Universidade de São Paulo
Instituto de Matemática e Estatística
Bachalerado em Ciência da Computação

Rafael Mota Gregorut

Título da monografia
se for longo ocupa esta linha também

São Paulo
Dezembro de 2015

Título da monografia
se for longo ocupa esta linha também

Monografia final da disciplina
MAC0499 – Trabalho de Formatura Supervisionado.

Orientadora: Prof. Dr. Ana Cristina Vieira de Melo

São Paulo
Dezembro de 2015

Resumo

Elemento obrigatório, constituído de uma sequência de frases concisas e objetivas, em forma de texto. Deve apresentar os objetivos, métodos empregados, resultados e conclusões. O resumo deve ser redigido em parágrafo único, conter no máximo 500 palavras e ser seguido dos termos representativos do conteúdo do trabalho (palavras-chave).

Palavras-chave: palavra-chave1, palavra-chave2, palavra-chave3.

Abstract

Elemento obrigatório, elaborado com as mesmas características do resumo em língua portuguesa.

Keywords: keyword1, keyword2, keyword3.

Contents

Chapter 1

Concepts

1.1 Statecharts and Automata

A software specification is a reference document which contains the requisites the program should satisfy. It can also be understood as a model of how the system should behave. A specification may be developed with natural language, with user cases for example, or usisng formal software engeering techniques.

Statechats are a type of formal software specification based on finite states machines (FSM) specially used in complex systems modeling, such as reactive systems and were defined in [1]. Similar to an automaton, a statechart has sets of states, transitions and input events that cause change of state. However, a statechart has additional features: orthogonality, hierarchy, broadcasting and history.

1.1.1 Nondeterministic finite automata

Definition: A nondeterministic finite automaton is a quintuple $M = (K, \Sigma, \Delta, s, F)$, where:

- K is the set of states
- Σ is the input alphabet
- Δ is the transition relation, a subset of $K \times (\Sigma \cup e) \times K$, where e is the empty string
- $s \in K$ is the initial state
- $F \subseteq K$ is the set of final states

Each tripe $(q, a, p) \in \Delta$ is a transition of M. If M is currently in state q and the next input is a, then M may follow any transition of the form (q, a, p) or (q, e, p). In the later case, no input is read. A configuration of M is an element of K^* and the relation $\vdash$ (yields one step) between two configuarions is defined as: $(q, w) \vdash (q', w') \Leftrightarrow$ there is a $u \in \Sigma \cup e$ such that $w = uw'$ and $(q, u, q') \in \Delta$.

Further information and formalism regarding finite automata can be obtained in [2].

A nondeterministic finite automaton example

1.1.2 Statechart models

A statechart model can be considered a extended finite state machine. The syntax of statechart is defined over the set of states, transitions, events, actions, conditions, expressions and labels [1]. In short terms:

- Transitions are the relations between states
- Events are the input that might cause a transition to happen
- Actions are generally triggered when a transition occurs
- Conditions are boolean verifications added to a transition in order to restrict the occurence of that transitions
- Expressions are composed are variables and algebraic operations over them
- A label is a pair made of an event and an action that is used to label a transition

Furthermore, it is worth to notice that a statechart model possess other features, described over the next sections of this chapter, that do not appear in a common finite state machine. These extra resources are useful, for instance, to model concurrency and different abstraction levels in a more practical way.

Orthogonality

More than one state may be active at the same time in a statechart, which is called orthogonality. It can be used to model concurrent and parallel situations. The set of active states in a certain moment is called configuration.

Hierarchy

It is possible that a state contains other states, called substates, and internal transitions, increasing the abstraction and encapsulation level of the model.

Guard conditions

In a transition, a guard condition is always verified before the change of states take place. If the condition is satisfied, in other words, e expression returns true, the transition will happen. Otherwise, that transition is not allowed to happen. A expression in a guard condition involve variables and operations. It is possible to check whether a state was entered for example.

Broadcasting

Besides an input event, a transition might have a action that is triggered when the transition is done. An action may cause another transition to happen, that is called broadcasting. This features makes it possible that chain reactions occurr in a statechart model.

History

A statechart is capable of remembering previously visited states by accessing the history state.

A statechart example

Find below an example of a statechart model. When compared to the nondeterministic finite automaton displayed previously, we can notice that the number of states is reduced due the increase in the abstraction level using hierarchy and orthogonality. Besides, in the statechart we can make usage of guard transitions and broadcasting, enriching the model with more information.

Appendix A

Título do apêndice

Texto texto.

Bibliography

[1] David Harel, Amir Pnueli, Jeanette Schmidt, and Rivi Sherman. On the formal semantics of statecharts. In *Proceedings of Symposium on Logic in Computer Science*, pages 55–64, 1987. 1, 2

[2] Harry R. Lewis and Christos H. Papadimitriou. *Elements of the Theory of Computation*. Prentice Hall, Inc, 2 edition, 1998. 1