WIKIPEDIA

Draft:Object state testing

Object state testing is a formal white-box software test design method testing object state behavior. State testing is complementary to local error detection of functional and structural testing. Interaction of class member functions are tested through object state transitions.[1] An object behavior is state based arranged due to method invocation sequence, which behavior is established due to parameter values and state, providing successor and return value. The ranges of values of a subset of member data of the object defines a state of an object. State transitions are deterministic finite due to the conditional execution of member functions. Object state tests can be determined from the source code (white-box) or from the specifications (black-box).

Contents

Requirements

1. Initial state
2. Parameter arguments
3. Expected return value
4. Expected object successor state

Defects

To ensure robustness, all void and valid method sequence invocation combinations need to be tested using state test cases[2]. Testing the object state brings out below defects.

1. Missing state transition
2. Void transitions
3. Wrong return value
4. Void successor state

Object States

The behavior of a class object is definable characterized on inhabited data member values. The run time execution path is determined due to evaluating member and parameter values in decisions (condition), effecting the class object behavior. Object states are segmented in independent domain member data interval combinations leading towards different execution paths.

State characterization

1. Examine all path conditions of member functions for conditional literals
2. For each conditional literal found, form domain intervals such that the conditional literal is evaluated to either True or False for all values in the interval.

Coverage

The amount of states and paths are circumscribing test state coverage. Due to possible infinite cyclic path sequences the path length needs limitation. The amount of methods n is a minimum criterion of path length for permutations of method sequences $n!$. Instead of limiting method permutation sequences an acyclic spanning tree is addressing systematically path length including negative and positive path combinations through a state space.

Example

A soda machine requires two 50c coins for a soda. It provides five methods and three class member variables. The variable tmp holds temporary coins till a customer withdraws a soda[3].

```python
#!/anaconda/bin/python
class SodaMachine:

    total = 0
    tmp = 0
    withDraw = False

    def __init__(self):
        self.reset()

    def add50c(self):
        self.tmp = self.tmp + 1
        if self.tmp > 1:
            self.withDraw = True

    def return50cs(self):
        self.tmp = 0

    def canWithDraw(self):
        return self.withDraw

    def draw(self):
        if self.withDraw:
            self.total = self.total + self.tmp
            self.tmp = 0
            self.withDraw = False

    def reset(self):
        self.total = 0
        self.tmp = 0
        self.withDraw = False
```

```python
if __name__ == '__main__':
    sodaM = SodaMachine()
    sodaM.add50c()
    sodaM.add50c()
    sodaM.draw()
```

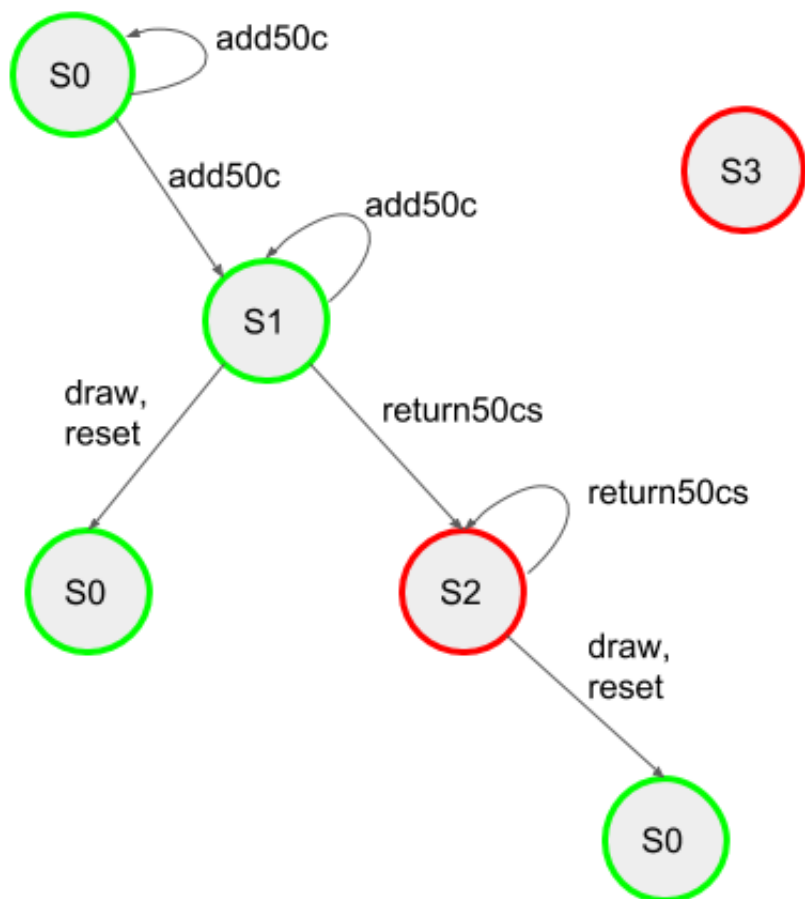## State Definition

The class contains the conditions:

1. ```
   add50c(): if self.tmp > 1
   ```

2. ```
   draw(): if self.withDraw
   ```

The two conditions are segmenting the class member variable values into four states S_1, S_2, S_3, S_4:

State	Member Interval		validity
	tmp	withDraw	
S_0	$x < 1$	False	✓
S_1	$x > 1$	True	✓
S_2	$x < 1$	True	✗
S_3	$x > 1$	False	✗

State transition spanning tree

See also

- Multiple condition coverage
- Control flow graph
- Decision to decision path
- Elementary comparison testing

References

1. Kung, D. C.; Suchak, N.; Gao and, J.; Hsia, P.; Toyoshima, Y.; Chen, C. (1994). "On Object State Testing". *Proceedings of Computer Software and Applications Conference.* IEEE Computer Society Press. pp. 222--227.

2. Tilo Linz (2014). *Testing in Scrum*, p. 51, Rocky Nook Inc., Santa Barbara. ISBN 978-1-937538-39-2

3. https://github.com/jbloemendal/SodaMachine *J. Bloemendal (2018), Test Case Design*

> ### Review waiting, please be patient.
>
> This may take 2–3 weeks or more, since drafts are reviewed in no specific order. There are 1363 pending submissions waiting for review.
>
> ---
>
> - If the submission is **accepted**, then this page will be moved into the article space.
> - If the submission is **declined**, then the reason will be

posted here.

- In the meantime, you can continue to improve this submission by editing normally.
- If you need extra help, please **ask us a question** (https ://en.wikipedia.org/w/index.php?title=Wikipedia:WikiProj ect_Articles_for_creation/Help_desk&action=edit&secti on=new&nosummary=1&preload=Template:AFC_submi ssion/draft/HD_preload&preloadparams%5B%5D=Draft :Object_state_testing) at the AfC Help Desk or get **live help** from experienced editors. *Please note that these help venues are only for assistance editing or submitting your article – not for requesting that your submission be reviewed.*

How to improve your article

- Wikipedia:Contributing to Wikipedia – a basic overview on how to edit Wikipedia.
- Help:Wikitext – how to use the markup
- Help:Referencing for beginners – how to include references
- Wikipedia:Article development – how to develop your article
- Wikipedia:Writing better articles – how to improve your article
- Wikipedia:Verifiability – make sure your article includes reliable third-party sources

---

Editor resources

- *Find sources:* Google (https://www.google.com/search?as _eq=wikipedia&q=%22Object+state+testing%22&num=50) (books (https://www.google.com/search?tbs=bks:1&q=%2 2Object+state+testing%22+-wikipedia) · news (https://ww w.google.com/search?tbm=nws&q=%22Object+state+testi ng%22+-wikipedia) · newspapers (https://www.google.com /search?&q=%22Object+state+testing%22+site:news.goo gle.com/newspapers&source=newspapers) · scholar (https ://scholar.google.com/scholar?q=%22Object+state+testing %22) · free images (https://www.google.com/search?safe= off&tbs=sur:fmc&tbm=isch&q=%22Object+state+testing% 22+-site:wikipedia.org+-site:wikimedia.org) · WP refs (http s://www.google.com/custom?hl=en&cx=00773483090829 5939403%3Agalkqgoksq0&cof=FORID%3A13%3BAH%3 Aleft%3BCX%3AWikipedia%2520Reference%2520Search &q=%22Object+state+testing%22)) · FENS (https://en.wiki pedia.org/wiki/Wikipedia:Free_English_newspaper_source s) · HighBeam (https://www.highbeam.com/Search?search Term=%22Object+state+testing%22) · JSTOR (https://ww w.jstor.org/action/doBasicSearch?Query=%22Object+state +testing%22&acc=on&wc=on) · NYT (https://www.nytimes.

com/search/%22Object+state+testing%22) · TWL (https://
wikipedialibrary.wmflabs.org/partners/)

- *Easy tools*: Citation bot (https://tools.wmflabs.org/citations/
  process_page.php?edit=toolbar&slow=1&page=Draft:Obje
  ct_state_testing) (help) I *Advanced:* Fix ambiguous links (h
  ttps://dispenser.info.tm/~dispenser/cgi-bin/dab_solver.py?p
  age=Draft:Object_state_testing&commonfixes=yes&client
  =Dablinks) · Fix bare URLs (https://tools.wmflabs.org/refill/
  result.php?page=Draft:Object_state_testing&defaults=y) ·
  Fix broken links (https://dispenser.info.tm/~dispenser/cgi-bi
  n/webchecklinks.py?page=Draft:Object_state_testing#vie
  w:0.0.0.1.1.1)

Reviewer tools

- Instructions · ⋈ · Object state testing (talk: + (https://en.wik
  ipedia.org/w/index.php?title=Talk:Object_state_testing&act
  ion=edit&editintro=Template:AFC_submission/banner_edit
  intro&preload=Template:AFC_submission/banner_preload
) bio (https://en.wikipedia.org/w/index.php?title=Talk:Objec
  t_state_testing&action=edit&editintro=Template:AFC_sub
  mission/banner_editintro&preload=Template:AFC_submis
  sion/banner_preload_(bio))) · (log (https://en.wikipedia.org/
  w/index.php?title=Index.php&title=Special:Log&page=Obj
  ect_state_testing)) · Earwig (https://tools.wmflabs.org/copy
  vios?lang=en&project=wikipedia&title=Draft:Object_state_t
  esting&url=) · reFill (https://tools.wmflabs.org/refill/result.ph
  p?page=Draft:Object_state_testing&defaults=y) · Citation
  Bot (https://tools.wmflabs.org/citations/process_page.php?
  edit=toolbar&slow=1&user=&page=Draft:Object_state_test
  ing) · (Search: Google (https://www.google.com/search?&
  q=Object+state+testing), Bing (https://www.bing.com/searc
  h?q=Object+state+testing), Wikipedia (https://en.wikipedia
  .org/w/index.php?title=Special:Search&search=Object+sta
  te+testing&fulltext=Search)) · Submitted 6 days ago by
  143.179.14.34 (talk: H (https://en.wikipedia.org/w/index.ph
  p?title=User_talk:143.179.14.34&action=edit&editintro=Te
  mplate:AFC_submission/user_talk_editintro_(hold)&preloa
  d=Template:AFC_submission/user_talk_preload_(hold)&pr
  eloadtitle=Your+submission+at+%5B%5BWP%3AAFC%7
  CArticles+for+creation%5D%5D&section=new) D (https://
  en.wikipedia.org/w/index.php?title=User_talk:143.179.14.3
  4&action=edit&editintro=Template:AFC_submission/user_t
  alk_editintro_(declined)&preload=Template:AFC_submissi
  on/user_talk_preload_(declined)&preloadtitle=Your+submi
  ssion+at+%5B%5BWP%3AAFC%7CArticles+for+creation
  %5D%5D&section=new) + (https://en.wikipedia.org/w/inde
  x.php?title=User_talk:143.179.14.34&action=edit&editintro
  =Template:AFC_submission/user_talk_editintro&preload=
  Template:AFC_submission/user_talk_preload&preloadtitle
  =Your+submission+at+%5B%5BWP%3AAFC%7CArticles
  +for+creation%5D%5D&section=new)) · Last edited 0

seconds ago by 145.137.107.54

Retrieved from "https://en.wikipedia.org/w/index.php?title=Draft:Object_state_testing&oldid=873685725"

This page was last edited on 14 December 2018, at 13:35 (UTC).