RefinedC

Automating the Foundational Verification of C Code with Refined Ownership types

RefinedC

Automated

Guide proof search via a type system

$$\Gamma \vdash e : \tau$$

Foundational

Semantic model in Coq / Iris

Ownership types

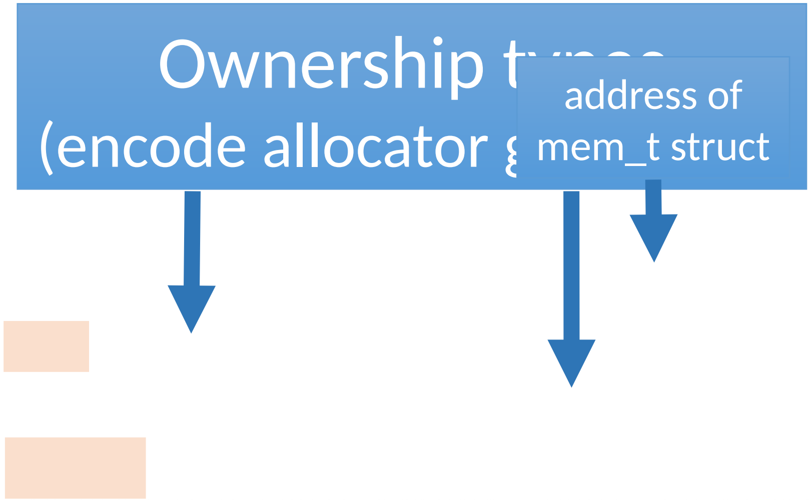
Handle pointers and memory management

Refinement types

Handle functional correctness

RefinedC in action

```
struct mem_t {
  size_t len;
  unsigned char* buffer;
};
```



Ownership types

(encode allocator g

address of mem_t struct

```
void* alloc(struct mem_t* d, size_t size) {
  if(size > d->len) return NULL;
  d->len -= size;
  return d->buffer + d->len;
}
```

RefinedC in action

```
struct mem_t {
  size_t len;
  unsigned char* buffer;
};
```

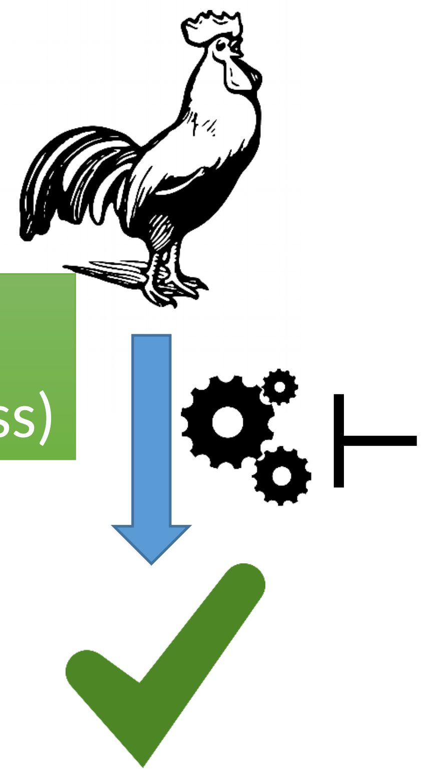
bytes available from allocator

requested allocation size

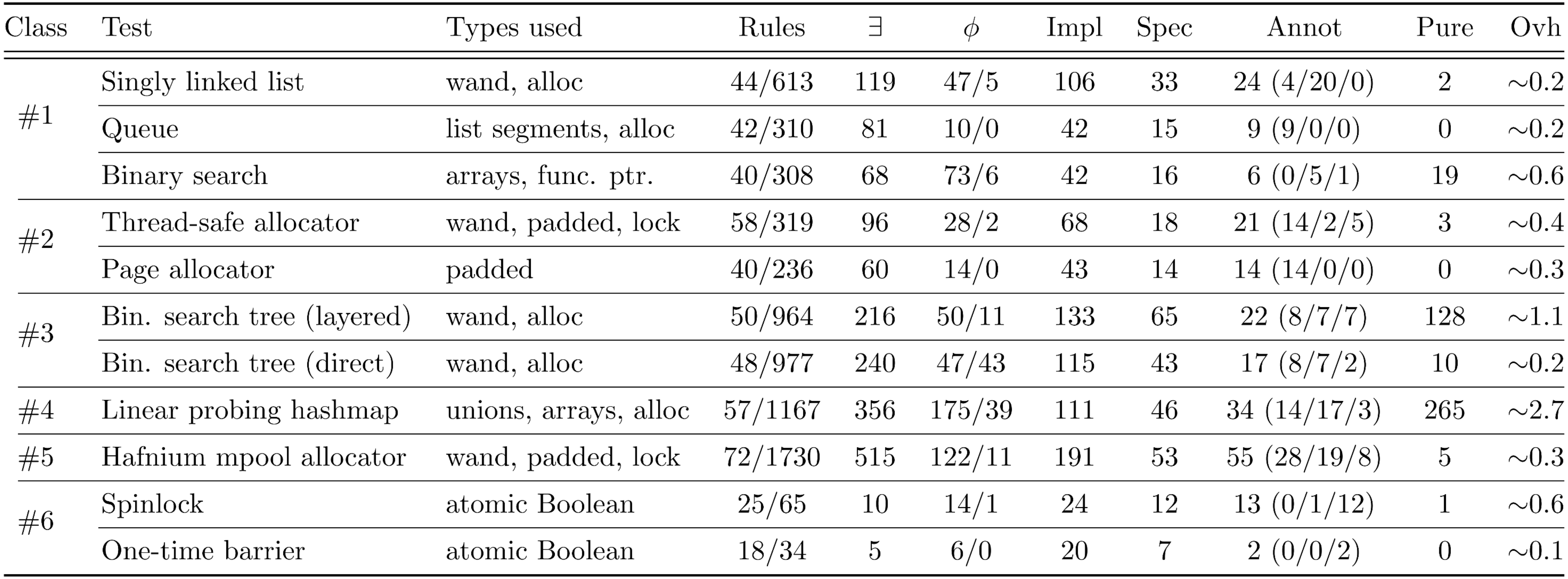
address of mem_t struct

```
[[rc::parameters(                           "p: loc")]]
[[rc::args      ("p @ &own<       mem_t>", "      int<size_t>")]]
[[rc::returns   ("           optional<&own<uninit    , null>>")]]
[[rc::ensures   ("own p                          mem_t")]]
void* alloc(struct mem_t* d, size_t size) {
  if(size > d->len) return NULL;
  d->len -= size;
  return d->buffer + d->len;
}
```

Refinement types
(encoding functional correctness)



Evaluation



Class	Test	Types used	Rules	$\exists$	ϕ	Impl	Spec	Annot	Pure	Ovh
#1	Singly linked list	wand, alloc	44/613	119	47/5	106	33	24 (4/20/0)	2	∼0.2
	Queue	list segments, alloc	42/310	81	10/0	42	15	9 (9/0/0)	0	∼0.2
	Binary search	arrays, func. ptr.	40/308	68	73/6	42	16	6 (0/5/1)	19	∼0.6
#2	Thread-safe allocator	wand, padded, lock	58/319	96	28/2	68	18	21 (14/2/5)	3	∼0.4
	Page allocator	padded	40/236	60	14/0	43	14	14 (14/0/0)	0	∼0.3
#3	Bin. search tree (layered)	wand, alloc	50/964	216	50/11	133	65	22 (8/7/7)	128	∼1.1
	Bin. search tree (direct)	wand, alloc	48/977	240	47/43	115	43	17 (8/7/2)	10	∼0.2
#4	Linear probing hashmap	unions, arrays, alloc	57/1167	356	175/39	111	46	34 (14/17/3)	265	∼2.7
#5	Hafnium mpool allocator	wand, padded, lock	72/1730	515	122/11	191	53	55 (28/19/8)	5	∼0.3
#6	Spinlock	atomic Boolean	25/65	10	14/1	24	12	13 (0/1/12)	1	∼0.6
	One-time barrier	atomic Boolean	18/34	5	6/0	20	7	2 (0/0/2)	0	∼0.1

☺

- Separation logic automation technique (Lithium)
- Reasoning about pointers / local variables using ownership types
- Reasonably accurate memory model (VIP, based on PNVI-ae-udi)
- Frontend for C code with annotations
- Using types to guide the proof search
- Extensibility of the type system via Iris
- Foundational proofs
- Duff’s device

☹

- Relatively young
- Amount of annotations
  → Annotation inference via biabduction
- Connection to assembly code
  → Translation validation
- Performance (large examples take minutes)
- Error messages currently expose state of type system
- Automation for pure sideconditions can be improved
- Missing features of C (floating point, strings, block scoped local variables, seq. points, …)
- Documentation / Tutorials

RefinedC

Available at

https://plv.mpi-sws.org/refinedc/

Automated

$$\Gamma \vdash e : \tau$$

Foundational

Iris