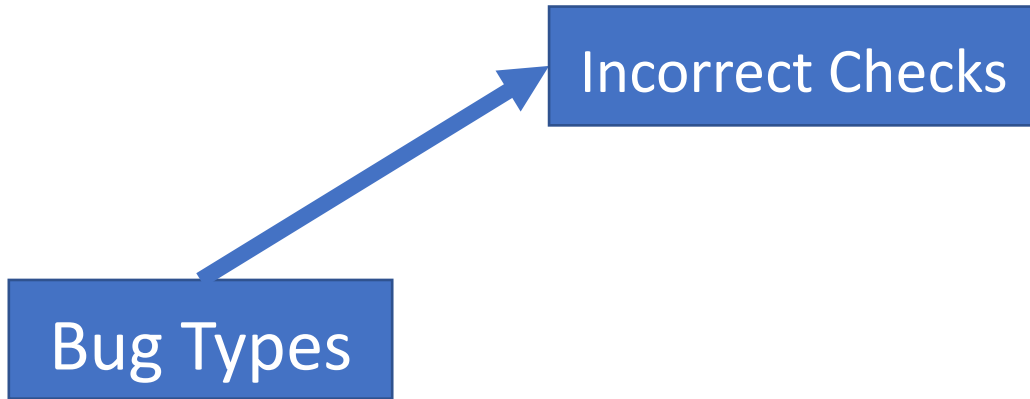


Inference of Error Specifications and Bug Detection Using Structural Similarities

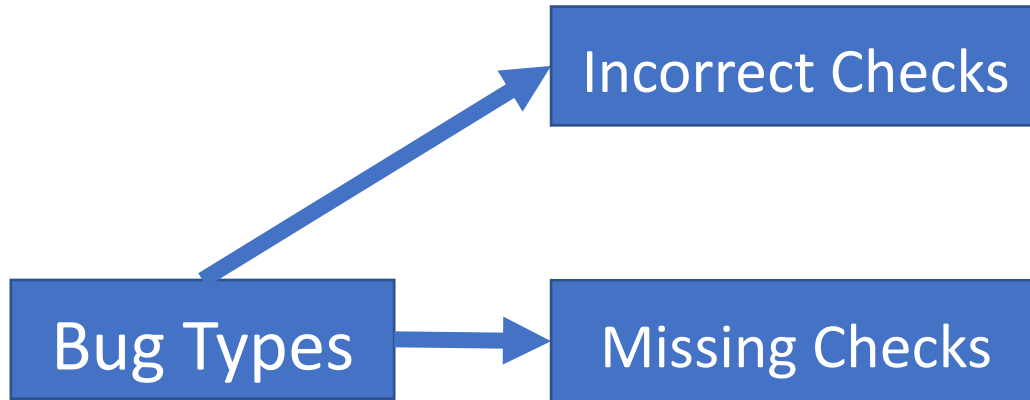
Error Check Bugs

Bug Types

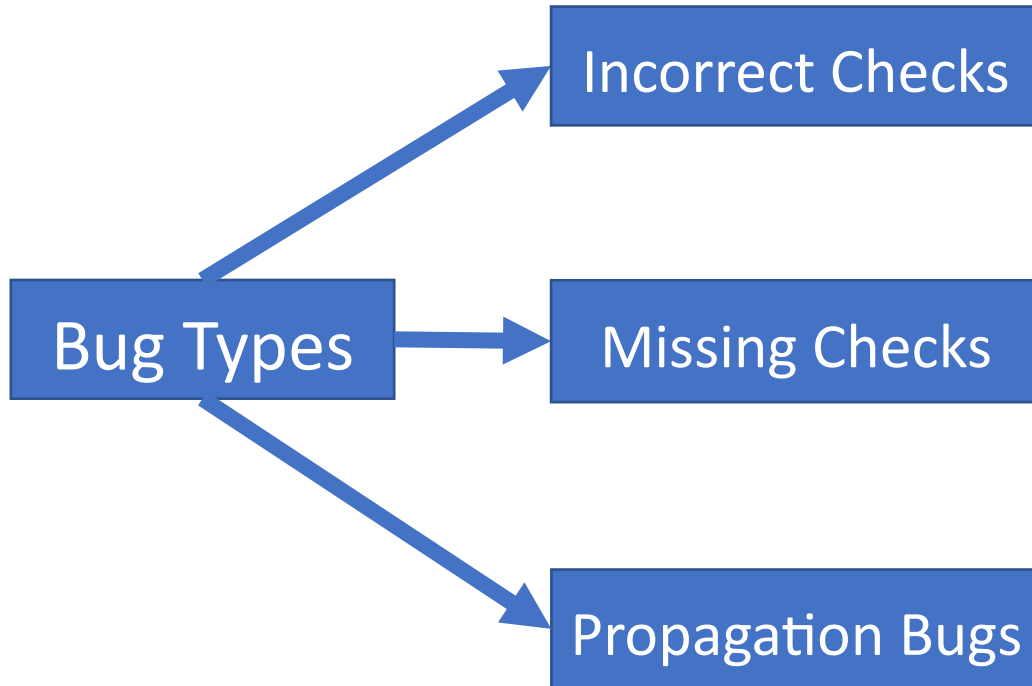
Error Check Bugs



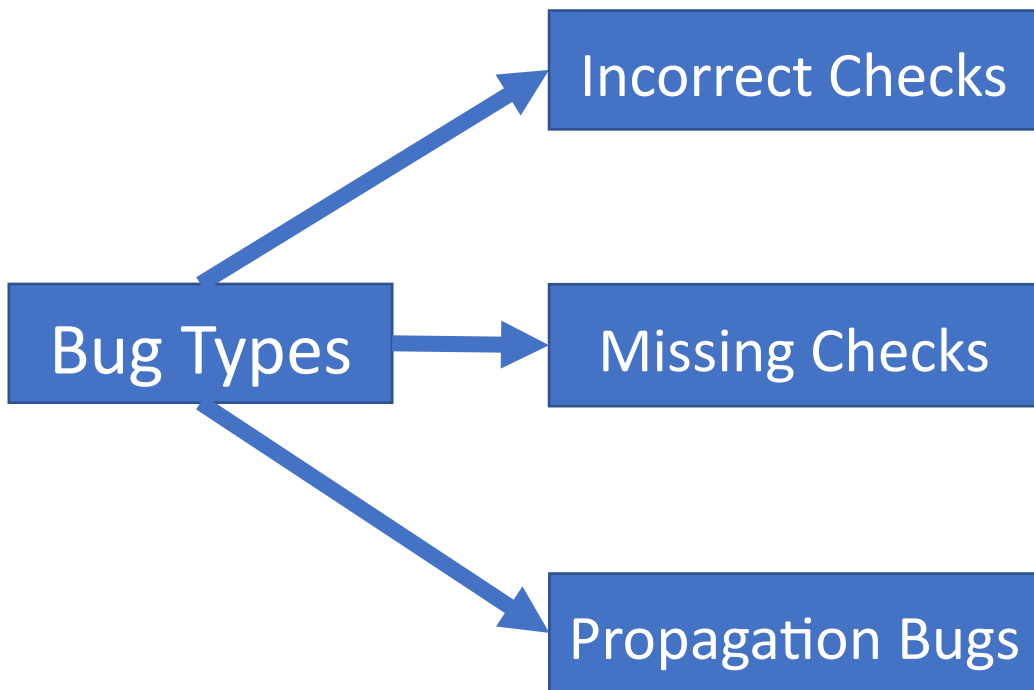
Error Check Bugs



Error Check Bugs

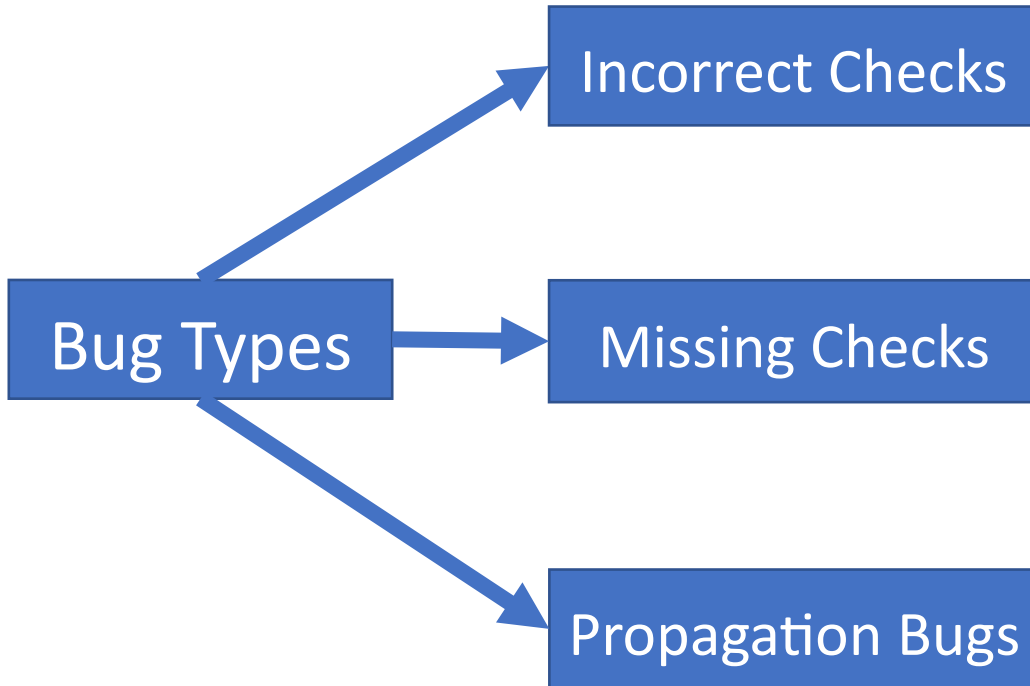


Error Check Bugs



```
if (generate_seed && RAND_bytes_ex(...) < 0)  
goto err;
```

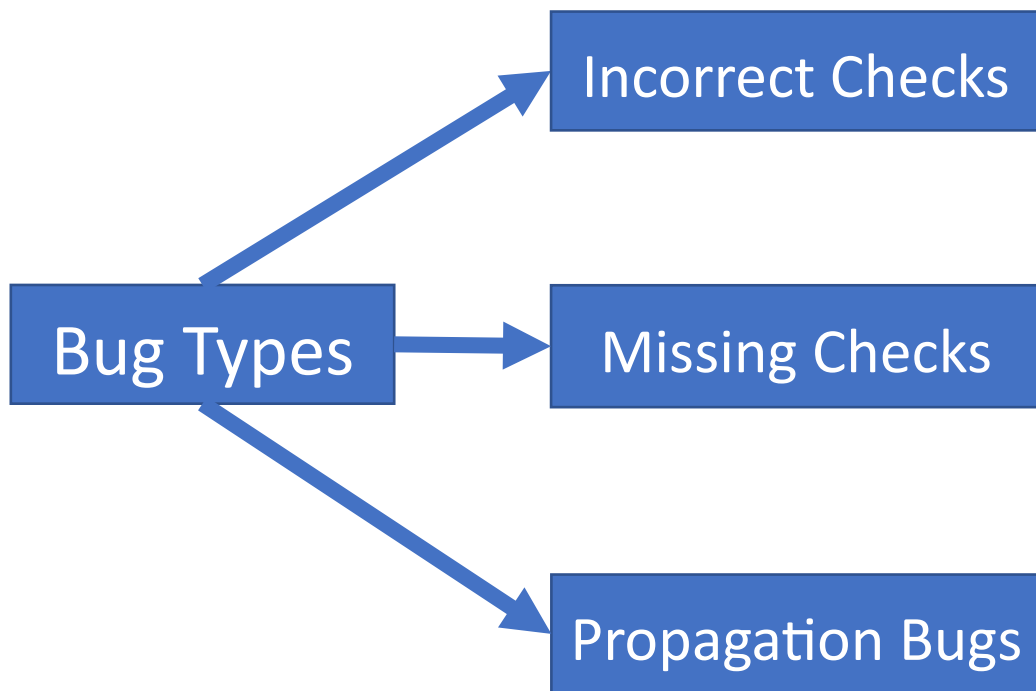
Error Check Bugs



```
if (generate_seed && RAND_bytes_ex(...) < 0)
    goto err;
```

```
retval = xmlTextWriterStartElement(...);
if (retval == -1)
    return false;
xmlTextWriterEndElement(...);
if (retval == -1)
    return false;
```

Error Check Bugs



```
if (generate_seed && RAND_bytes_ex(...) < 0)
    goto err;
```

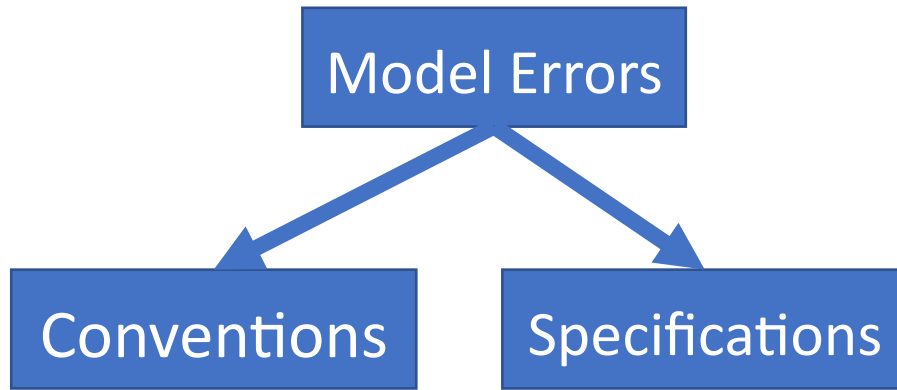
```
retval = xmlTextWriterStartElement(...);
if (retval == -1)
    return false;
xmlTextWriterEndElement(...);
if (retval == -1)
    return false;
```

```
zend_result zend_update_static_property(...) {
    ...
    bool retval = zend_update_static_property_ex(...);
    ...
    return retval;
}
```

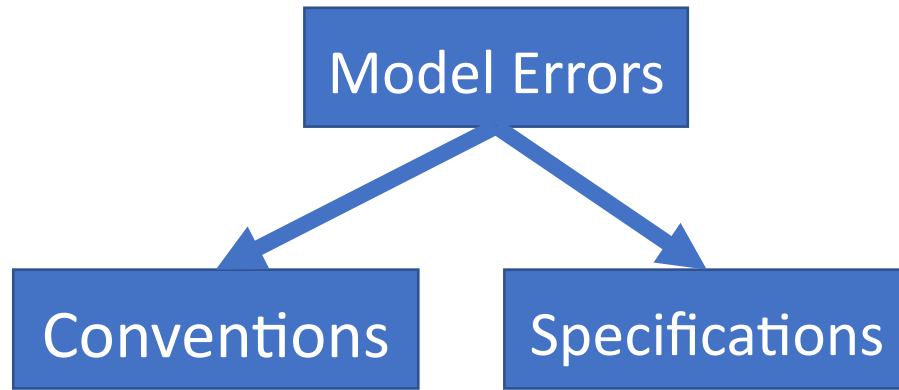

Static Approaches

Model Errors

Static Approaches

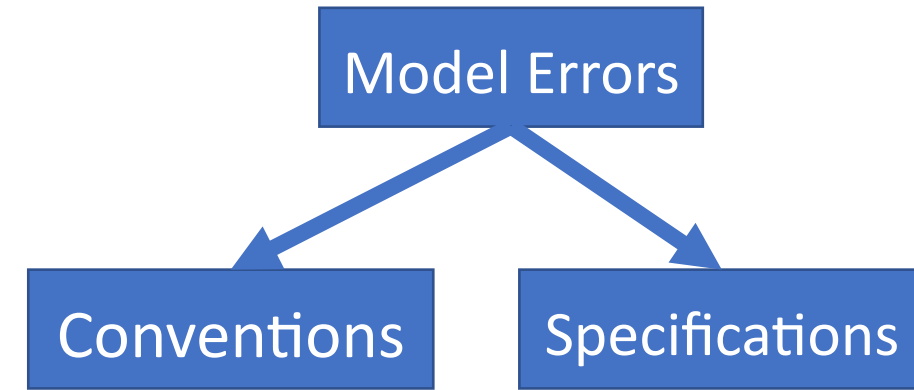


Static Approaches



e.g. NULL, EINVAL, goto error

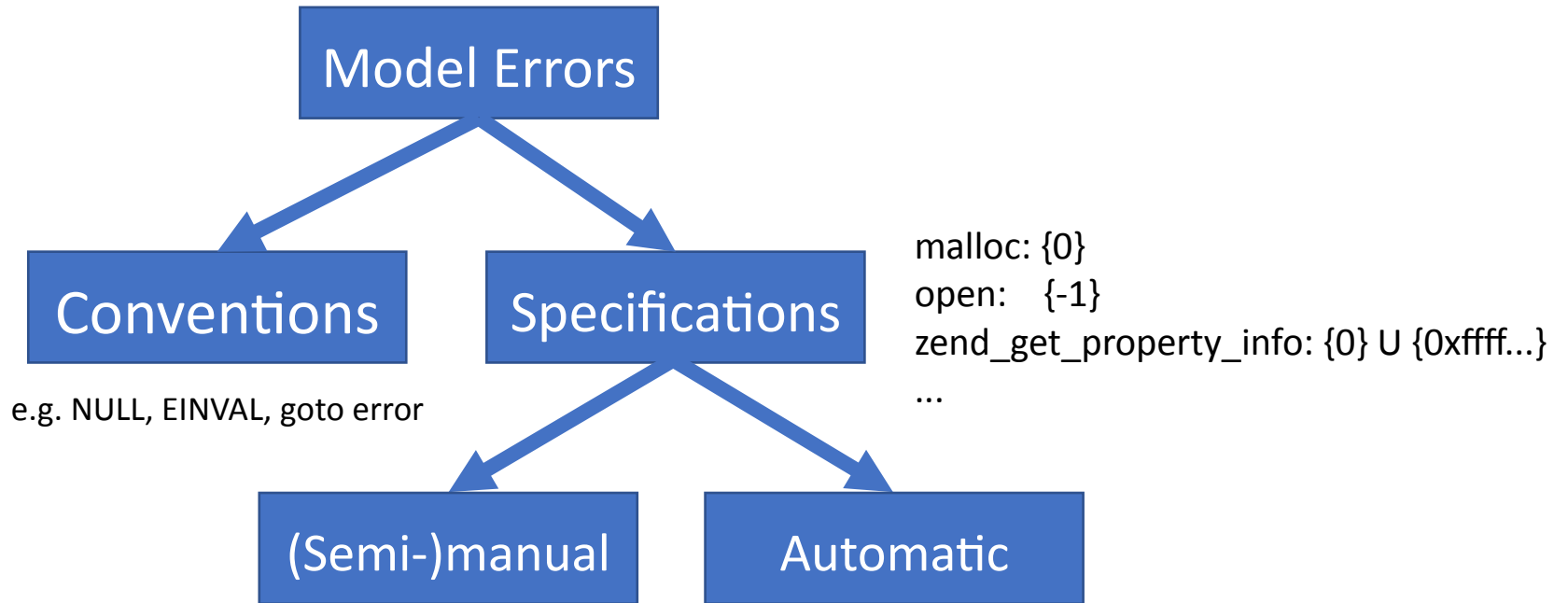
Static Approaches



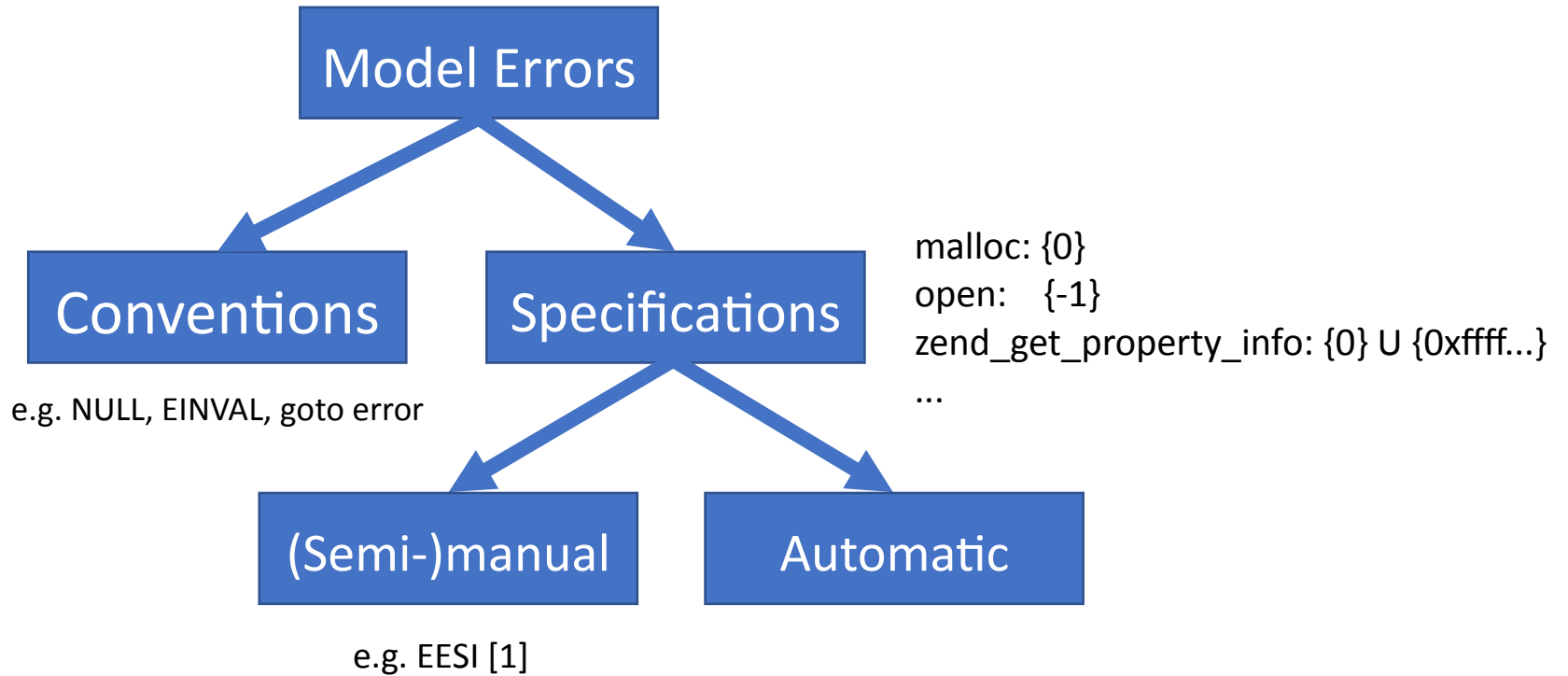
e.g. NULL, EINVAL, goto error

malloc: {0}
open: {-1}
zend_get_property_info: {0} U {0xffff...}
...

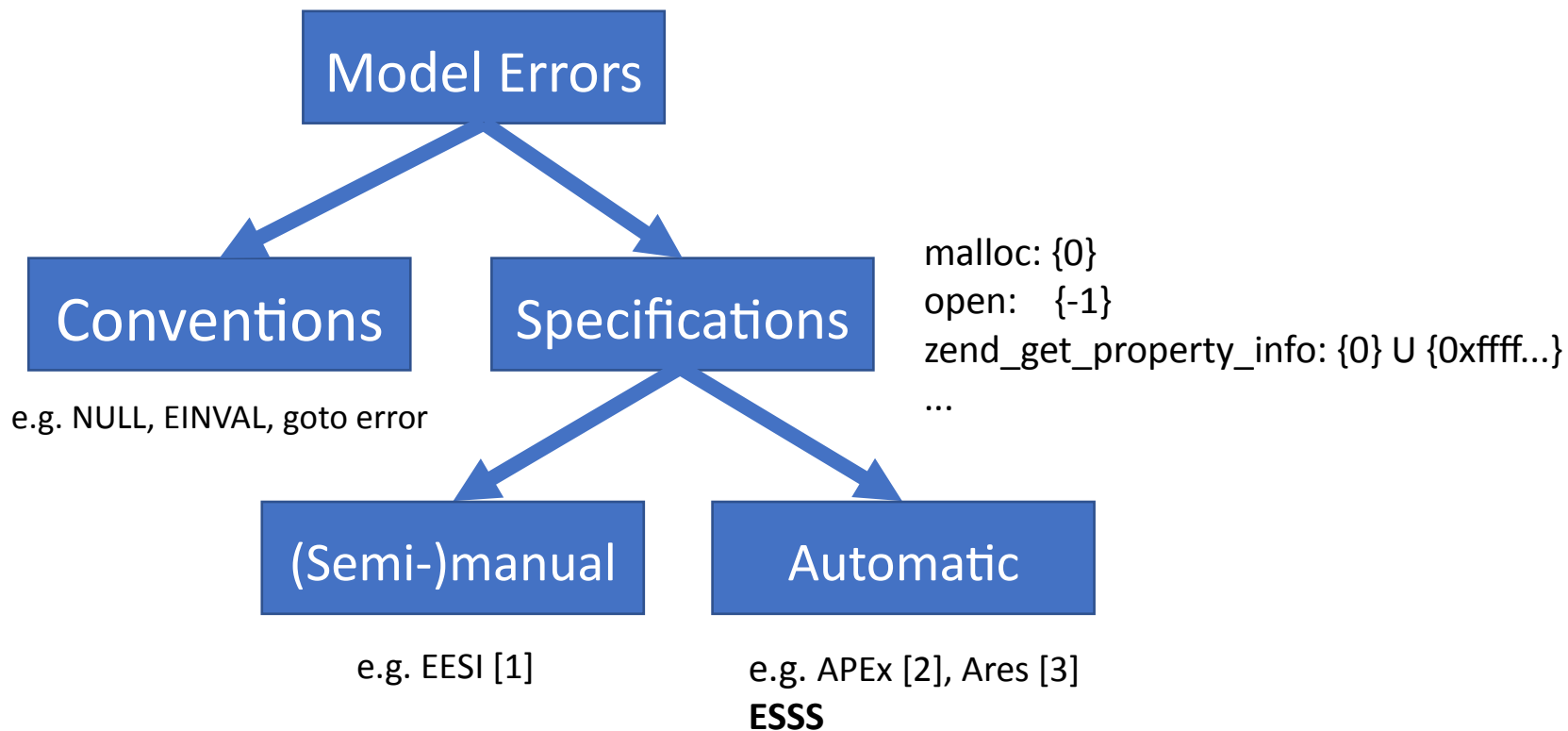
Static Approaches



Static Approaches



Static Approaches



Convention-based Approach

```
int ossl_rsa_sp800_56b_check_keypair(...) {
    int ret = 0;
    if (!rsa_check_public_exponent(rsa->e)) {
        ERR_raise(ERR_LIB_RSA, RSA_R_PUB_EXPONENT_OUT_OF_RANGE);
        return 0;
    }
    BN_CTX *ctx = BN_CTX_new_ex(rsa->libctx);
    if (ctx == NULL)
        return 0;
    if (!BN_mul(...))
        goto err;
    ret = rsa_check_prime_factor(...) && rsa_check_crt_components(...);
    if (ret != 1)
        ERR_raise(ERR_LIB_RSA, RSA_R_INVALID_KEYPAIR);
err:
    BN_CTX_free(ctx);
    return ret;
}
```


Convention-based Approach

```
int ossl_rsa_sp800_56b_check_keypair(...) {
    int ret = 0;
    if (!rsa_check_public_exponent(rsa->e)) {
        ERR_raise(ERR_LIB_RSA, RSA_R_PUB_EXPONENT_OUT_OF_RANGE);
        return 0;
    }
    BN_CTX *ctx = BN_CTX_new_ex(rsa->libctx);
    if (ctx == NULL)
        return 0;
    if (!BN_mul(...))
        goto err;
    ret = rsa_check_prime_factor(...) && rsa_check_crt_components(...);
    if (ret != 1)
        ERR_raise(ERR_LIB_RSA, RSA_R_INVALID_KEYPAIR);
err:
    BN_CTX_free(ctx);
    return ret;
}
```



Our Approach: Similarities

```
int ossl_rsa_sp800_56b_check_keypair(...) {
    int ret = 0;
    if (!rsa_check_public_exponent(rsa->e)) {
        ERR_raise(ERR_LIB_RSA, RSA_R_PUB_EXPONENT_OUT_OF_RANGE);
        return 0;
    }
    BN_CTX *ctx = BN_CTX_new_ex(rsa->libctx);
    if (ctx == NULL)
        return 0;
    if (!BN_mul(...))
        goto err;
    ret = rsa_check_prime_factor(...) && rsa_check_crt_components(...);
    if (ret != 1)
        ERR_raise(ERR_LIB_RSA, RSA_R_INVALID_KEYPAIR);
err:
    BN_CTX_free(ctx);
    return ret;
}
```



Our Approach: Similarities

```
int ossl_rsa_sp800_56b_check_keypair(...) {  
    int ret = 0;  
    if (!rsa_check_public_exponent(rsa->e)) {  
        { ERR_raise(ERR_LIB_RSA, RSA_R_PUB_EXPONENT_OUT_OF_RANGE);  
        return 0;  
    }  
    BN_CTX *ctx = BN_CTX_new_ex(rsa->libctx);  
    if (ctx == NULL)  
        return 0;  
    if (!BN_mul(...))  
        goto err;  
    ret = rsa_check_prime_factor(...) && rsa_check crt components(...);  
    if (ret != 1)  
        { ERR_raise(ERR_LIB_RSA, RSA_R_INVALID_KEYPAIR);  
    err:  
        BN_CTX_free(ctx);  
        return ret;  
    }  
}
```



Our Approach: Similarities

```
int ossl_rsa_sp800_56b_check_keypair(...) {
    int ret = 0;
    if (!rsa_check_public_exponent(rsa->e)) {
        ERR_raise(ERR_LIB_RSA, RSA_R_PUB_EXPONENT_OUT_OF_RANGE);
        return 0;
    }
    BN_CTX *ctx = BN_CTX_new_ex(rsa->libctx);
    if (ctx == NULL)
        return 0;
    if (!BN_mul(...))
        goto err;
    ret = rsa_check_prime_factor(...) && rsa_check crt_components(...);
    if (ret != 1)
        ERR_raise(ERR_LIB_RSA, RSA_R_INVALID_KEYPAIR);
err:
    BN_CTX_free(ctx);
    return ret;
}
```

CALL ERR_raise
RETURN 0

CALL ERR_raise
CALL BN_CTX_free
RETURN 0



Our Approach: Similarities

```
int ossl_rsa_sp800_56b_check_keypair(...) {
    int ret = 0;
    if (!rsa_check_public_exponent(rsa->e)) {
        ERR_raise(ERR_LIB_RSA, RSA_R_PUB_EXPONENT_OUT_OF_RANGE);
        return 0;
    }
    BN_CTX *ctx = BN_CTX_new_ex(rsa->libctx);
    if (ctx == NULL)
        return 0;
    if (!BN_mul(...))
        goto err;
    ret = rsa_check_prime_factor(...) && rsa_check crt_components(...);
    if (ret != 1)
        ERR_raise(ERR_LIB_RSA, RSA_R_INVALID_KEYPAIR);
err:
    BN_CTX_free(ctx);
    return ret;
}
```

CALL ERR_raise
RETURN 0

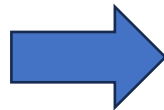
CALL ERR_raise
CALL BN_CTX_free
RETURN 0

Our Approach: Similarities

```
int ossl_rsa_sp800_56b_check_keypair(...) {  
    int ret = 0;  
    if (!rsa_check_public_exponent(rsa->e)) {  
        ERR_raise(ERR_LIB_RSA, RSA_R_PUB_EXPONENT_OUT_OF_RANGE);  
        return 0;  
    }  
    BN_CTX *ctx = BN_CTX_new_ex(rsa->libctx);  
    if (ctx == NULL)  
        return 0;  
    if (!BN_mul(...))  
        goto err;  
    ret = rsa_check_prime_factor(...) && rsa_check crt components(...);  
    if (ret != 1)  
        ERR_raise(ERR_LIB_RSA, RSA_R_INVALID_KEYPAIR);  
err:  
    BN_CTX_free(ctx);  
    return ret;  
}
```

CALL ERR_raise
RETURN 0

CALL ERR_raise
CALL BN_CTX_free
RETURN 0



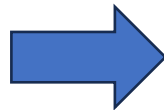
Subsequence match of 2

Our Approach: Similarities

```
int ossl_rsa_sp800_56b_check_keypair(...) {
    int ret = 0;
    if (!rsa_check_public_exponent(rsa->e)) {
        ERR_raise(ERR_LIB_RSA, RSA_R_PUB_EXPONENT_OUT_OF_RANGE);
        return 0;
    }
    BN_CTX *ctx = BN_CTX_new_ex(rsa->libctx);
    if (ctx == NULL)
        return 0;
    if (!BN_mul(...))
    {
        goto err;
        ret = rsa_check_prime_factor(...) && rsa_check crt_components(...);
        if (ret != 1)
            ERR_raise(ERR_LIB_RSA, RSA_R_INVALID_KEYPAIR);
    err:
        BN_CTX_free(ctx);
        return ret;
    }
}
```

CALL BN_CTX_free
RETURN 0

CALL ERR_raise
CALL BN_CTX_free
RETURN 0



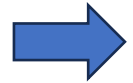
Subsequence match of 2

Expanding Error Specifications

```
int f() {  
    int x = fallible();  
    if (x != 0) {  
        ERR_raise(...);  
        return x;  
    } else {  
        return 0;  
    }  
}
```


Expanding Error Specifications

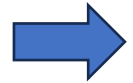
Single check
No similar code



```
int f() {  
    int x = fallible();  
    if (x != 0) {  
        ERR_raise(...);  
        return x;  
    } else {  
        return 0;  
    }  
}
```

Expanding Error Specifications

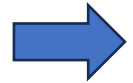
Single check
No similar code



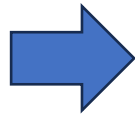
```
int f() {  
    int x = fallible();  
    if (x != 0) {  
        ERR_raise(...);  
        return x;  
    } else {  
        return 0;  
    }  
}
```

Expanding Error Specifications

Single check
No similar code



```
int f() {  
    int x = fallible();  
    if (x != 0) {  
        ERR_raise(...);  
        return x;  
    } else {  
        return 0;  
    }  
}
```



Association analysis

Expanding Error Specifications

```
int f() {  
    void *x = malloc(...);  
    if (x == NULL) {  
        return -1;  
    }  
    ...  
}
```

Expanding Error Specifications

```
int f() {  
    void *x = malloc(...);  
    if (x == NULL) {  
        return -1;  
    }  
    ...  
}
```



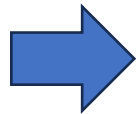
malloc: {0}

Expanding Error Specifications

```
int f() {  
    void *x = malloc(...);  
    if (x == NULL) {  
        return -1;  
    }  
    ...  
}
```



malloc: {0}



f == -1

Pipeline

Pipeline

LLVM bitcode

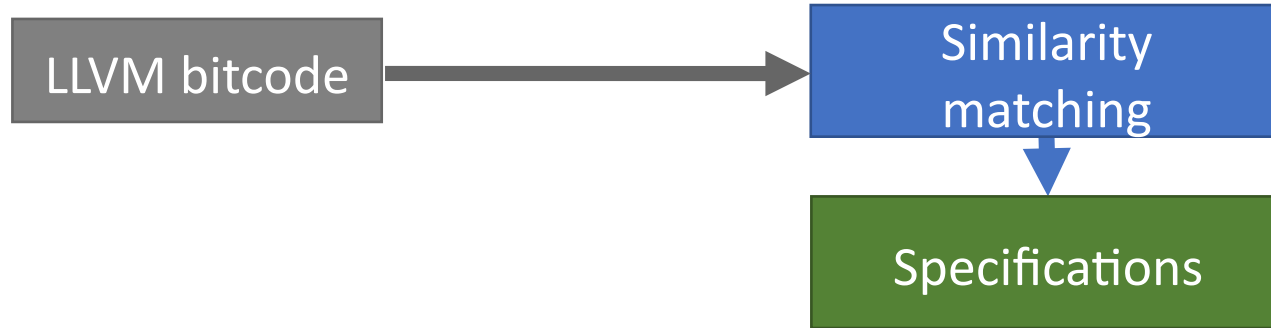
Pipeline



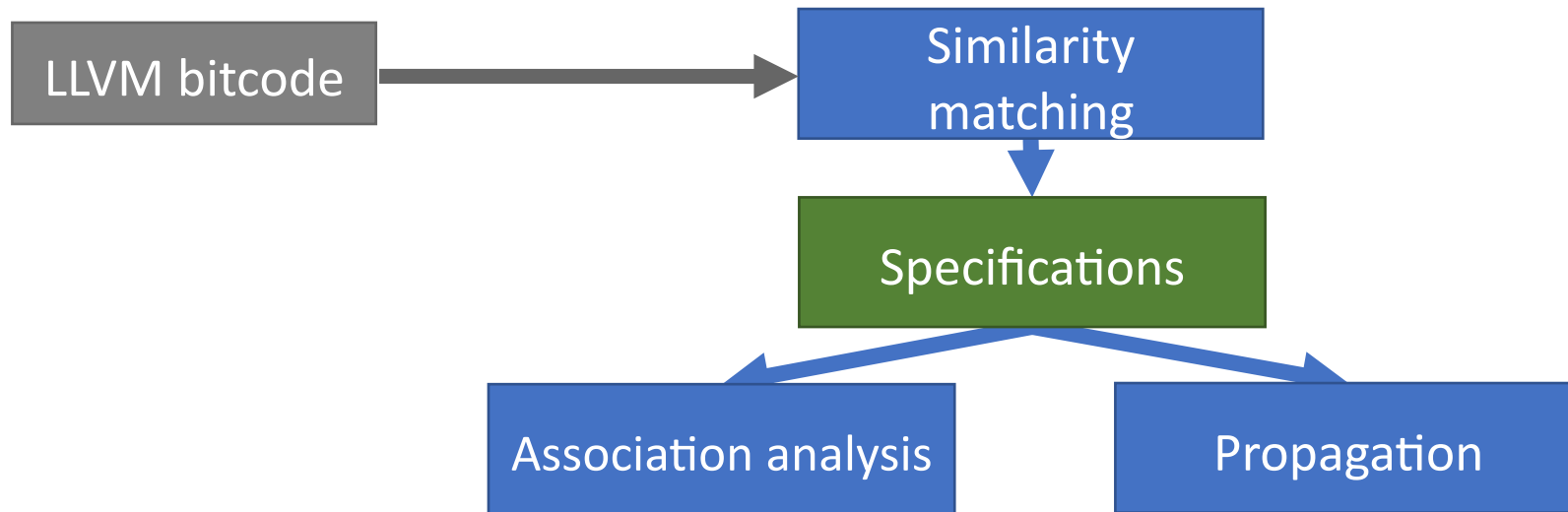
Pipeline



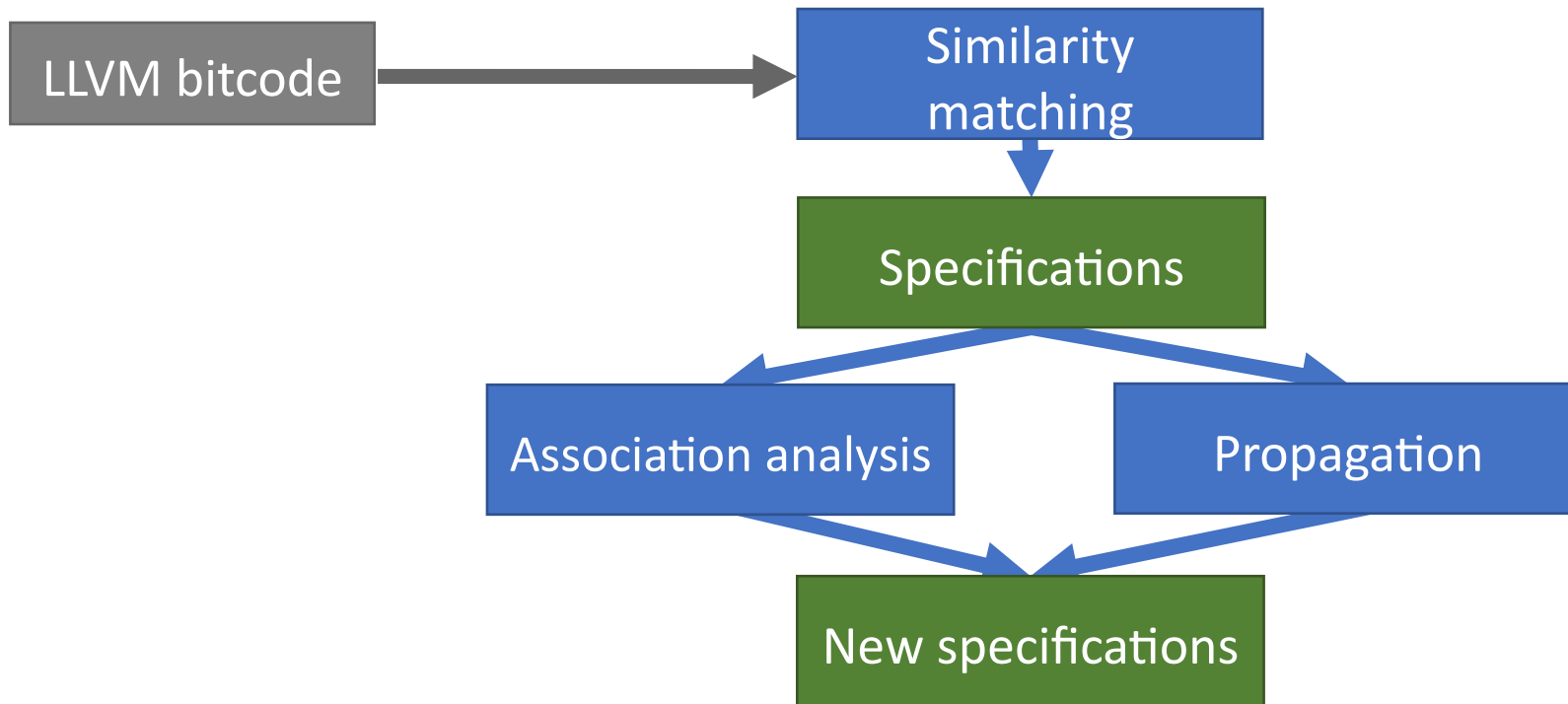
Pipeline



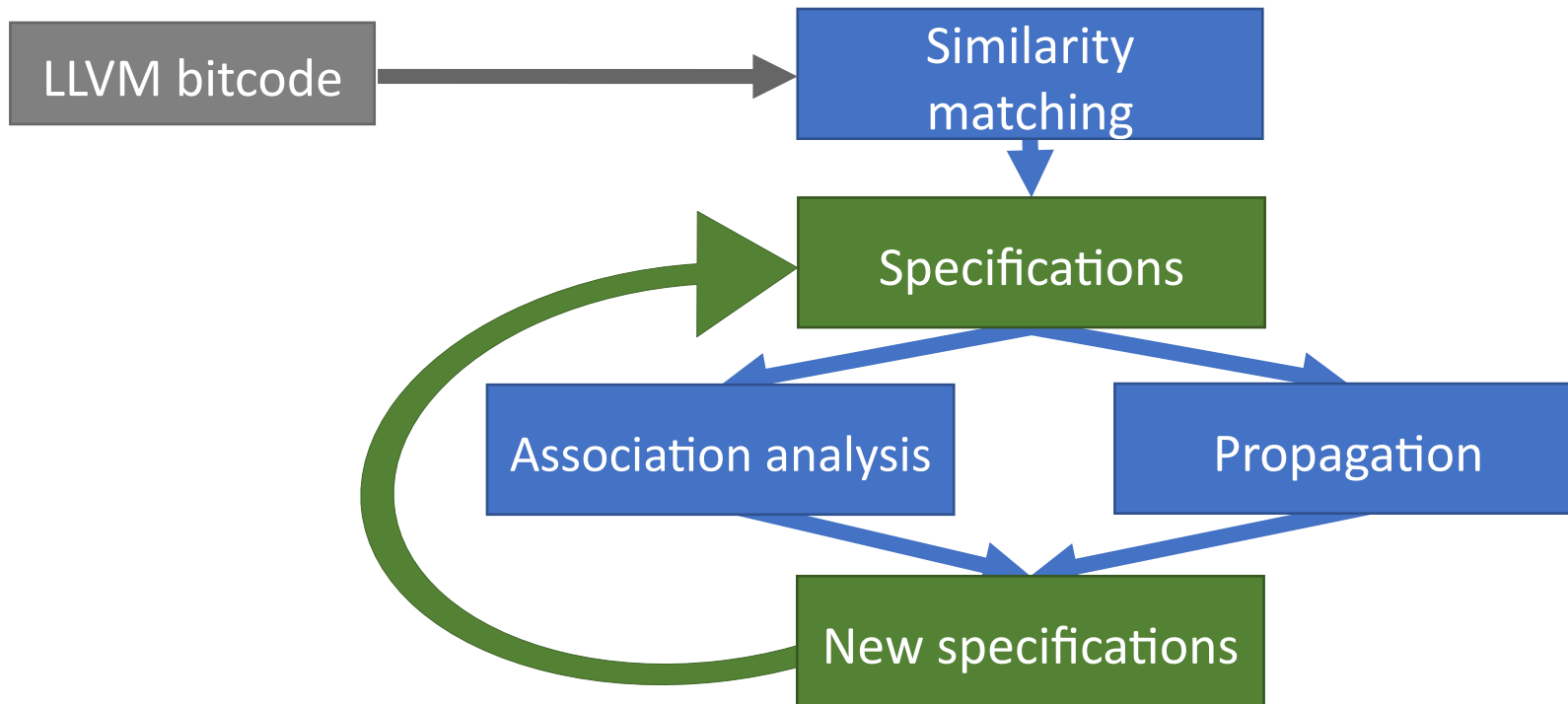
Pipeline



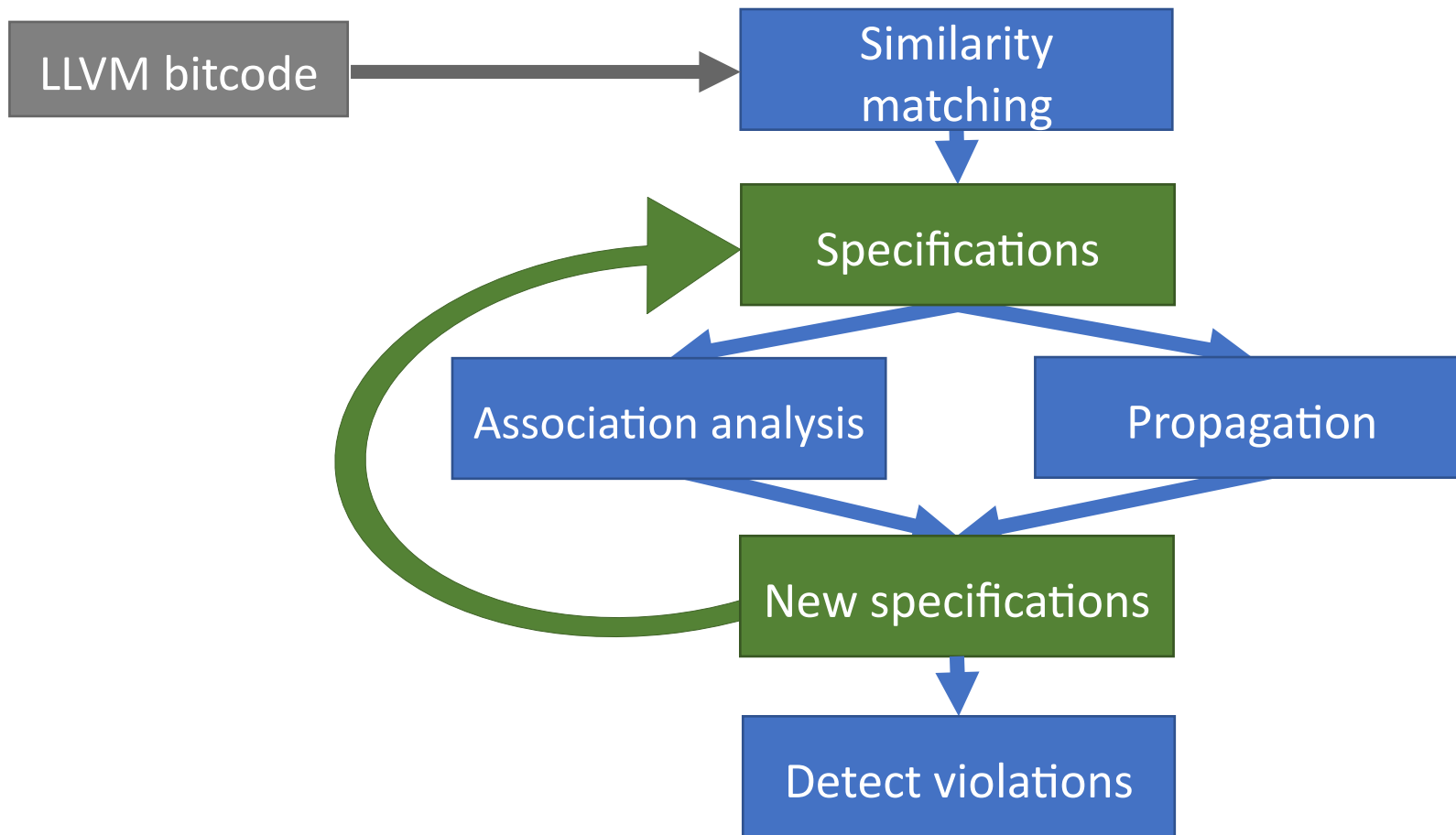
Pipeline



Pipeline



Pipeline



Bug Report Results

Project	Reports	MC	IC	PB	Unknown	FPR MC	FPR IC	FPR total
OpenSSL								
OpenSSH								
PHP								
zlib								
libpng								
freetype2								
libwebp								

Bug Report Results

Project	Reports	MC	IC	PB	Unknown	FPR MC	FPR IC	FPR total
OpenSSL	366							
OpenSSH	61							
PHP	355							
zlib	4							
libpng	1							
freetype2	25							
libwebp	15							

Bug Report Results

Project	Reports	MC	IC	PB	Unknown	FPR MC	FPR IC	FPR total
OpenSSL	366	238	84	0				
OpenSSH	61	49	11	0				
PHP	355	303	29	3				
zlib	4	4	0	0				
libpng	1	1	0	0				
freetype2	25	20	4	0				
libwebp	15	13	0	0				

Bug Report Results

Project	Reports	MC	IC	PB	Unknown	FPR MC	FPR IC	FPR total
OpenSSL	366	238	84	0	44	46.22%	39.29%	44.41%
OpenSSH	61	49	11	0	1	16.33%	54.55%	23.33%
PHP	355	303	29	3	20	26.40%	44.83%	28.01%
zlib	4	4	0	0	0	25.00%	-	25.00%
libpng	1	1	0	0	0	0.00%	-	0.00%
freetype2	25	20	4	0	1	0.00%	50.00%	8.33%
libwebp	15	13	0	0	2	18.18%	-	18.18%

Inference Performance

Project	EESI		ESSS		SLOC
	Time	Memory	Time	Memory	
OpenSSL					
OpenSSH					
PHP					
zlib					
libpng					
freetype2					
libwebp					

Time in seconds, memory in GiB

Inference Performance

Project	EESI		ESSS		SLOC
	Time	Memory	Time	Memory	
OpenSSL	176.79	20.46			
OpenSSH	81.70	4.02			
PHP	-	-			
zlib	4.09	0.25			
libpng	21.44	4.50			
freetype2	91.66	10.56			
libwebp	12.65	2.20			

Time in seconds, memory in GiB

Inference Performance

Project	EESI		ESSS		SLOC
	Time	Memory	Time	Memory	
OpenSSL	176.79	20.46	3.22	0.75	542K
OpenSSH	81.70	4.02	1.44	0.22	120K
PHP	-	-	14.50	1.85	1.46M
zlib	4.09	0.25	0.38	0.05	30K
libpng	21.44	4.50	0.49	0.09	63K
freetype2	91.66	10.56	0.94	0.19	141K
libwebp	12.65	2.20	0.40	0.14	75K

Time in seconds, memory in GiB

Results

827 inspected bugs in 7 projects

541 true positives

46 fixed in 3 projects

Results

827 inspected bugs in 7 projects

541 true positives

46 fixed in 3 projects

OpenSSL
Cryptography and SSL/TLS Toolkit

16 fixes



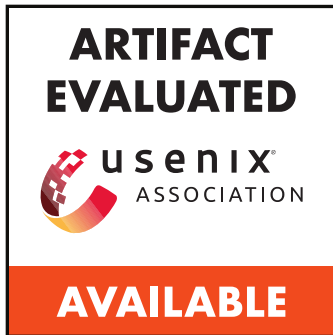
1 fix



29 fixes
(1 security bug)

And More!

- More experiments and details in paper
- Open-Source on GitHub: <https://github.com/csl-ugent/ESSS>
- Artifact evaluated



References

- [1] Daniel DeFreez, Haaken Martinson Baldwin, Cindy Rubio-González, and Aditya V Thakur. Effective Error-Specification Inference via Domain-Knowledge Expansion. In Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering
- [2] Yuan Kang, Baishakhi Ray, and Suman Jana. APEx: Automated Inference of Error Specifications for C APIs. In Proceedings of the 31st IEEE/ACM International Conference on Automated Software Engineering
- [3] Chi Li, Min Zhou, Zuxing Gu, Ming Gu, and Hongyu Zhang. Ares: Inferring Error Specifications through Static Analysis. In 2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE)