Automatic Generation of Learning Assignments for Software Engineering Formalisms

Michael Steinle and Bernd Westphal

Software Engineering im Unterricht der Hochschulen 2020

February 27, 2020

Learning Assignments

- Definition Seel [1981]
  Learning assignments are “[...]selected and prepared learning objects with the aim to initiate, control, and organise learning processes.”

Software Engineering Formalism: Decision Tables cf. Balzert [2009]

T	r_1	r_2	r_3
c_1	$-$	$\times$	$-$
c_2	$-$	$\times$	$-$
c_3	$\times$	$*$	$-$
a_1	$\times$	$-$	$-$
a_2	$\times$	$\times$	$\times$

Decision Tables, Syntax

- Let C be a set of conditions and A be a set of actions, $C \cap A = \emptyset$
- A decision table T over C and A is a labelled $(m + k) \times n$ matrix

T	r_1	$\dots$	r_n
c_1	$v_{1,1}$	$\dots$	$v_{1,n}$
$\vdots$	$\vdots$	$\ddots$	$\vdots$
c_m	$v_{m,1}$	$\dots$	$v_{m,n}$
a_1	$w_{1,1}$	$\dots$	$w_{1,n}$
$\vdots$	$\vdots$	$\ddots$	$\vdots$
a_k	$w_{m,1}$	$\dots$	$w_{m,n}$

- where
  - $v_{1,1}, \dots, v_{m,n} \in \{\times, -, *\}$
  - $w_{1,1}, \dots, w_{k,n} \in \{\times, -\}$

Decision Tables, Semantics

Each rule $r \in \{r_1, \dots, r_n\}$ of table T

T	r_1	$\dots$	r_n
c_1	$v_{1,1}$	$\dots$	$v_{1,n}$
$\vdots$	$\vdots$	$\ddots$	$\vdots$
c_m	$v_{m,1}$	$\dots$	$v_{m,n}$
a_1	$w_{1,1}$	$\dots$	$w_{1,n}$
$\vdots$	$\vdots$	$\ddots$	$\vdots$
a_k	$w_{m,1}$	$\dots$	$w_{m,n}$

is assigned a propositional logical formula $\mathcal{F}(r)$ as follows:

Decision Tables, Semantics

Each rule $r \in \{r_1, \ldots, r_n\}$ of table T

T	r_1	$\ldots$	r_n
c_1	$v_{1,1}$	$\ldots$	$v_{1,n}$
$\vdots$	$\vdots$	$\ddots$	$\vdots$
c_m	$v_{m,1}$	$\ldots$	$v_{m,n}$
a_1	$w_{1,1}$	$\ldots$	$w_{1,n}$
$\vdots$	$\vdots$	$\ddots$	$\vdots$
a_k	$w_{m,1}$	$\ldots$	$w_{m,n}$

is assigned a propositional logical formula $\mathcal{F}(r)$ as follows:

- Let $(v_1, \ldots, v_m)$ and $(w_1, \ldots w_k)$ be premise and effect of r.
- Then

$$\mathcal{F}(r) := \underbrace{\bigwedge_{1 \leq i \leq m} F(v_i, c_i)}_{=:\mathcal{F}_{pre}(r)} \wedge \underbrace{\bigwedge_{1 \leq j \leq k} F(w_j, a_j)}_{=:\mathcal{F}_{eff}(r)}$$

where

$$F(v, x) = \begin{cases} x & \text{if } v = \times \\ \neg x & \text{if } v = - \\ true & \text{if } v = * \end{cases}$$

Decision Tables, Example

T	r_1	r_2	r_3
c_1	$-$	$\times$	$-$
c_2	$-$	$\times$	$-$
c_3	$\times$	$*$	$-$
a_1	$\times$	$-$	$-$
a_2	$\times$	$\times$	$\times$

- $\mathcal{F}(r_1) = (\neg c_1 \wedge \neg c_2 \wedge c_3) \wedge (a_1 \wedge a_2)$

Decision Tables, Determinism

A decision table T is called deterministic if and only if the premises of all rules are pairwise disjoint, i.e. if

$$\forall r_1 \neq r_2 \in T \bullet \models \neg(\mathcal{F}_{pre}(r_1) \wedge \mathcal{F}_{pre}(r_2)).$$

Otherwise, T is called non-deterministic.

SE-Formalism: Decision Tables, Example

T	r_1	r_2	r_3
c_1	$-$	$\times$	$-$
c_2	$-$	$\times$	$-$
c_3	$\times$	$*$	$-$
a_1	$\times$	$-$	$-$
a_2	$\times$	$\times$	$\times$

SE-Formalism: Decision Tables, Example

T	r_1	r_2	r_3
c_1	−	×	−
c_2	−	×	−
c_3	×	*	−
a_1	×	−	−
a_2	×	×	×

- T is deterministic, because no two rules can be enabled at the same time.

Decision Table Learning Assignments

T	r_1	r_2	r_3
c_1	$\times$	$\times$	$-$
c_2	$\times$	$-$	$*$
c_3	$-$	$\times$	$*$
a_1	$\times$	$-$	$-$
a_2	$-$	$\times$	$-$

1. Give the rule formulae for r_1, r_2, r_3.
2. It is claimed that T is deterministic. Prove this claim.

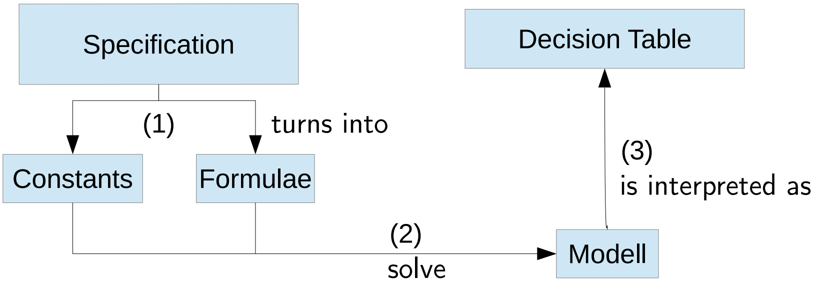
Decision Table Learning Assignments

T	r_1	r_2	r_3
c_1	×	×	—
c_2	×	—	*
c_3	—	×	*
a_1	×	—	—
a_2	—	×	—

1. Give the rule formulae for r_1, r_2, r_3.
2. It is claimed that T is deterministic. Prove this claim.

- Creating learning assignments is hard.
  - Consistency constraints
  - Quality can vary
- Consistency constraints are formally and precisely defined
- Decision tables are mathematical objects

Generating Decision Tables, Overview



Generating Decision Tables, Consistency

T	r_1	r_2
c_1	$v_{1,1}$	$v_{1,2}$
c_2	$v_{2,1}$	$v_{2,2}$

Generating Decision Tables, Consistency

T	r_1	r_2
c_1	$v_{1,1}$	$v_{1,2}$
c_2	$v_{2,1}$	$v_{2,2}$

$$\forall r_1 \neq r_2 \in T \bullet \models \neg(\mathcal{F}_{pre}(r_1) \wedge \mathcal{F}_{pre}(r_2))$$

Generating Decision Tables, Consistency

T	r_1	r_2
c_1	$v_{1,1}$	$v_{1,2}$
c_2	$v_{2,1}$	$v_{2,2}$

$$\forall r_1 \neq r_2 \in T \bullet \forall c_1, c_2 \bullet \neg(\mathcal{F}_{pre}(r_1) \wedge \mathcal{F}_{pre}(r_2))$$

Generating Decision Tables, Consistency

T	r_1	r_2
c_1	$v_{1,1}$	$v_{1,2}$
c_2	$v_{2,1}$	$v_{2,2}$

$$\forall r_1 \neq r_2 \in T \bullet \forall c_1, c_2 \bullet \neg((F(v_{1,1}, c_1) \wedge F(v_{2,1}, c_2)) \wedge (F(v_{1,2}, c_1) \wedge F(v_{2,2}, c_2)))$$

Generating Decision Tables, Consistency

T	r_1	r_2
c_1	$v_{1,1}$	$v_{1,2}$
c_2	$v_{2,1}$	$v_{2,2}$

$$\forall r_1 \neq r_2 \in T \bullet \forall c_1, c_2 \bullet \neg((G_{1,1} \wedge G_{2,1}) \wedge (G_{1,2} \wedge G_{2,2}))$$

with

$$G_{i,j} := (v_{i,j} = \times \wedge c_i) \vee (v_{i,j} = - \wedge \neg c_i) \vee (v_{i,j} = *)$$

Generating Decision Tables, Consistency

T	r_1	r_2
c_1	$v_{1,1}$	$v_{1,2}$
c_2	$v_{2,1}$	$v_{2,2}$

$$\forall r_1 \neq r_2 \in T \bullet \forall c_1, c_2 \bullet \neg((G_{1,1} \wedge G_{2,1}) \wedge (G_{1,2} \wedge G_{2,2}))$$

with

$$G_{i,j} := (v_{i,j} = \times \wedge c_i) \vee (v_{i,j} = - \wedge \neg c_i) \vee (v_{i,j} = *)$$

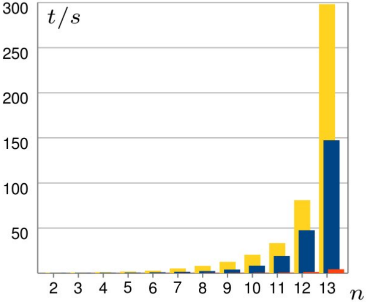
satisifable with $v_{1,1}, \ldots, v_{2,2}$ free
results in some deterministic decision table

Generating Decision Tables, Quality

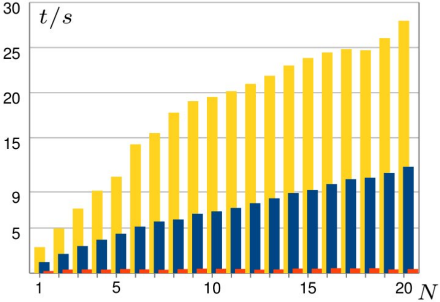
- each symbol should appear
  $\forall s \in \{\times, -, *\} \bullet (v_{1,1} = s) \vee (v_{1,2} = s) \vee (v_{2,1} = s) \vee (v_{2,2} = s)$
- rules not the same
  $(v_{1,1} \neq v_{1,2}) \vee (v_{2,1} \neq v_{2,2})$
- counting $*$
  $n = (v_{1,1} = *) + (v_{1,2} = *) + (v_{2,1} = *) + (v_{2,2} = *)$
- ...

Results in some good decision table

Evaluation



(a) Generation of one decision table of size $(n+n) \times n$.



(b) Generation of a new bunch of 100 decision tables of size $(3+3) \times 3$.

Applications and Future Work

Possible Applications:

- check own learning assignment
- generate a learning assignment
- generate a different learning assignment
- for exams
- for exercise sheets
- for self-learning
- ...

Applications and Future Work

Possible Applications:

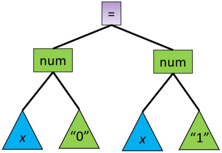
- check own learning assignment
- generate a learning assignment
- generate a different learning assignment
- for exams
- for exercise sheets
- for self-learning
- ...

Future Work:

- more decision table properties
- more quality formalizations
- other formalisms
- ...

Related Works

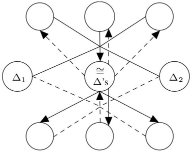
Shenoy, Aparanji, Sripradha, and Kumar [2016]



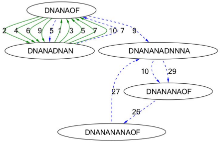
Singh, Gulwani, and Rajamani [2012]

$$\lim_{n \to \infty} \sum_{i=0}^{n} \frac{C_0 i^2 + C_1 i + C_2}{(C_3)^i} = \frac{C_4}{C_5}$$

Alvin, Gulwani, Majumdar, and Mukhopadhyay [2015]



Andersen, Gulwani, and Popovic [2013]



Fin

Thank you - Questions?

References

Chris Alvin, Sumit Gulwani, Rupak Majumdar, and Supratik Mukhopadhyay. Automatic synthesis of geometry problems for an intelligent tutoring system. *CoRR*, abs/1510.08525, 2015. URL http://arxiv.org/abs/1510.08525.

Erik Andersen, Sumit Gulwani, and Zoran Popovic. A trace-based framework for analyzing and synthesizing educational progressions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '13, pages 773–782, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-1899-0. doi: 10.1145/2470654.2470764. URL http://doi.acm.org/10.1145/2470654.2470764.

Helmut Balzert. *Lehrbuch der softwaretechnik: Basiskonzepte und requirements engineering*. Springer-Verlag, 2009.

Norbert M. Seel. *Lernaufgaben und Lernprozesse*. Studienbuch Pädagogik. W. Kohlhammer, Stuttgart, 1981. ISBN 3-17-007198-X.

Varun Shenoy, Ullas Aparanji, K. Sripradha, and Viraj Kumar. Generating DFA construction problems automatically. In *International Conference on Learning and Teaching in Computing and Engineering, LaTICE 2016, Mumbai, India, March 31 - April 3, 2016*, pages 32–37, 2016. doi: 10.1109/LaTiCE.2016.8. URL https://doi.org/10.1109/LaTiCE.2016.8.

Rohit Singh, Sumit Gulwani, and Sriram K. Rajamani. Automatically generating algebra problems. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence, July 22-26, 2012, Toronto, Ontario, Canada.*, 2012. URL http://www.aaai.org/ocs/index.php/AAAI/AAAI12/paper/view/5133.