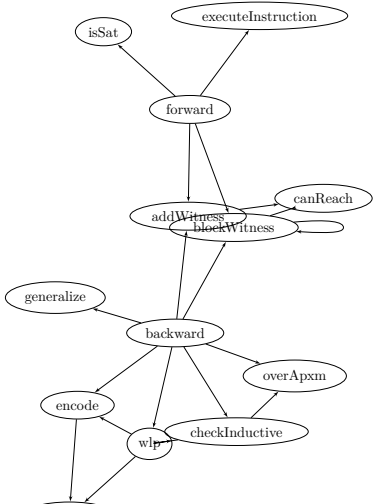
По статье «Staged Selective Parser Combinators»
с конференции IFCP 2020

June 17, 2021

Build date: June 17, 2021


executeInstruction
isSat
forward
canReach
addWitness
blockWitness
generalize
backward
overApxm
encode
checkInductive
wlp

L

```
val propDirSymExec: locs:(loc set) -> program -> unit
(* возвращает хрен знает что
*)

val start: loc -> unit
(* обновляет startState, Qf, T(loc)
и много раз вызывает addWitness
*)
```

```
type pob = { loc : loc; φ : formula; lvl : level }
type state = {
loc : loc; PC : formula; store : store;
loc0 : formula; lvl : level; pobs : Set<pob> }
```

```
curLvl : level

mainPobs : pob set
(* главные запросы *)
pobs : pob set

Qf : state set

Qb : (pob * state) set

witnesses : pob -> state set
blockedLocs : (pob, loc set) map
pobsLocs : loc set
T : loc -> state set
L : loc × level -> formula set
```

```
val propDirSymExec: loc set -> program -> unit
(* изменяет pobs, curLvl
  вызывает ChooseAction, forward, backward, start, nextLevel
*)

val forward: state -> unit
(* обновляет Qf,T
    вызывает blockWitness
*)

val backward: pob -> state -> program -> level -> unit
(* обновляет Qb, lvl, принятый pob, child
    вызывает WLP, overApxm, addWitness, encode,
              checkInductive, generalize, answerYes
*)
```

```
val answerYes: pob -> 'wtf
val answerNo: pob -> 'wtf
(* выдают финальный ответ, не реализованы,
  Должны смотреть на pob, с помощью child находит его детей (? или родителей)
  и обрушать их во множествах pobs и mainPobs
 *)
val child: (pob, pob set) map
(* Так себе описан *)

val canReach: loc -> loc -> locs -> bool
(* проверяет наличие межпроцедурного пути в CFG от локации loc1 до loc2,
не посещающего ни одну из локаций locs
*)
val checkInductive: level -> unit
(* обновляет L
  вызывает WLP, overApx
*)
```

```
val addWitness: state -> pob -> unit
(* вызывает canReach
  обновляет s.pobs и witnesses[state]
*)

val blockWitness: state -> pob -> unit
(* обновляет witnesses, blockedLocs
  вызывает blockWitness, canReach
*)
```

```
val eL: loc -> level -> formula
```

Сильно использует в себе функцию

```
val over_apxm : loc -> lvl:level -> cur_lvl:level -> formula
val encode: state -> formula
(* Использует mkPrime *)

val wlp: state -> formula -> formula
(* Использует encode & mkPrime *)
```

Ссылки I


Staged Selective Parser Combinators
Jamie Willis & Nicolas Wu & Matthew Pickering
https://doi.org/10.1145/3409002


Garnishing Parsec With Parsley: A Staged Selective Parser Combinator Library
Jamie Willis
https://www.youtube.com/watch?v=tJcyY9L2z84


Selective Applicative Functors
Andrey Mokhov & Georgy Lukyanov & Simon Marlow & Jeremie Dimino
https://doi.org/10.1145/3341694


Библиотека FastParse для Scala
Documentation

Ссылки II


Try vs. lookahead
https://stackoverflow.com/questions/20020350