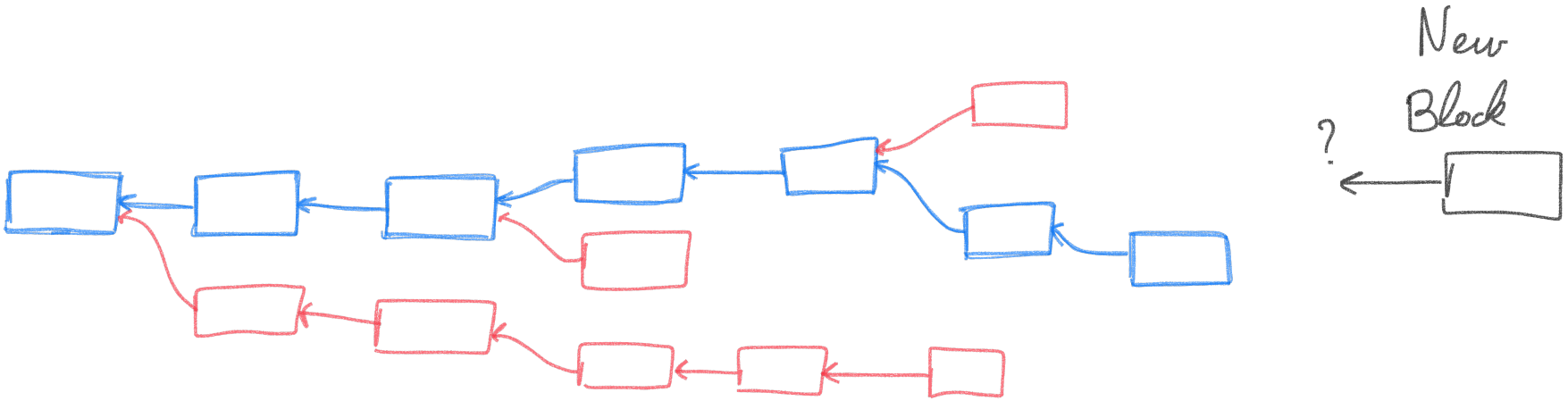
Computing the **Heaviest Conflict-free Sub-DAG** in DAG-based DLTs

Quentin Bramas
ICUBE, University of Strasbourg

anr agence nationale de la recherche
ANR-22-CE25-0012
ICUBE
Université de Strasbourg

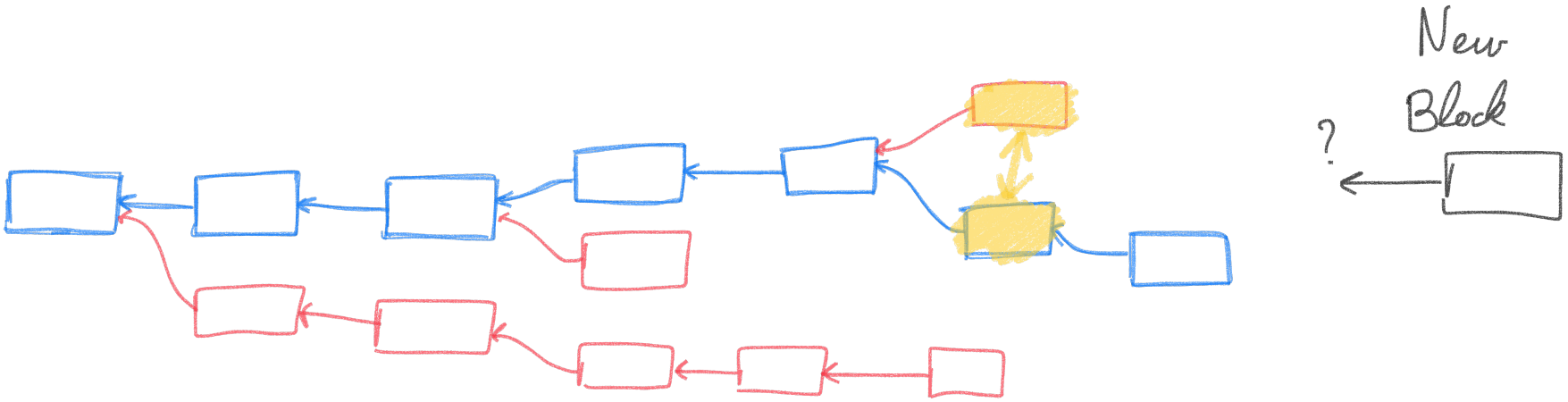
Context

Current state = parent I would select for a new block

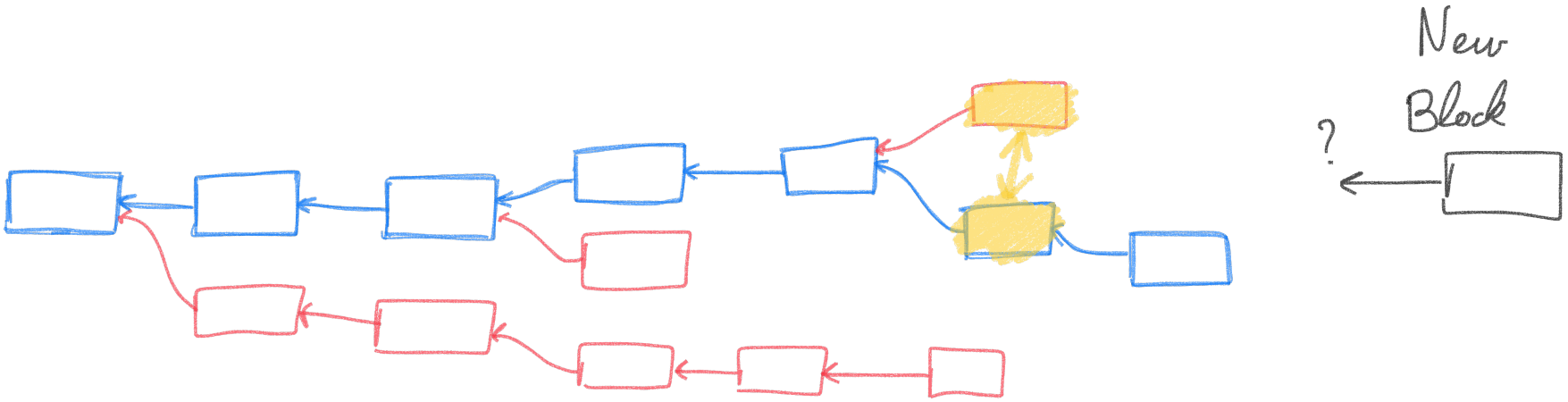


= heaviest branch

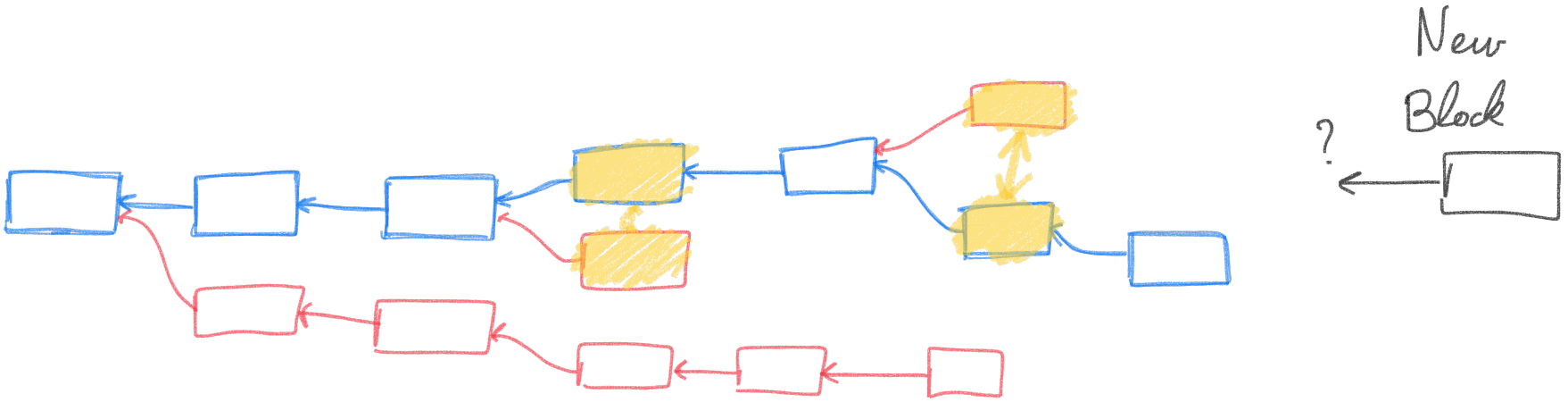
Resolving conflicts in Blockchains



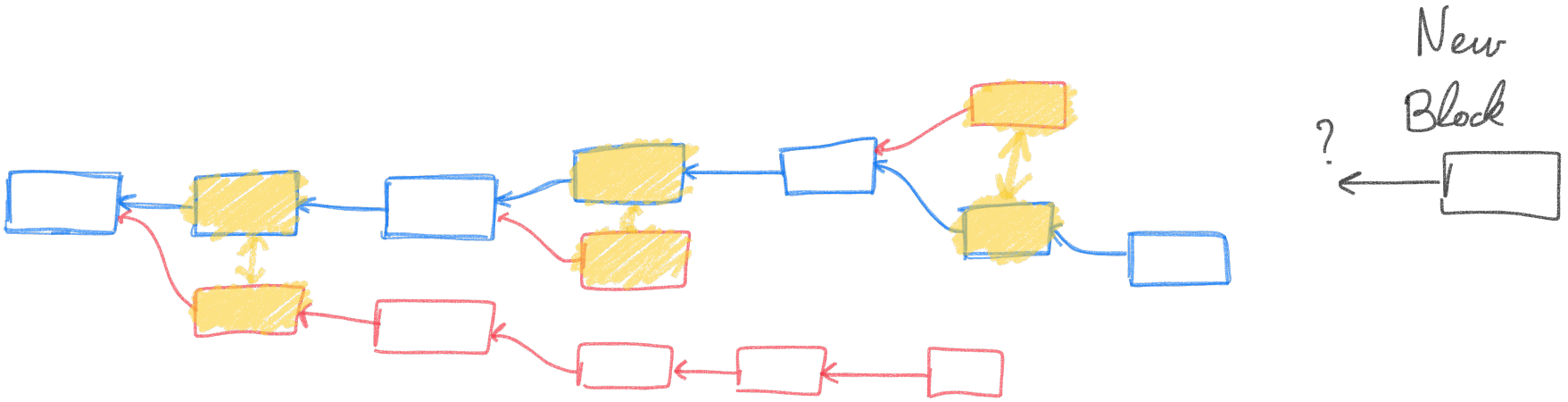
Resolving conflicts in Blockchains



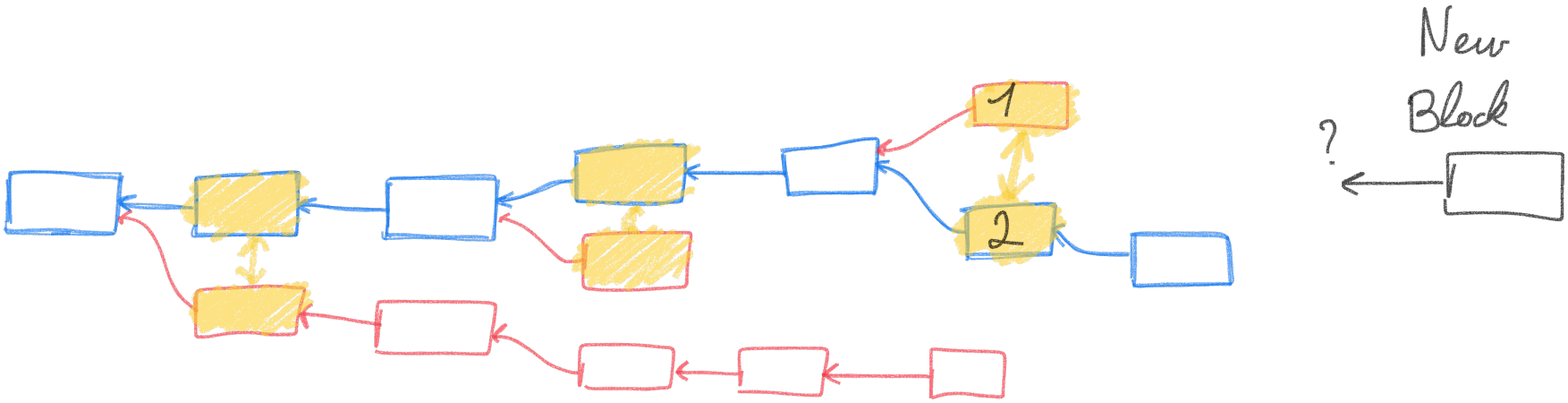
Resolving conflicts in Blockchains



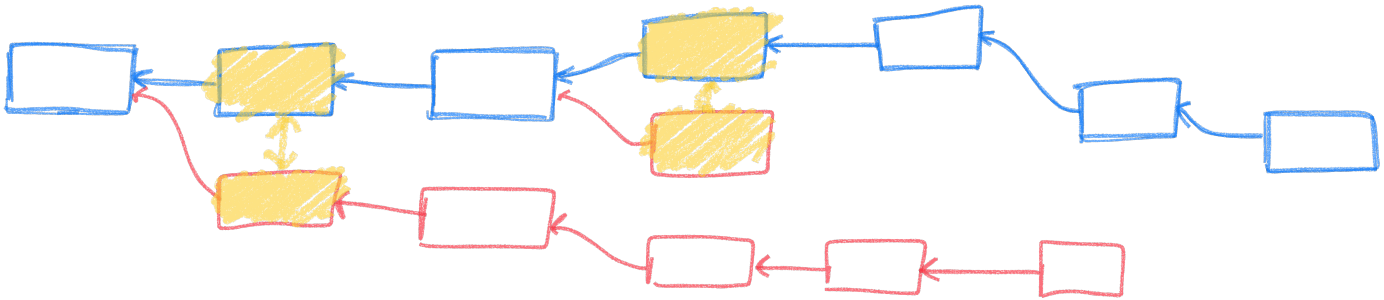
Resolving conflicts in Blockchains

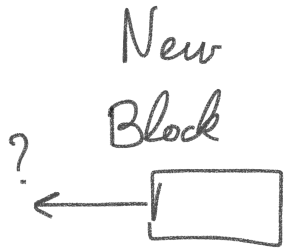


Resolving conflicts in Blockchains

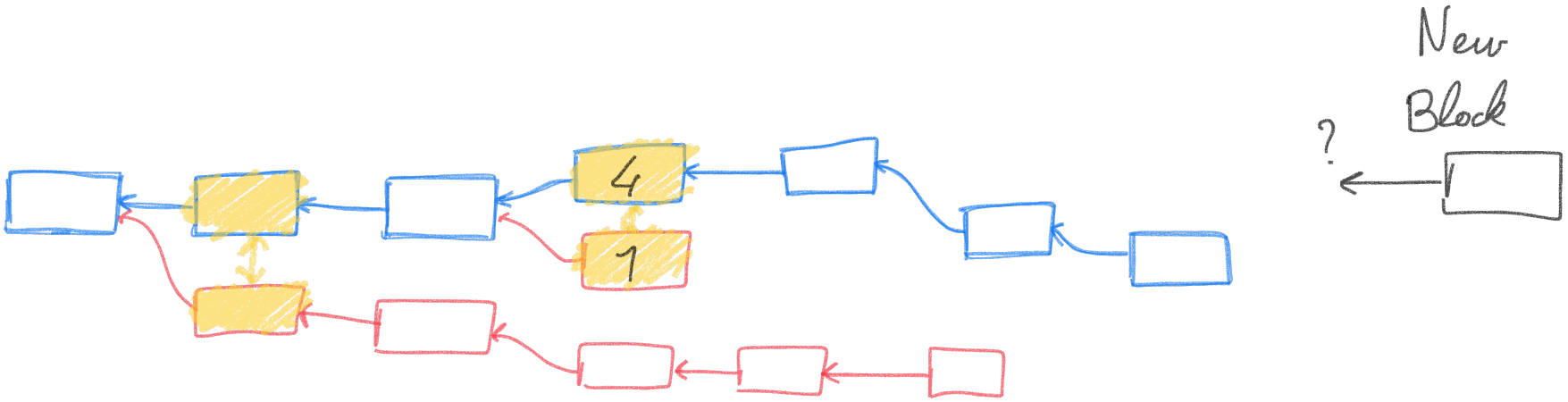


Resolving conflicts in Blockchains

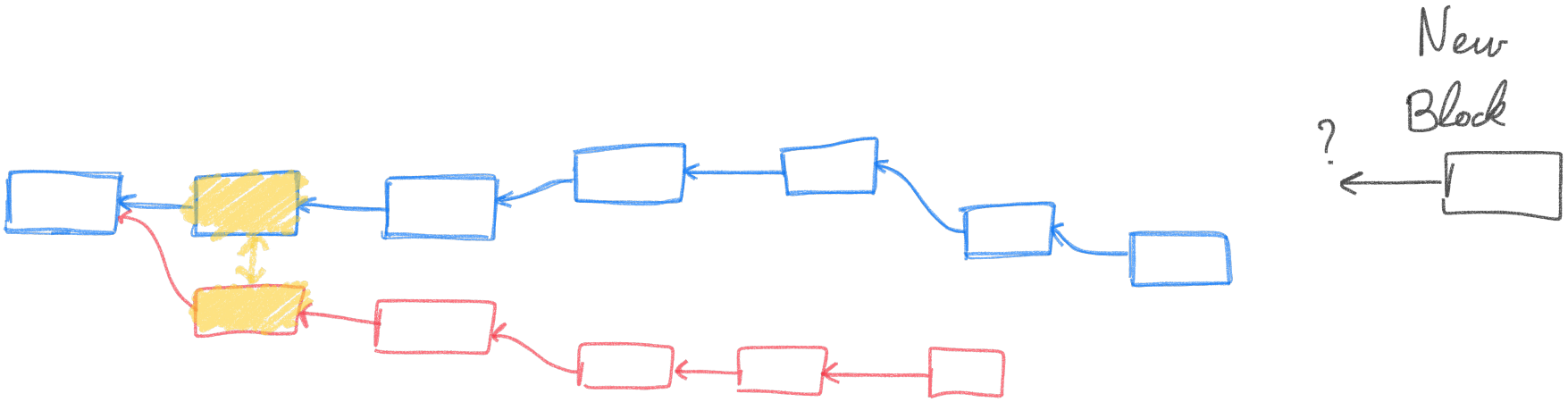




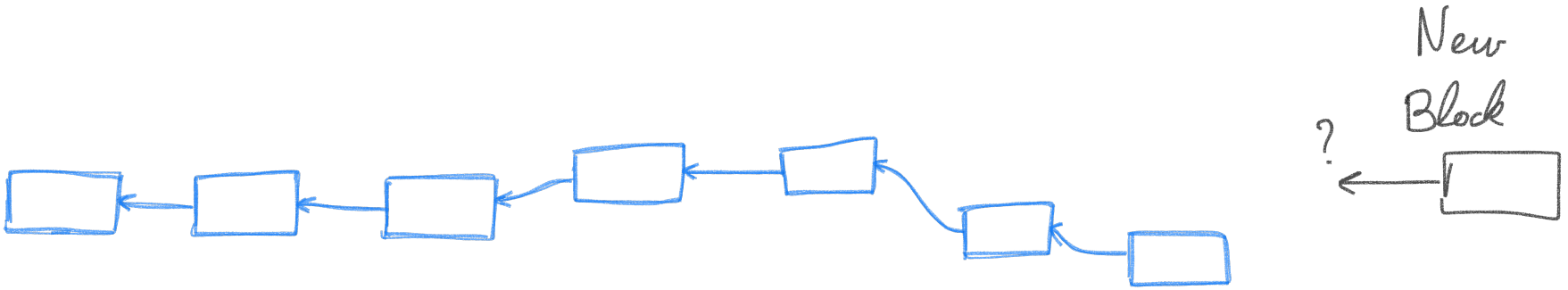
Resolving conflicts in Blockchains



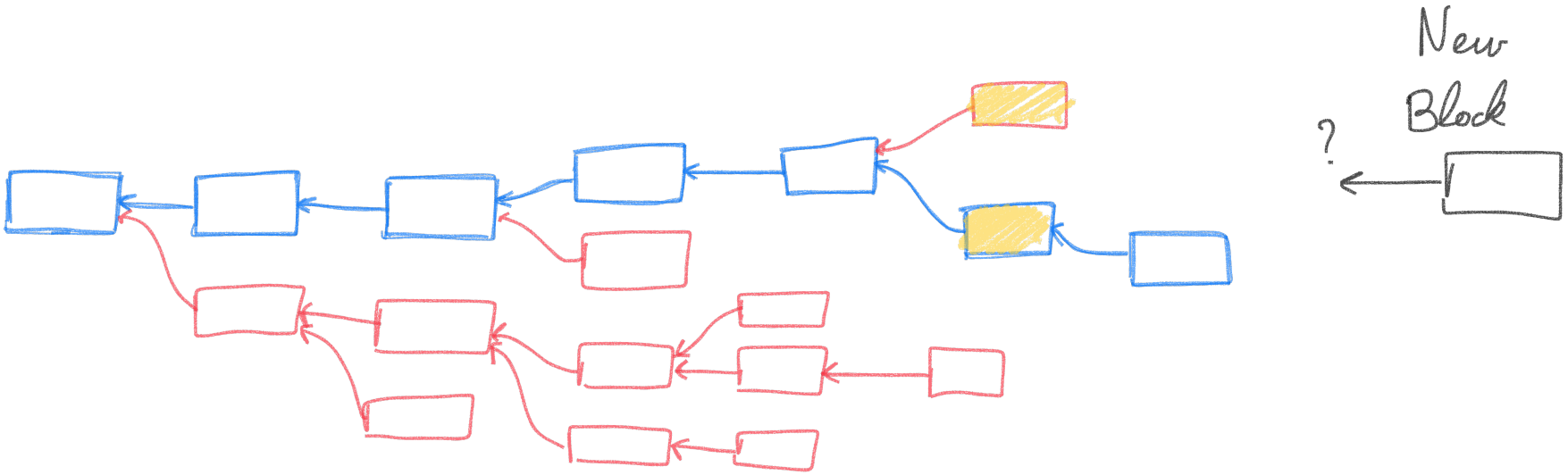
Resolving conflicts in Blockchains



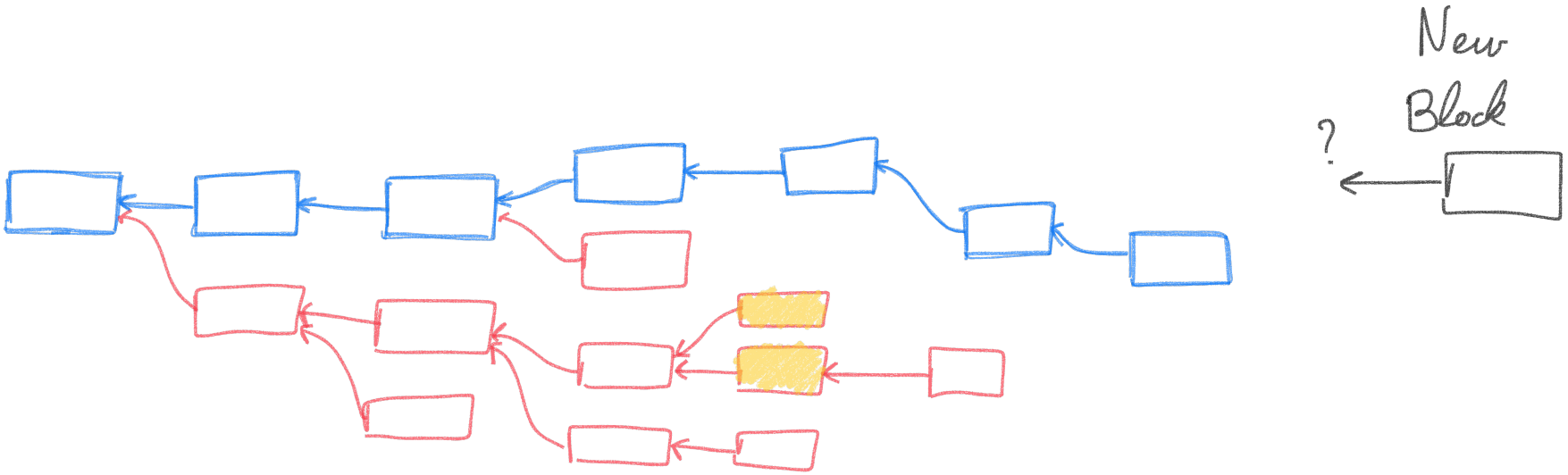
Resolving conflicts in Blockchains



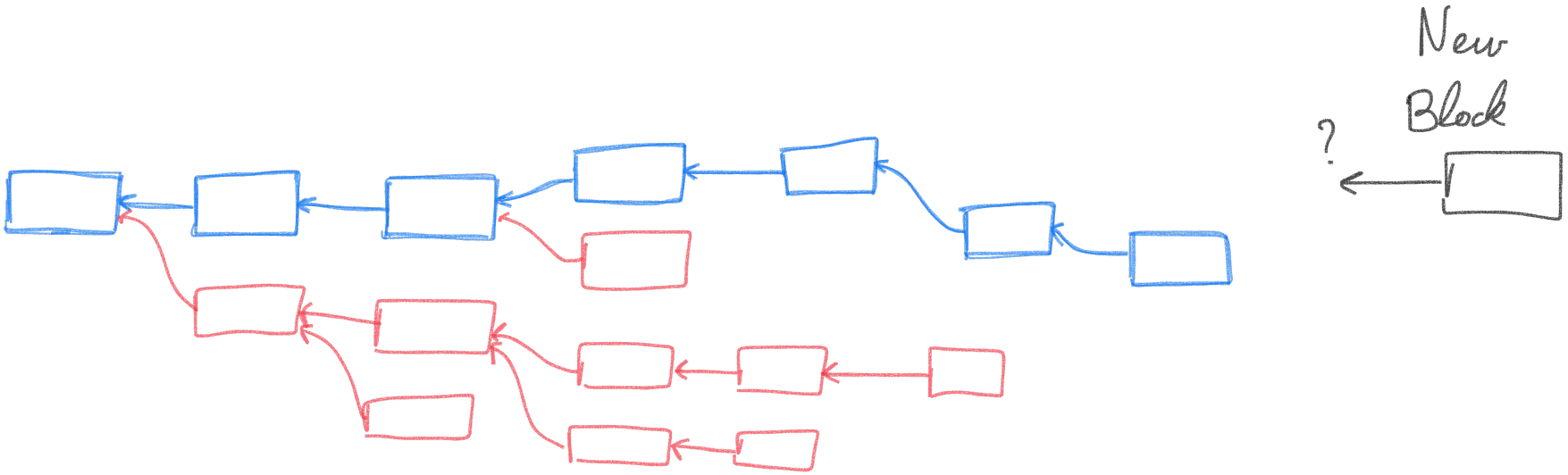
Resolving conflicts in Blockchains



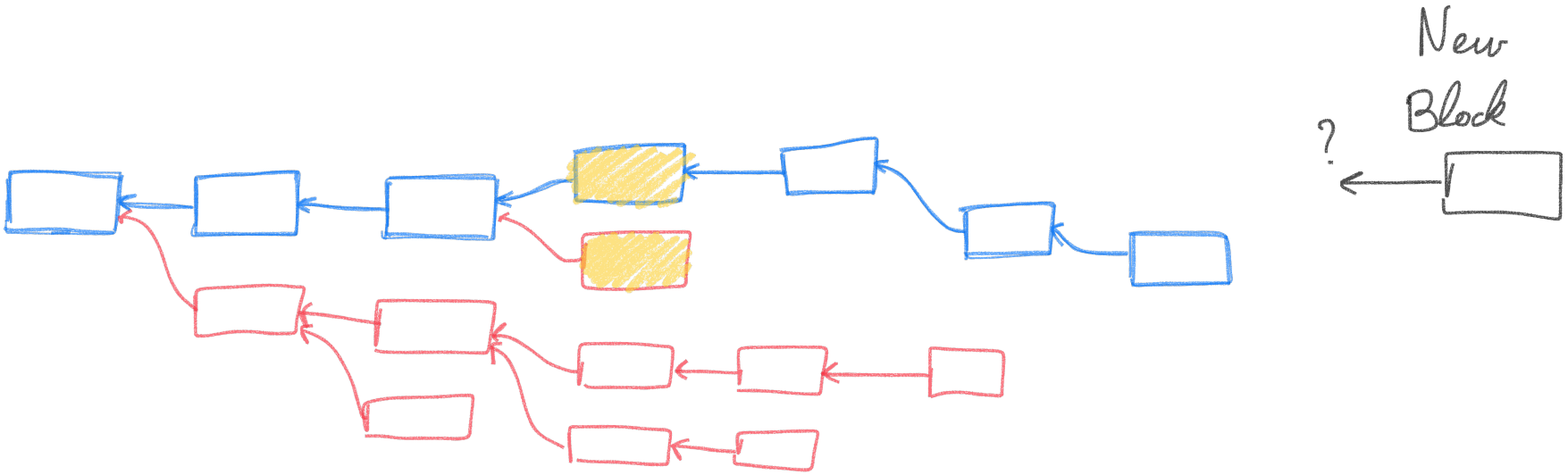
Resolving conflicts in Blockchains



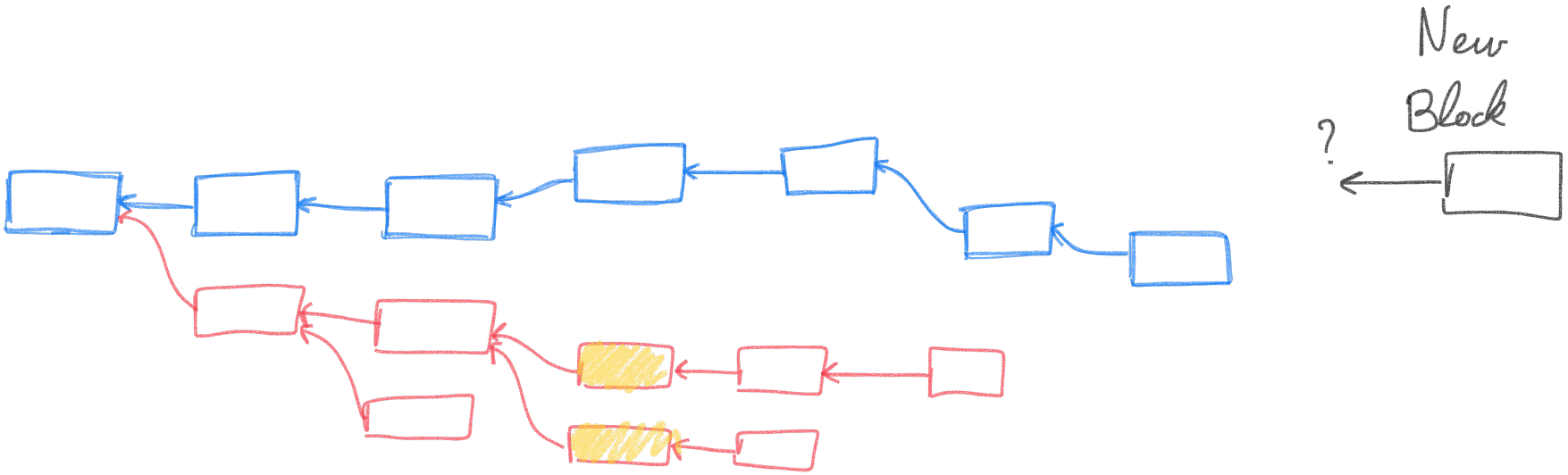
Resolving conflicts in Blockchains



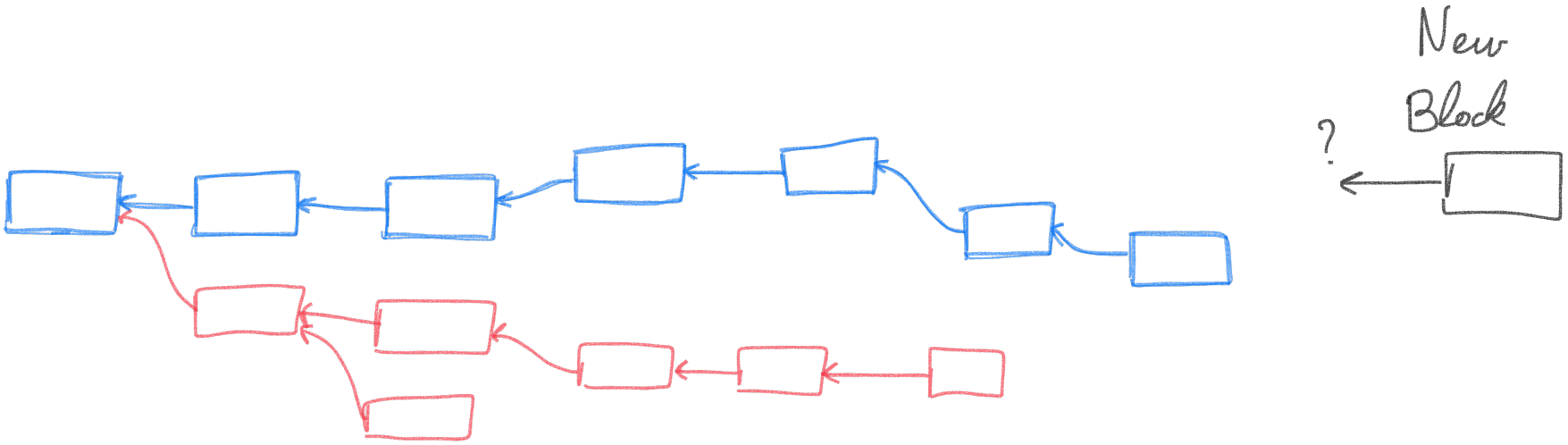
Resolving conflicts in Blockchains



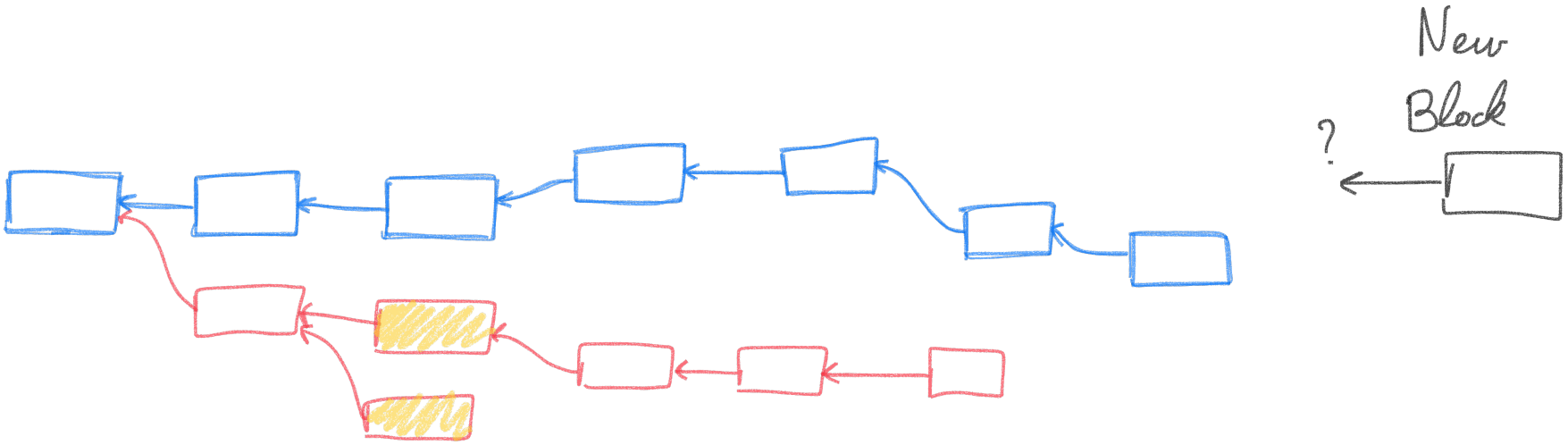
Resolving conflicts in Blockchains



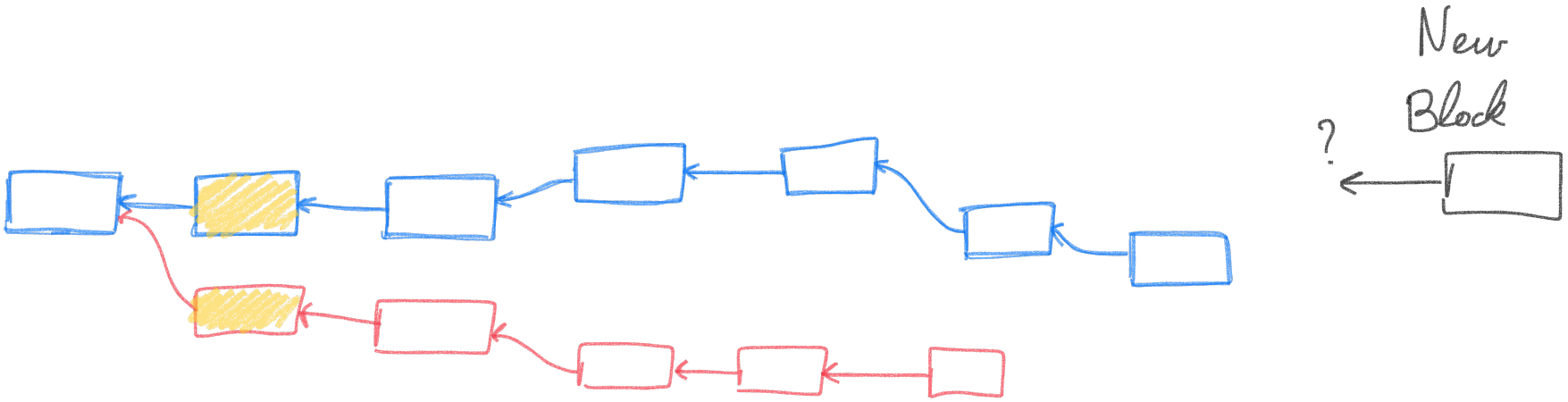
Resolving conflicts in Blockchains



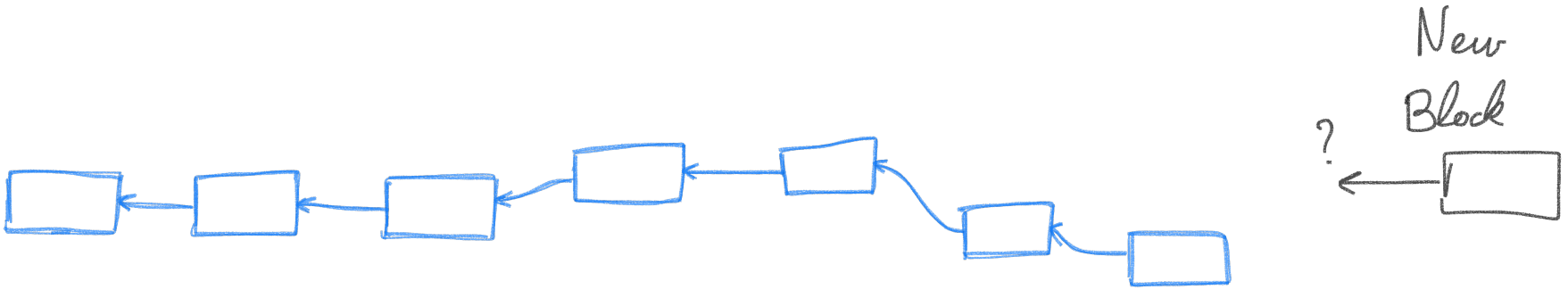
Resolving conflicts in Blockchains



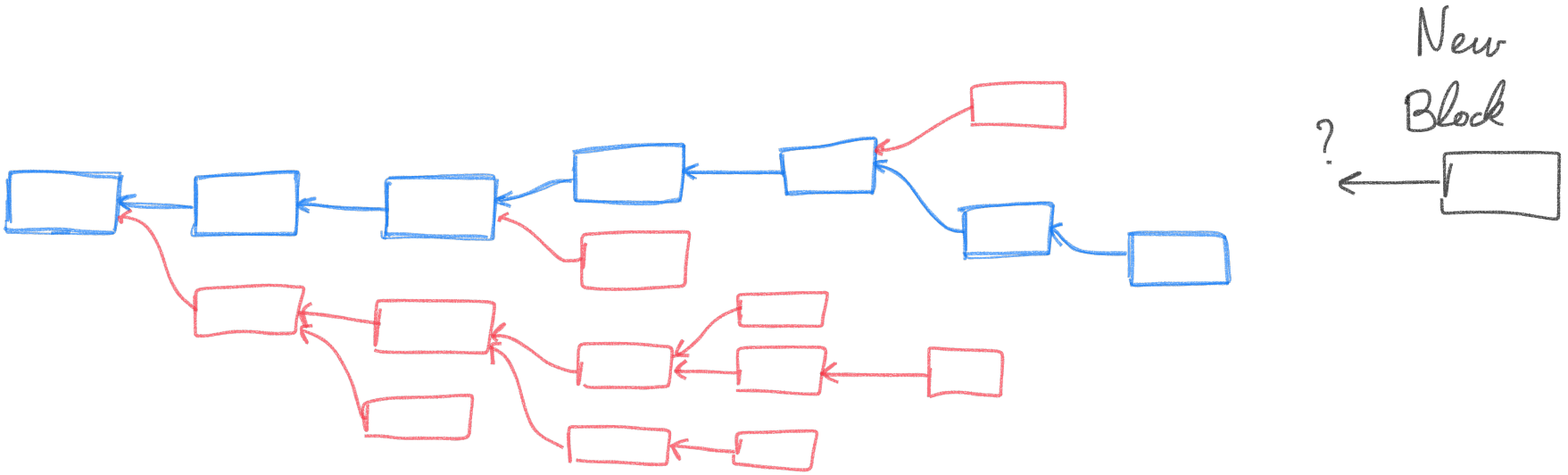
Resolving conflicts in Blockchains



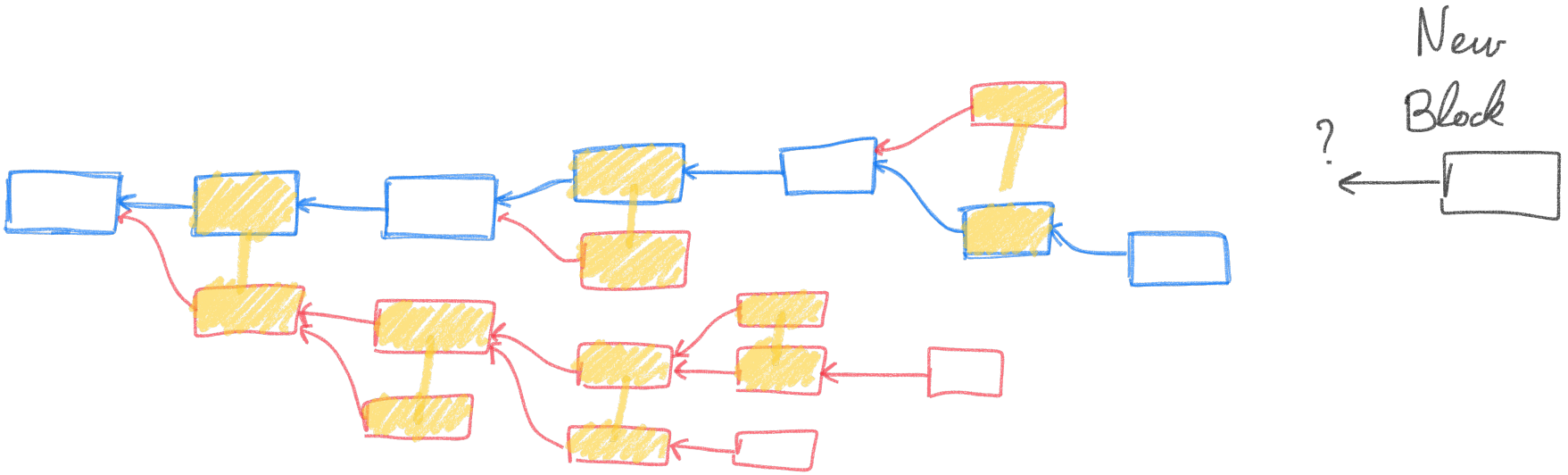
Resolving conflicts in Blockchains



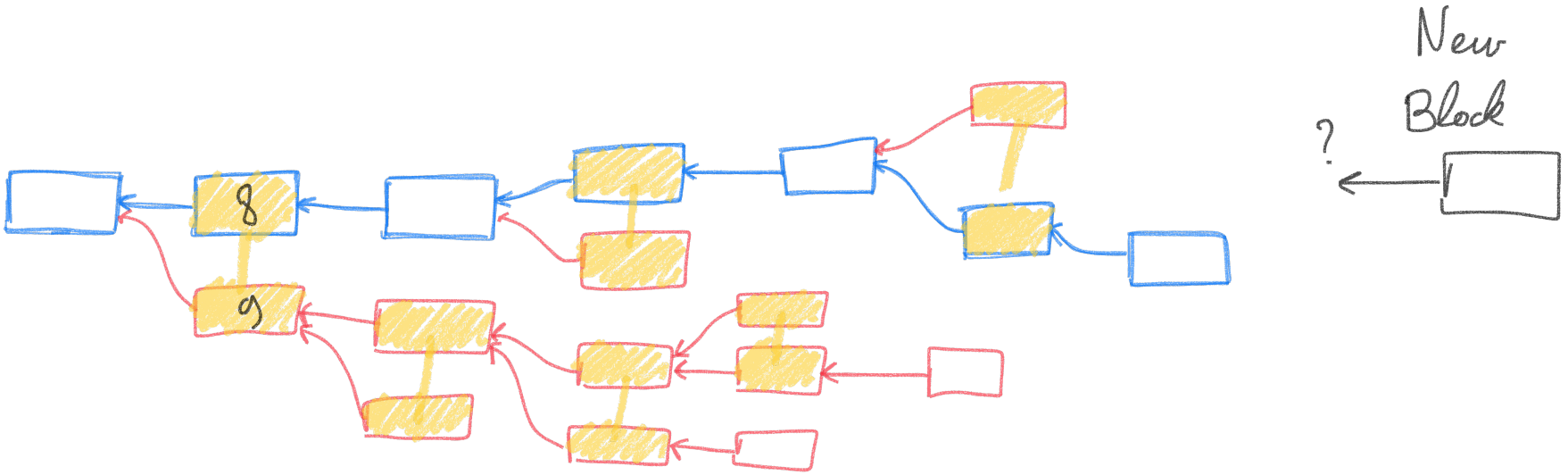
Resolving conflicts in Blockchains



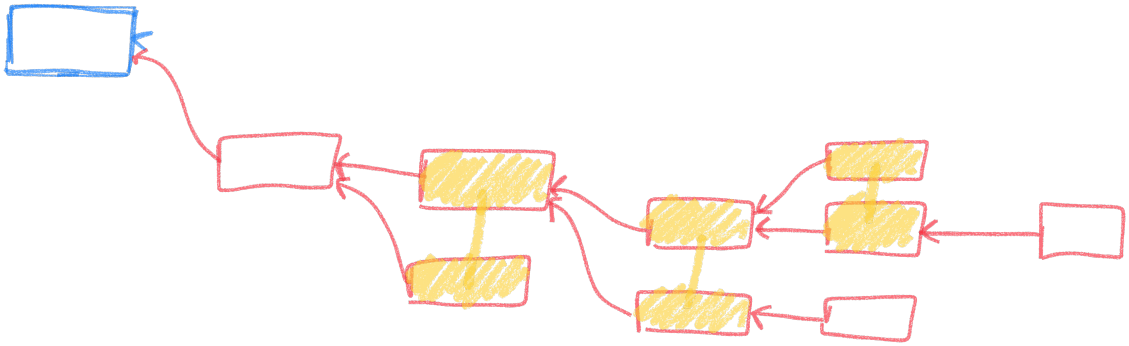
Resolving conflicts in Blockchains

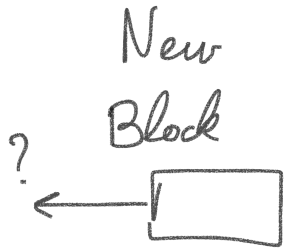


Resolving conflicts in Blockchains

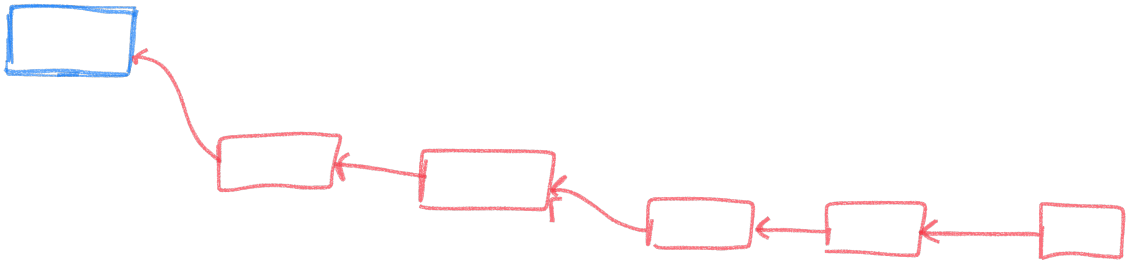


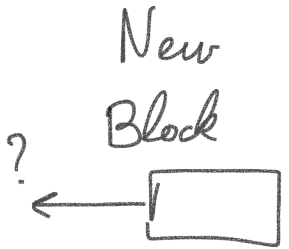
Resolving conflicts in Blockchains



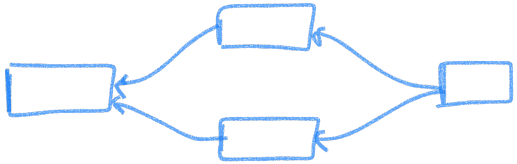


Resolving conflicts in Blockchains

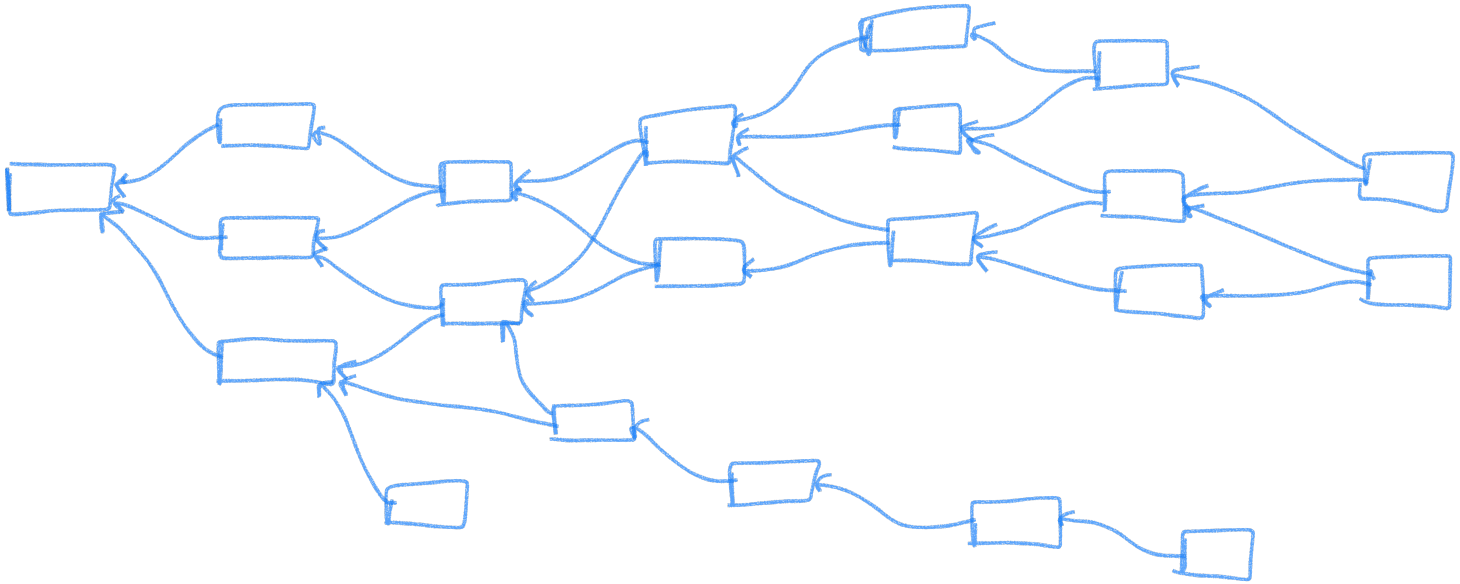




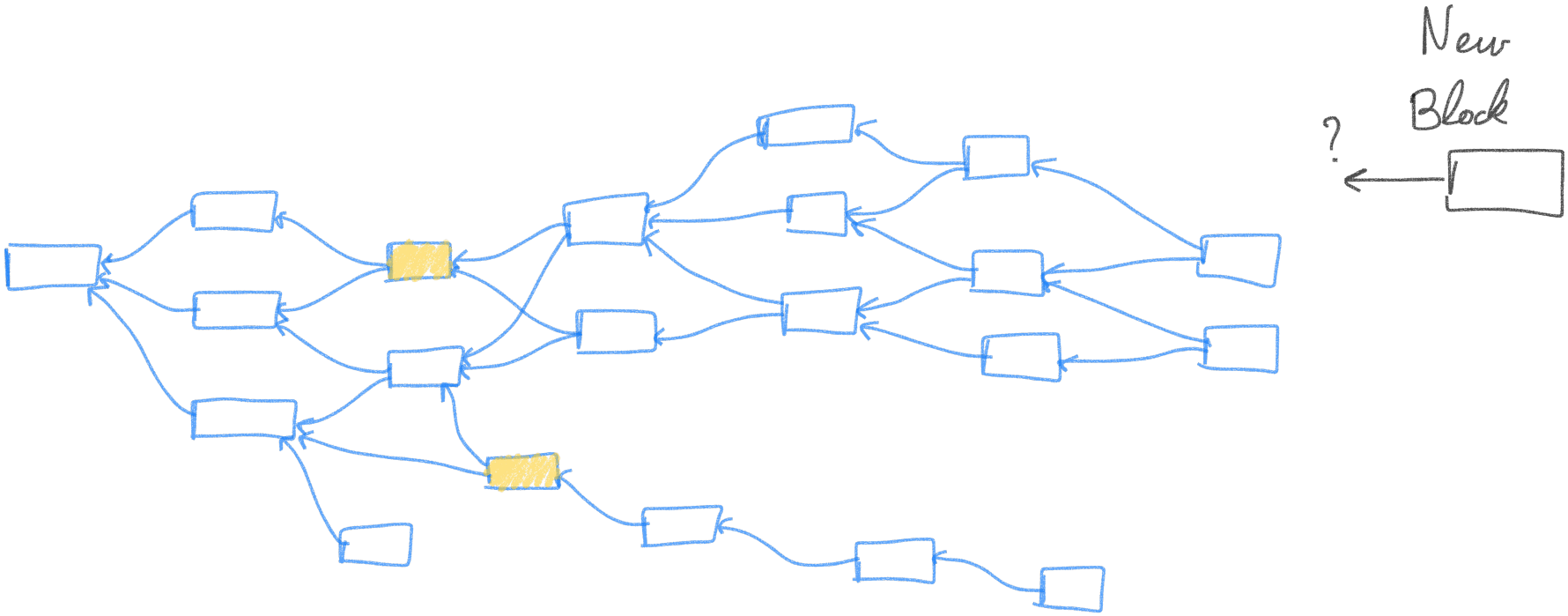
Resolving conflicts in BlockDAGs



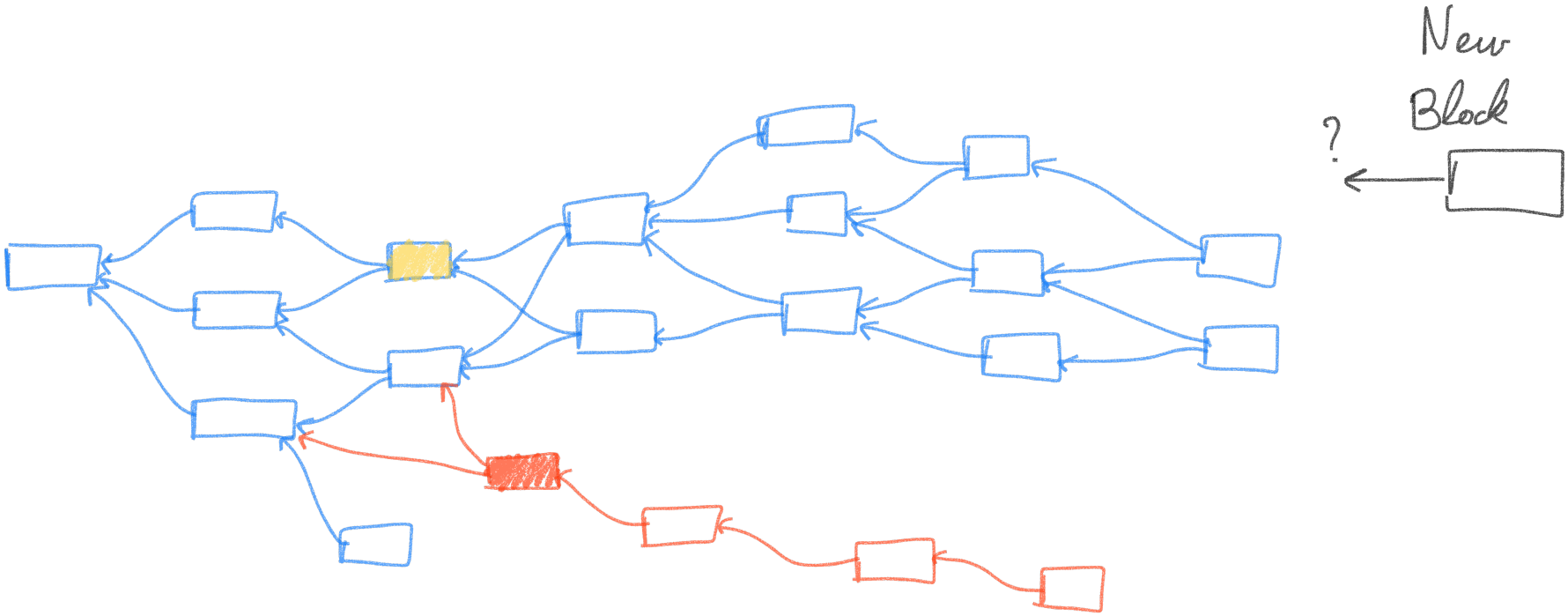
Resolving conflicts in BlockDAGs



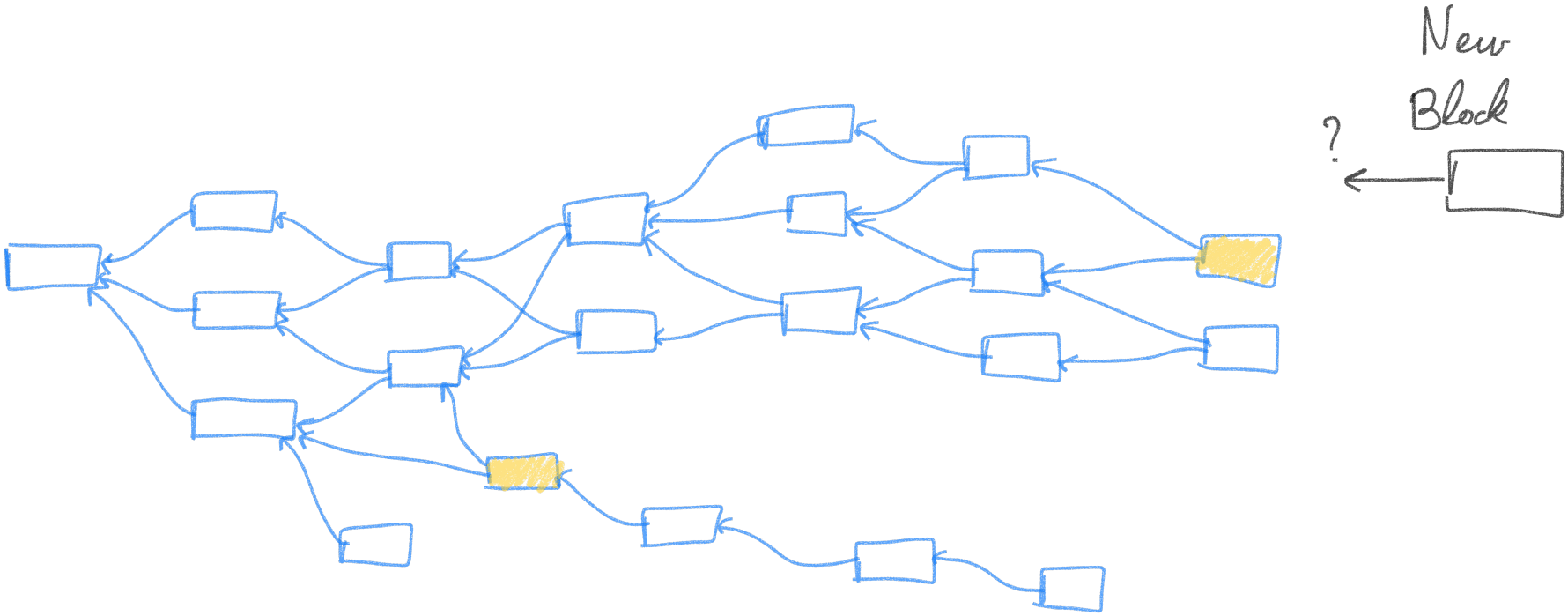
Resolving conflicts in BlockDAGs



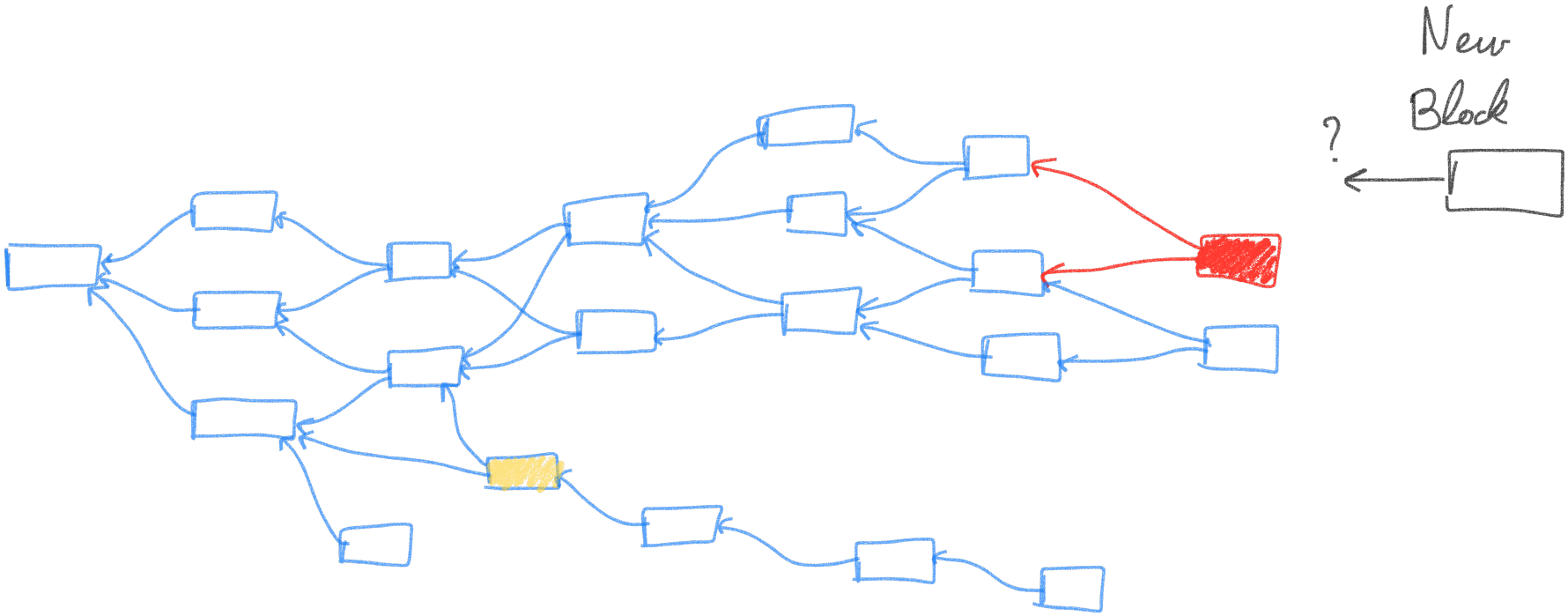
Resolving conflicts in BlockDAGs



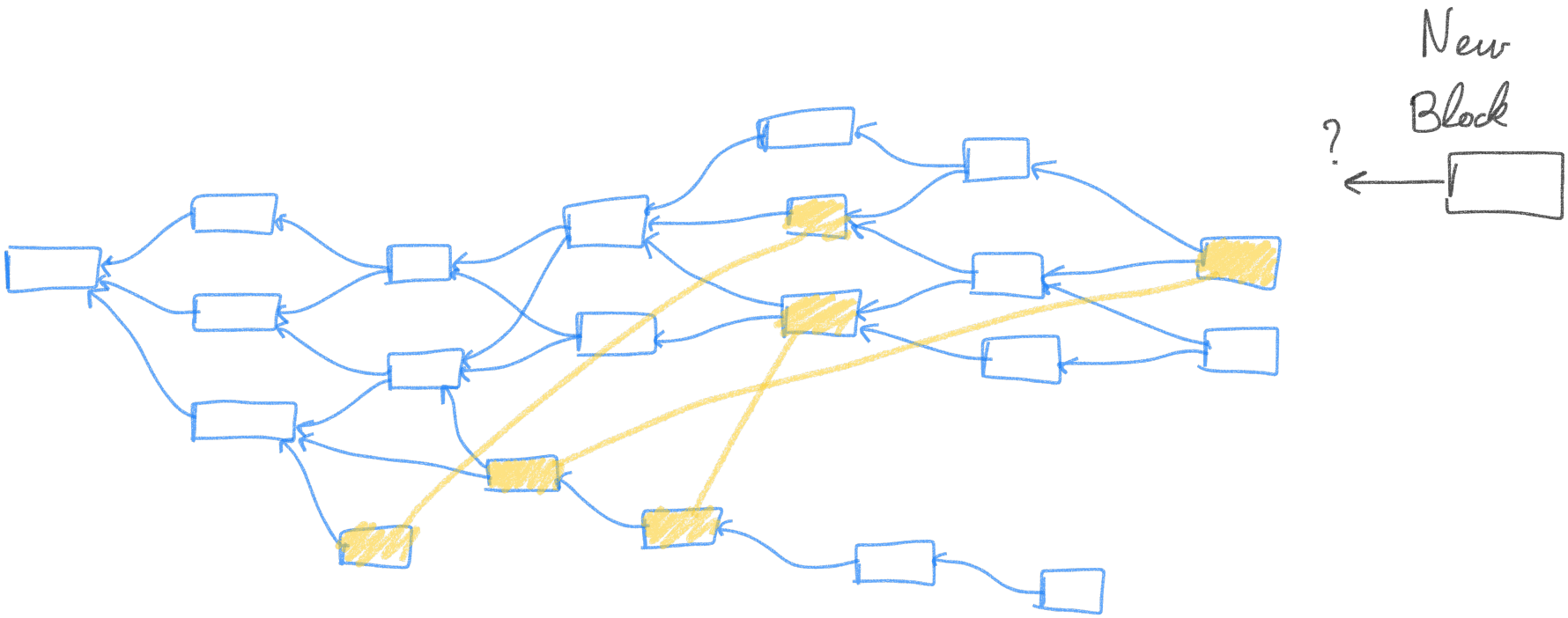
Resolving conflicts in BlockDAGs



Resolving conflicts in BlockDAGs



Problem: How to resolve conflicts?



Related Work

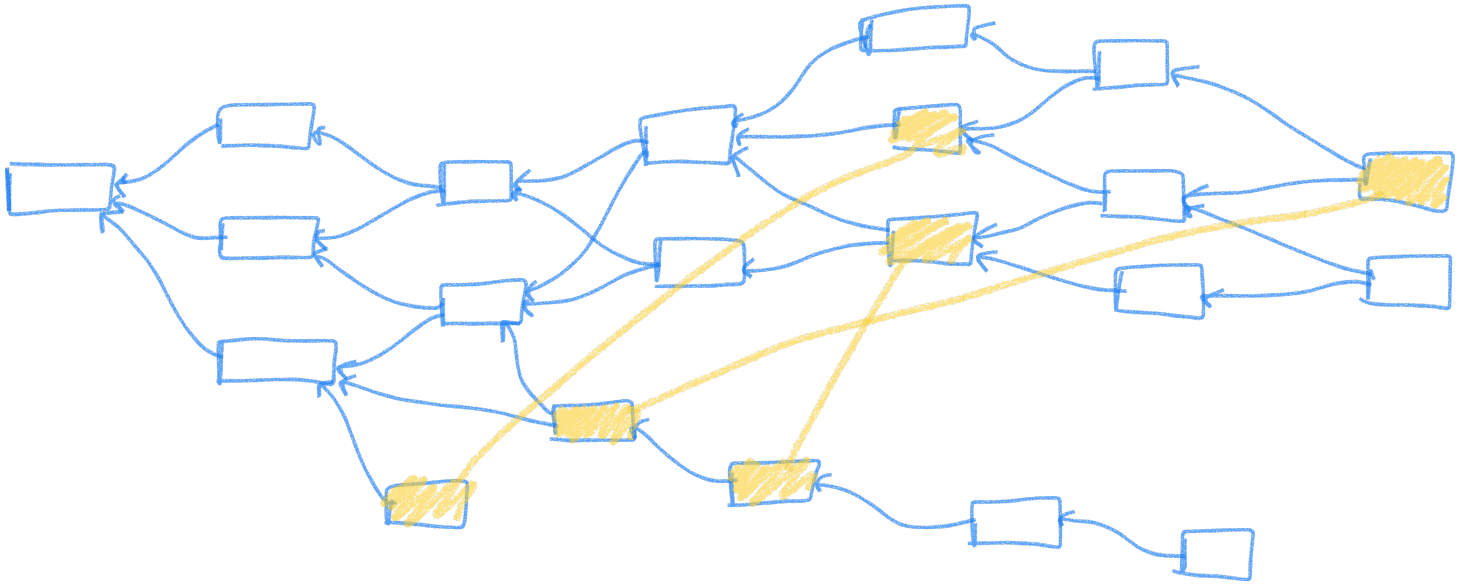
IOTA Tangle (v1): randomized, based on weight

GHOSTDAG: select a subDAG with « small » width (k-cluster)

Several other protocols using PoS or voting mechanisms (out of scope)

Sompolinsky, Yonatan, Shai Wyborski, and Aviv Zohar. "PHANTOM GHOSTDAG: a scalable generalization of Nakamoto consensus: September 2, 2021." Proceedings of the 3rd ACM Conference on Advances in Financial Technologies. 2021.

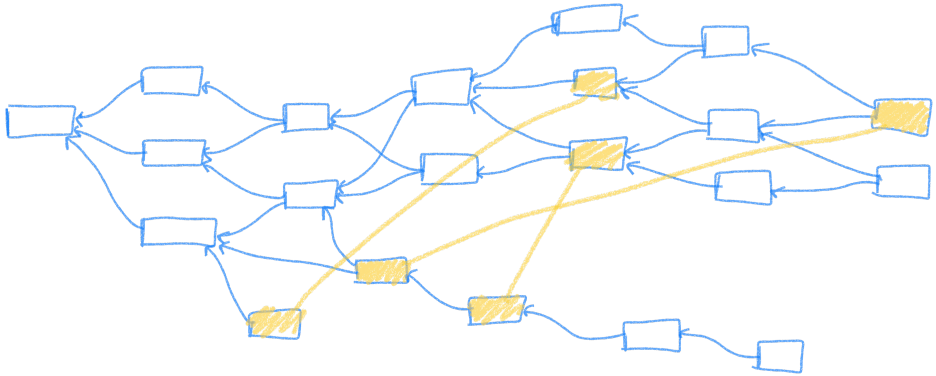
Heaviest blockDAG



Heaviest blockDAG

Given:

- A BlockDAG G
- A set of conflicting set $Conflicts$



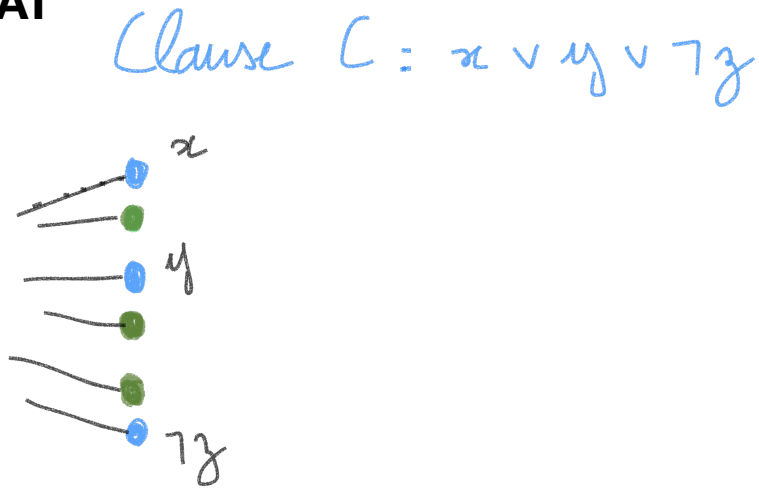
We want:

- The heaviest sub-blockDAG S that have no conflicts

Heaviest blockDAG

Theorem: NP-complete

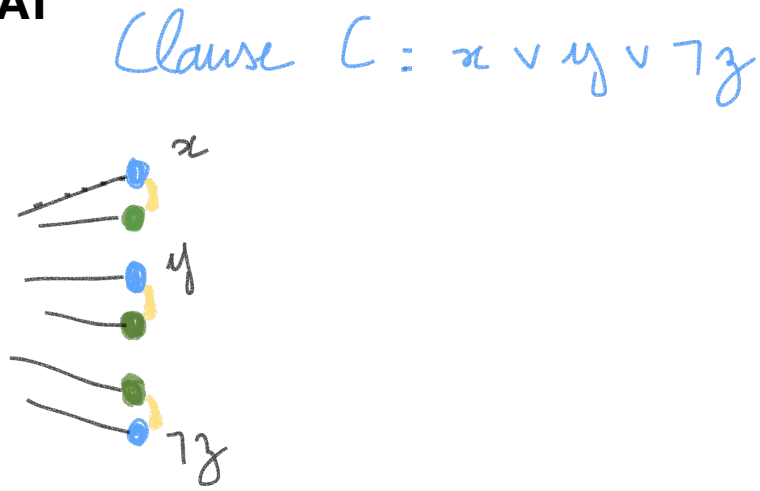
Proof: From 3-SAT



Heaviest blockDAG

Theorem: NP-complete

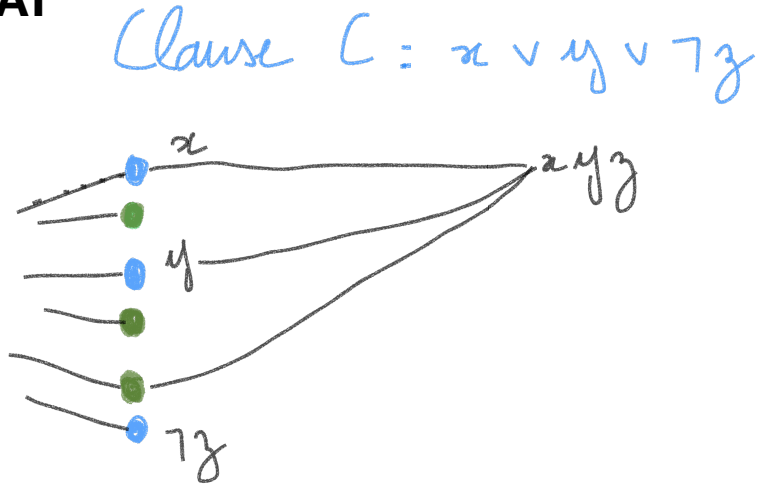
Proof: From 3-SAT



Heaviest blockDAG

Theorem: NP-complete

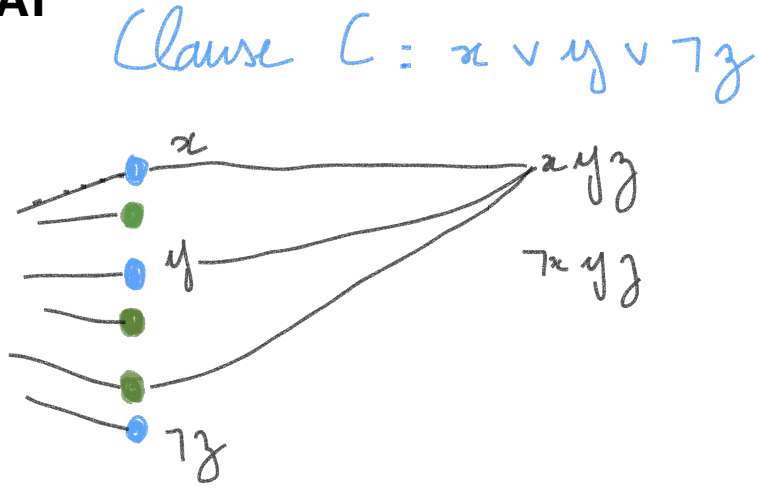
Proof: From 3-SAT



Heaviest blockDAG

Theorem: NP-complete

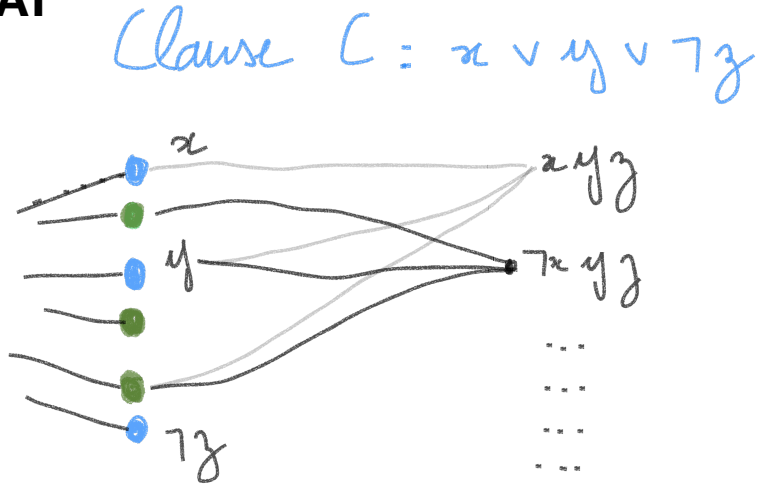
Proof: From 3-SAT



Heaviest blockDAG

Theorem: NP-complete

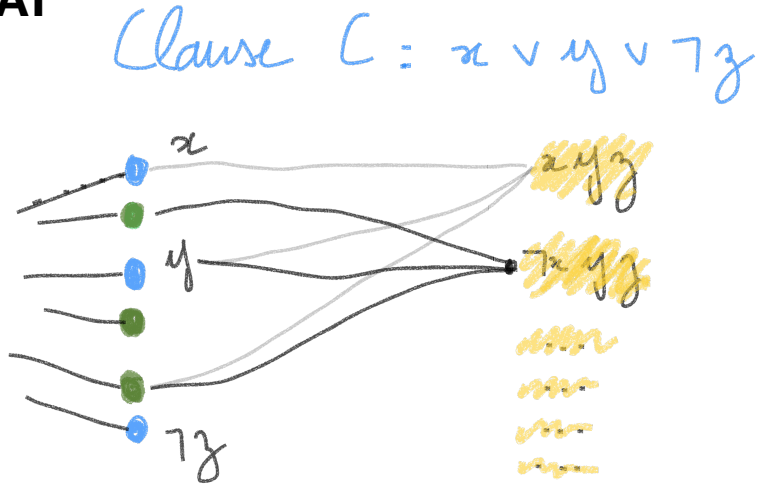
Proof: From 3-SAT



Heaviest blockDAG

Theorem: NP-complete

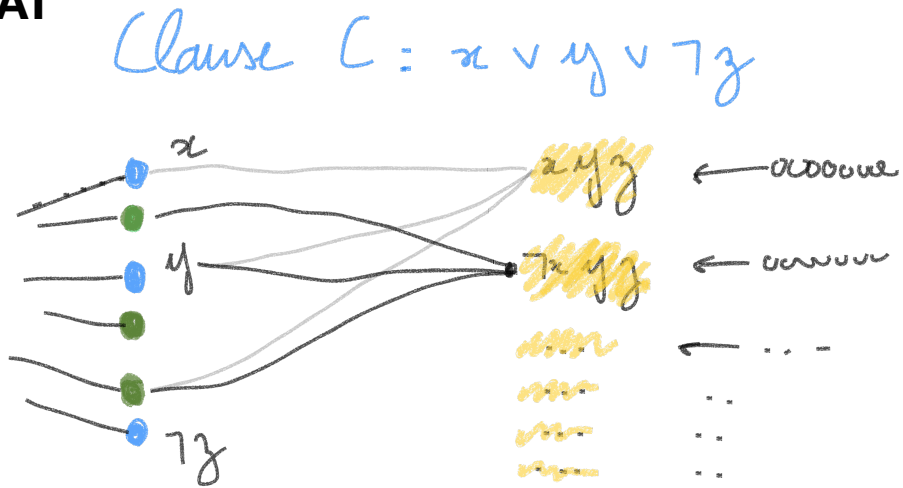
Proof: From 3-SAT



Heaviest blockDAG

Theorem: NP-complete

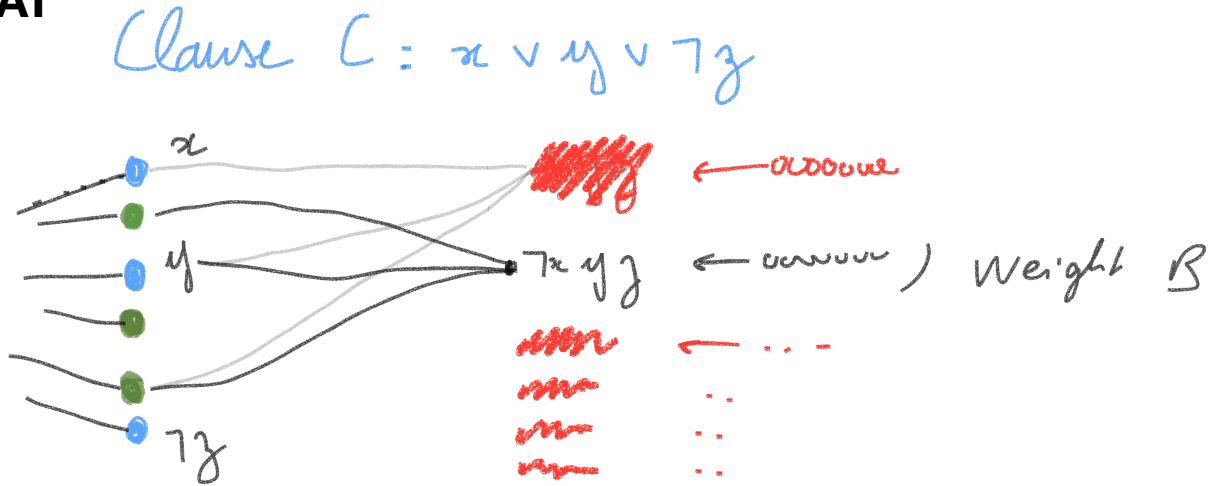
Proof: From 3-SAT



Heaviest blockDAG

Theorem: NP-complete

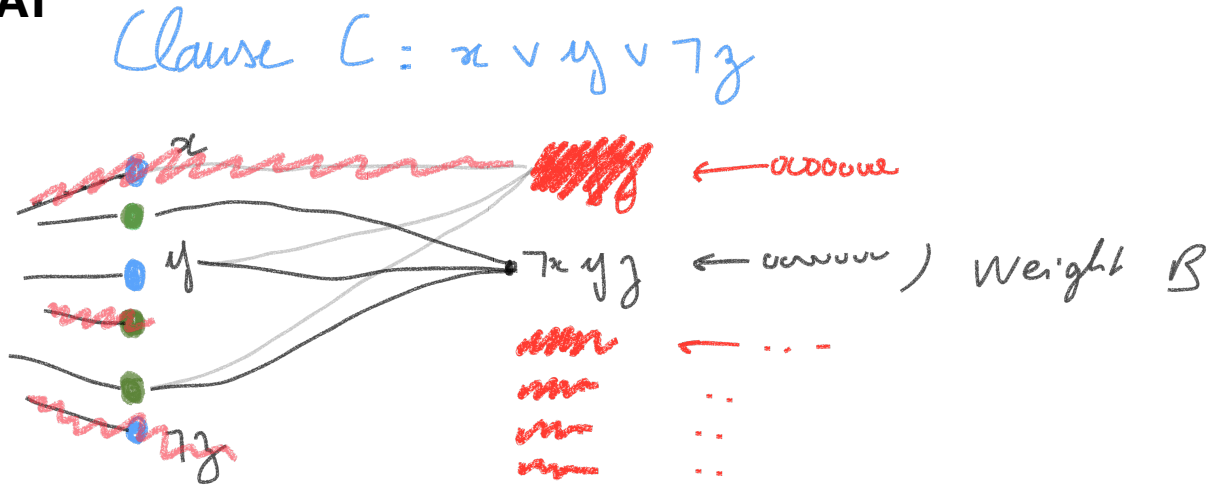
Proof: From 3-SAT



Heaviest blockDAG

Theorem: NP-complete

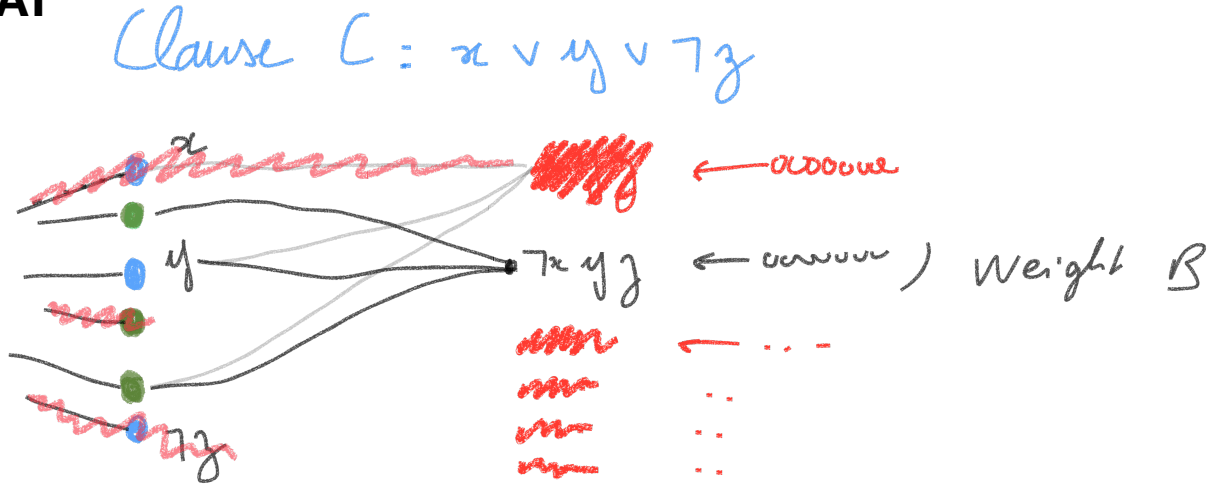
Proof: From 3-SAT



Heaviest blockDAG

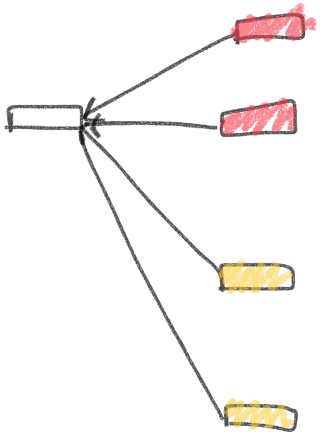
Theorem: NP-complete

Proof: From 3-SAT

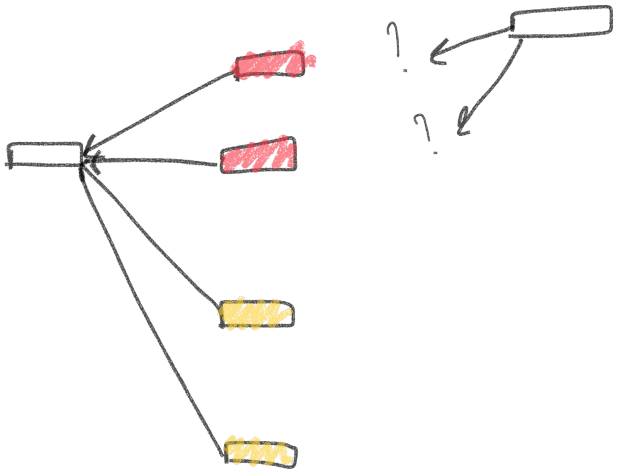


Boolean affectation Satisfying all clauses ⟺ total weight ≥ |clauses| × B

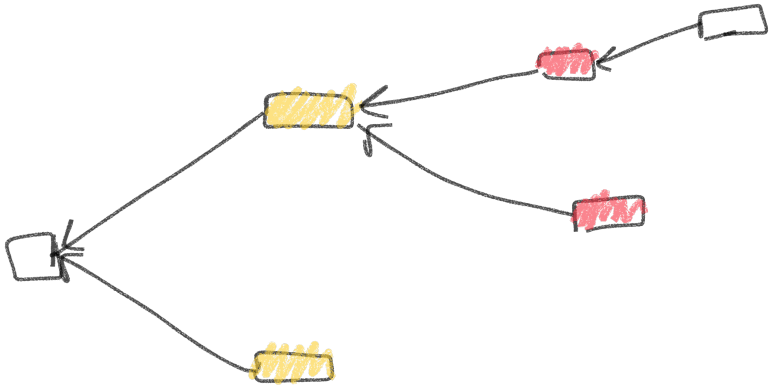
Concurrent conflicts



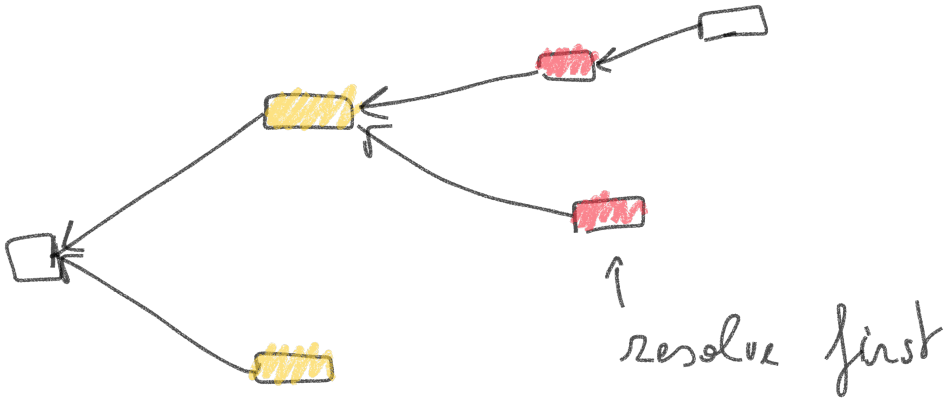
Concurrent conflicts



Concurrent conflicts



Concurrent conflicts



SeHes Algorithm

SeHes Algorithm

"SEquential" HEaviest SubDAG

SeHes Algorithm

"SEquential" HEaviest SubDAG

- find a "small" set of conflicts C (not concurrent with the other conflicts)

SeHes Algorithm

"SEquential" HEaviest SubDAG

- find a "small" set of conflicts $\mathcal{C}$ (not concurrent with the other conflicts)
- Resolve $\mathcal{C}$ (exponential in $|\mathcal{C}|$)

SeHes Algorithm

"SEquential" HEaviest SubDAG

- find a "small" set of conflicts $\mathcal{C}$ (not concurrent with the other conflicts)
- Resolve $\mathcal{C}$ (exponential in $|\mathcal{C}|$)
- if conflicts remain, goto step 1

Evaluation

Evaluation

Simulation :

Evaluation

Simulation :

- SeHeS algorithm ;

Evaluation

Simulation :

- SeHeS algorithm ;
- GHOSTDAG (with $k = \infty$) ;

Evaluation

Simulation :

- SeHeS algorithm ;
- GHOSTDAG (with $k = \infty$) ;
- two greedy approaches (min and max)

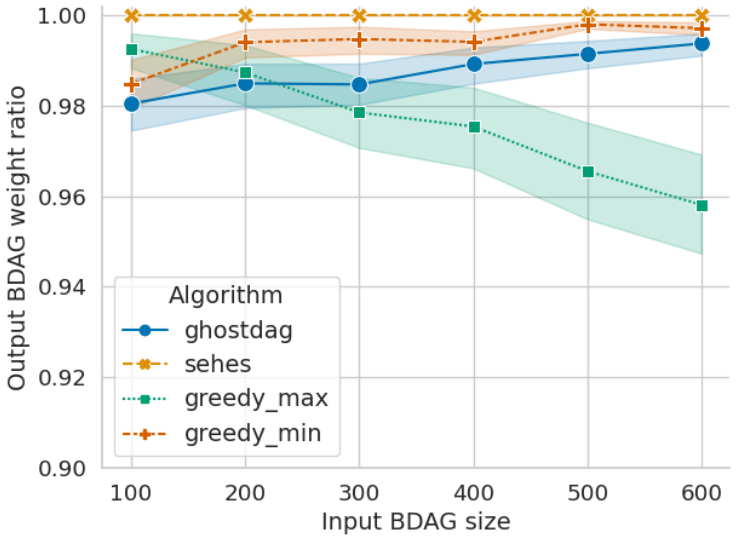
Evaluation

Simulation :

- SeHeS algorithm ;
- GHOSTDAG (with $k = \infty$) ;
- two greedy approaches (min and max)

Random DAG generated by adding blocks 10 by 10 with a constant probability to have conflicts and a constant probability to be forgotten

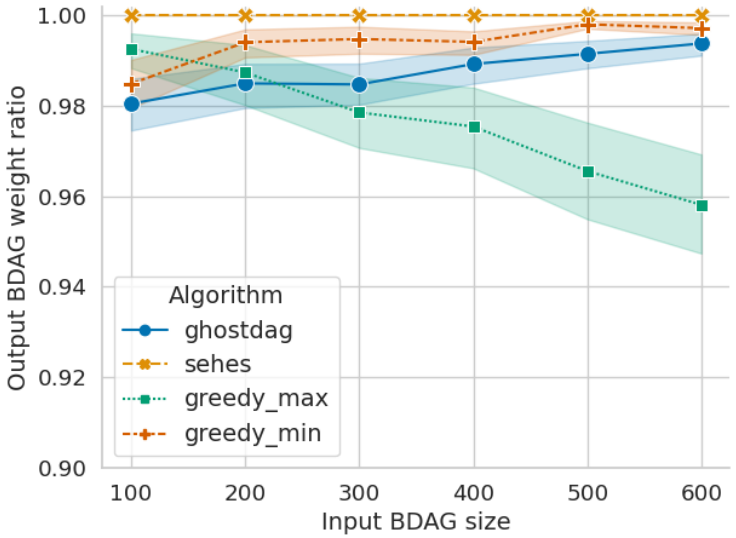
Evaluation

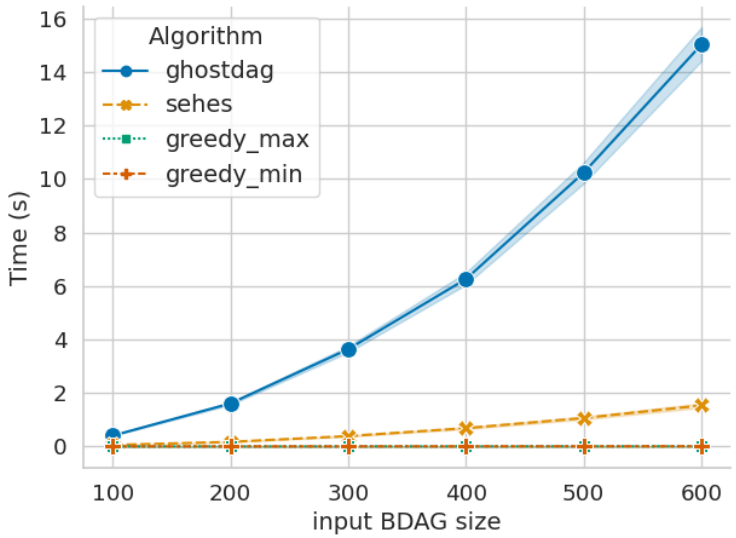


Run on MACbook pro 2,6 GHz Intel Core i7 6-core. 200 random BlockDAGs per size

Code available online: https://doi.org/10.5281/zenodo.8052827

Evaluation





Run on MACbook pro 2,6 GHz Intel Core i7 6-core. 200 random BlockDAGs per size

Code available online: https://doi.org/10.5281/zenodo.8052827

Conclusion

We proposed an efficient algorithm to find the heaviest sub-blockDAG

Future work

Find an efficient incremental algorithm

Evaluation on more topologies

Conclusion

We proposed an efficient algorithm to find the heaviest sub-blockDAG

Future work

Find an efficient incremental algorithm

Evaluation on more topologies

Merci !

Slides available at https://bramas.fr