which a request is pending without satisfying the request.”

\( \forall t_1: T \bullet \forall t_2: T \bullet \forall n: N \bullet \)

\((\text{req}(n, t_1) \land t_1 \leq t_2 \land \text{pos}(t_2) \neq n \land \)

\((\exists t_{\text{trav}}: T \bullet t_1 \leq t_{\text{trav}} \leq t_2 \land \text{pos}(t_{\text{trav}}) = n) \Rightarrow \)

\((\exists t_{\text{serv}}: T \bullet t_1 \leq t_{\text{serv}} \leq t_2 \land \text{serv}(n, t_{\text{serv}}))\)

Describing Dynamic Behaviour

Another example:

“The elevator never traverses a floor for which a request is pending without satisfying the request.”

\( \forall t_1: T \bullet \forall t_2: T \bullet \forall n: N \bullet \)

\((\text{req}(n, t_1) \land t_1 \leq t_2 \land \text{pos}(t_2) \neq n \land \)

\((\exists t_{\text{trav}}: T \bullet t_1 \leq t_{\text{trav}} \leq t_2 \land \text{pos}(t_{\text{trav}}) = n) \Rightarrow \)

\((\exists t_{\text{serv}}: T \bullet t_1 \leq t_{\text{serv}} \leq t_2 \land \text{serv}(n, t_{\text{serv}}))\)

Temporal Logic — Examples

\( \mathbf{G} (\text{rain} \lor \text{sun}) \)

Every day there will be rain or sunshine.

\( \mathbf{F} \) rain

It will rain some day.

\( \mathbf{G} \neg \text{rain} \)

It will never rain.

\( \mathbf{G} \mathbf{F} \) rain

Every day will be followed by a rainy day, i.e., it will rain infinitely often.
Atomic Propositions

- Propositional logic is built from atomic propositions.
- Examples:
  - rain, error, floor = 3, …
- We need to extend automata to include propositions.

Kripke Structures

A Kripke structure is a tuple

\[ K = (Q, \text{Prop}, T, q_0, l) \]

with
- finite set of states \( Q = \{q_1, q_2, q_3, \ldots\} \)
- finite set of propositions \( \text{Prop} = \{p_1, \ldots\} \)
- transition relation \( T \subseteq Q \times Q \)
- initial state \( q_0 \in Q \)
- labelling function \( l: Q \rightarrow \text{Prop} \)

Kripke Structure for Digicode

- \( p_A \): “Button A has been pressed.”
- \( p_B \): “Button B has been pressed.”
- \( g \): “Passage is granted.”

Each formula characterises a set of states.

Temporal Combinators

Temporal combinators enable us to describe a single execution sequence.

\[ \begin{align*}
G p & \quad \text{“Globally p”} \\
F p & \quad \text{“Finally p”}
\end{align*} \]
The Until Combinator

\[ p \rightarrow p \rightarrow p \rightarrow q \rightarrow \]

\( p \mathcal{U} q \), “p until q”, is true for an execution if
- q is true at some state, and
- p is true at all states between the start state and the state where q holds.

The Weak Until Combinator

\[ p \rightarrow p \rightarrow p \rightarrow q \rightarrow \]

\( p \mathcal{W} q \equiv (p \mathcal{U} q) \lor \mathcal{G}p \)

“p waiting for q” or “p weak-until q”

p does not become false before a state where q holds is reached.

Path Quantifiers

A\( \varphi \) – holds in a state from which all executions satisfy the path formula \( \varphi \).

E\( \varphi \) – holds in a state from which there exists an execution satisfying the path formula \( \varphi \).

Path Quantifiers in CTL

AG \( p \)
EG \( q \)
AF \( r \)
EF \( s \)

Examples

\[ q_0 \]
\[ \text{warm} \Rightarrow \neg \text{warm} \]
\[ \mathcal{G} (\text{warm} \Rightarrow \neg \text{warm}) \]
\[ \mathcal{G} (\text{warm} \Rightarrow \mathcal{X} \neg \text{warm}) \]
\[ \text{warm} \Rightarrow \mathcal{F} \text{ error} \]
\[ \text{ok} \mathcal{U} \text{error} \]
\[ \text{ok} \mathcal{W} \text{error} \]
\[ \text{warm} \mathcal{W} \text{error} \]

Reading

Béard et. al.:
2.1 – The Language of Temporal Logic