

# **Unqomp**: Synthesizing Uncomputation in Quantum Circuits

**Anouk Paradis**, Benjamin Bischel, Samuel Steffen, Martin Vechev

Contact us via [firstname.lastname@inf.ethz.ch](mailto:firstname.lastname@inf.ethz.ch)

# Uncomputation - Intuition

```
def library(x) {  
    a = 0;           // initialize a  
    auxiliary( a ); // modify a  
    x = a ^ x       // use a  
  
}  
  
def root() {  
    x = 0;          // initialize x  
    library( x );  // modify x  
    ...            // use x  
  
}
```

# Uncomputation - Intuition

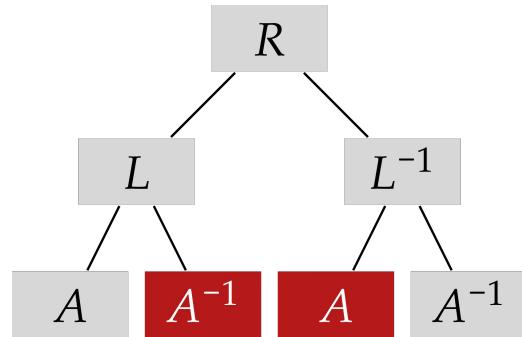
```
def library(x) {  
    a = 0;          // initialize a  
    auxiliary( a ); // modify a  
    x = a ^ x      // use a  
    auxiliary-1( a ); // reset a to 0  
}
```

```
def root() {  
    x = 0;          // initialize x  
    library( x );   // modify x  
    ...            // use x  
    library-1( x ); // reset x to 0  
}
```

# Uncomputation - Intuition

```
def library(x) {  
    a = 0;          // initialize a  
    auxiliary( a ); // modify a  
    x = a ^ x      // use a  
    auxiliary-1( a ); // reset a to 0  
}
```

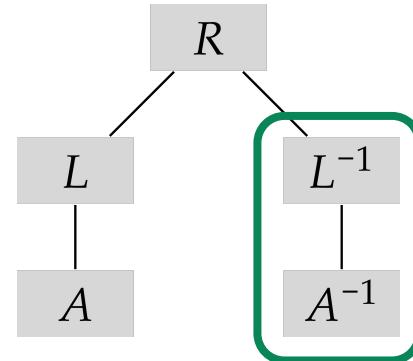
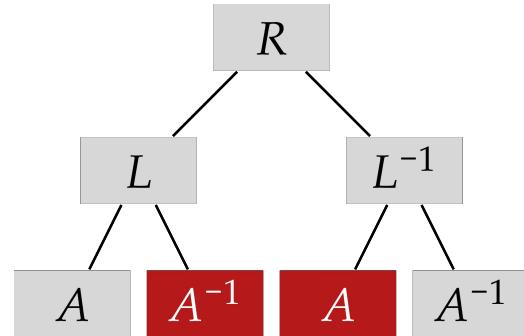
```
def root() {  
    x = 0;          // initialize x  
    library( x );  // modify x  
    ...            // use x  
    library-1( x ); // reset x to 0  
}
```



# Uncomputation - Intuition

```
def library(x) {  
    a = 0;          // initialize a  
    auxiliary( a ); // modify a  
    x = a ^ x      // use a  
    auxiliary-1( a ); // reset a to 0  
}
```

```
def root() {  
    x = 0;          // initialize x  
    library( x );  // modify x  
    ...            // use x  
    library-1( x ); // reset x to 0  
}
```



# Uncomputation - In practice



Manual uncomputation  
(language constructs)

Qiskit [2019]  
Q# [RWDSL2018]  
Quipper [PLDI13]

Inefficient or  
non-modular

# Uncomputation - In practice



Manual uncomputation  
(language constructs)

Qiskit [2019]  
Q# [RWDSL2018]  
Quipper [PLDI13]

Inefficient or  
non-modular

Only classical

Classical uncomputation

Quipper [PLDI13]  
Revs [CAV17]



# Uncomputation - In practice



Manual uncomputation  
(language constructs)

Qiskit [2019]  
Q# [RWDSL2018]  
Quipper [PLDI13]

Inefficient or  
non-modular

Only classical

Classical uncomputation

Quipper [PLDI13]  
Revs [CAV17]



Type system

Silq [PLDI20]

No compilation

# Unqomp - Overview

```
def library(x) {  
    a = 0;  
    auxiliary( a );  
    x = a ^ x  
}
```

```
def root(y) {  
    x = 0;  
    library( x );  
    ...  
}
```

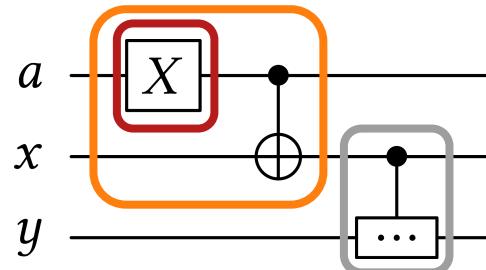
# Unqomp - Overview

```
def library(x) {  
    a = 0;  
    auxiliary( a );  
    x = a ^ x  
}
```

```
def root(y) {  
    x = 0;  
    library( x );  
    ...
```



Quantum Compiler



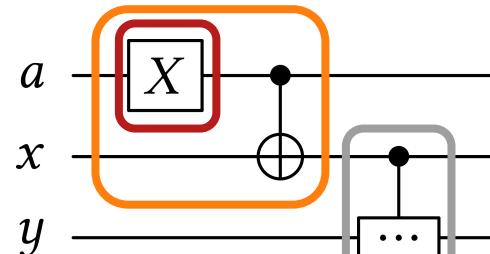
# Unqomp - Overview

```
def library(x) {  
    a = 0;  
    auxiliary( a );  
    x = a ^ x  
}
```

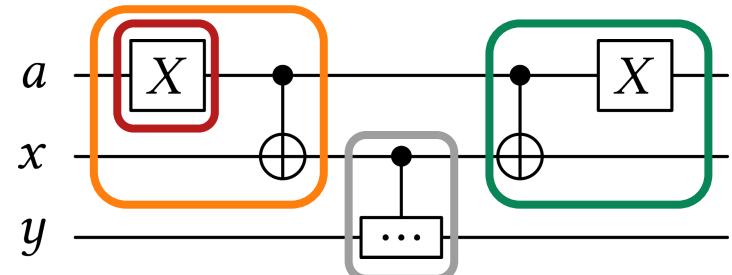
```
def root(y) {  
    x = 0;  
    library( x );  
    ...  
}
```



Quantum Compiler



Unqomp



# Unqomp - Overview

```
def library(x) {  
    a = 0;  
    auxiliary( a );  
    x = a ^ x  
}
```

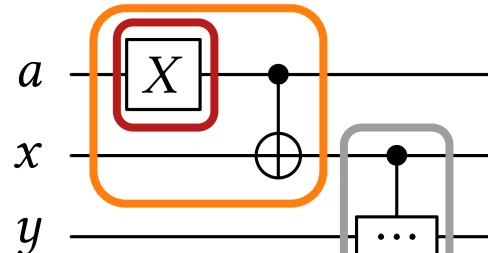
```
def root(y) {  
    x = 0;  
    library( x );  
    ...  
}
```

```
def library(x) {  
    a = 0;  
    aux...  
    x = ...  
}
```

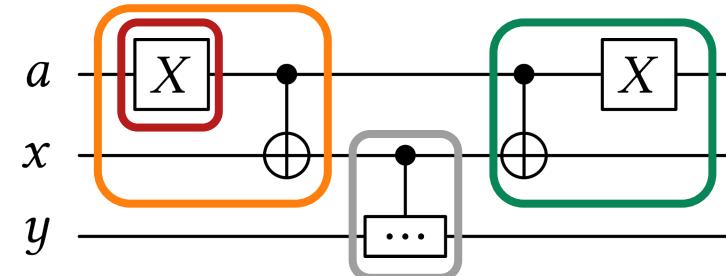
```
def root(y) {  
    ...  
    +  uncomputation  
    auxiliary( x );  
    ...  
}
```



Quantum Compiler



Unqomp



# Unqomp - Overview

```
def library(x) {  
    a = 0;  
    auxiliary( a );  
    x = a ^ x  
}
```

```
def root(y) {  
    x = 0;  
    library( x );  
    ...  
}
```

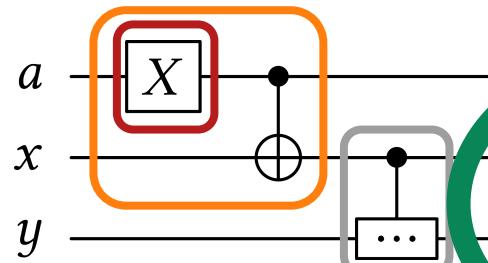
```
def library(x) {  
    a = 0;  
    aux...  
    x = ...  
}
```

```
def root(y) {  
    ...  
    auxiliary( x );  
    ...  
}
```

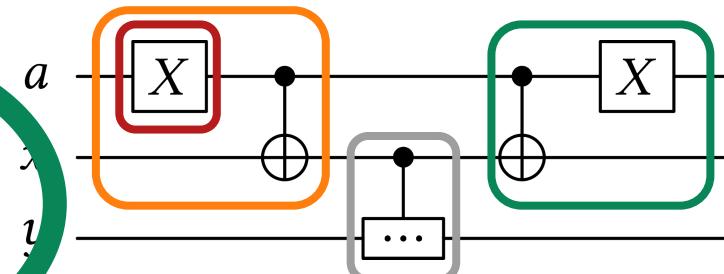
+      uncomputation



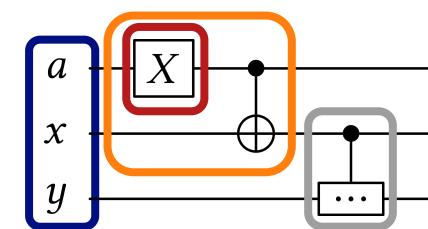
Quantum Compiler



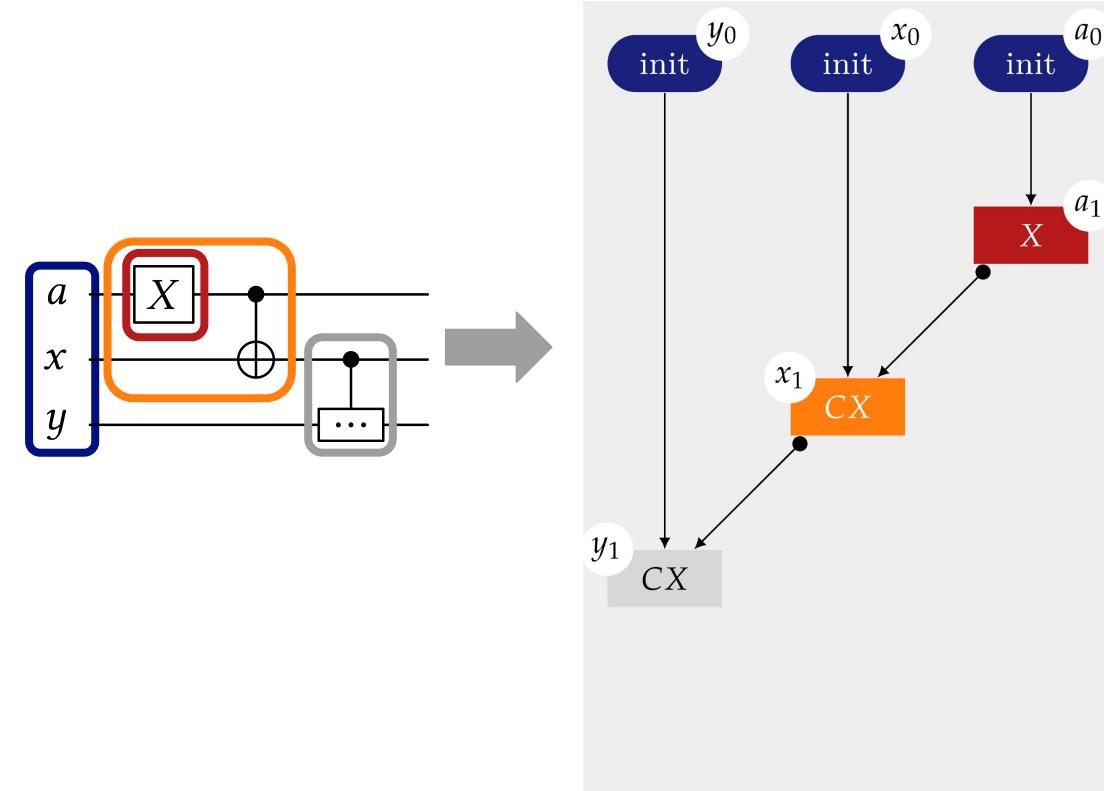
Unqomp



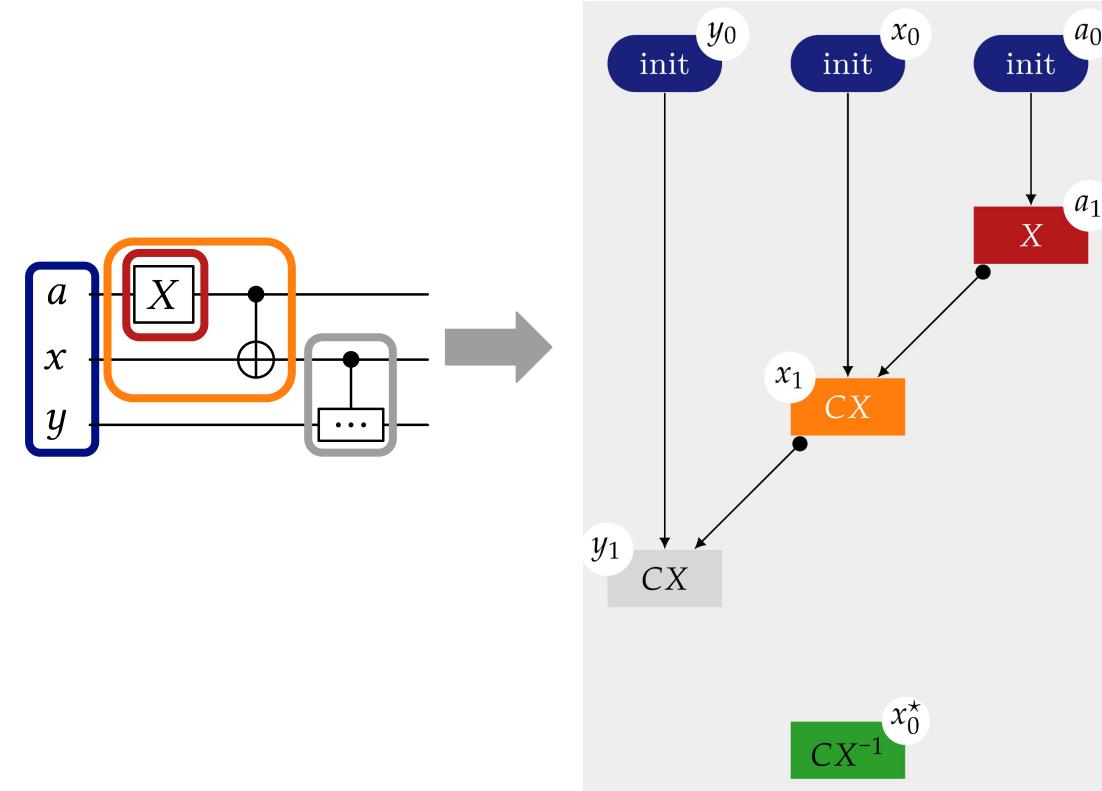
# Unqomp - Automatic uncomputation for circuits



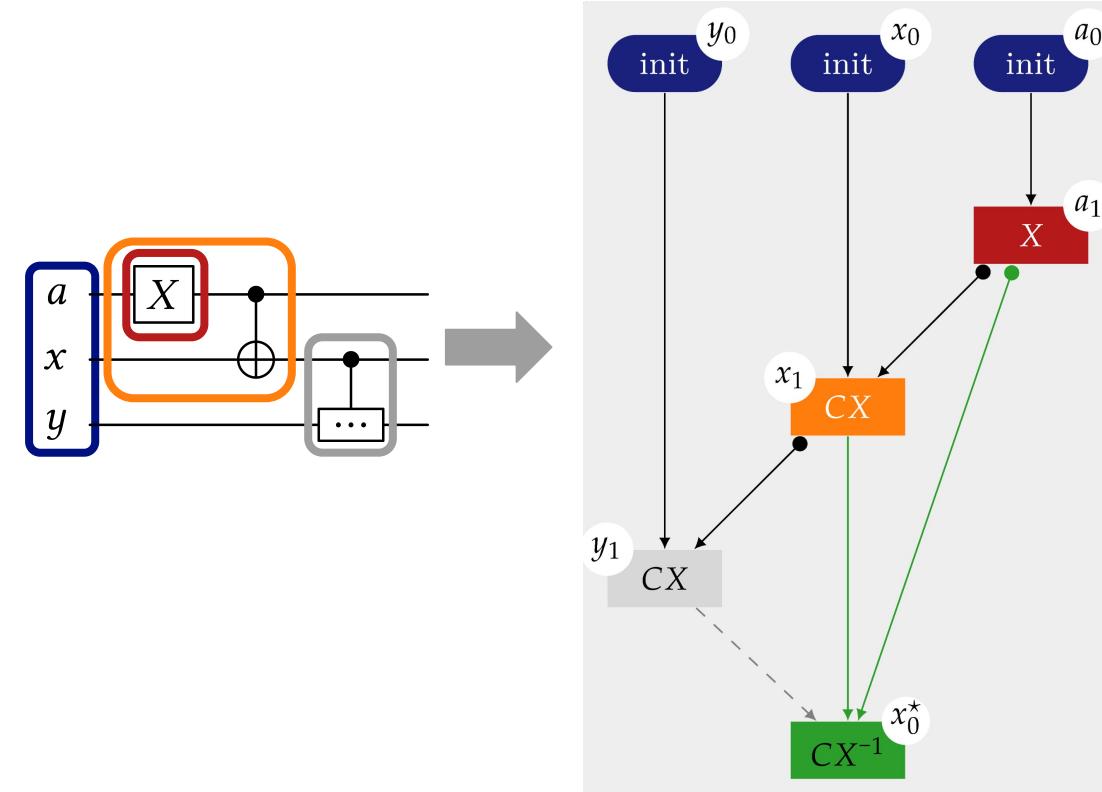
# Unqomp - Automatic uncomputation for circuits



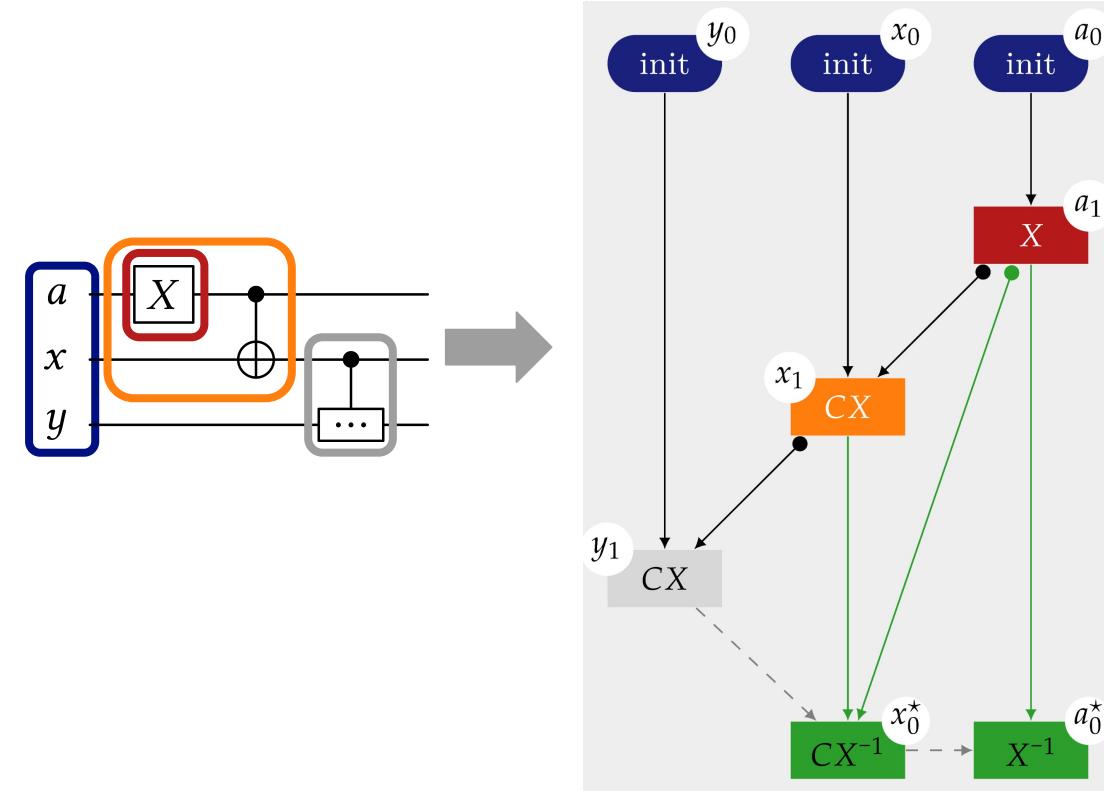
# Unqomp - Automatic uncomputation for circuits



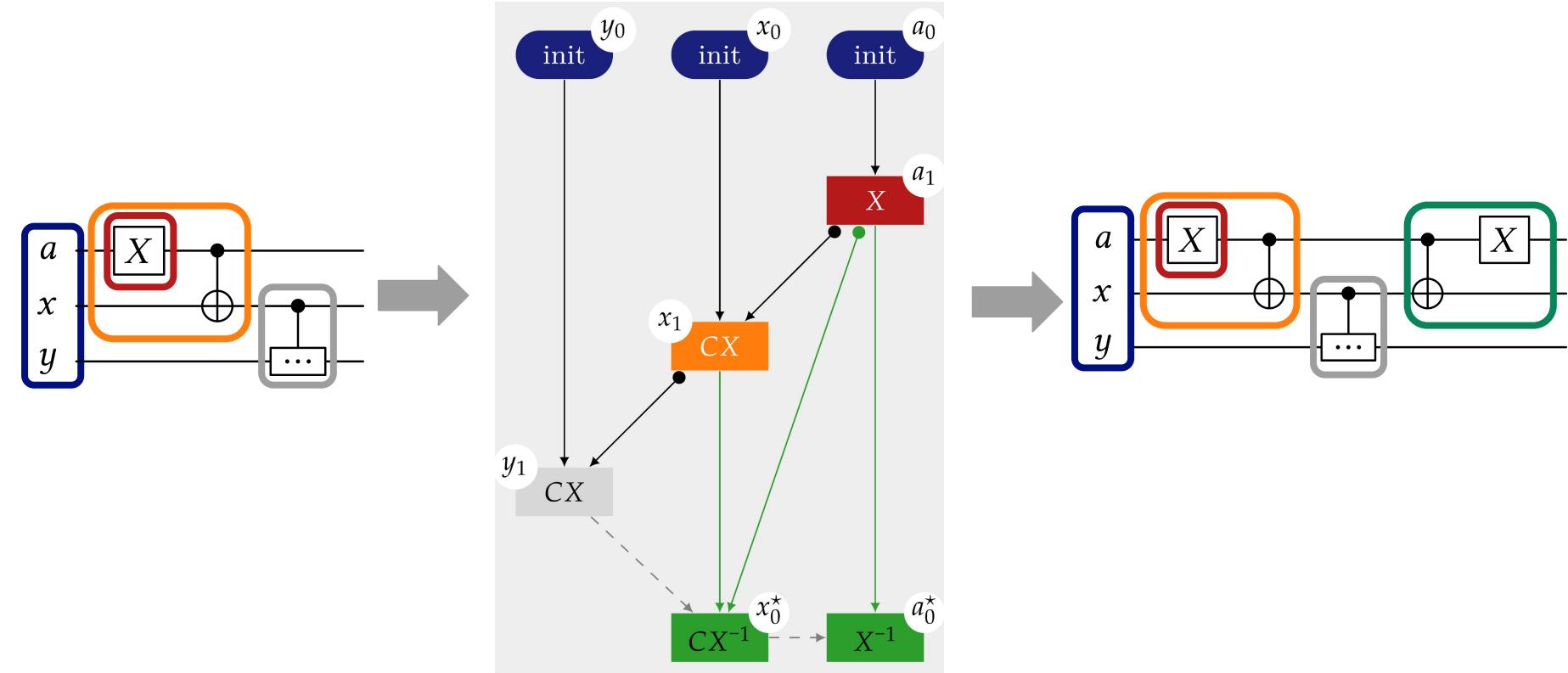
# Unqomp - Automatic uncomputation for circuits



# Unqomp - Automatic uncomputation for circuits



# Unqomp - Automatic uncomputation for circuits



# Evaluation

Code Complexity?

# Evaluation

Code Complexity?

-19% program length

Reported 3 Qiskit bugs

Simpler code

# Evaluation

Code Complexity?

Circuit Efficiency?

-19% program length

Reported 3 Qiskit bugs

Simpler code

# Evaluation

## Code Complexity?

-19% program length

Reported 3 Qiskit bugs

## Circuit Efficiency?

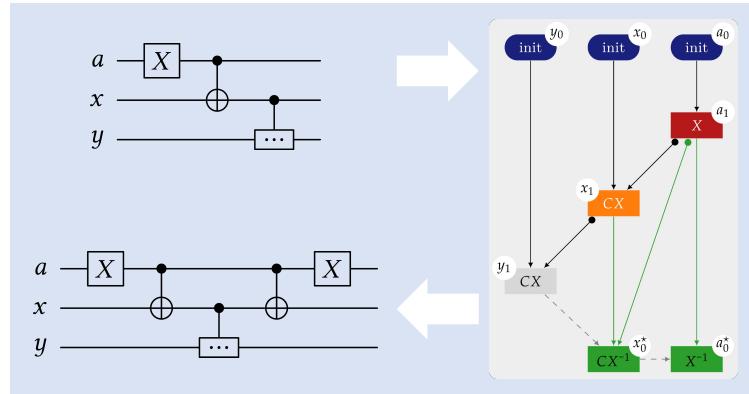
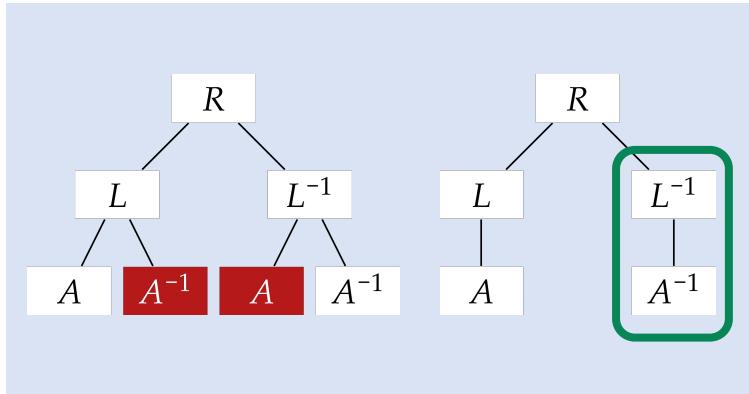
-71% gates  
-19% qubits

Compared to Qiskit  
and Quipper

Simpler code

Smaller circuits

# Summary



Qiskit implementation available  
<https://github.com/eth-sri/unqomp>

Simpler code

Smaller circuits