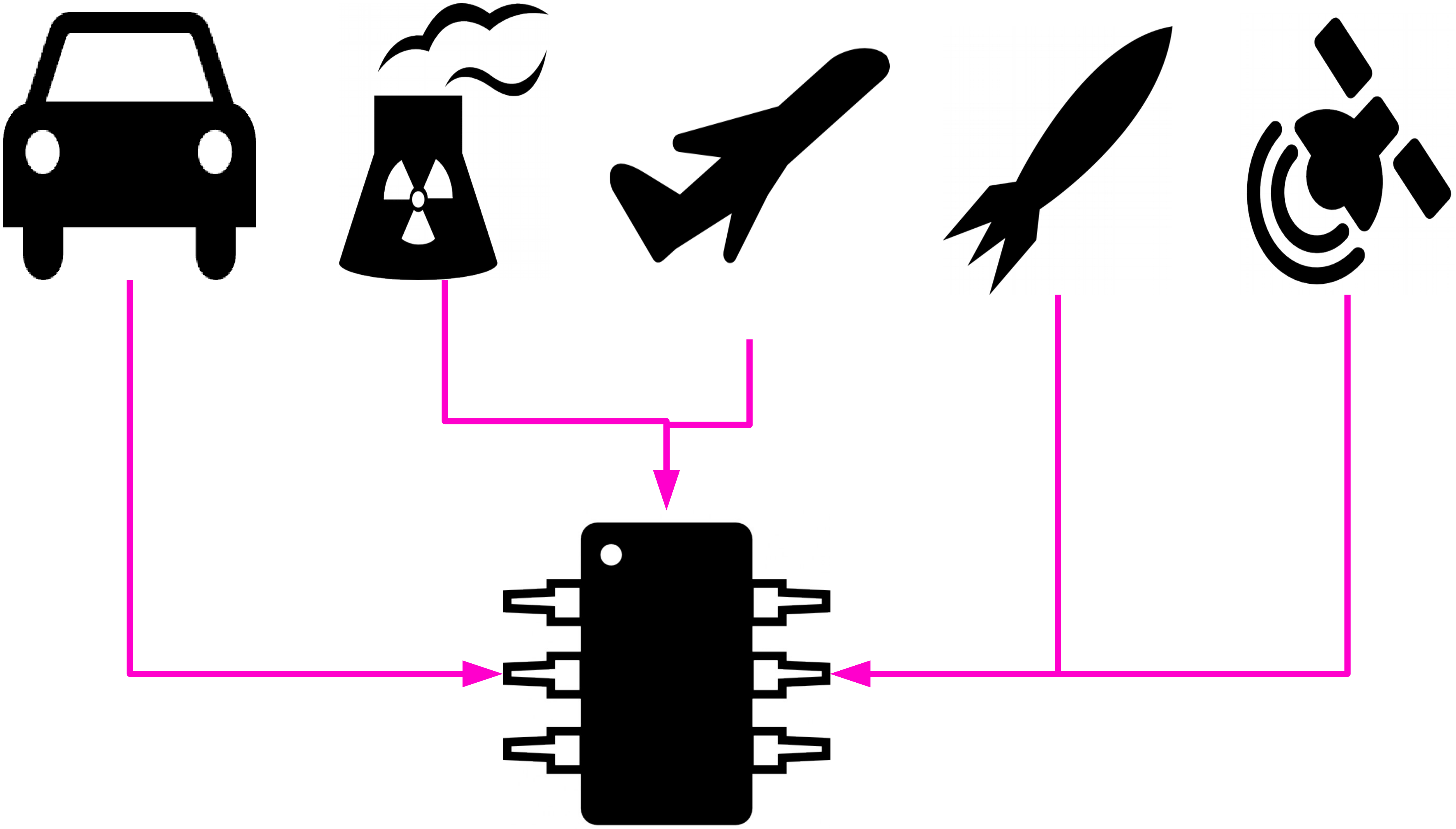
A Generalized Reduction of Ordered Binary Decision Diagram (GroBdd)

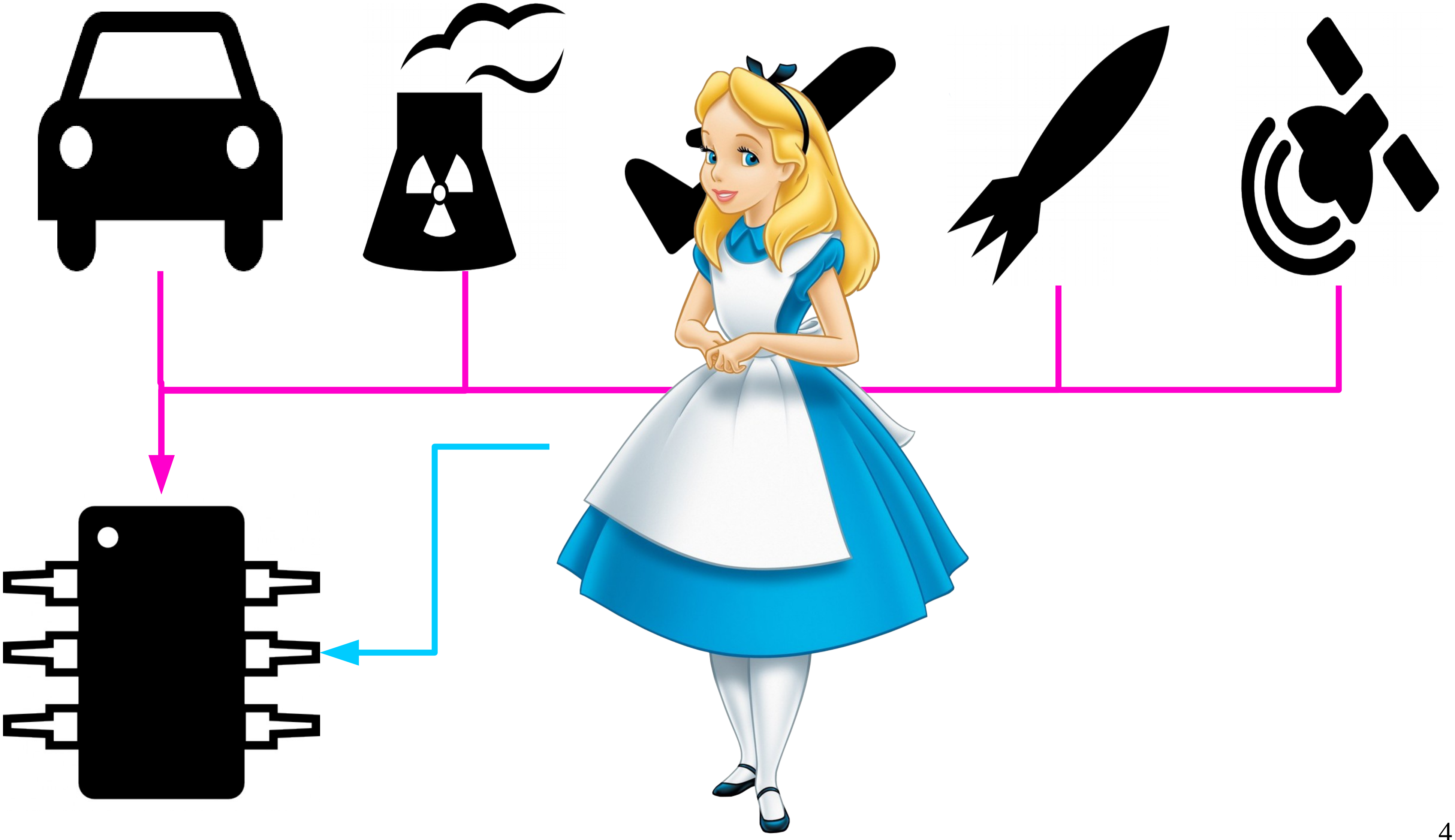
Joan Thibault

Most critical systems ...

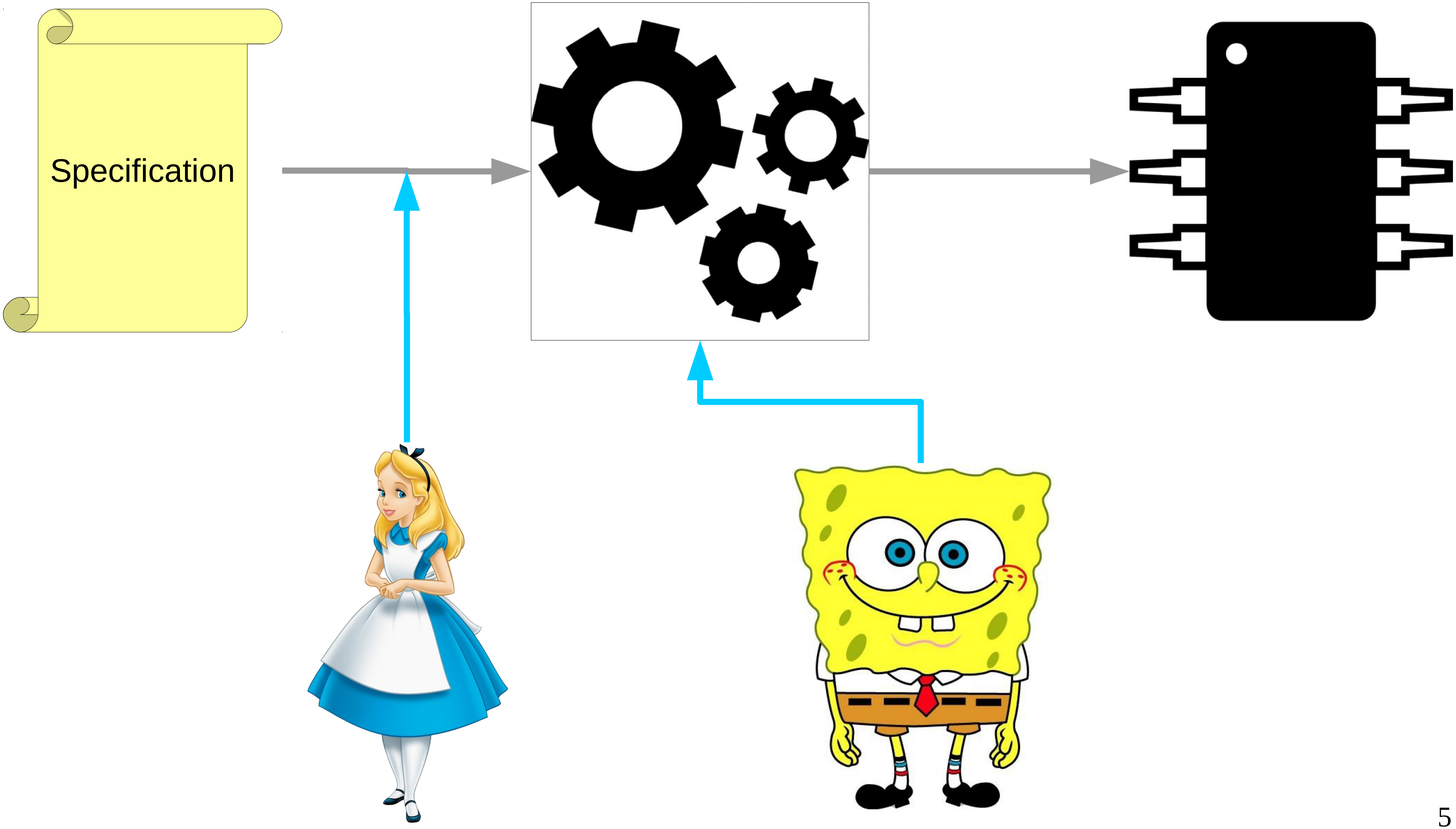
… relies on chips ...



… designed by Alice ...

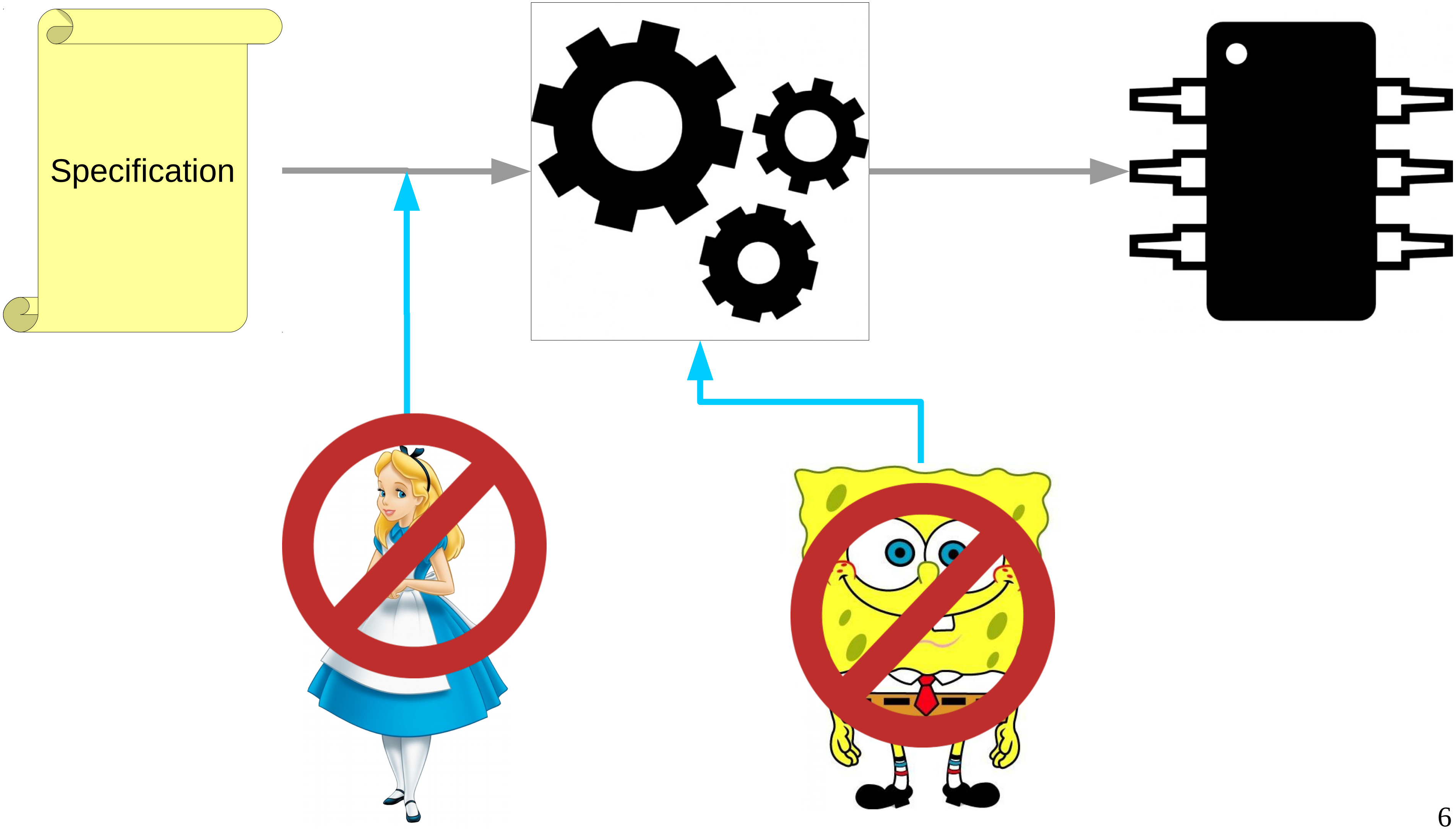


… using Bob's software.

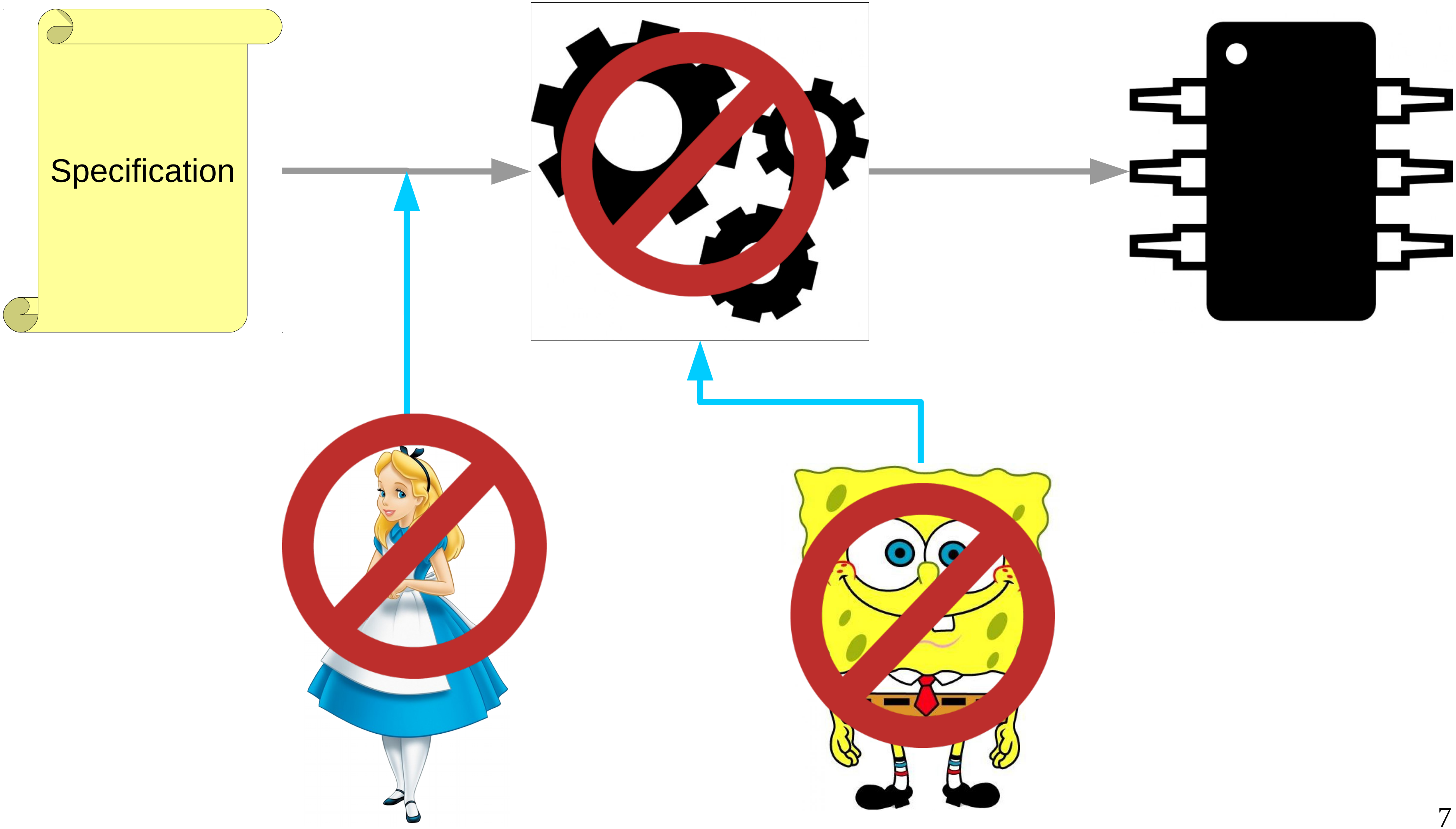


Specification

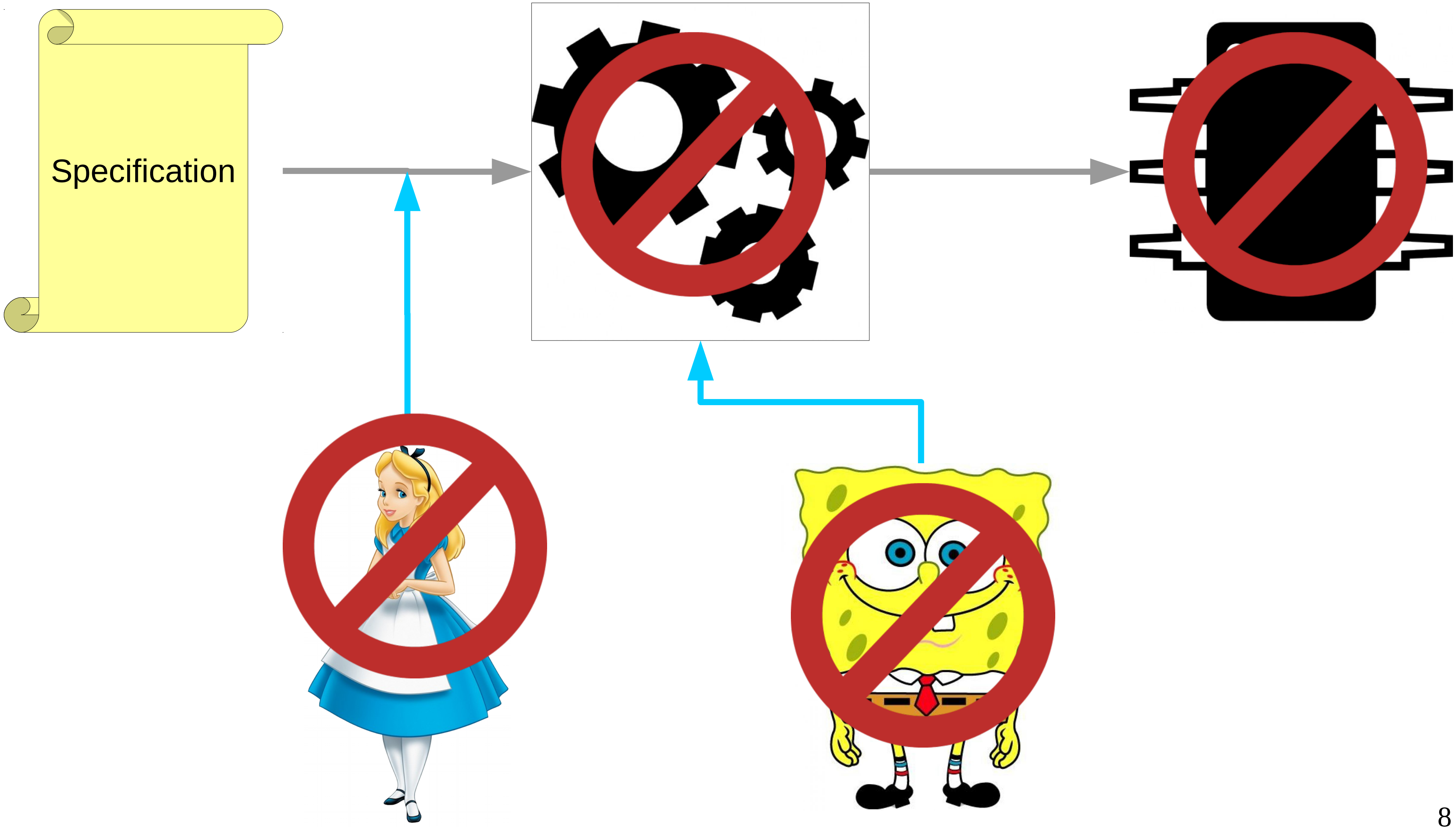
Lack of reliability



Lack of reliability

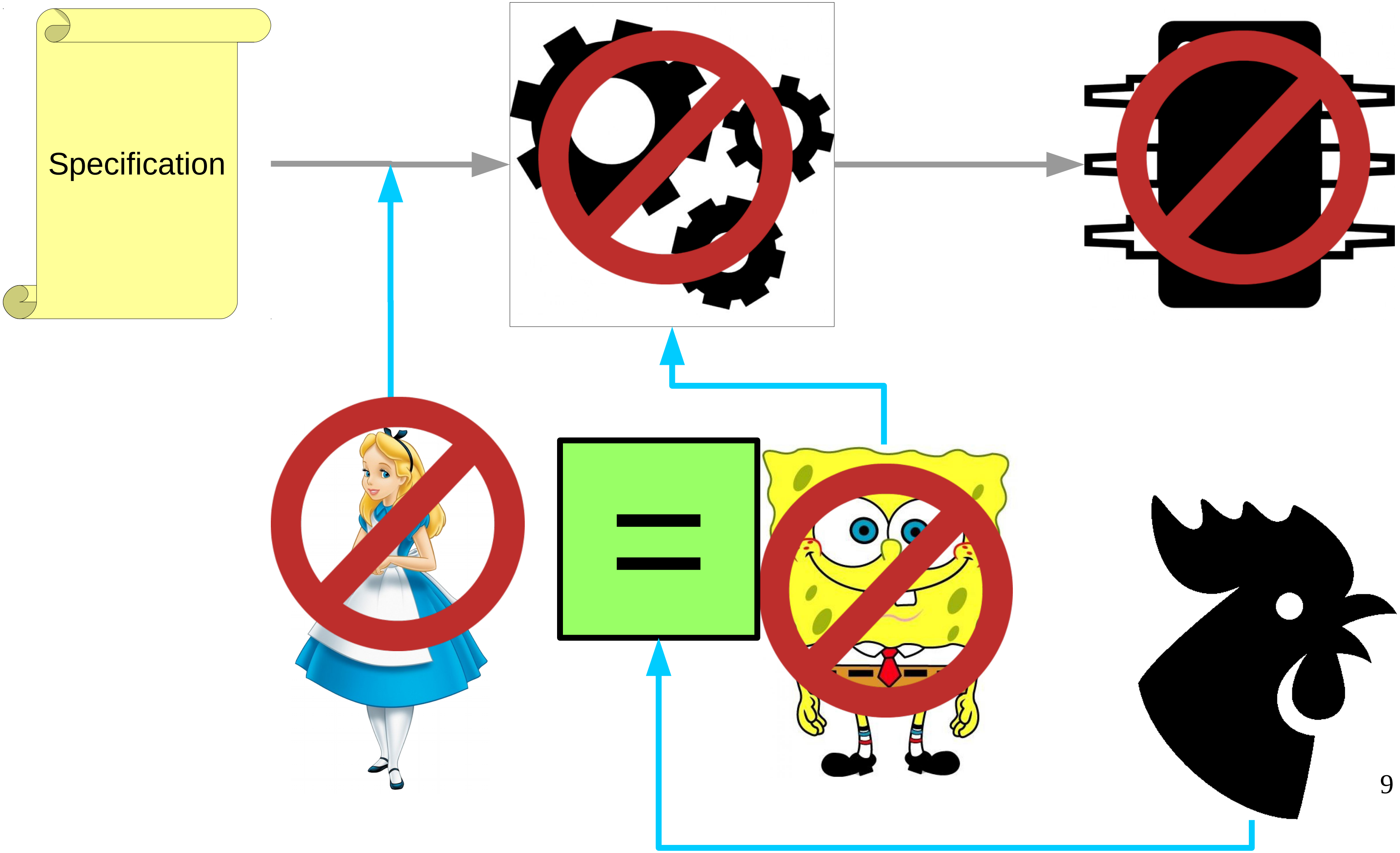


Lack of reliability

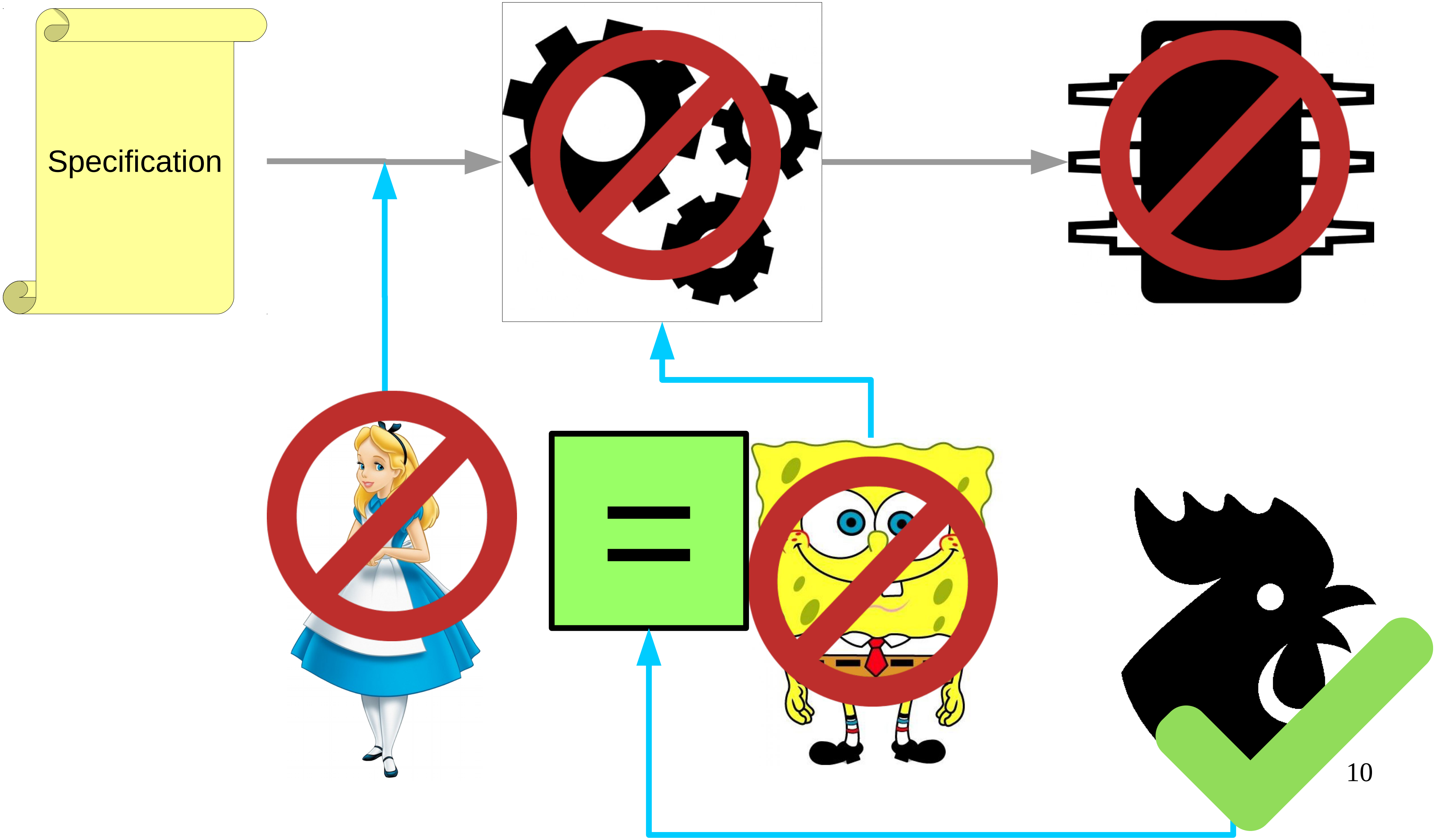


Specification

Circuit Equivalence Checking

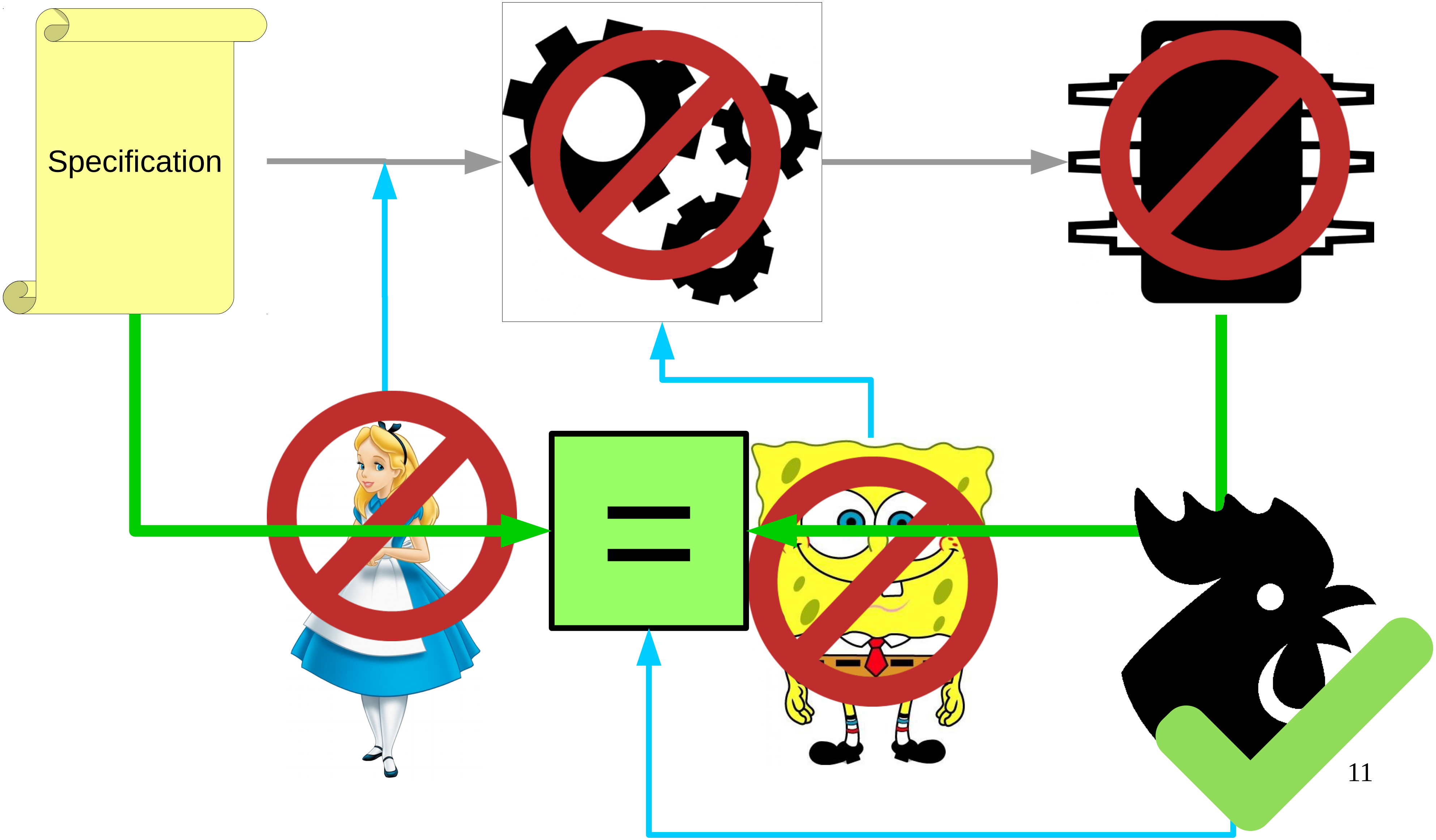


Circuit Equivalence Checking

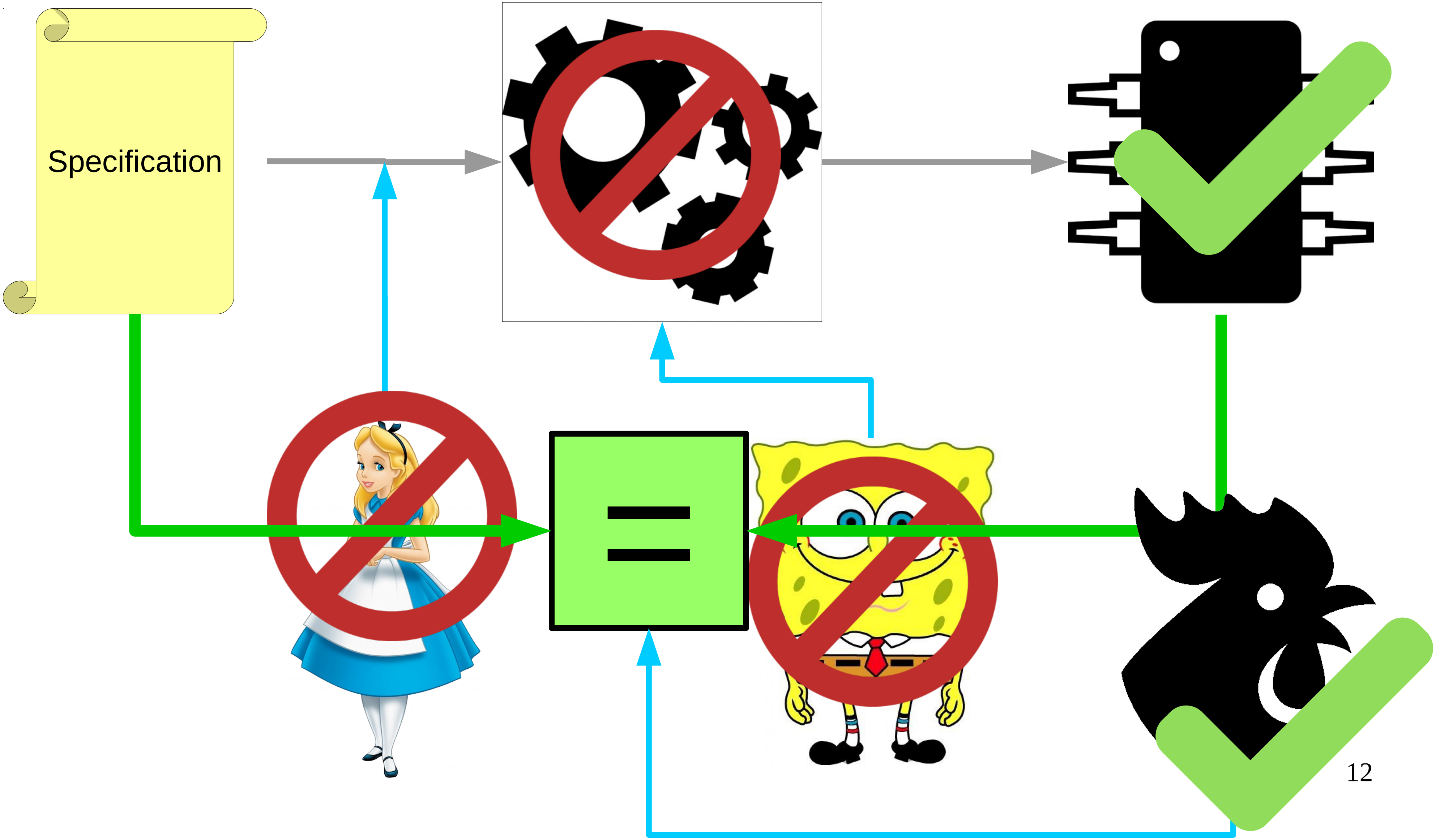


Specification

Circuit Equivalence Checking



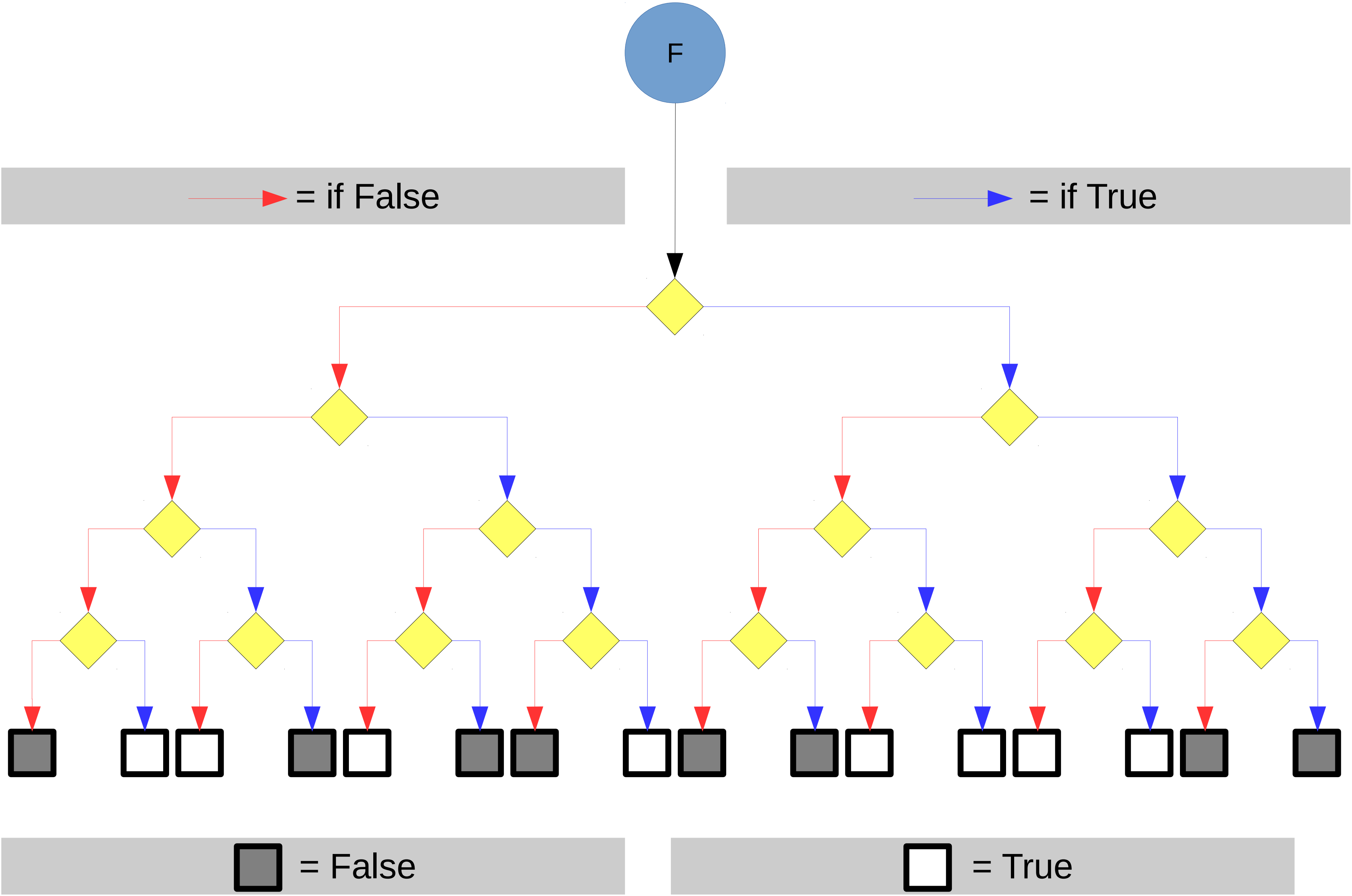
Design matches specification



Reduced Ordered Binary Decision Diagram (ROBDD)

- Applications
  - Computer Aided Design (e.g. equivalence checking)
  - Knowledge Representation (e.g. Artificial Intelligence)
  - Combinatorial Problems (e.g. CNF SAT problem)
- What are required operation ?
  - Compact representation
  - Operations (e.g. composing, concatening, evaluation)
  - Operators (e.g. AND, XOR, ITE, NOT)
  - Reductions (e.g. quantification, partial evaluation, SAT)

Shanon’s Binary Decision Tree



F

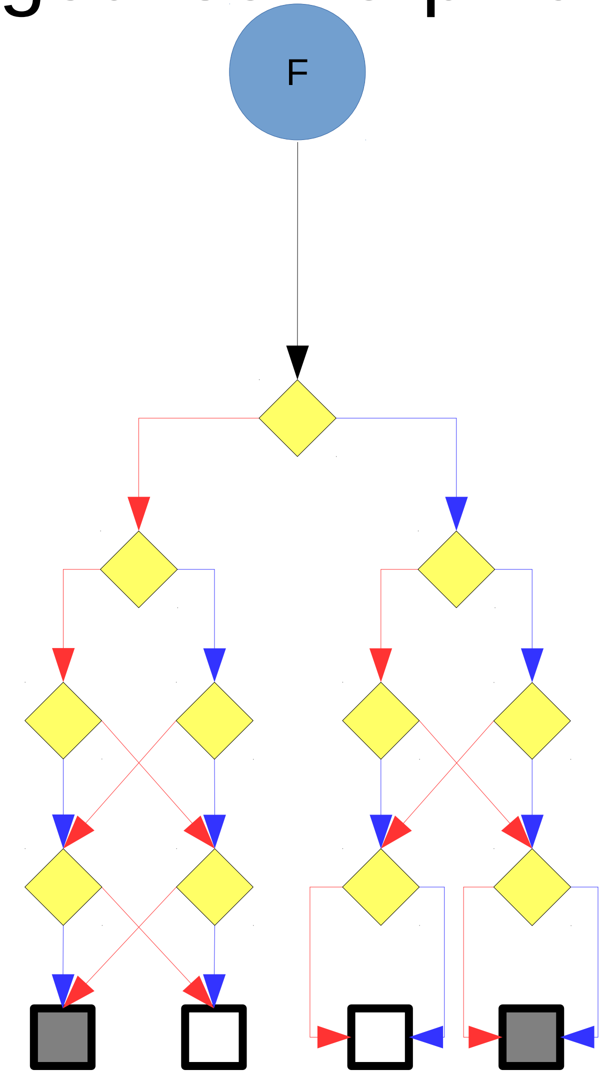
→ = if False

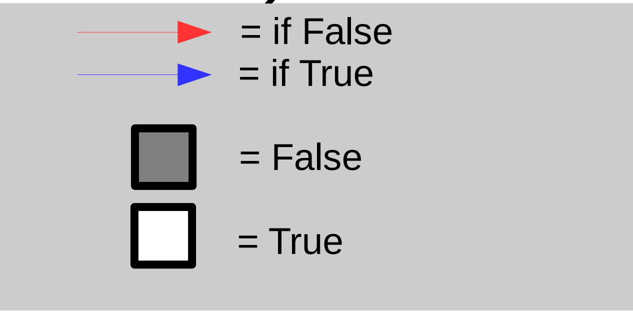
→ = if True

▨ = False

□ = True

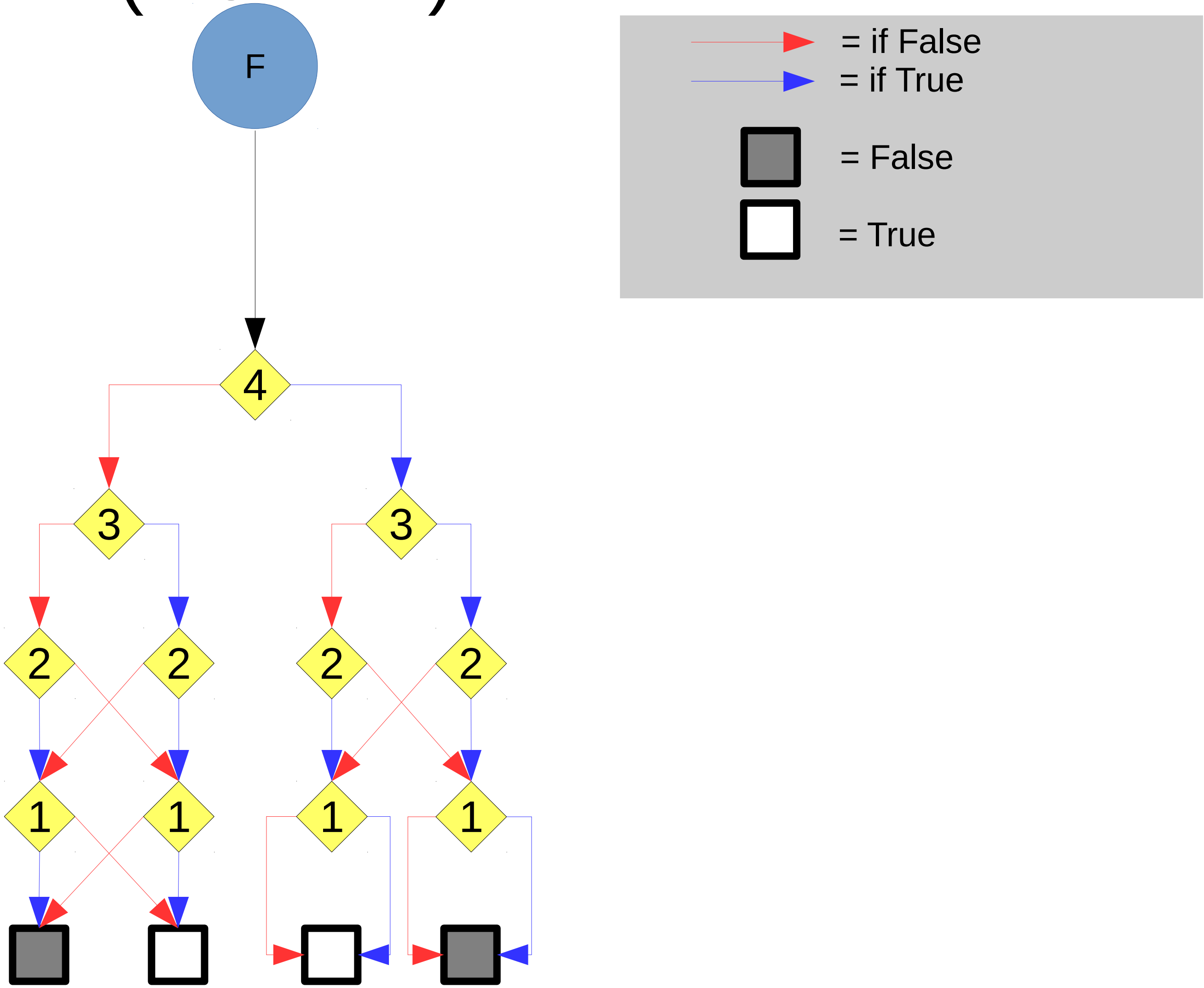
Shannon’s Decision Diagram
(We merged isomorphic sub-trees)





For aesthetic reasons, we do not merge all terminals

Reduced Ordered Binary Decision Diagram (ROBDD)

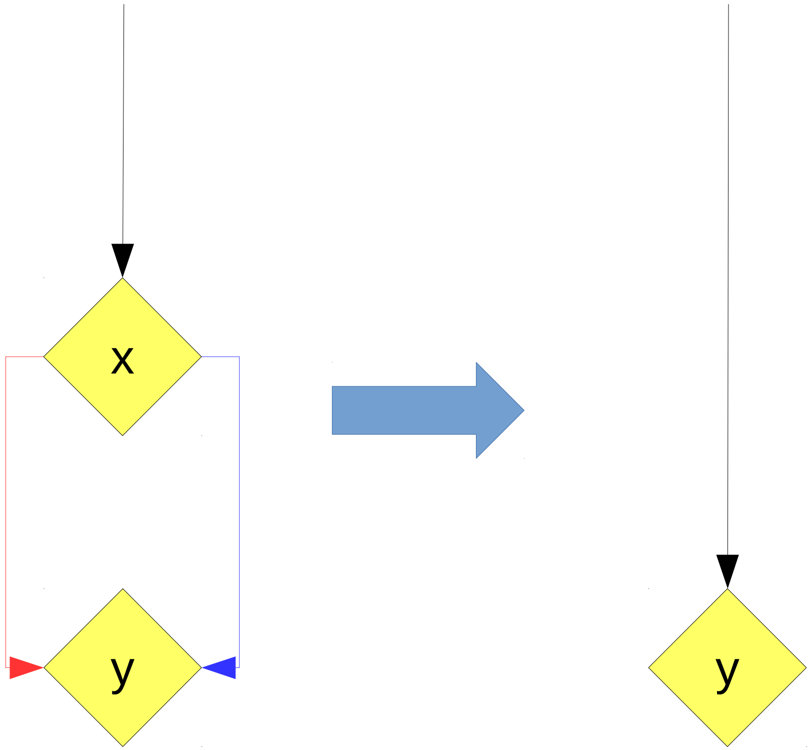


For aesthetic reasons, we do not merge all terminals

ROBDD : reduction rule

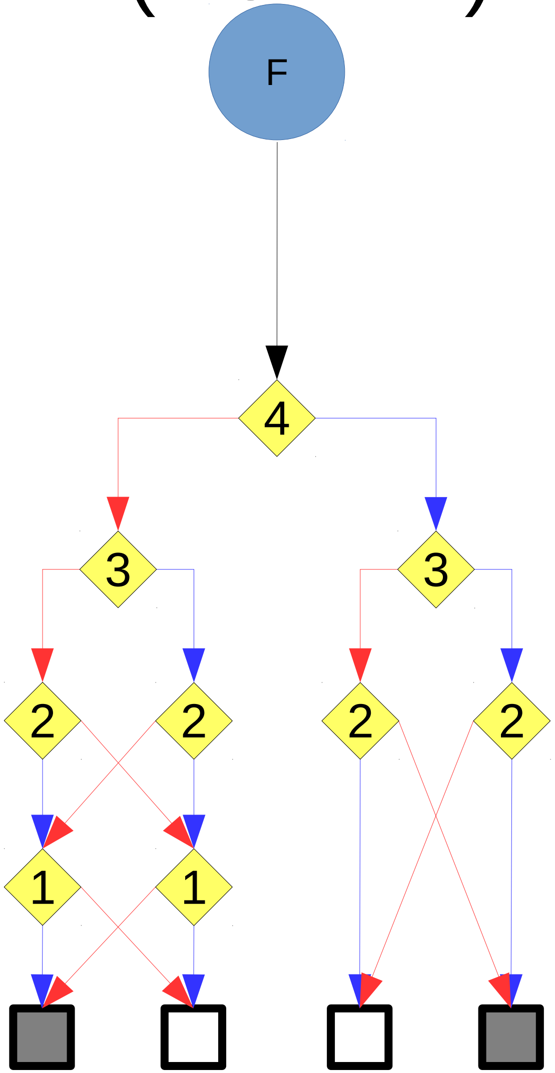
→ = if False
→ = if True

■ = False
□ = True



For aesthetic reasons, we do not merge all terminals

Reduced Ordered Binary Decision Diagram (ROBDD)

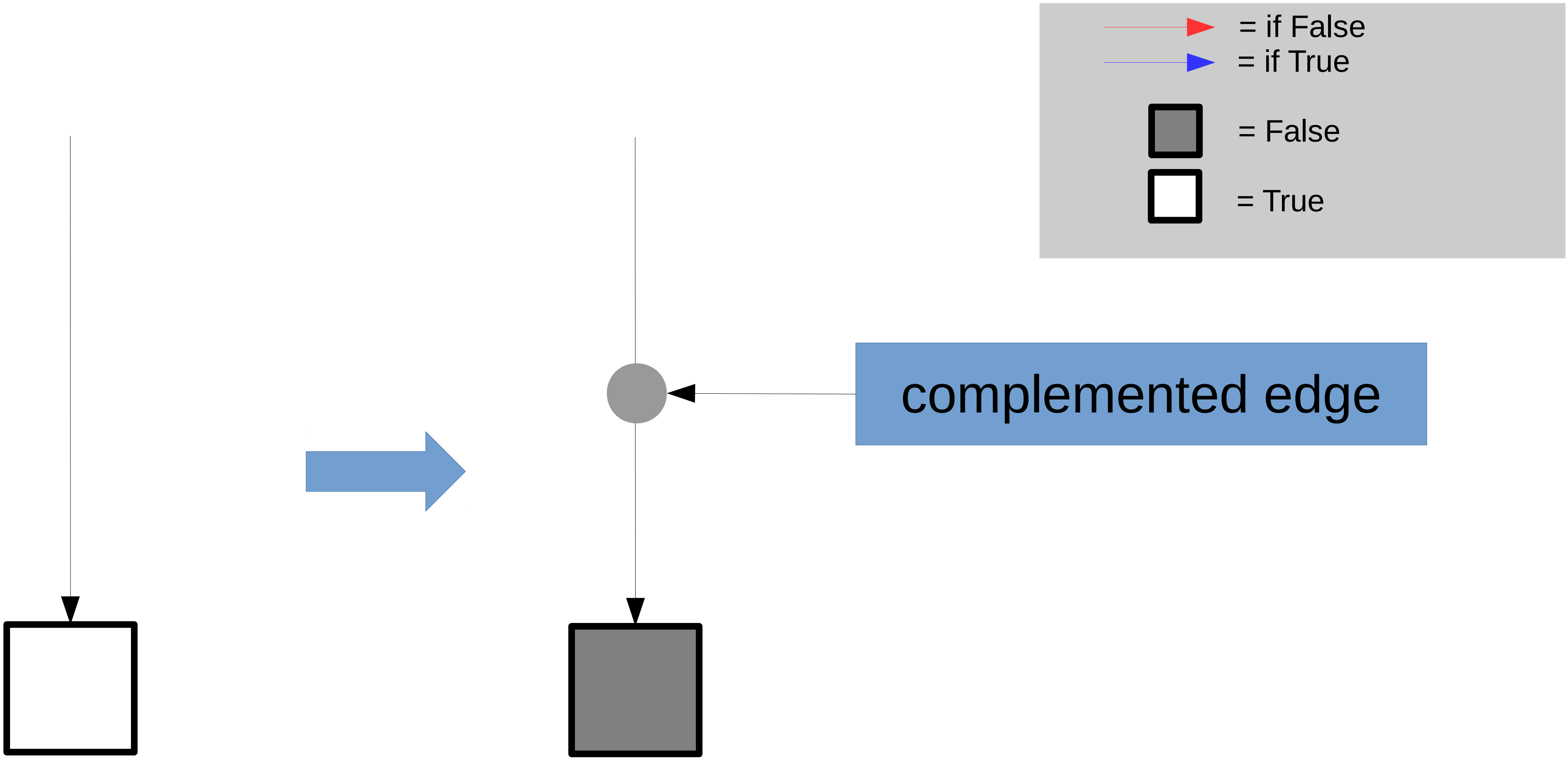


→ = if False
→ = if True

■ = False

□ = True

“output negation” : reduction rule (N1)



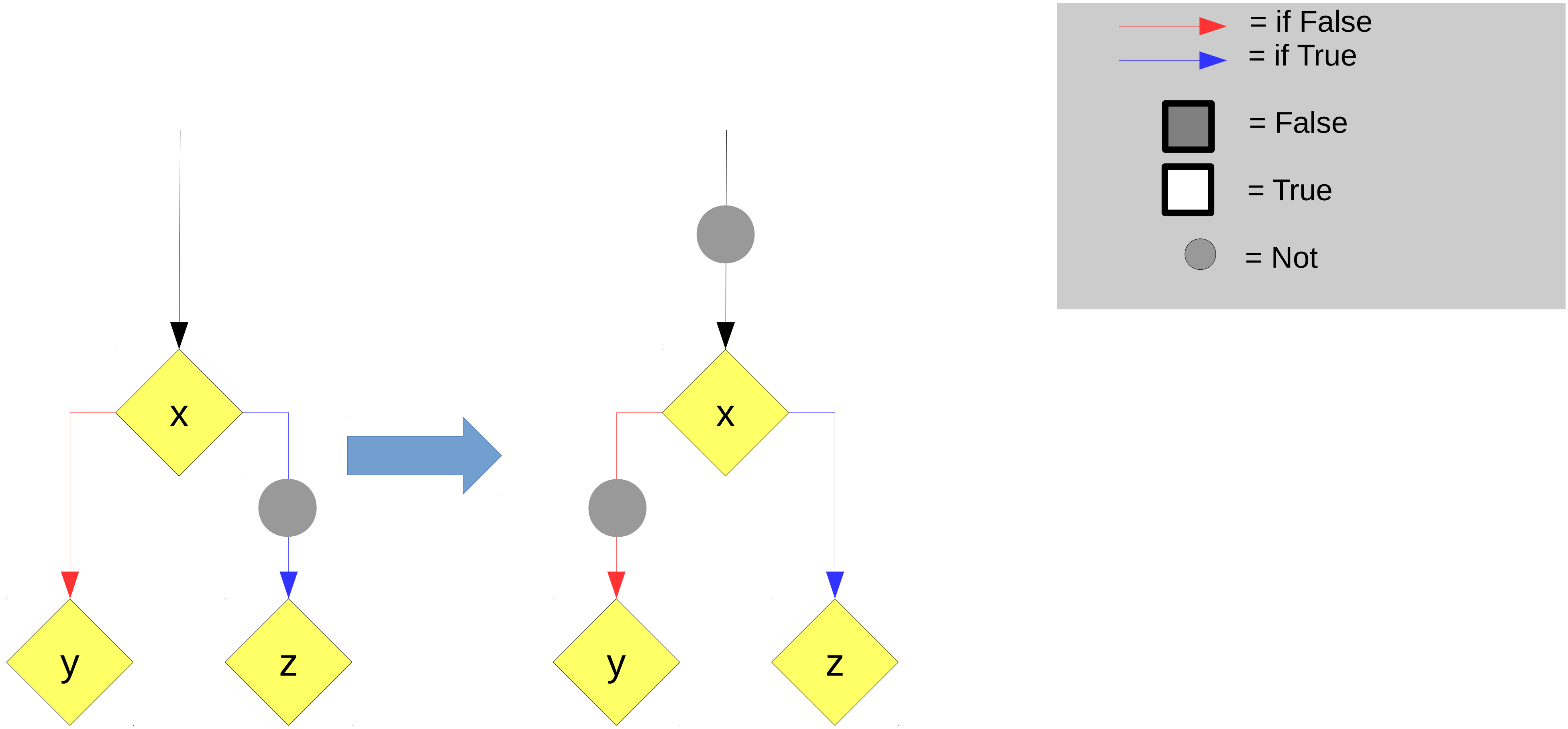
→ = if False
→ = if True

■ = False
□ = True

complemented edge

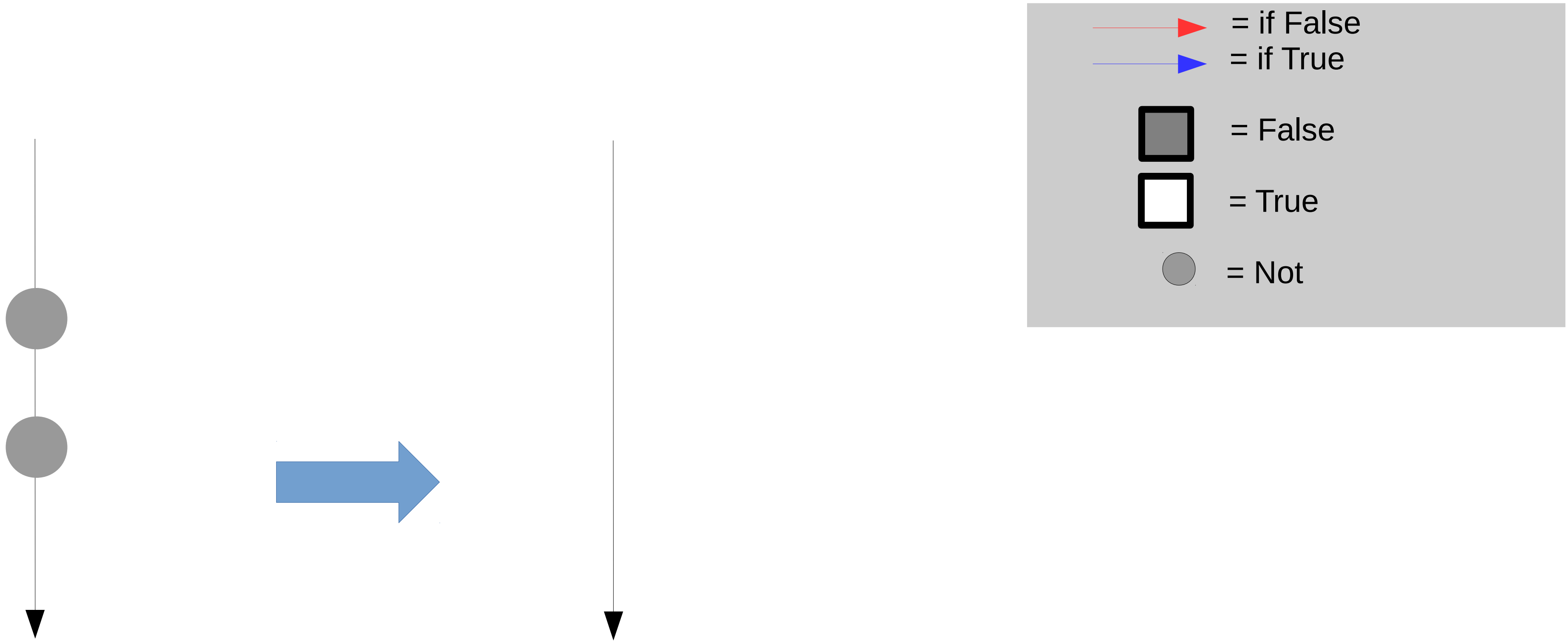
For aesthetic reasons, we do not merge all terminals

“output negation” : reduction rule (N2)



For aesthetic reasons, we do not merge all terminals

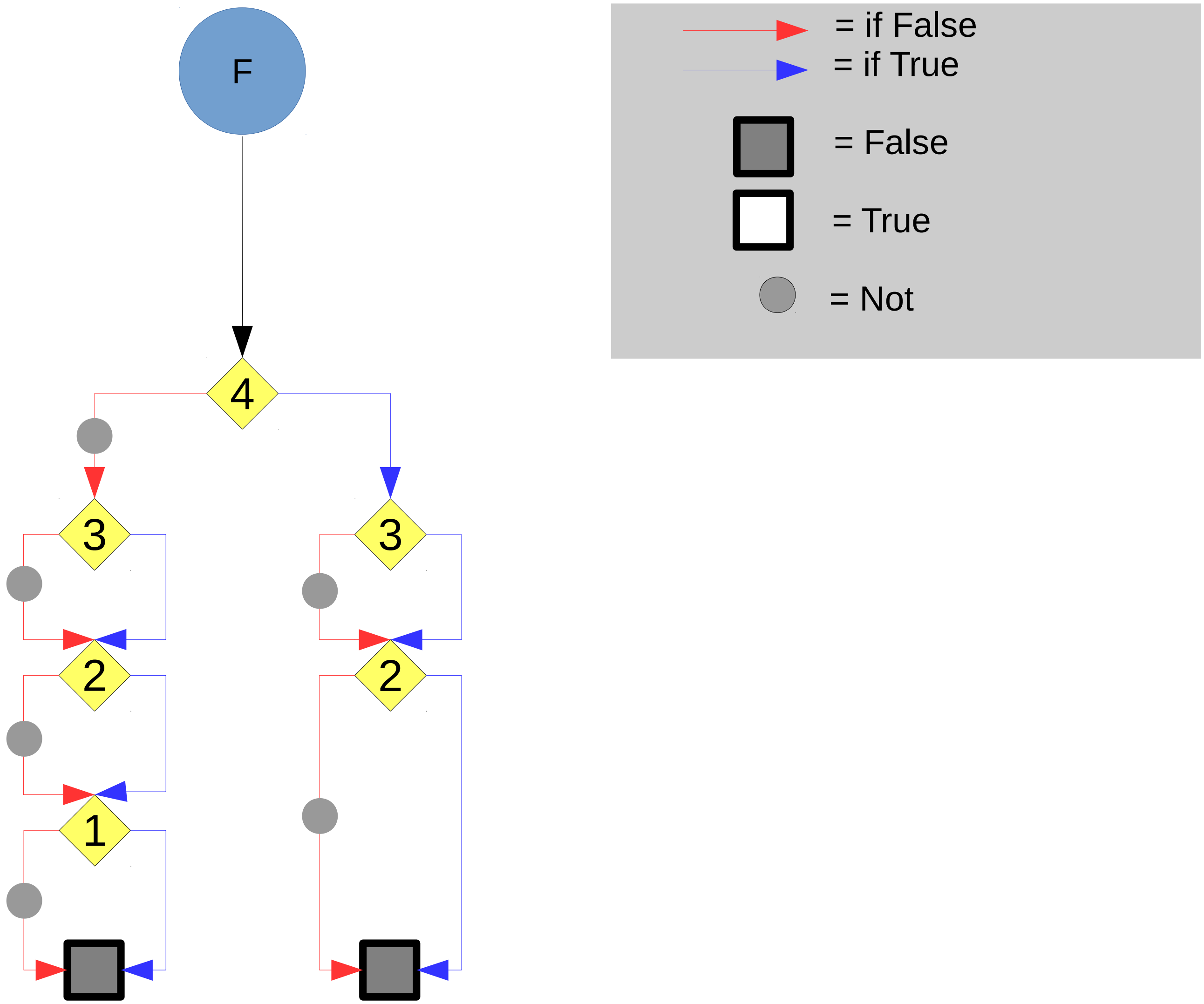
“output negation” : reduction rule (N3)



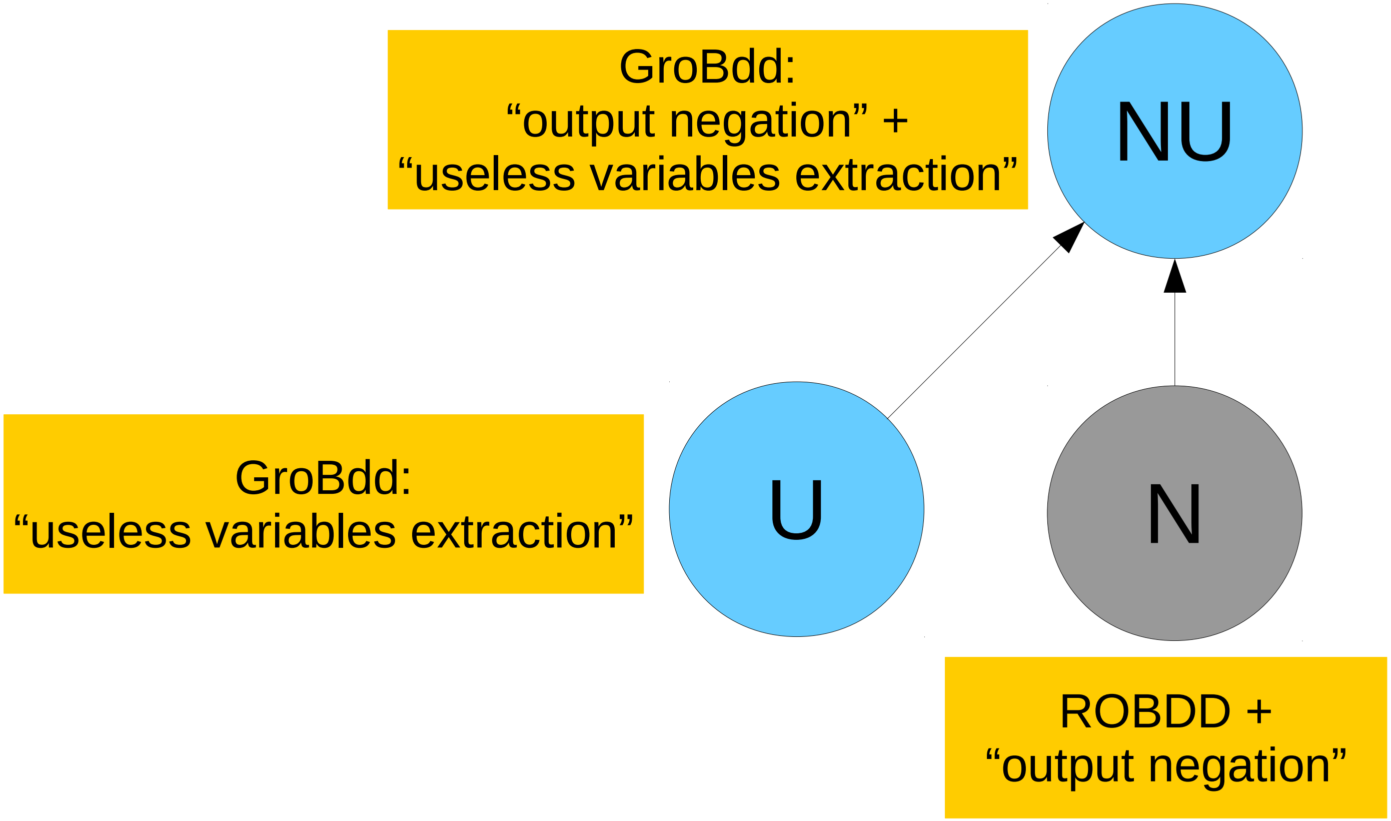
= if False
= if True
= False
= True
= Not

For aesthetic reasons, we do not merge all terminals

ROBDD + “output negation”



Generalized Reduction of Ordered Binary Decision Diagram (GroBdd)

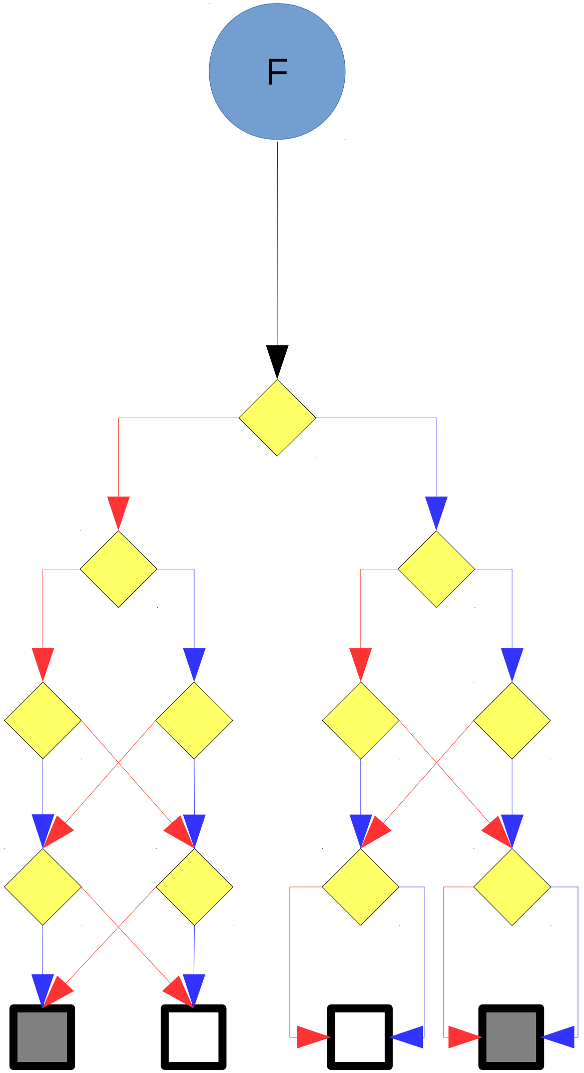


GroBdd:
“output negation” +
“useless variables extraction”

GroBdd:
“useless variables extraction”

ROBDD +
“output negation”

Shannon’s Decision Diagram



→ = if False
→ = if True

■ = False

□ = True

For aesthetic reasons, we do not merge all terminals

Section 1 : model U

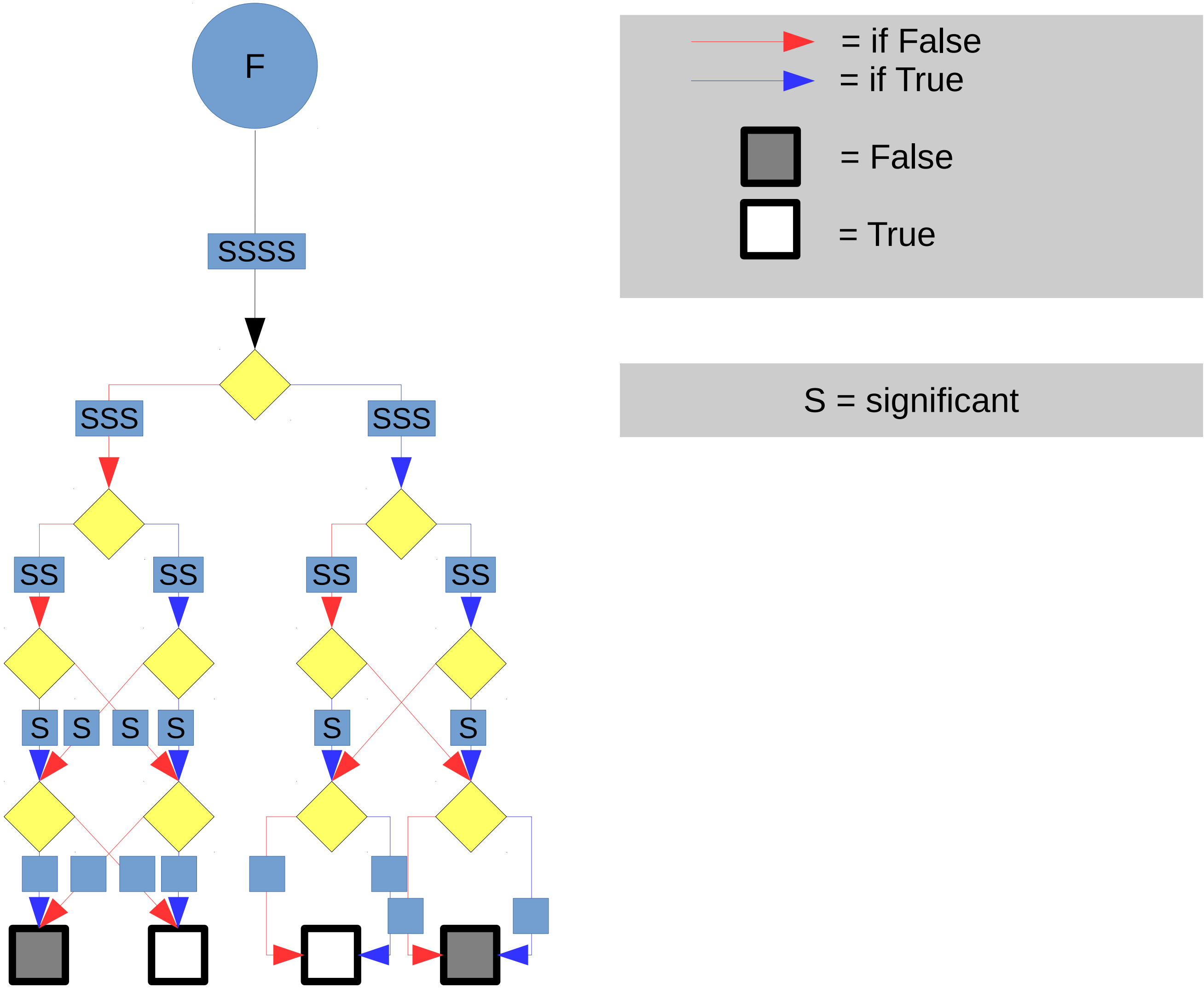
Functions of arity 4

Functions of arity 3

Functions of arity 2

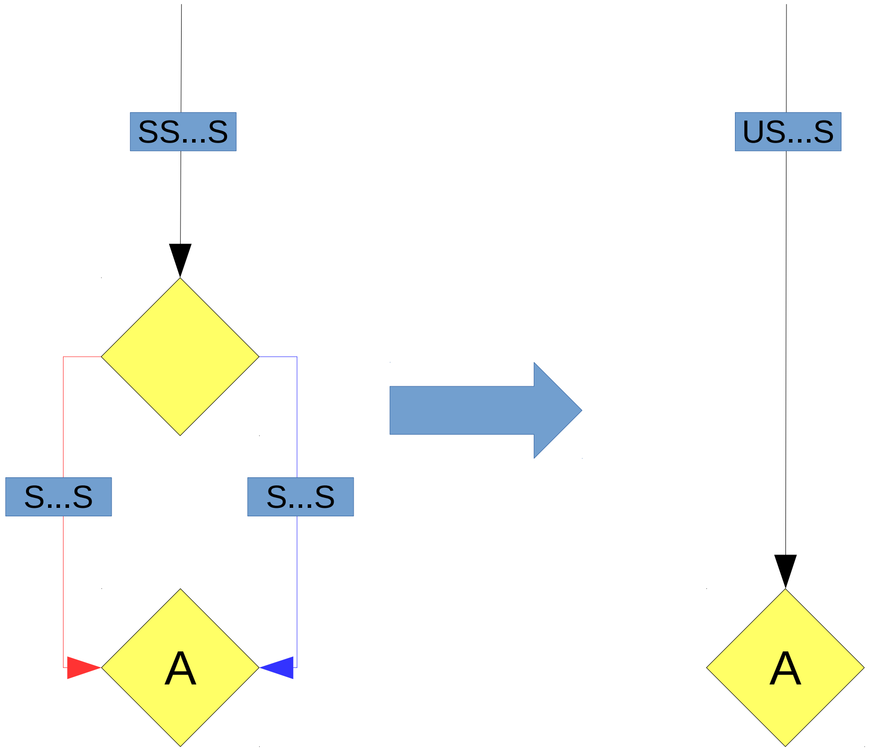
Functions of arity 1

Functions of arity 0



For aesthetic reasons, we do not merge all terminals

“Useless variables extraction” : reduction rule (U1)



→ = if False
→ = if True

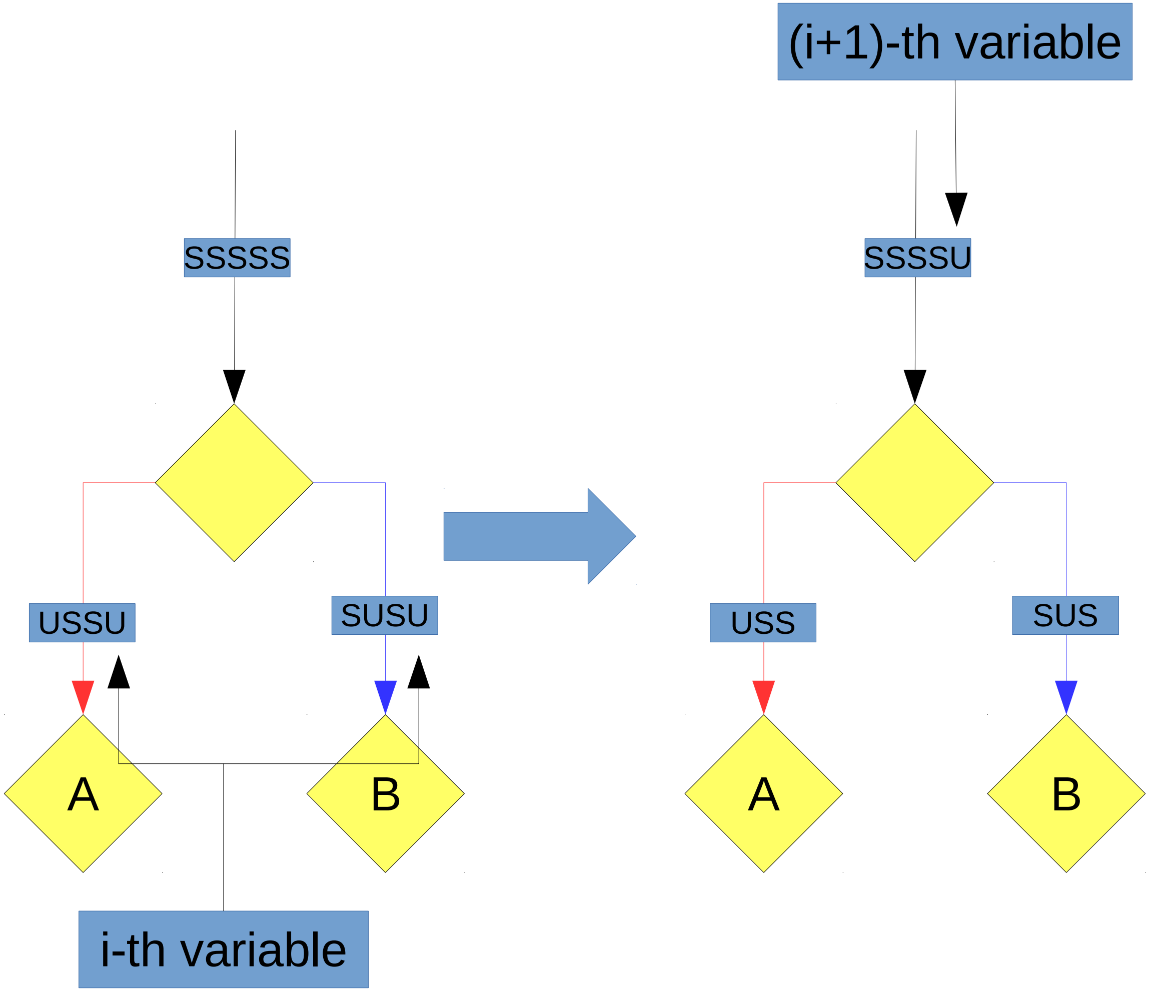
■ = False

□ = True

S = significant
U = useless

For aesthetic reasons, we do not merge all terminals

“Useless variables extraction” : reduction rule (U2)



(i+1)-th variable

SSSSS

SSSSU

USSU

SUSU

USS

SUS

A

B

A

B

i-th variable

🟥 = if False
🟦 = if True

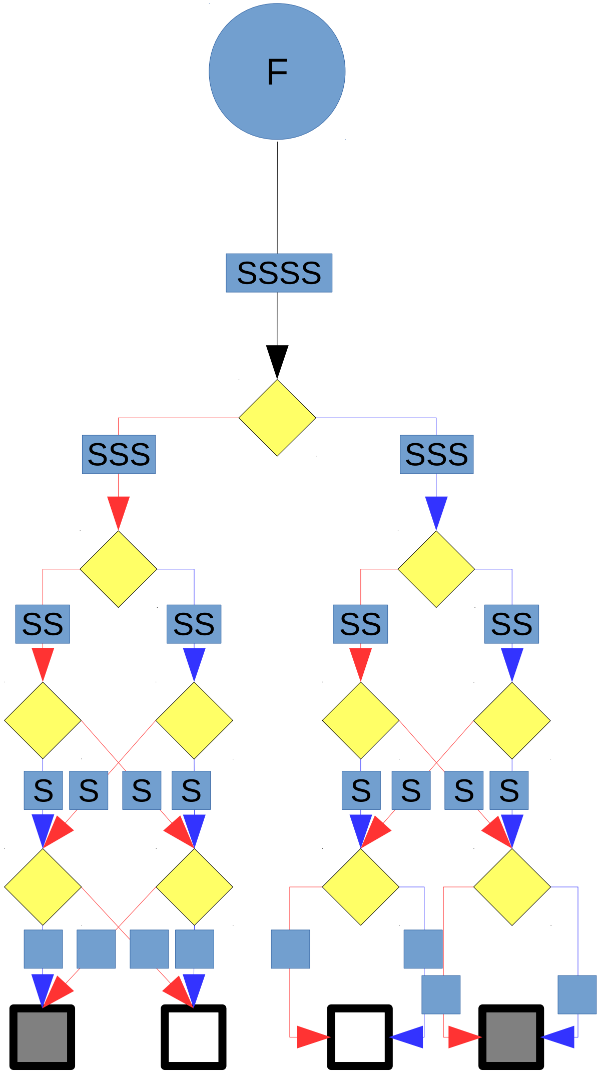
■ = False

□ = True

S = significant
U = useless

For aesthetic reasons, we do not merge all terminals

Section 1 : model U



→ = if False
→ = if True

■ = False

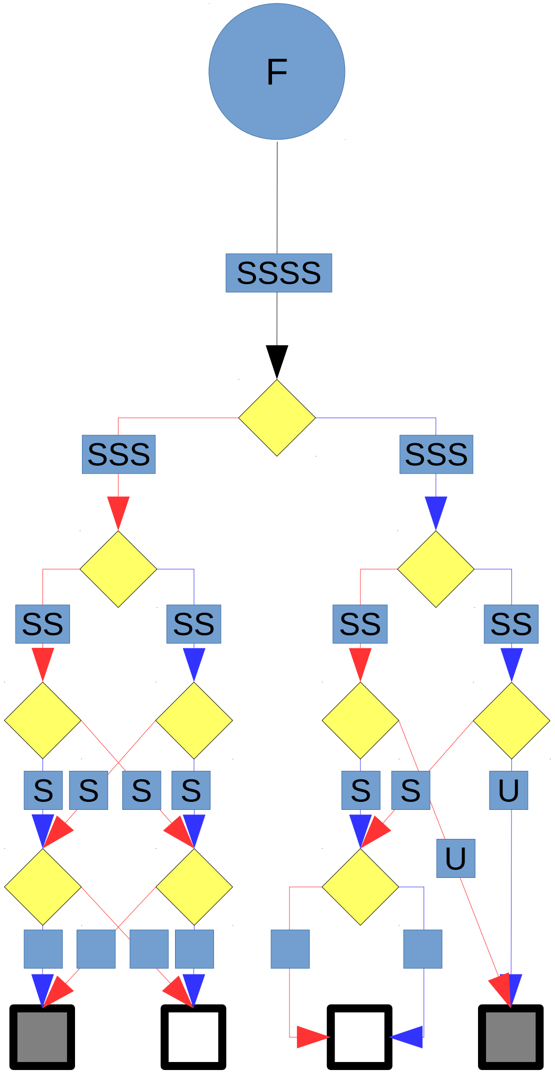
□ = True

S = significant
U = useless

Apply U1

For aesthetic reasons, we do not merge all terminals

Section 1 : model U



→ = if False
→ = if True

■ = False

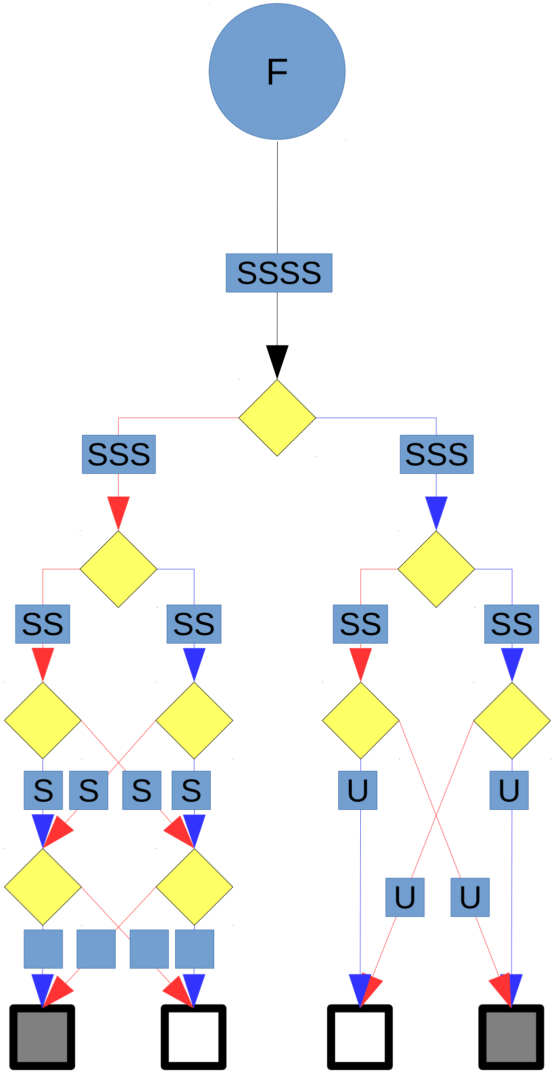
□ = True

S = significant
U = useless

Apply U1

For aesthetic reasons, we do not merge all terminals

Section 1 : model U



→ = if False
→ = if True

■ = False

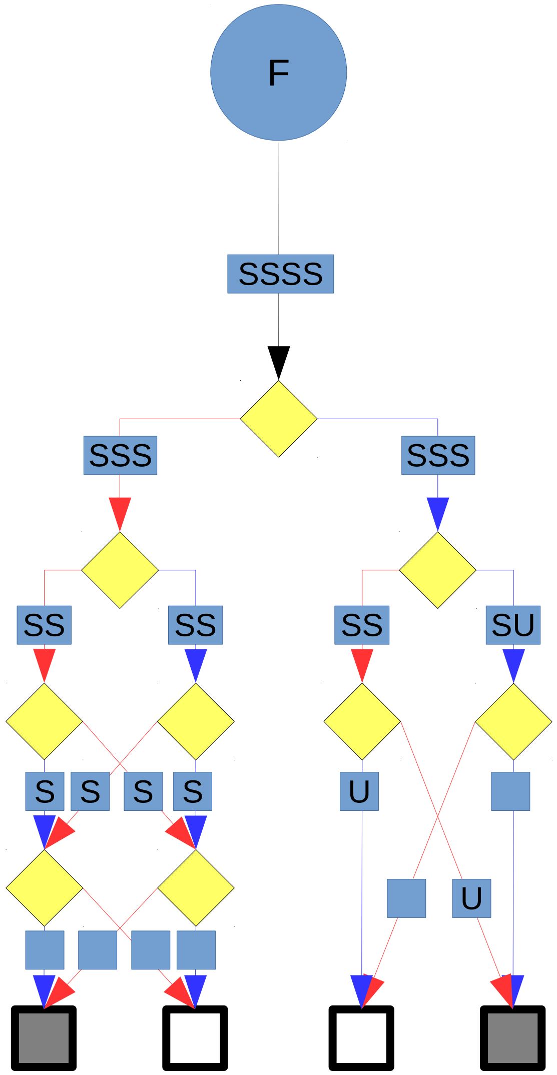
□ = True

S = significant
U = useless

Apply U2

For aesthetic reasons, we do not merge all terminals

Section 1 : model U



🟥 = if False
🟦 = if True

■ = False

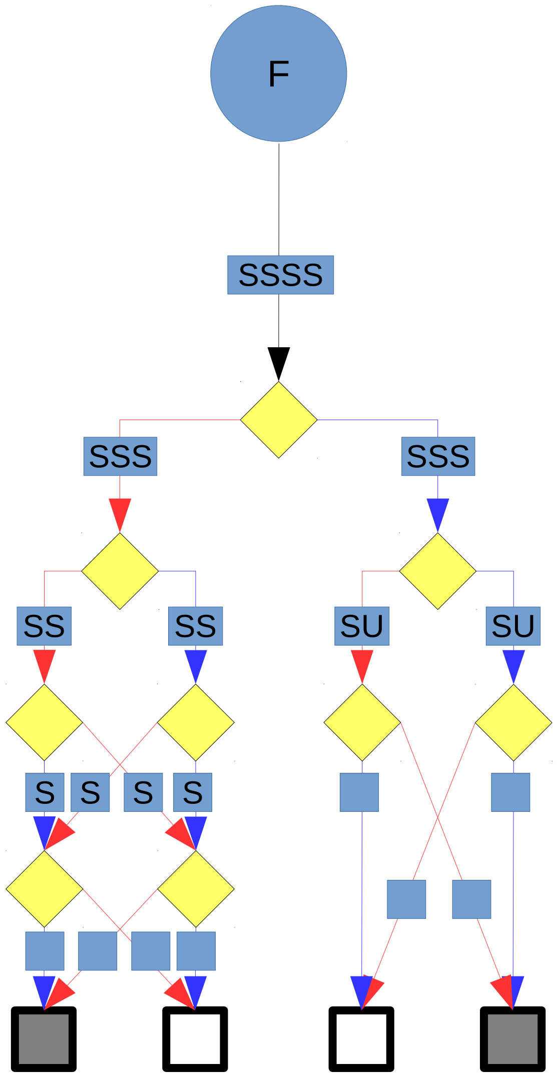
□ = True

S = significant
U = useless

Apply U2

For aesthetic reasons, we do not merge all terminals

Section 1 : model U



→ = if False
→ = if True

■ = False

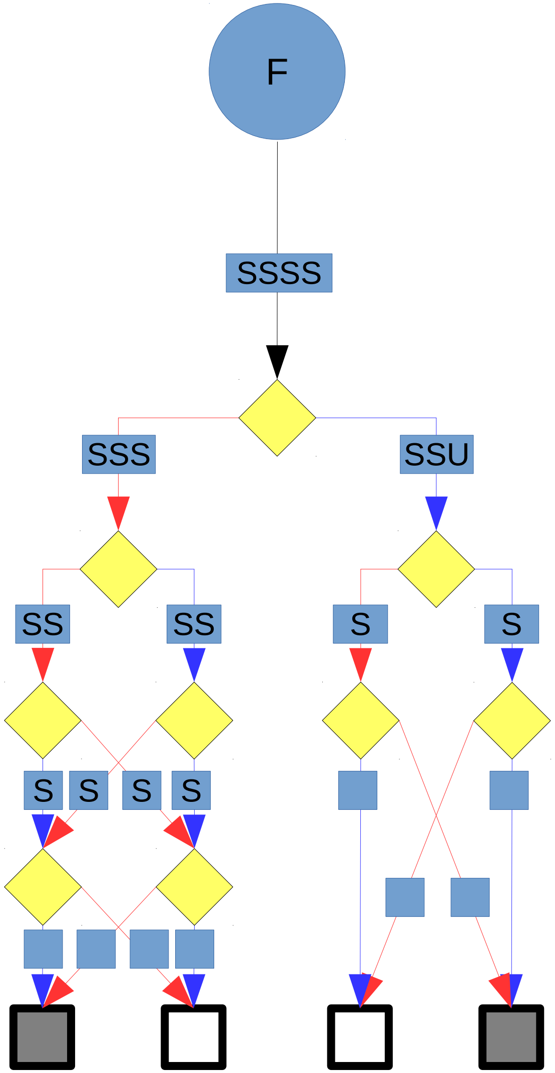
□ = True

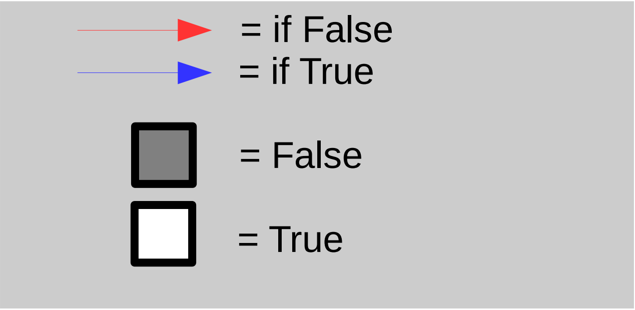
S = significant
U = useless

Apply U2

For aesthetic reasons, we do not merge all terminals

Section 1 : model U

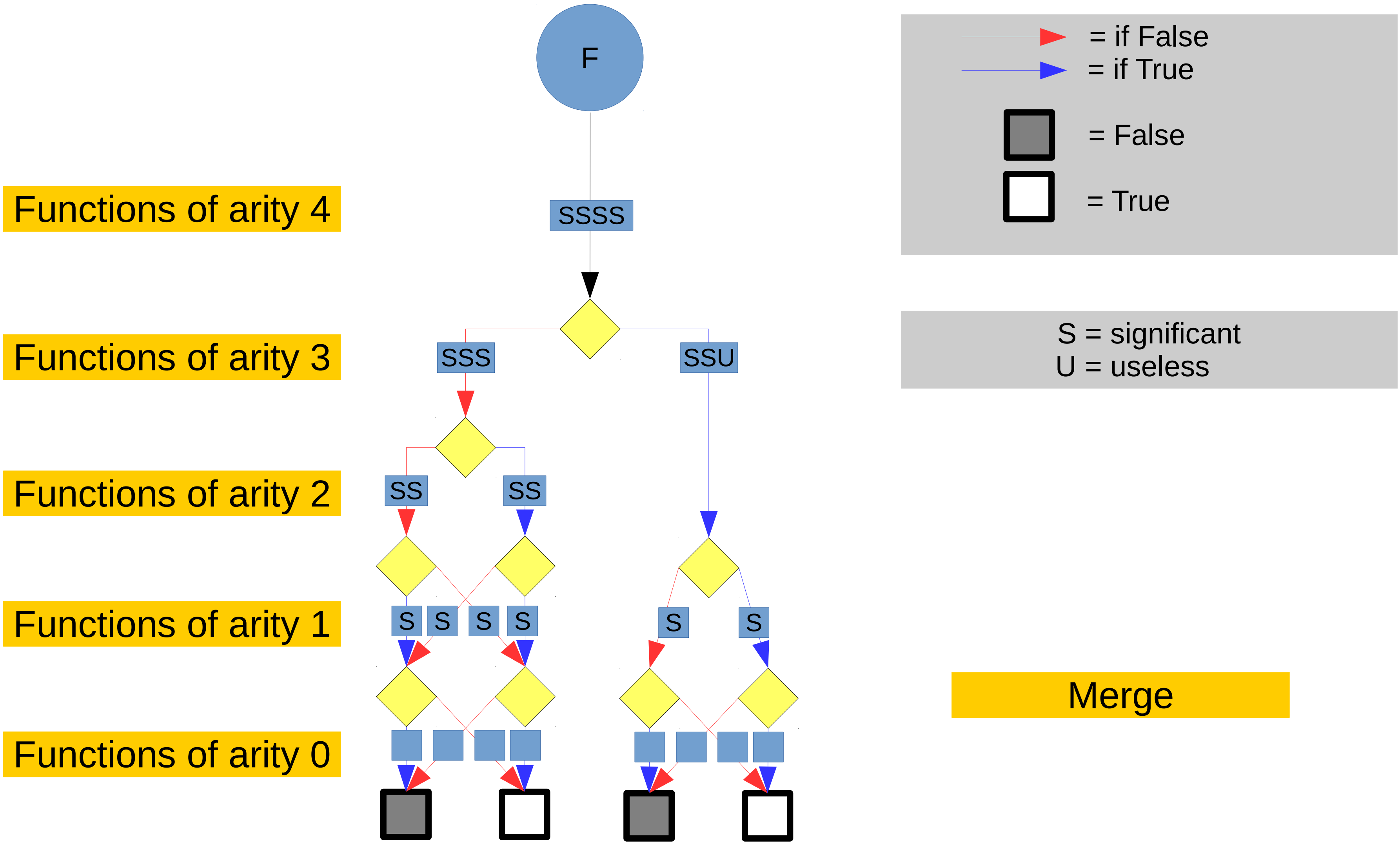




S = significant
U = useless

For aesthetic reasons, we do not merge all terminals

Section 1 : model U



Functions of arity 4

Functions of arity 3

Functions of arity 2

Functions of arity 1

Functions of arity 0

S = significant
U = useless

Merge

For aesthetic reasons, we do not merge all terminals

Section 1 : model U

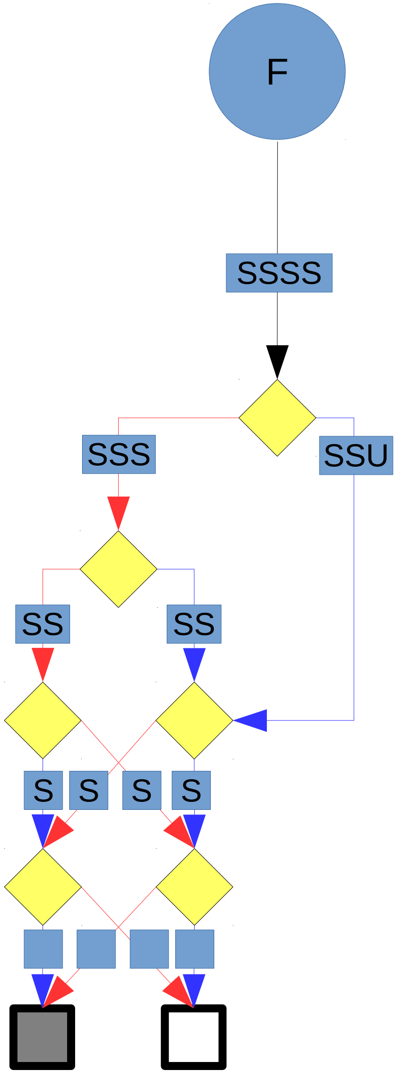
Functions of arity 4

Functions of arity 3

Functions of arity 2

Functions of arity 1

Functions of arity 0



→ = if False
→ = if True
■ = False
□ = True

S = significant
U = useless

For aesthetic reasons, we do not merge all terminals

Section 1 : model U

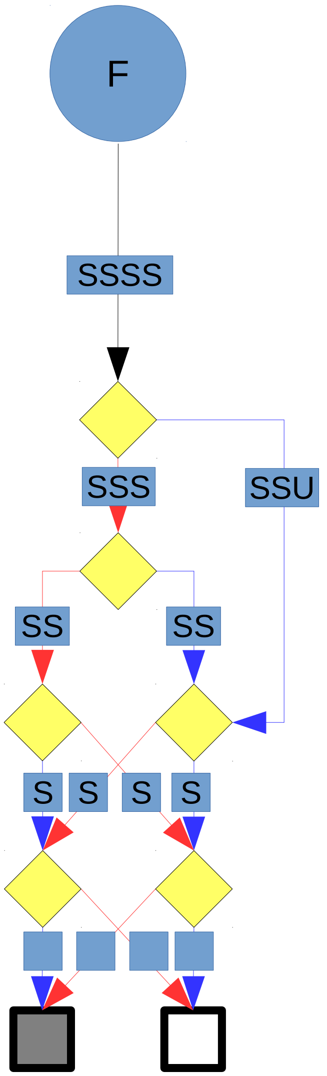
Functions of arity 4

Functions of arity 3

Functions of arity 2

Functions of arity 1

Functions of arity 0

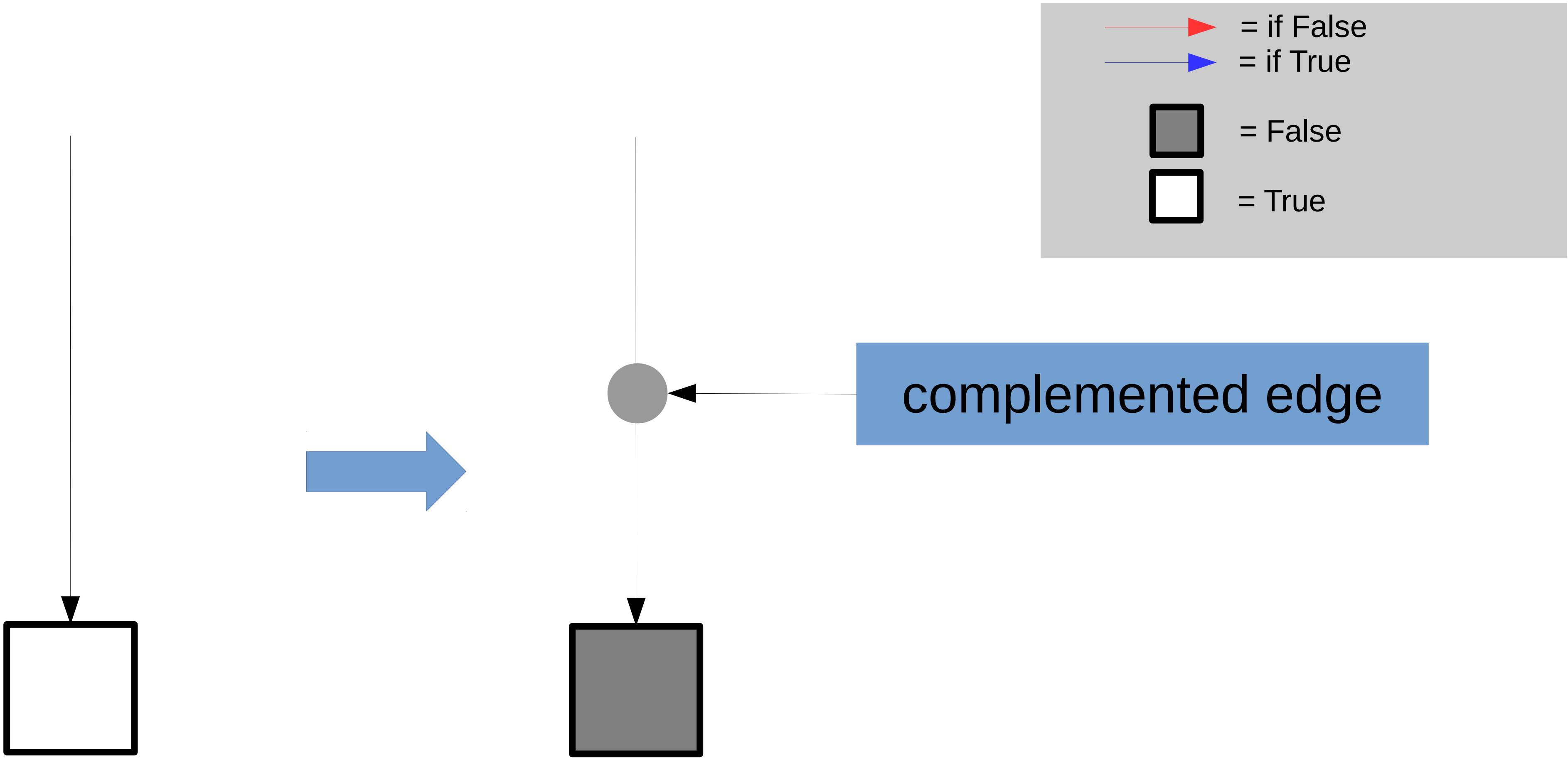


→ = if False
→ = if True
■ = False
□ = True

S = significant
U = useless

For aesthetic reasons, we do not merge all terminals

“output negation” : reduction rule (N1)



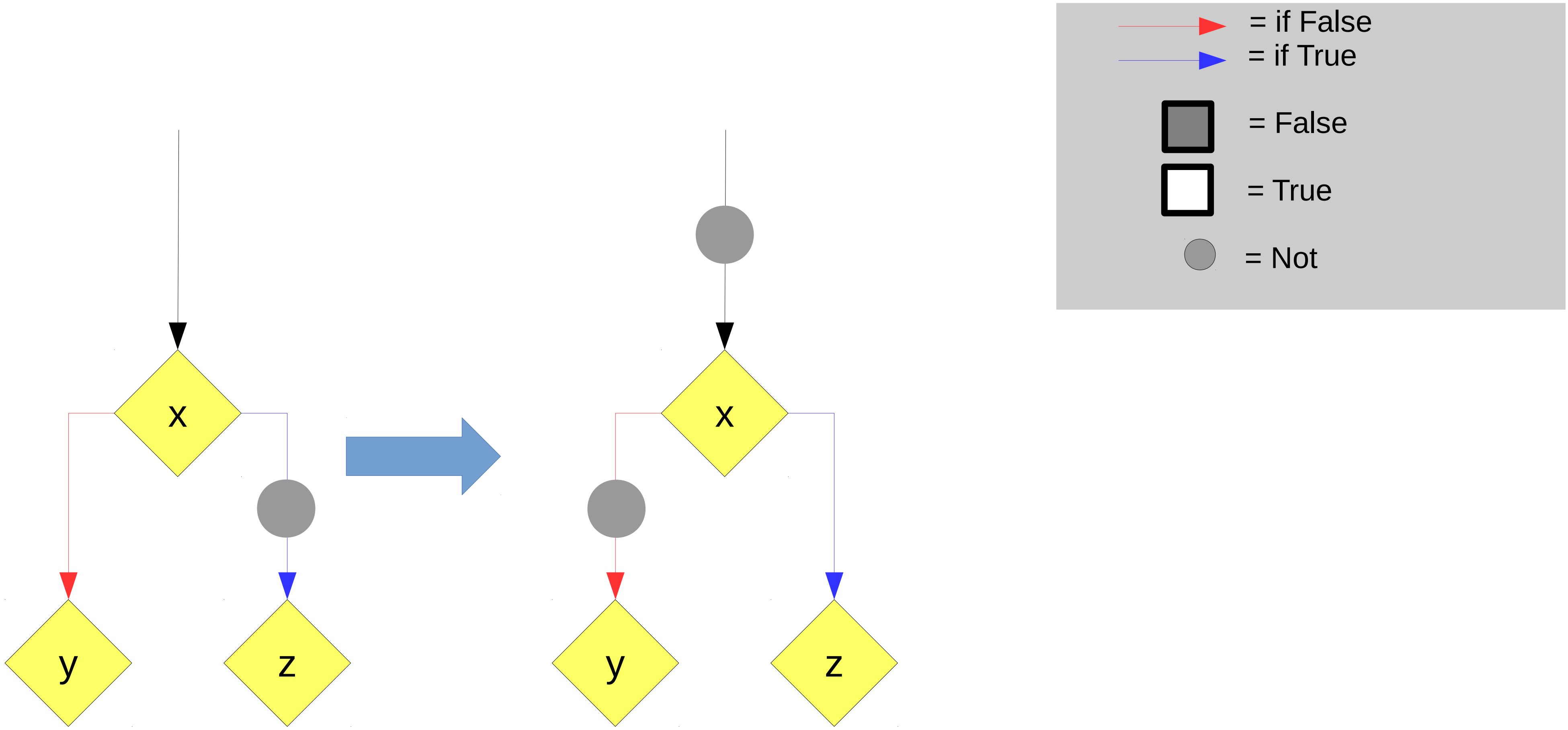
→ = if False
→ = if True

▨ = False
□ = True

complemented edge

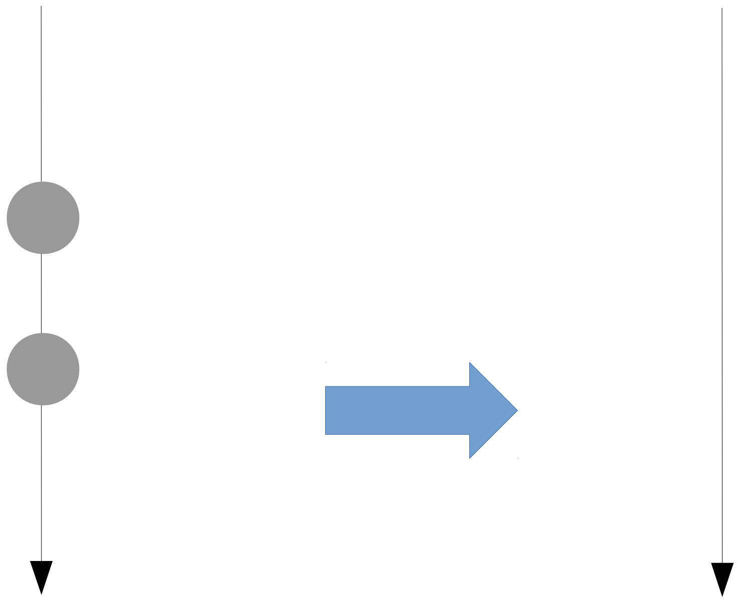
For aesthetic reasons, we do not merge all terminals

“output negation” : reduction rule (N2)



For aesthetic reasons, we do not merge all terminals

“output negation” : reduction rule (N3)



→ = if False
→ = if True

▨ = False

□ = True

● = Not

For aesthetic reasons, we do not merge all terminals

Section 2 : model NU

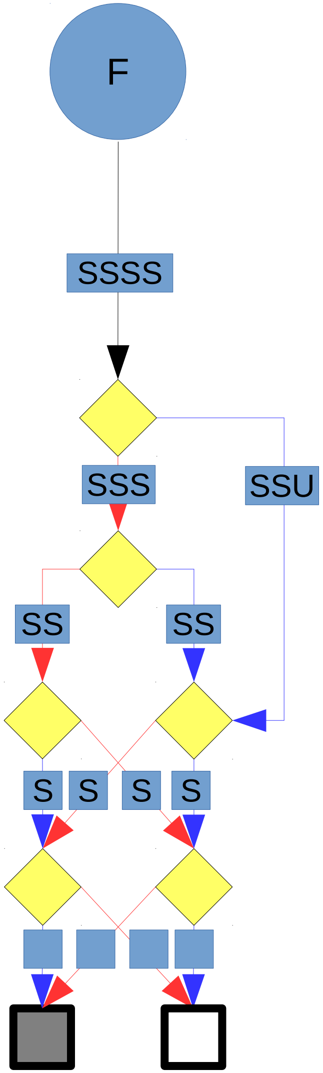
Functions of arity 4

Functions of arity 3

Functions of arity 2

Functions of arity 1

Functions of arity 0



→ = if False
→ = if True
■ = False
□ = True
● = Not

S = significant
U = useless

Apply N1

For aesthetic reasons, we do not merge all terminals

Section 2 : model NU

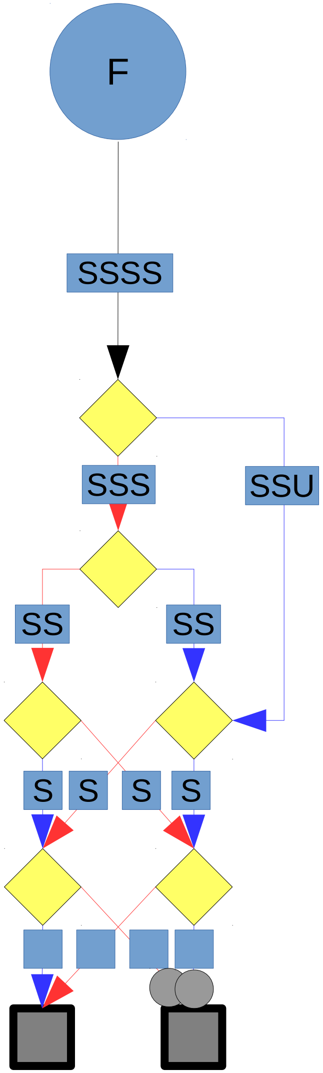
Functions of arity 4

Functions of arity 3

Functions of arity 2

Functions of arity 1

Functions of arity 0



→ = if False
→ = if True
■ = False
□ = True
● = Not

S = significant
U = useless

Apply N2

For aesthetic reasons, we do not merge all terminals

Section 2 : model NU

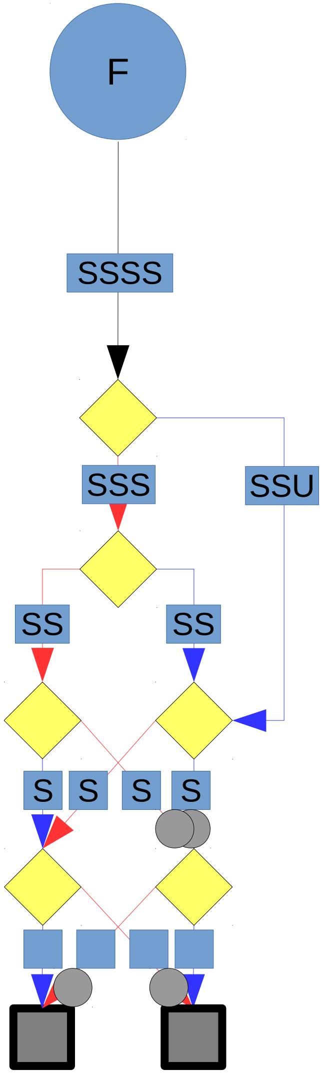
Functions of arity 4

Functions of arity 3

Functions of arity 2

Functions of arity 1

Functions of arity 0



→ = if False
→ = if True
■ = False
□ = True
● = Not

S = significant
U = useless

Apply N2

For aesthetic reasons, we do not merge all terminals

Section 2 : model NU

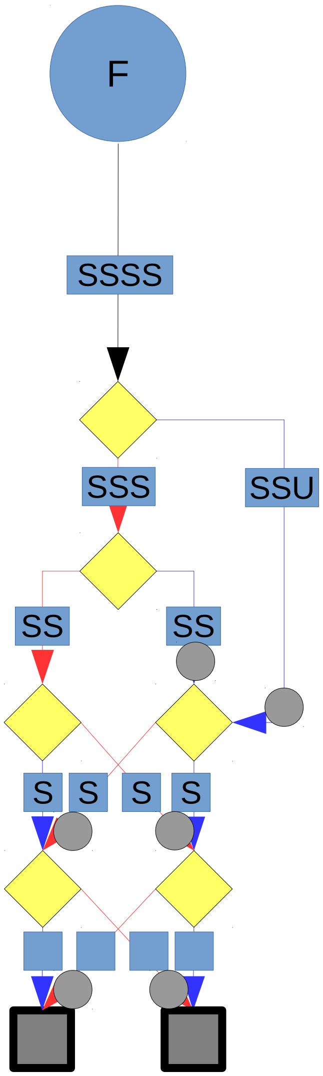
Functions of arity 4

Functions of arity 3

Functions of arity 2

Functions of arity 1

Functions of arity 0



→ = if False
→ = if True
■ = False
□ = True
● = Not

S = significant
U = useless

Apply N2

For aesthetic reasons, we do not merge all terminals

Section 2 : model NU

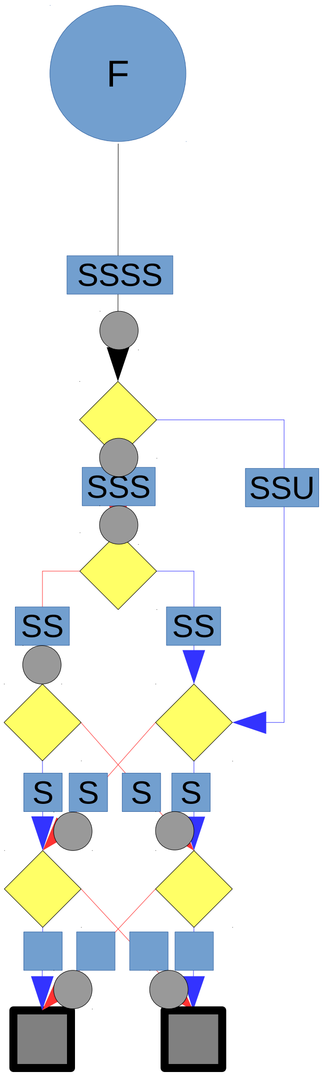
Functions of arity 4

Functions of arity 3

Functions of arity 2

Functions of arity 1

Functions of arity 0



▶ = if False
▶ = if True
■ = False
□ = True
● = Not

S = significant
U = useless

Apply N3

For aesthetic reasons, we do not merge all terminals

Section 2 : model NU

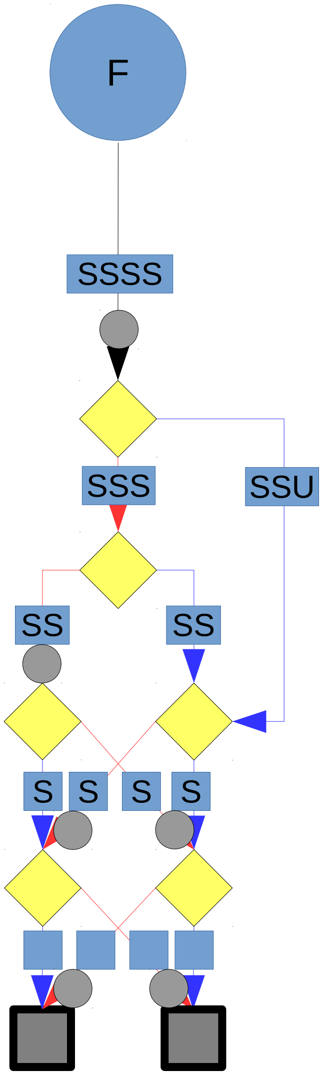
Functions of arity 4

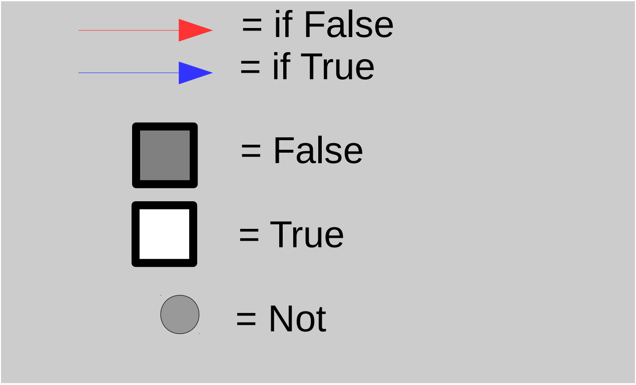
Functions of arity 3

Functions of arity 2

Functions of arity 1

Functions of arity 0





S = significant
U = useless

Merge

For aesthetic reasons, we do not merge all terminals

Section 2 : model NU

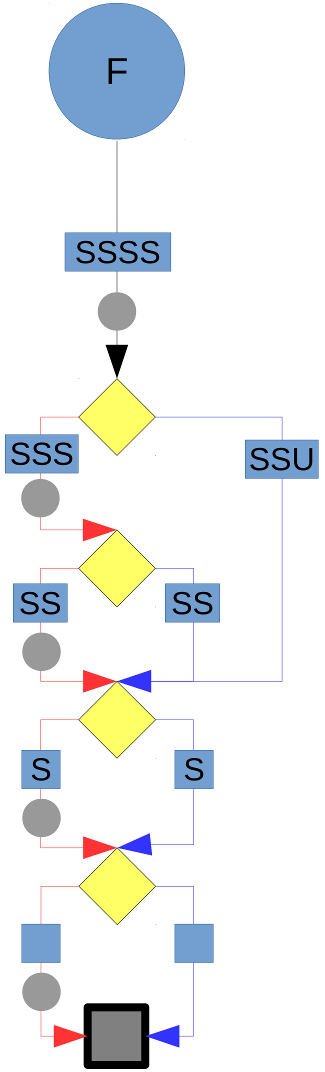
Functions of arity 4

Functions of arity 3

Functions of arity 2

Functions of arity 1

Functions of arity 0



→ = if False

→ = if True

■ = False

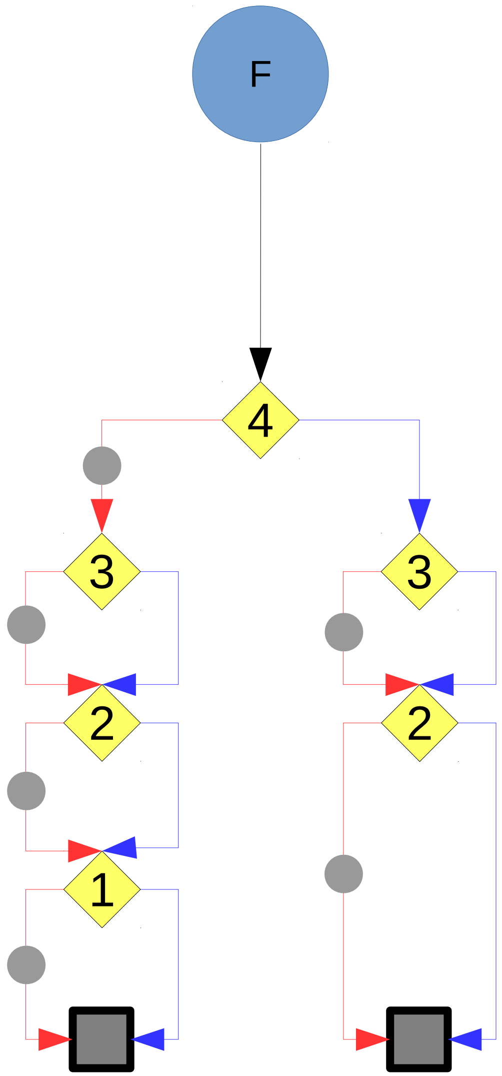
□ = True

● = Not

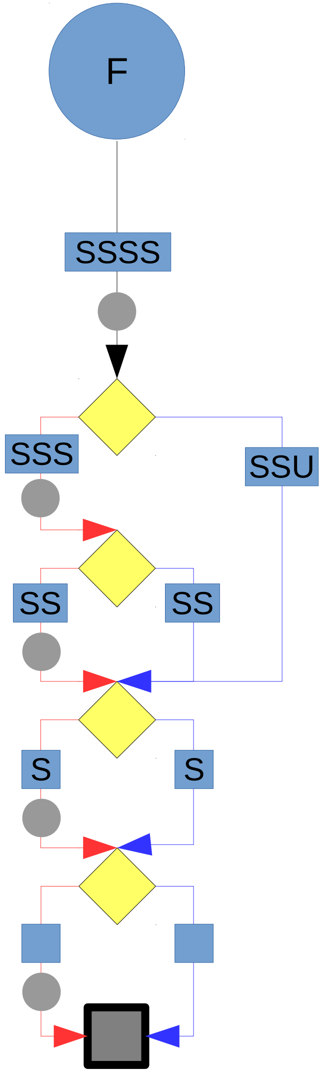
S = significant
U = useless

For aesthetic reasons, we do not merge all terminals

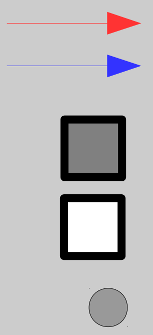
N vs NU



N =
ROBDD + “output negation”



Model NU



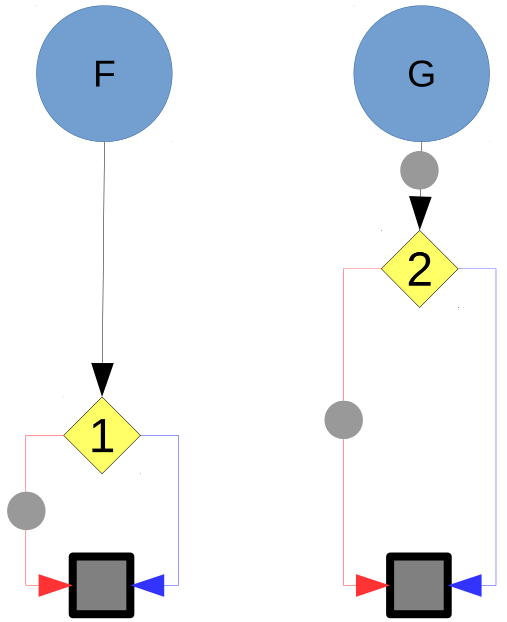
= if False
= if True
= False
= True
= Not

S = significant
U = useless

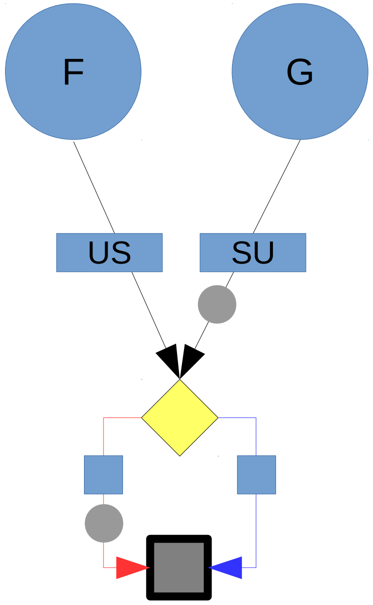
N vs NU: Example 2

→ = if False
→ = if True
■ = False
□ = True
● = Not

S = significant
U = useless

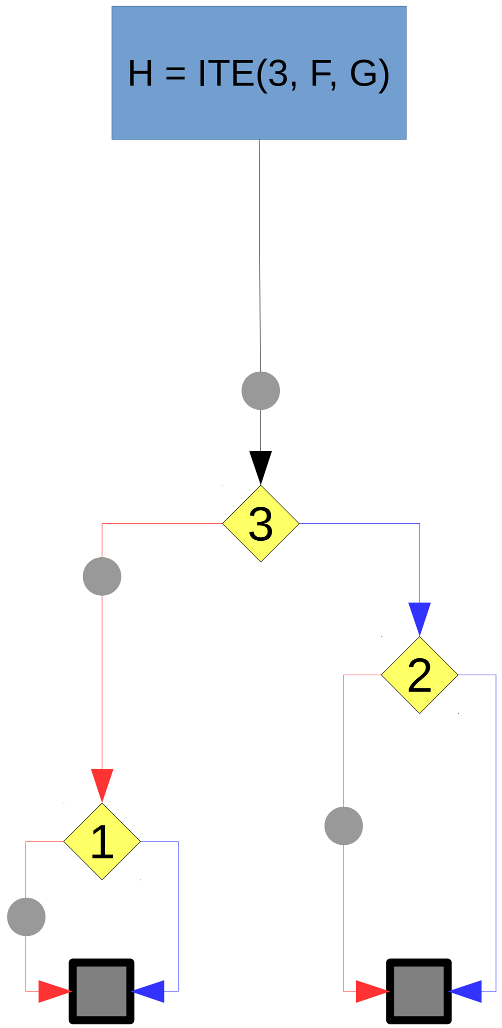


N =
ROBDD + “output negation”

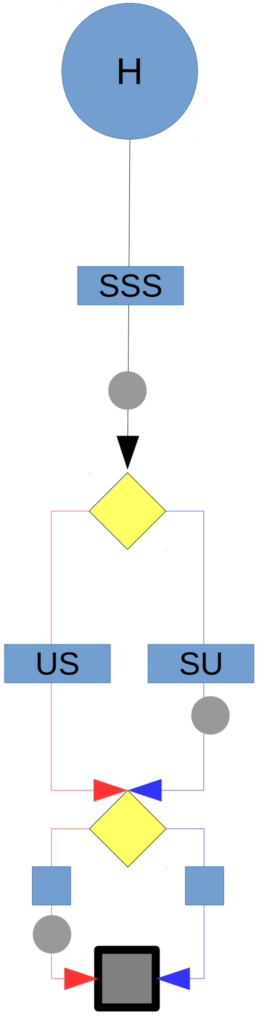


Model NU

N vs NU: Example 3



N =
ROBDD + “output negation”

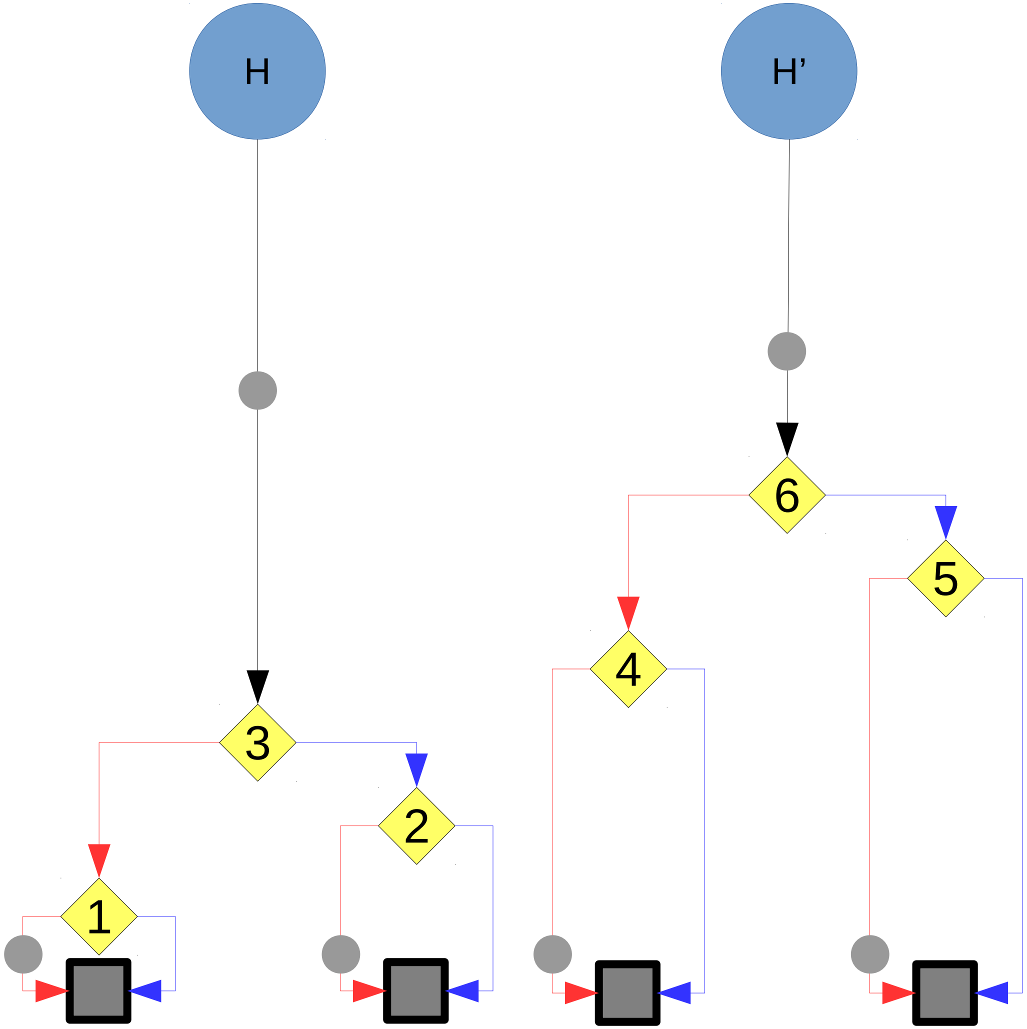


Model NU

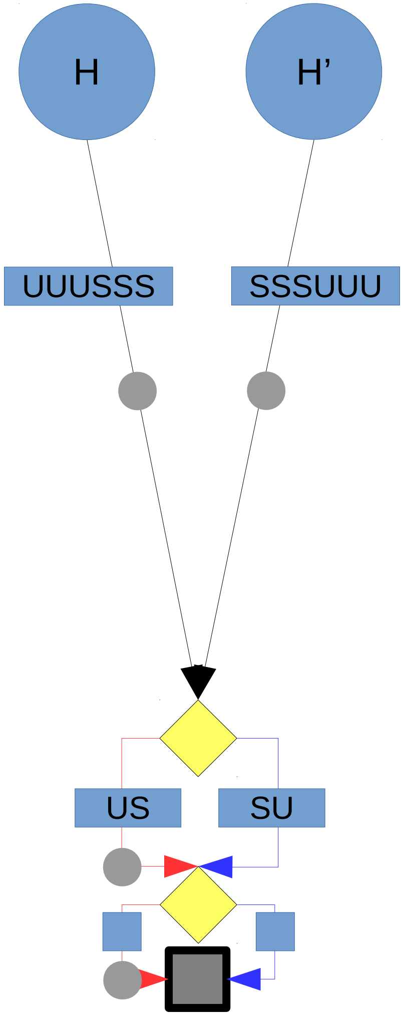
→ = if False
→ = if True
■ = False
□ = True
● = Not

S = significant
U = useless

N vs NU: Example 4



N =
ROBDD + “output negation”



Model NU

Results

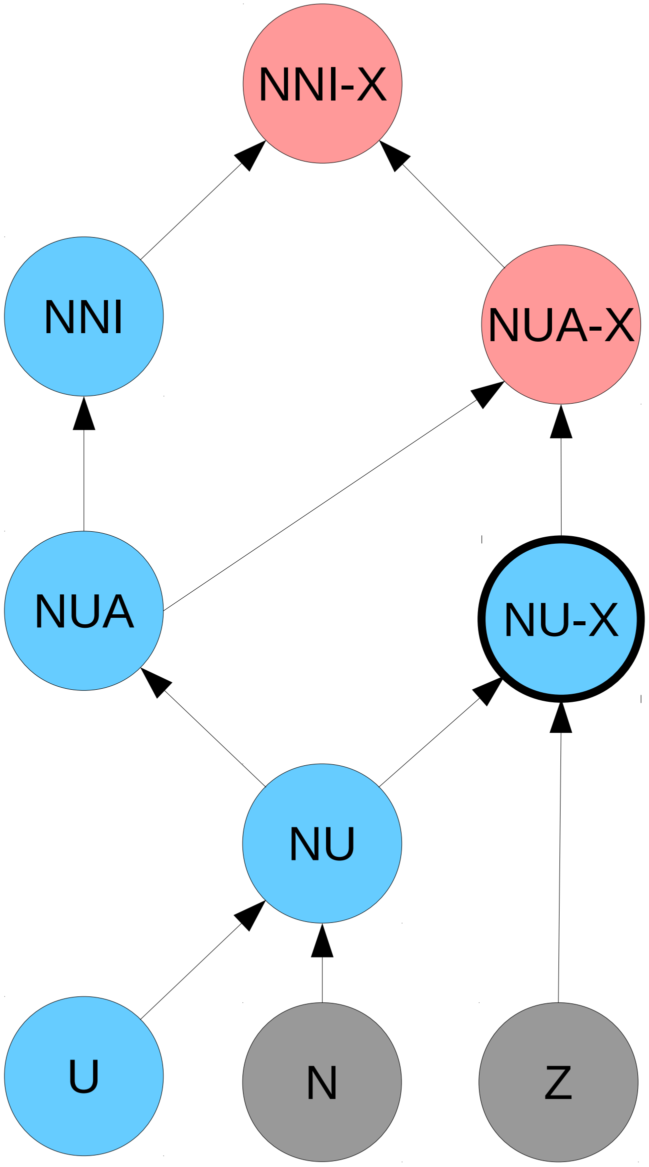
Average reduction of the {number of nodes / **estimated** memory cost} on four benchmarks

	Circuits (dense functions)				CNF formulas (sparse functions)			
	lgsynth91		iscas99		uf20-91		uf50-218	
variants	#node	mem	#node	mem	#node	mem	#node	mem
NU	-26%	-21%	-25%	-20%	-3%	+7%	-3%	+22%

Can we go further ?

“input/output negation”

“output negation” + “useless variables extraction” + “anti-variables extraction”



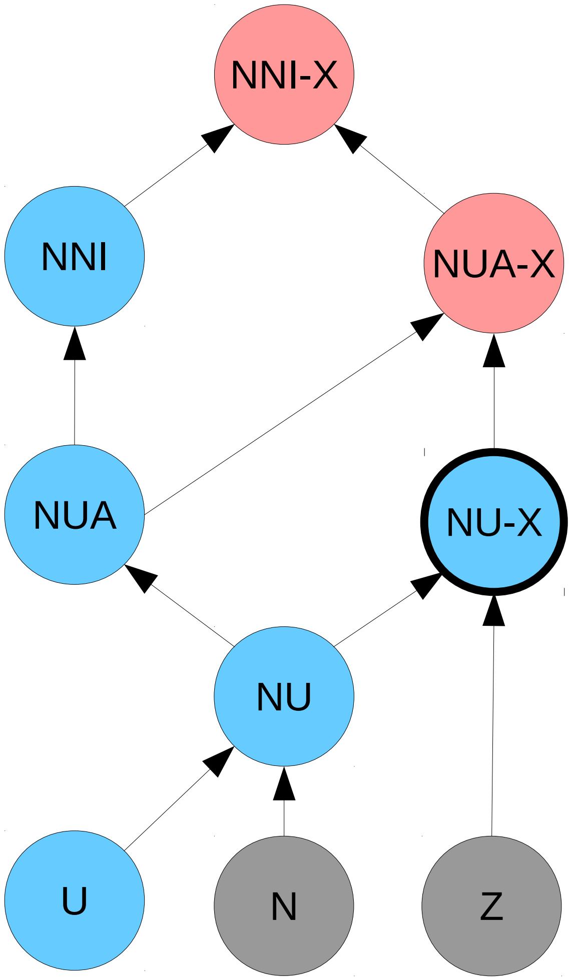
“input/output negation” + “1-prediction extraction”

“output negation” + “useless variables extraction” + “anti-variables extraction” + “1-prediction extraction”

“output negation” + “useless variables extraction” + “1-prediction extraction”

Conclusion

- Software implemented in OCaml:
  - https://github.com/JoanThibault/DAGaml/tree/grobdd-dev
  - ~ 12 000 lines of OCaml
- Fewer nodes & Less memory
- Future Work
  - Quantify the dependency between variables' order and #node
  - Solve & Implement NUA-X and NNI-X versions
- TO DO
  - Quantification Operators
  - Variable Reordering
  - Parallelism & hardware acceleration
- Other Applications
  - Apply similar strategies to compress other DAG
    - DAG / Graph isomorphism

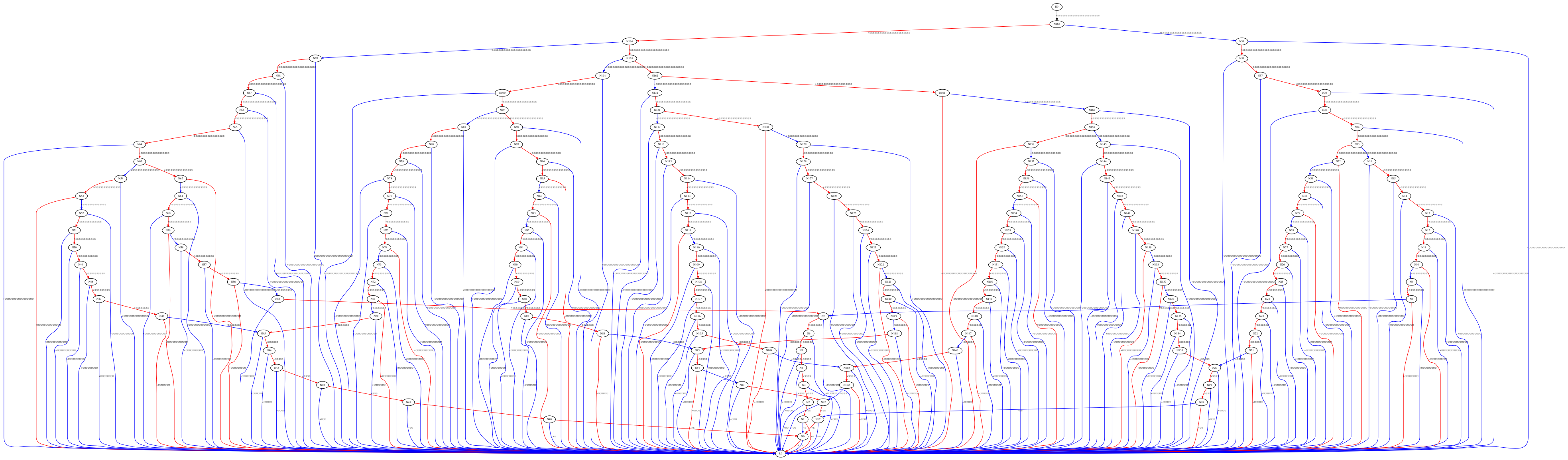


Results

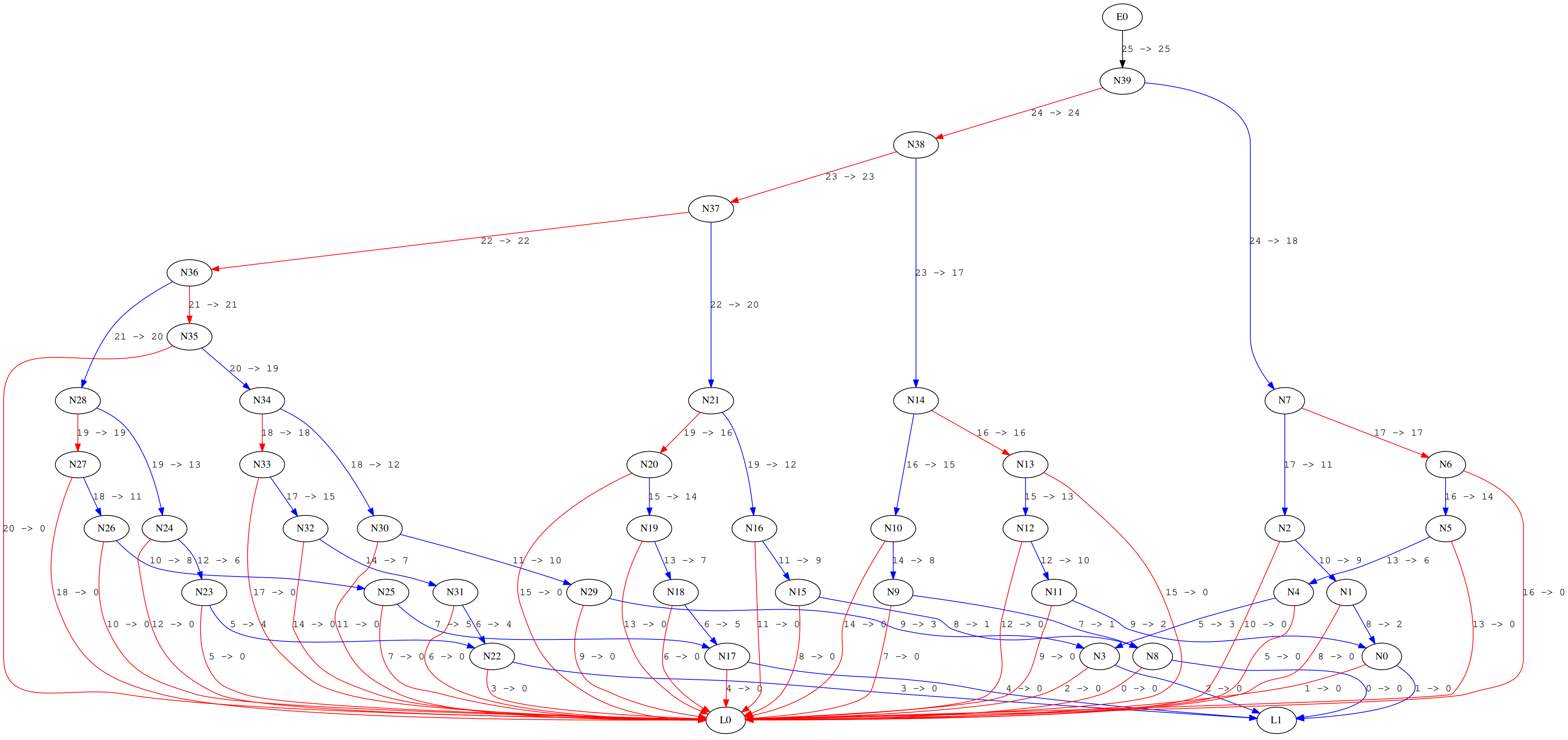
Average reduction of the {number of nodes / **estimated** memory cost} on four benchmarks

	Circuits (dense functions)				CNF formulas (sparse functions)			
	lgsynth91		iscas99		uf20-91		uf50-218	
variants	#node	mem	#node	mem	#node	mem	#node	mem
Z	+233%	+233%	+162%	+162%	-41%	-41%	-42%	-42%
NU	-26%	-21%	-25%	-20%	-3%	+7%	-3%	+22%
NNI	-60%	-53%	-56%	-49%	-30%	-10%	-39%	+5%
NU-X	-64%	-58%	-55%	-46%	-96%	-95%	-97%	-96%

5-Queens: N or NU

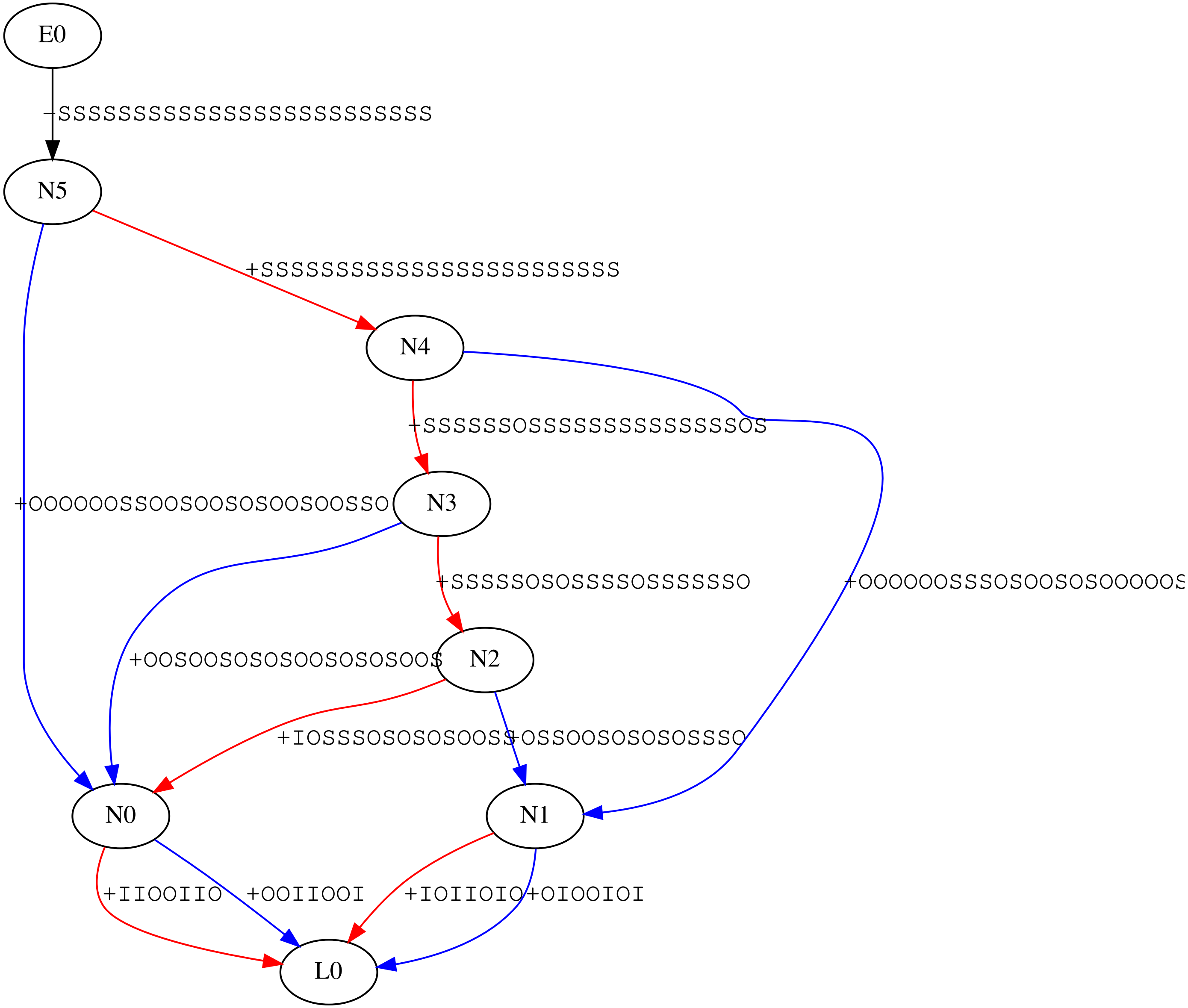


5-Queens: Z

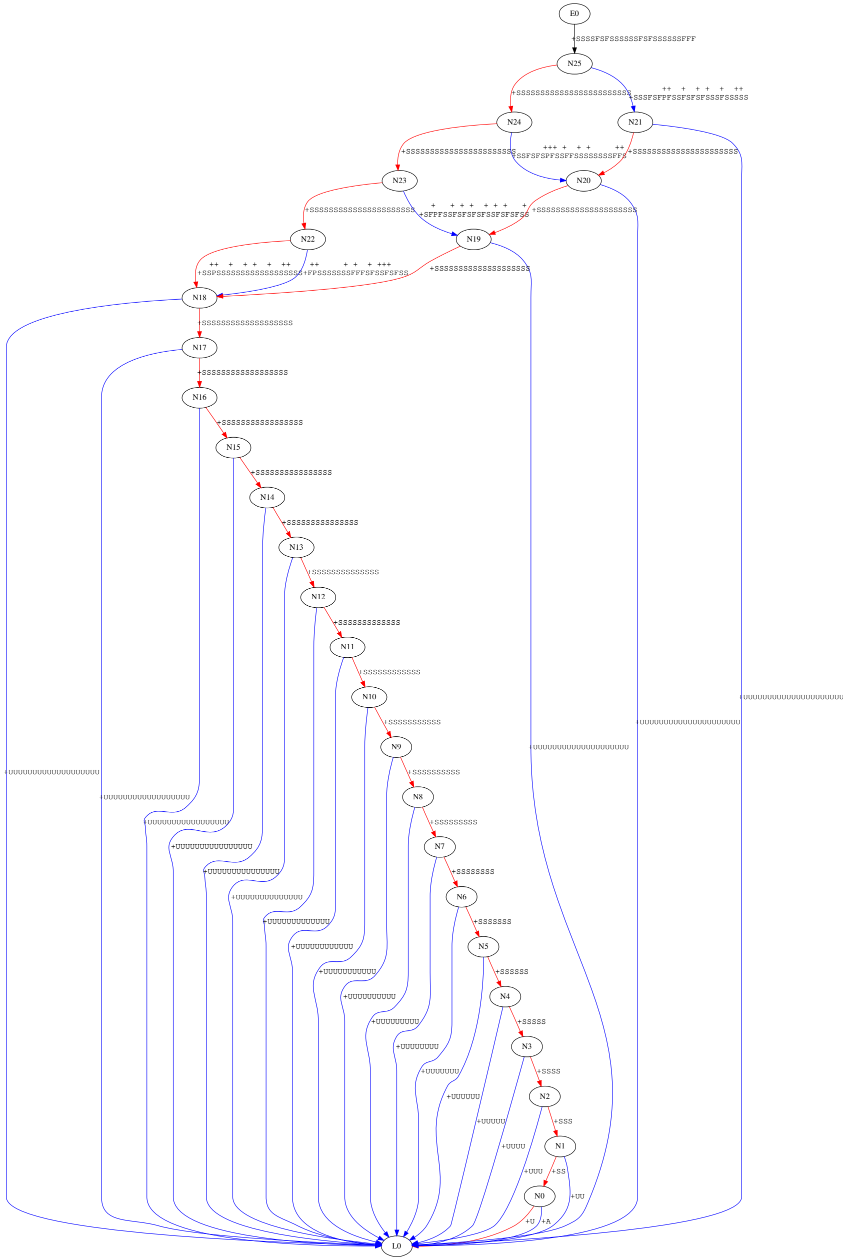


E0
25 -> 25
N39
24 -> 24
N38
23 -> 23
N37
22 -> 22
24 -> 18
N36
23 -> 17
21 -> 21
22 -> 20
21 -> 20
N35
20 -> 19
N28
N34
N21
N14
N7
19 -> 19
18 -> 18
19 -> 16
16 -> 16
17 -> 17
N27
19 -> 13
N33
18 -> 12
N20
19 -> 12
16 -> 15
N13
17 -> 11
N6
18 -> 11
17 -> 15
15 -> 14
15 -> 13
16 -> 14
20 -> 0
N26
N24
N32
N30
N19
N16
N10
N12
N2
N5
10 -> 8
12 -> 6
14 -> 7
11 -> 10
13 -> 7
11 -> 9
14 -> 8
12 -> 10
10 -> 9
13 -> 6
18 -> 0
N23
17 -> 0
N25
N31
15 -> 0
N29
N18
N15
N9
N11
15 -> 0
N4
N1
16 -> 0
10 -> 0
12 -> 0
5 -> 4
14 -> 0
11 -> 0
7 -> 5
6 -> 4
13 -> 0
6 -> 5
11 -> 0
14 -> 0
9 -> 3
8 -> 1
12 -> 0
7 -> 1
9 -> 2
5 -> 3
10 -> 0
8 -> 2
13 -> 0
5 -> 0
7 -> 0
6 -> 0
N22
9 -> 0
6 -> 0
N17
8 -> 0
7 -> 0
9 -> 0
N3
N8
5 -> 0
8 -> 0
N0
3 -> 0
4 -> 0
3 -> 0
4 -> 0
2 -> 0
0 -> 0
2 -> 0
1 -> 0
0 -> 0
1 -> 0
L0
L1

5-Queens: NU-X


E0
+SSSSSSSSSSSSSSSSSSSSSSSSS
N5
+SSSSSSSSSSSSSSSSSSSSSSSSS
N4
+SSSSSSOSSSSSSSSSSSSSSSSOS
+OOOOOOSSOOSOOSOSOOSOOSSO
N3
+SSSSSOSOSSSSOSSSSSSO
+OOOOOOSSSOSOOSOSOOOOOS
+OOSOOSOSOSOOSOSOSOOS
N2
+IOSSSOSOSOSOOSS
+OSSOOSOSOSOSSSO
N0
N1
+IIOOIIO
+OOIIOOI
+IOIIOIO
+OIOOIOI
L0

5-Queens: NNI



E0
+SSSSFSFSSSSSSFSFSSSSSSFFF
N25
+SSSSSSSSSSSSSSSSSSSSSSSS
++   +   + +   +   ++
+SSSFSFPFSSFSFSFSSSFSSSSS
N24
N21
+SSSSSSSSSSSSSSSSSSSSSSS
+++ +   + +   ++
+SSFSFSPFSSFFSSSSSSSSFFS
+SSSSSSSSSSSSSSSSSSSSS
N23
N20
+SSSSSSSSSSSSSSSSSSSSSS
+   + + +   + + +   +
+SFPFSSFSFSFSFSSFSFSFSS
+SSSSSSSSSSSSSSSSSSSS
N22
N19
++   +   + +   +   ++
+SSPSSSSSSSSSSSSSSSSSS
++   + +   + +++
+FPSSSSSSSFFFSFSSFSFSS
+SSSSSSSSSSSSSSSSSSS
N18
+SSSSSSSSSSSSSSSSSSS
N17
+SSSSSSSSSSSSSSSSSS
N16
+SSSSSSSSSSSSSSSSS
N15
+SSSSSSSSSSSSSSSS
N14
+SSSSSSSSSSSSSSS
N13
+SSSSSSSSSSSSSS
N12
+SSSSSSSSSSSSS
N11
+SSSSSSSSSSSS
N10
+SSSSSSSSSSS
N9
+SSSSSSSSSS
N8
+SSSSSSSSS
N7
+SSSSSSSS
N6
+SSSSSSS
N5
+SSSSSS
N4
+SSSSS
N3
+SSSS
N2
+SSS
N1
+SS
N0
+UUUUUUUUUUUUUUUUUUUUUU
+UUUUUUUUUUUUUUUUUUUUU
+UUUUUUUUUUUUUUUUUUUU
+UUUUUUUUUUUUUUUUUUU
+UUUUUUUUUUUUUUUUUU
+UUUUUUUUUUUUUUUUU
+UUUUUUUUUUUUUUUU
+UUUUUUUUUUUUUUU
+UUUUUUUUUUUUUU
+UUUUUUUUUUUUU
+UUUUUUUUUUUU
+UUUUUUUUUUU
+UUUUUUUUUU
+UUUUUUUUU
+UUUUUUUU
+UUUUUUU
+UUUUUU
+UUUUU
+UUUU
+UUU
+UU
+U
+A
L0